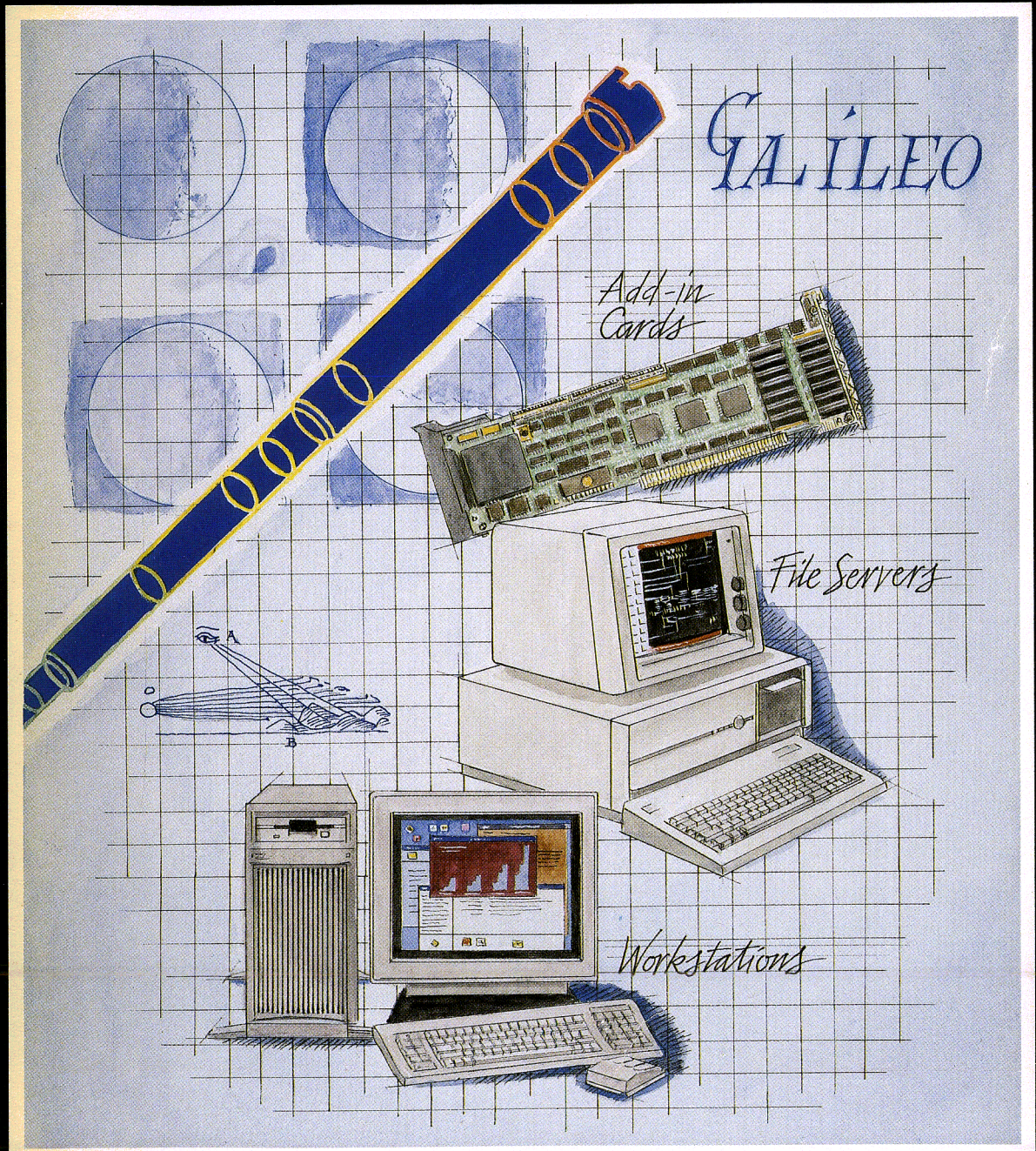
intel®

Peripheral Components



Benedict Norbert, Wong

Order Number: 296467-002

LITERATURE

To order Intel Literature or obtain literature pricing information in the U.S. and Canada call or write Intel Literature Sales. In Europe and other international locations, please contact your ***local*** sales office or distributor.

INTEL LITERATURE SALES
P.O. BOX 7641
Mt. Prospect, IL 60056-7641

In the U.S. and Canada
call toll free
(800) 548-4725

CURRENT HANDBOOKS

Product line handbooks contain data sheets, application notes, article reprints and other design information. All handbooks can be ordered individually, and most are available in a pre-packaged set in the U.S. and Canada.

TITLE	INTEL ORDER NUMBER	ISBN
SET OF THIRTEEN HANDBOOKS (Available in U.S. and Canada)	**231003**	**N/A**
CONTENTS LISTED BELOW FOR INDIVIDUAL ORDERING:		
COMPONENTS QUALITY/RELIABILITY	210997	1-55512-132-2
EMBEDDED APPLICATIONS	270648	1-55512-123-3
8-BIT EMBEDDED CONTROLLERS	270645	1-55512-121-7
16-BIT EMBEDDED CONTROLLERS	270646	1-55512-120-9
16/32-BIT EMBEDDED PROCESSORS	270647	1-55512-122-5
MEMORY PRODUCTS	210830	1-55512-117-9
MICROCOMMUNICATIONS	231658	1-55512-119-5
MICROCOMPUTER PRODUCTS	280407	1-55512-118-7
MICROPROCESSORS	230843	1-55512-115-2
PACKAGING	240800	1-55512-128-4
PERIPHERAL COMPONENTS	296467	1-55512-127-6
PRODUCT GUIDE (Overview of Intel's complete product lines)	210846	1-55512-116-0
PROGRAMMABLE LOGIC	296083	1-55512-124-1
ADDITIONAL LITERATURE: (Not included in handbook set)		
AUTOMOTIVE HANDBOOK	231792	1-55512-125-x
INTERNATIONAL LITERATURE GUIDE (Available in Europe only)	E00029	N/A
CUSTOMER LITERATURE GUIDE	210620	N/A
MILITARY HANDBOOK (2 volume set)	210461	1-55512-126-8
SYSTEMS QUALITY/RELIABILITY	231762	1-55512-046-6

intel®

U.S. and CANADA LITERATURE ORDER FORM

NAME: _____

COMPANY: _____

ADDRESS: _____

CITY: _____ STATE: _____ ZIP: _____

COUNTRY: _____

PHONE NO.: ( ) _____

ORDER NO.	TITLE	QTY.	PRICE	TOTAL
			×	=
			×	=
			×	=
			×	=
			×	=
			×	=
			×	=
			×	=
			×	=
			×	=

Subtotal _____

Must Add Your Local Sales Tax _____

Include postage:
Must add 15% of Subtotal to cover U.S. and Canada postage. (20% all other.)

→ Postage _____

Total _____

Pay by check, money order, or include company purchase order with this form ($100 minimum). We also accept VISA, MasterCard or American Express. Make payment to Intel Literature Sales. Allow 2-4 weeks for delivery.

☐ VISA ☐ MasterCard ☐ American Express Expiration Date _____

Account No. _____

Signature _____

Mail To: Intel Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

International Customers outside the U.S. and Canada should use the International order form on the next page or contact their local Sales Office or Distributor.

For phone orders in the U.S. and Canada
Call Toll Free: (800) 548-4725

Prices good until 12/31/91.
Source HB

intel®

INTERNATIONAL LITERATURE ORDER FORM

NAME: _____

COMPANY: _____

ADDRESS: _____

CITY: _____ STATE: _____ ZIP: _____

COUNTRY: _____

PHONE NO.: ( ) _____

ORDER NO.	TITLE	QTY.		PRICE	TOTAL
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=

Subtotal _____

Must Add Your Local Sales Tax _____

Total _____

PAYMENT

Cheques should be made payable to your ***local*** Intel Sales Office (see inside back cover).

Other forms of payment may be available in your country. Please contact the Literature Coordinator at your ***local*** Intel Sales Office for details.

The completed form should be marked to the attention of the LITERATURE COORDINATOR and returned to your ***local*** Intel Sales Office.

Intel Corporation is a leading supplier of microcomputer components, modules and systems. When Intel invented the microprocessor in 1971, it created the era of the microcomputer. Today, Intel architectures are considered world standards. Whether used in embedded applications such as automobiles, printers and microwave ovens, or as the CPU in personal computers, client servers or supercomputers, Intel delivers leading-edge technology.

PERIPHERAL COMPONENTS

1991

About Our Cover:

Thinkers, inventors, and artists throughout history have breathed life into their ideas by converting them into rough working sketches, models, and products. This series of covers shows a few of these creations, along with the applications and products created by Intel customers.

Additional copies of this manual or other Intel literature may be obtained from:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

CUSTOMER SUPPORT

INTEL'S COMPLETE SUPPORT SOLUTION WORLDWIDE

Customer Support is Intel's complete support service that provides Intel customers with hardware support, software support, customer training, consulting services and network management services. For detailed information contact your local sales offices.

After a customer purchases any system hardware or software product, service and support become major factors in determining whether that product will continue to meet a customer's expectations. Such support requires an international support organization and a breadth of programs to meet a variety of customer needs. As you might expect, Intel's customer support is extensive. It can start with assistance during your development effort to network management. 100 Intel sales and service offices are located worldwide – in the U.S., Canada, Europe and the Far East. So wherever you're using Intel technology, our professional staff is within close reach.

HARDWARE SUPPORT SERVICES

Intel's hardware maintenance service, starting with complete on-site installation will boost your productivity from the start and keep you running at maximum efficiency. Support for system or board level products can be tailored to match your needs, from complete on-site repair and maintenance support to economical carry-in or mail-in factory service.

Intel can provide support service for not only Intel systems and emulators, but also support for equipment in your development lab or provide service on your product to your end-user/customer.

SOFTWARE SUPPORT SERVICES

Software products are supported by our Technical Information Service (TIPS) that has a special toll free number to provide you with direct, ready information on known, documented problems and deficiencies, as well as work-arounds, patches and other solutions.

Intel's software support consists of two levels of contracts. Standard support includes TIPS (Technical Information Phone Service), updates and subscription service (product-specific troubleshooting guides and; *COMMENTS Magazine*). Basic support consists of updates and the subscription service. Contracts are sold in environments which represent product groupings (e.g., iRMX® environment).

NETWORK SERVICE AND SUPPORT

Today's broad spectrum of powerful networking capabilities are only as good as the customer support provided by the vendor. Intel offers network services and support structured to meet a wide variety of end-user computing needs. From a ground up design of your network's physical and logical design to implementation, installation and network wide maintenance. From software products to turn-key system solutions; Intel offers the customer a complete networked solution. With over 10 years of network experience in both the commercial and Government arena; network products, services and support from Intel provide you the most optimized network offering in the industry.

CONSULTING SERVICES

Intel provides field system engineering consulting services for any phase of your development or application effort. You can use our system engineers in a variety of ways ranging from assistance in using a new product, developing an application, personalizing training and customizing an Intel product to providing technical and management consulting. Systems Engineers are well versed in technical areas such as microcommunications, real-time applications, embedded microcontrollers, and network services. You know your application needs; we know our products. Working together we can help you get a successful product to market in the least possible time.

CUSTOMER TRAINING

Intel offers a wide range of instructional programs covering various aspects of system design and implementation. In just three to ten days a limited number of individuals learn more in a single workshop than in weeks of self-study. For optimum convenience, workshops are scheduled regularly at Training Centers worldwide or we can take our workshops to you for on-site instruction. Covering a wide variety of topics, Intel's major course categories include: architecture and assembly language, programming and operating systems, BITBUS™ and LAN applications.

DATA SHEET DESIGNATIONS

Intel uses various data sheet markings to designate each phase of the document as it relates to the product. The marking appears in the upper, right-hand corner of the data sheet. The following is the definition of these markings:

Data Sheet Marking	Description
Product Preview	Contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product becomes available.
Advanced Information	Contains information on products being sampled or in the initial production phase of development.*
Preliminary	Contains preliminary information on new products in production.*
No Marking	Contains information on products in full production.*

*Specifications within these data sheets are subject to change without notice. Verify with your local Intel sales office that you have the latest data sheet before finalizing a design.

Chip Sets 1

Memory Controllers 2

Support Peripherals 3

Floppy Disk Controller 4

Hard Disk Controller 5

Universal Peripheral Interface Slave Microcontrollers 6

Table of Contents

Alphanumeric Index xi
Overview xii

CHAPTER 1
Chip Sets
ISA CHIP SETS
82341 ISA Peripheral Combo Chip 1-1
82340DX ISA Chip Set 1-8
82340SX ISA Chip Set 1-10
82343 System Controller/Data Buffer 1-12
82344 ISA Bus Controller 1-24
82345 Data Buffer 1-37
82346 System Controller 1-43
82347 Power Management PC Peripheral 1-55
82335 SX High-Integration Interface Device for 386 SX Microprocessor Based PC-AT Systems 1-81
82230/82231 High-Integration AT*-Compatible Chip Set 1-137
EISA CHIP SET
82350 EISA Chip Set 1-174
82357 Integrated System Peripheral (ISP) 1-183
82358 EISA Bus Controller (EBC) 1-272
82352 EISA Bus Buffer (EBB) 1-389
82355 Bus Master Interface Controller (BMIC) 1-424
82077AA CHMOS Single-Chip Floppy Disk Controller 1-508
MICRO CHANNEL CHIP SETS
82311 Micro Channel* Compatible Peripheral Family 1-509
82303 Local I/O Support Chip 1-526
82304 Local I/O Support Chip 1-536
82307 DMA/Micro Channel Arbitration Controller 1-551
82308 Micro Channel Bus Controller 1-563
82309 Address Bus Controller 1-594
82077AA CHMOS Single-Chip Floppy Disk Controller 1-624

CHAPTER 2
Memory Controllers
DATA SHEETS
8206 Error Detection and Correction Unit 2-1
8207 Dual-Port Dynamic RAM Controller 2-23
82C08 CHMOS Dynamic RAM Controller 2-70
APPLICATION NOTES
AP-167 Interfacing the 8207 Dynamic RAM Controller to the iAPX 186 2-99
AP-168 Interfacing the 8207 Advanced Dynamic RAM Controller to the iAPX 286 .. 2-106

CHAPTER 3
Support Peripherals
DATA SHEETS
8231A Arithmetic Processing Unit 3-1
8237A High Performance Programmable DMA Controller (8237A, 8237A-4, 8237A-5) 3-14
82C37A-5 CHMOS High Performance Programmable DMA Controller 3-33
8253/8253-5 Programmable Interval Timer 3-51
8254 Programmable Interval Timer 3-62
82C54 CHMOS Programmable Interval Timer 3-83
8255A/8255A-5 Programmable Peripheral Interface 3-100
82C55A CHMOS Programmable Peripheral Interface 3-124
8256AH Multifunction Microprocessor Support Controller 3-147

Table of Contents (Continued)

8259A Programmable Interrupt Controller (8259A/8259A-2) 3-171
82C59A-2 CHMOS Programmable Interrupt Controller 3-195
8279/8279-5 Programmable Keyboard/Display Interface 3-215
82389 Message Passing Coprocessor A MULTIBUS II Bus Interface Controller 3-231

CHAPTER 4
Floppy Disk Controller
DATA SHEET
82077AA CHMOS Single-Chip Floppy Disk Controller 4-1

CHAPTER 5
Hard Disk Controller
DATA SHEET
82064 CHMOS Winchester Disk Controller with On-Chip Error Detection and Correction 5-1
APPLICATION NOTE
AP-402 Multimodule Winchester Controller Using the CHMOS 82064 5-33

CHAPTER 6
Universal Peripheral Interface Slave Microcontrollers
Microprocessor Peripherals UPI-41A/41AH/42/42AH User's Manual 6-1
DATA SHEETS
UPI-452 CHMOS Programmable I/O Processor 6-69
UPI-41AH/42AH Universal Peripheral Interface 8-Bit Slave Microcontroller 6-122
8741A Universal Peripheral Interface 8-Bit Microcomputer 6-142
8742 Universal Peripheral Interface 8-Bit Slave Microcontroller 6-154
APPLICATION NOTE
AP-281 UPI-452 Accelerates iAPX 286 Bus Performance 6-166
SYSTEM SUPPORT
iUP-200A/iUP-201A Universal PROM Programmers 6-188

Alphanumeric Index

8206 Error Detection and Correction Unit 2-1
82064 CHMOS Winchester Disk Controller with On-Chip Error Detection and Correction .. 5-1
8207 Dual-Port Dynamic RAM Controller 2-23
82077AA CHMOS Single-Chip Floppy Disk Controller 1-624
82077AA CHMOS Single-Chip Floppy Disk Controller 1-508
82077AA CHMOS Single-Chip Floppy Disk Controller 4-1
82230/82231 High-Integration AT*-Compatible Chip Set 1-137
82303 Local I/O Support Chip 1-526
82304 Local I/O Support Chip 1-536
82307 DMA/Micro Channel Arbitration Controller 1-551
82308 Micro Channel Bus Controller 1-563
82309 Address Bus Controller 1-594
82311 Micro Channel* Compatible Peripheral Family 1-509
8231A Arithmetic Processing Unit 3-1
82335 SX High-Integration Interface Device for 386 SX Microprocessor Based PC-AT Systems 1-81
82340DX ISA Chip Set 1-8
82340SX ISA Chip Set 1-10
82341 ISA Peripheral Combo Chip 1-1
82343 System Controller/Data Buffer 1-12
82344 ISA Bus Controller 1-24
82345 Data Buffer 1-37
82346 System Controller 1-43
82347 Power Management PC Peripheral 1-55
82350 EISA Chip Set 1-174
82352 EISA Bus Buffer (EBB) 1-389
82355 Bus Master Interface Controller (BMIC) 1-424
82357 Integrated System Peripheral (ISP) 1-183
82358 EISA Bus Controller (EBC) 1-272
8237A High Performance Programmable DMA Controller (8237A, 8237A-4, 8237A-5) 3-14
82389 Message Passing Coprocessor A MULTIBUS II Bus Interface Controller 3-231
8253/8253-5 Programmable Interval Timer 3-51
8254 Programmable Interval Timer 3-62
8255A/8255A-5 Programmable Peripheral Interface 3-100
8256AH Multifunction Microprocessor Support Controller 3-147
8259A Programmable Interrupt Controller (8259A/8259A-2) 3-171
8279/8279-5 Programmable Keyboard/Display Interface 3-215
82C08 CHMOS Dynamic RAM Controller 2-70
82C37A-5 CHMOS High Performance Programmable DMA Controller 3-33
82C54 CHMOS Programmable Interval Timer 3-83
82C55A CHMOS Programmable Peripheral Interface 3-124
82C59A-2 CHMOS Programmable Interrupt Controller 3-195
8741A Universal Peripheral Interface 8-Bit Microcomputer 6-142
8742 Universal Peripheral Interface 8-Bit Slave Microcontroller 6-154
AP-167 Interfacing the 8207 Dynamic RAM Controller to the iAPX 186 2-99
AP-168 Interfacing the 8207 Advanced Dynamic RAM Controller to the iAPX 286 2-106
AP-281 UPI-452 Accelerates iAPX 286 Bus Performance 6-166
AP-402 Multimodule Winchester Controller Using the CHMOS 82064 5-33
iUP-200A/iUP-201A Universal PROM Programmers 6-188
Microprocessor Peripherals UPI-41A/41AH/42/42AH User's Manual 6-1
Overview xii
UPI-41AH/42AH Universal Peripheral Interface 8-Bit Slave Microcontroller 6-122
UPI-452 CHMOS Programmable I/O Processor 6-69

INTRODUCTION

Intel microprocessors and peripherals provide a complete solution in increasingly complex application environments. Quite often, a single peripheral device will replace anywhere from 20 to 100 TTL devices (and the associated design time that goes with them).

Built-in functions and standard Intel microprocessor/peripheral interface deliver very real *time* and *performance* advantages to the designer of microprocessor-based systems.

REDUCED TIME TO MARKET

When you can purchase an off-the-shelf solution that replaces a number of discrete devices, you're also replacing all the design, testing, and debug *time* that goes with them.

INCREASED RELIABILITY

At Intel, the rate of failure for devices is carefully tracked. Highest reliability is a tangible goal that translates to higher reliability for your product, reduced downtime, and reduced repair costs. And as more and more functions are integrated on a single VLSI device, the resulting system requires less power, produces less heat, and requires fewer mechanical connections — again resulting in greater system reliability.

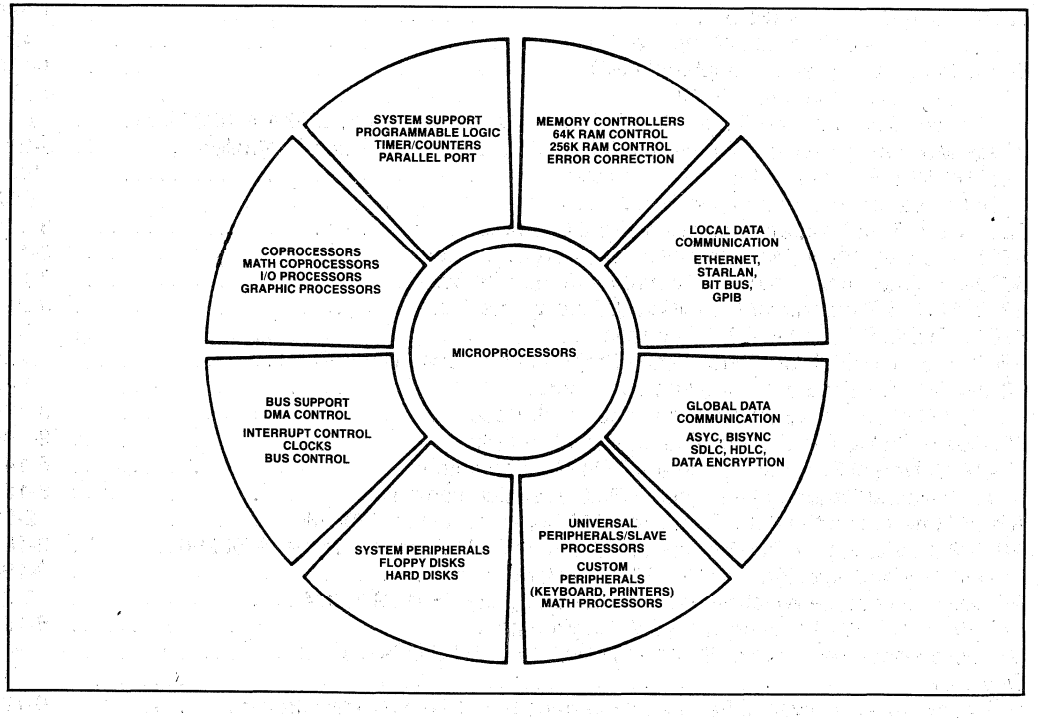
LOWER PRODUCTION COST

By minimizing design time, increasing reliability, and replacing numerous parts, microprocessor and peripheral solutions can contribute dramatically to lower product costs.

HIGHER SYSTEM PERFORMANCE

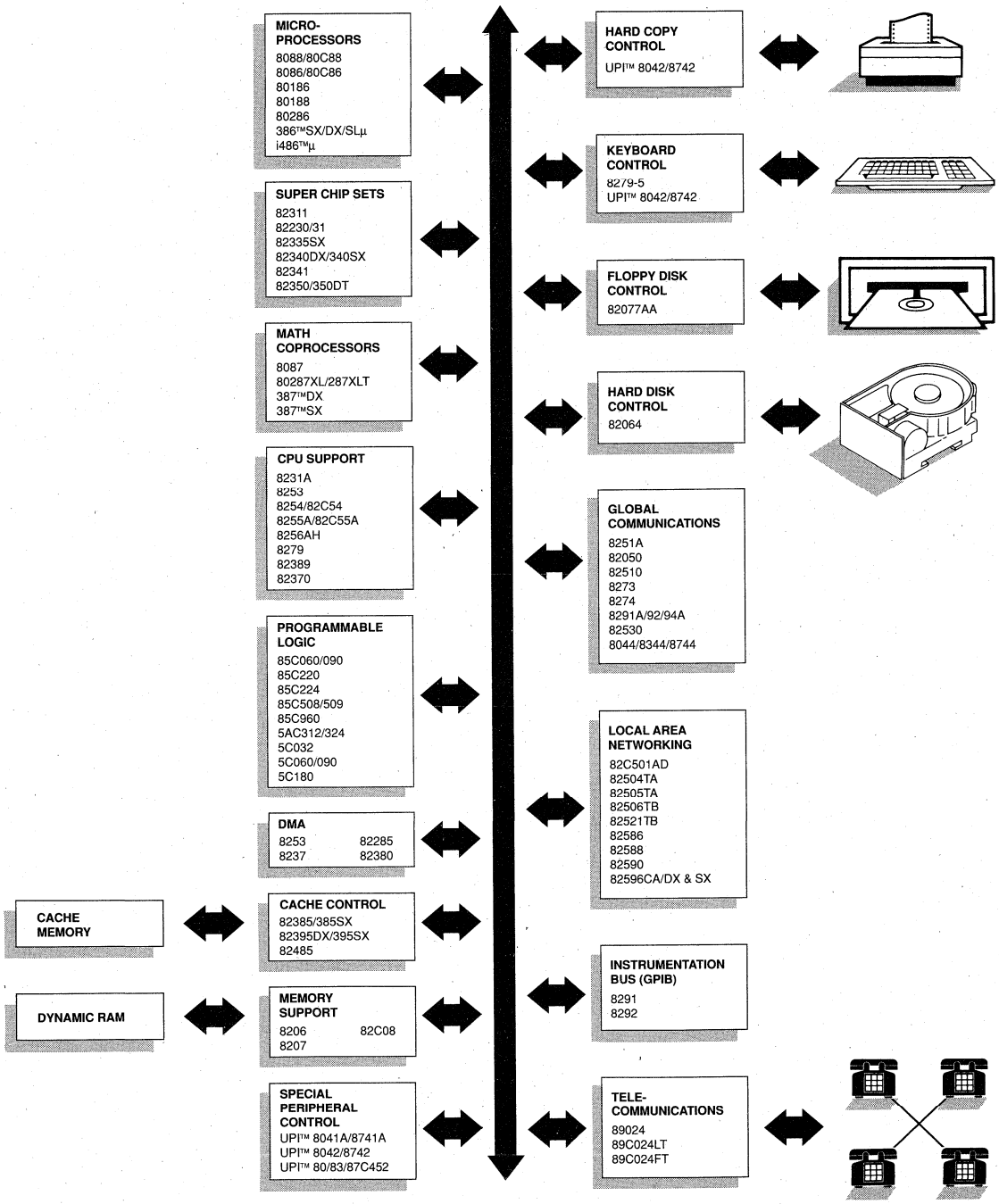
Intel microprocessors and peripherals provide the highest system performance for the demands of today's (and tomorrow's) microprocessor-based applications. For example, the Intel386/i486™ Microprocessor Family offers 32-bit performance for multitasking, multiuser systems. Intel's peripheral products have been designed with the future in mind. They support all of Intel's 8, 16 and 32-bit processors.

The Intel microprocessor and peripherals family provides a broad range of time-saving, high performance solutions.



Get Your Kit Together!

Intel's Microsystem Components Kit Solution



August 1990
Order Number: 230664-009

Chip Sets

1

82341
ISA PERIPHERAL COMBO CHIP

- **Combines the following PC/AT Peripheral Chips:**
  - **16C450 UART - COM1:**
  - **16C450 UART - COM2:**
  - **Parallel Printer Port - LPT1:**
  - **Keyboard/Mouse Ctrl. - KBD**
  - **Real Time Clock**
- **Serial Ports Fully 16C450 Compatible**
- **Bidirectional Line Printer Port**
- **Software Control of PS/2*-Compatible Enhancements (LPT Port, Mouse)**
- **CMOS Direct Drive of Centronics-Type Parallel Interface**
- **PC/AT- or PS/2-Compatible Keyboard and Mouse Controller**
- **146818A-Compatible Real Time Clock (RTC)**
- **16 Bytes of Additional Standby RAM (66 Bytes Total)**
- **IDE Bus Control Signals Included (Two External 74LS245 and One 74LS244—or Equivalent—Buffers are Required)**
- **Seven Battery-Backed Programmable Chip Select Registers for Auto Configuration**
- **Preprogrammed Default Chip Selects**
- **Programmable Wait State Generation**
- **5 μA Standby Current for RTC, RAM, and Chip Select Registers**
- **Single 128-Pin Plastic Quad Flatpack**

1

The 82341 Peripheral Combo chip replaces with a single 128-pin chip, several of the commonly used peripherals found in PC/AT-compatible computers. This chip when used with the Intel PC/AT-compatible chip set allows designers to implement a very cost effective, minimum chip count motherboard containing functions that are common to virtually all PCs.

The on-chip UARTs are completely software compatible with the 16C450 Asynchronous Communication Element.

The bidirectional parallel port provides a PS/2 software compatible interface between a Centronics-type printer and the 82341. Direct drive is provided so that all that is necessary to interface to the line printer port is a resistor-capacitor network. The bidirectional feature (option) is software programmable for backwards PC/AT-compatibility.

The keyboard/mouse controller is selectable as PC/AT-or PS/2-compatible.

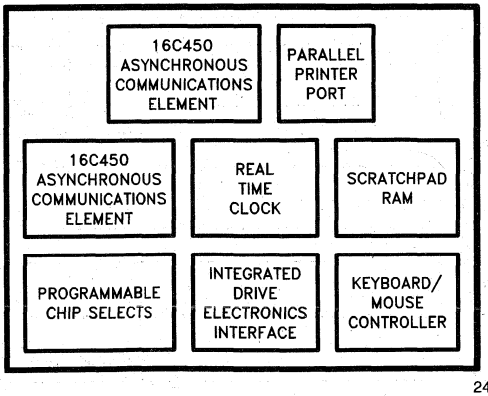
The Real Time Clock is 146818A-compatible and offers a standby current drain of 5 μA at 3.0V.

Ordering Information

Part Number	Package
82341	128-Pin Plastic Quad Flatpack

NOTE:
Operating temperature range is 0°C to +70°C.

Internal Functional Diagram



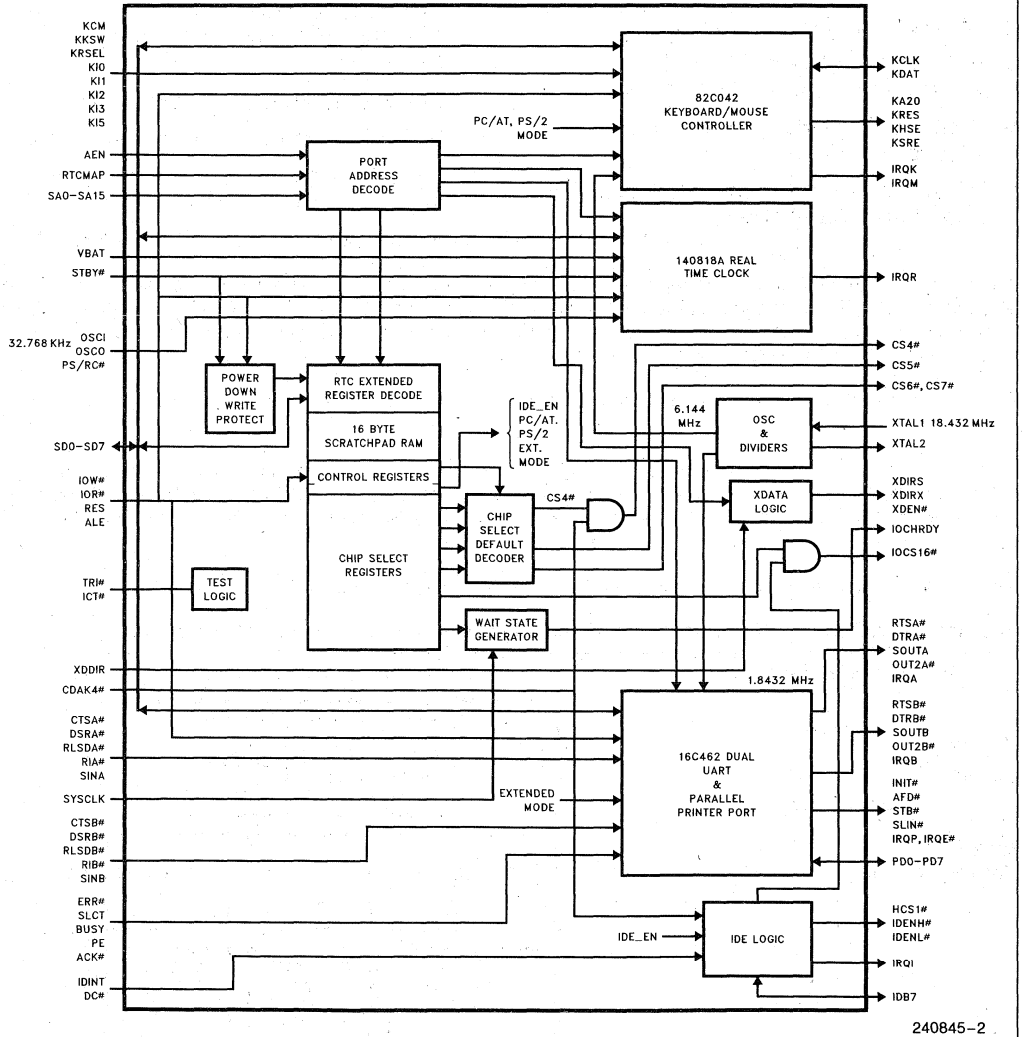
240845-1

*PC/AT® and PS/2® are registered trademarks of International Business Machines Corp.

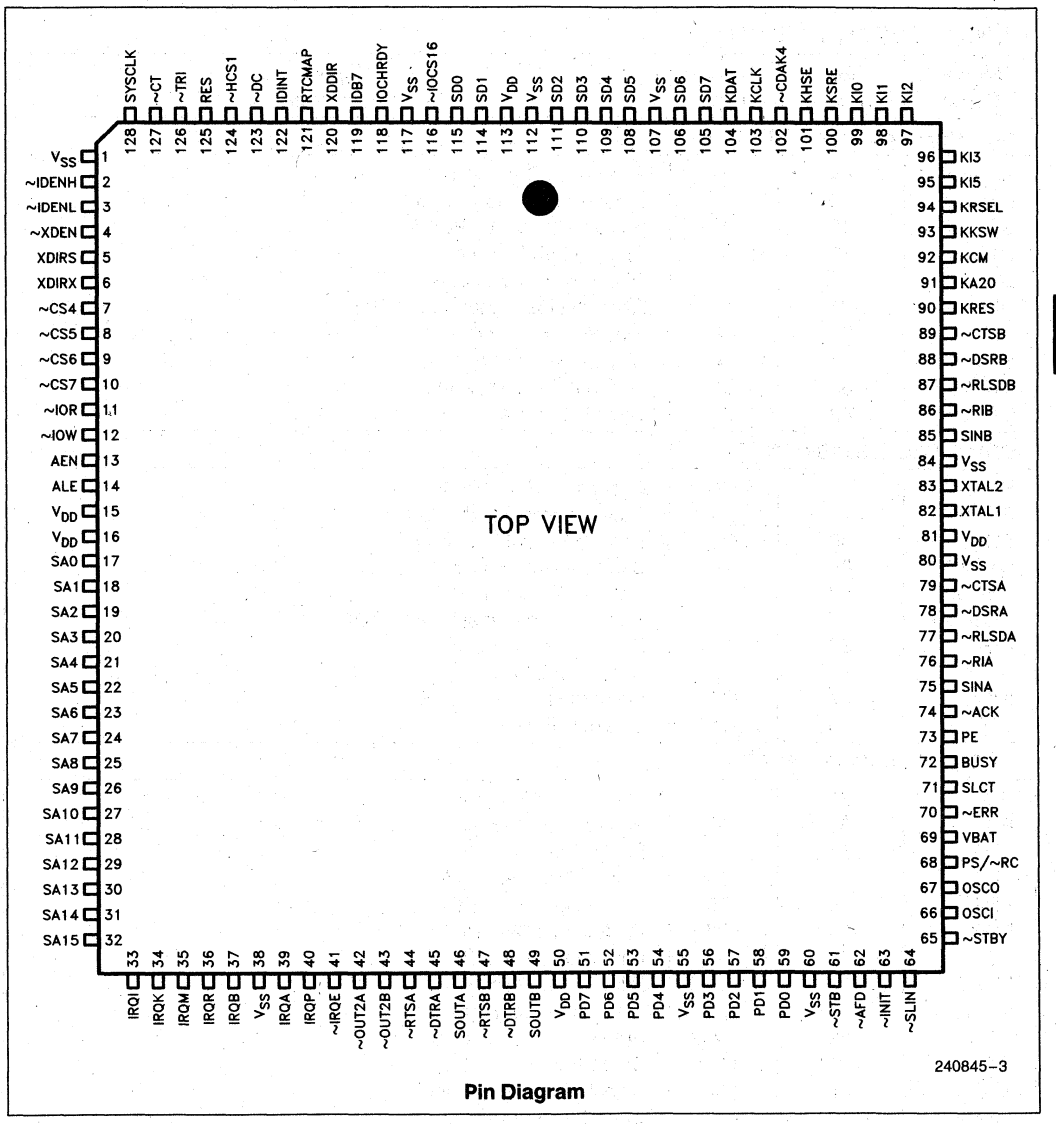
October 1990
Order Number: 240845-001

Included is the control logic necessary for the support of the Integrated Drive Electronics (IDE) hard disk bus interface.

The Peripheral Combo chip also includes seven programmable chip selects, three internal and four external. Each chip select has a programmable 16-bit base address and a mask register that allows the number of bytes corresponding to each chip select to be programmed (e.g. 3F8H–3FFH has a base address of 3F8H and a range of 8 bytes). Each chip select can be programmed for number of wait states (0–7) and 8- or 16-bit operation. 16-bit decoding is used for all I/O addresses. A default fixed decode is provided on reset for the on-chip serial ports, printer port, and off-chip floppy and hard disk controllers, which may be changed to battery-backed programmable chip selects via control bit.



Block Diagram


TOP VIEW
1 VSS
2 ~IDENH
3 ~IDENL
4 ~XDEN
5 XDIRS
6 XDIRX
7 ~CS4
8 ~CS5
9 ~CS6
10 ~CS7
11 ~IOR
12 ~IOW
13 AEN
14 ALE
15 VDD
16 VDD
17 SA0
18 SA1
19 SA2
20 SA3
21 SA4
22 SA5
23 SA6
24 SA7
25 SA8
26 SA9
27 SA10
28 SA11
29 SA12
30 SA13
31 SA14
32 SA15
33 IRQI
34 IRQK
35 IRQM
36 IRQR
37 IRQB
38 VSS
39 IRQA
40 IRQP
41 ~IRQE
42 ~OUT2A
43 ~OUT2B
44 ~RTSA
45 ~DTRA
46 SOUTA
47 ~RTSB
48 ~DTRB
49 SOUTB
50 VDD
51 PD7
52 PD6
53 PD5
54 PD4
55 VSS
56 PD3
57 PD2
58 PD1
59 PD0
60 VSS
61 ~STB
62 ~AFD
63 ~INIT
64 ~SLIN
65 ~STBY
66 OSCI
67 OSCO
68 PS/~RC
69 VBAT
70 ~ERR
71 SLCT
72 BUSY
73 PE
74 ~ACK
75 SINA
76 ~RIA
77 ~RLSDA
78 ~DSRA
79 ~CTSA
80 VSS
81 VDD
82 XTAL1
83 XTAL2
84 VSS
85 SINB
86 ~RIB
87 ~RLSDB
88 ~DSRB
89 ~CTSB
90 KRES
91 KA20
92 KCM
93 KKSW
94 KRSEL
95 KI5
96 KI3
97 KI2
98 KI1
99 KI0
100 KSRE
101 KHSE
102 ~CDAK4
103 KCLK
104 KDAT
105 SD7
106 SD6
107 VSS
108 SD5
109 SD4
110 SD3
111 SD2
112 VSS
113 VDD
114 SD1
115 SD0
116 ~IOCS16
117 VSS
118 IOCHRDY
119 IDB7
120 XDDIR
121 RTCMAP
122 IDINT
123 ~DC
124 ~HCS1
125 RES
126 ~TRI
127 ~CT
128 SYSCLK
240845-3

Pin Diagram

SIGNAL DESCRIPTIONS

Signal Name	Pin Number	Signal Type	Signal Description
COMMUNICATIONS PORT A			
RTSA#	44	O1	Request to Send, Port A
DTRA#	45	O1	Data Terminal Ready, Port A
SOUTA	46	O1	Serial Data Output, Port A
CTSA#	79	I4	Clear to Send, Port A
DSRA#	78	I4	Data Set Ready, Port A
RLSDA#	77	I4	Receive Line Signal Detect, Port A
RIA#	76	I4	Ring Indicator, Port A
SINA	75	I4	Serial Input, Port A
IRQA	39	O6	Interrupt Request, Port A
OUT2A#	42	O1	Output 2, Port A
COMMUNICATIONS PORT B			
RTSB#	47	O1	Request to Send, Port B
DTRB#	48	O1	Data Termina Ready, Port B
SOUTB	49	O1	Serial Data Output, Port B
CTSB#	89	I4	Clear to Send, Port B
DSRB#	88	I4	Data Set Ready, Port B
RLSDB#	87	I4	Receive Line Signal Detect, Port B
RIB#	86	I4	Ring Indicator, Port B
SINB	85	I4	Serial Input, Port B
IRQB	37	O6	Interrupt Request, Port B
OUT2B#	43	O1	Output 2, Port B
PARALLEL PRINTER PORT			
PD0	59	IO5	Printer Data Port, Bit 0
PD1	58	IO5	Printer Data Port, Bit 1
PD2	57	IO5	Printer Data Port, Bit 2
PD3	56	IO5	Printer Data Port, Bit 3
PD4	54	IO5	Printer Data Port, Bit 4
PD5	53	IO5	Printer Data Port, Bit 5
PD6	52	IO5	Printer Data Port, Bit 6
PD7	51	IO5	Printer Data Port, Bit 7
INIT#	63	O4	Initialize Printer Signal
AFD#	62	O4	Autofeed Printer Signal
STB#	61	O4	Data Strobe to Printer
SLIN#	64	O4	Select Signal from Printer
ERR#	70	I4	Error Signal from Priner
SLCT	71	I4	Select Signal from Printer

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
PARALLEL PRINTER PORT (Continued)			
BUSY	72	I4	Busy Signal from Printer
PE	73	I4	Paper Error Signal from Printer
ACK#	74	I4	Acknowledge Signal from Printer
IRQP	40	O6	Printer Interrupt Request Output
IRQE#	41	O1	Printer Interrupt Request Enable Signal
REAL TIME CLOCK PORT			
VBAT	69	NA	Standby Power—Normally 3V to 5V, battery backed
STBY#	65	15	Power Down Control
OSCI	66	NA	Crystal Connection Input—32 kHz
OSCO	67	NA	Crystal Connection Output—32 kHz
PS/RC#	68	I5	Power Sense/RAM Clear Input
IRQR	36	O1	Real Time Clock Interrupt Request Output
RTCMAP	121	I4	High—RTC is mapped to 70H and 71H, Low—RTC is mapped to 170H and 171H
KEYBOARD CONTROLLER PORT			
KCLK	103	IO4	Keyboard Clock
KDAT	104	IO4	Keyboard Data
KCM	92	I4	General Purpose Input, Normally Color/Monochrome
KKSW	93	I4	General Purpose Input, Normally Keyboard Switch
KA20	91	O1	General Purpose Output, Normally A20 Gate
KRES	90	O1	General Purpose Output, Normally Reset
KHSE	101	O1/IO4	General Purpose Input, Normally Speed Select
KSRE	100	O1/IO4	General Purpose Output, Normally Shadow RAM Enable
IRQK	34	O1	Keyboard Interrupt Request
IRQM	35	O1	Mouse Interrupt Request
KRSEL	94	I4	General Purpose Input, Normally RAM Select
K10	99	I4	General Purpose Input, Bit 0
K11	98	I4	General Purpose Input, Bit 1
K12	97	I4	General Purpose Input, Bit 2
K13	96	I4	General Purpose Input, Bit 3
K15	95	I4	General Purpose Input, Bit 5
IDE BUS I/O			
IDENH#	2	O1	IDE Bus Transceiver High Byte Enable
IDENL#	3	O1	IDE Bus Transceiver Low Byte Enable
IDINT	122	I4	IDE Bus Interrupt Request Input
IDB7	119	IO6	IDE Bus Data Bit 7
DC#	123	I4	Floppy Disk Change Signal
HCS1#	124	O1	IDE Host Chip Select 1
IRQI#	33	O6	IDE Interrupt Request Output

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
COMMON BUS I/O			
SD0	115	IO2	System Bus Data, Bit 0
SD1	114	IO2	System Bus Data, Bit 1
SD2	111	IO2	System Bus Data, Bit 2
SD3	110	IO2	System Bus Data, Bit 3
SD4	109	IO2	System Bus Data, Bit 4
SD5	108	IO2	System Bus Data, Bit 5
SD6	106	IO2	System Bus Data, Bit 6
SD7	105	IO2	System Bus Data, Bit 7
SA0	17	I1	System Bus Address, Bit 0
SA1	18	I1	System Bus Address, Bit 1
SA2	19	I1	System Bus Address, Bit 2
SA3	20	I1	System Bus Address, Bit 3
SA4	21	I1	System Bus Address, Bit 4
SA5	22	I1	System Bus Address, Bit 5
SA6	23	I1	System Bus Address, Bit 6
SA7	24	I1	System Bus Address, Bit 7
SA8	25	I1	System Bus Address, Bit 8
SA9	26	I1	System Bus Address, Bit 9
SA10	27	I1	System Bus Address, Bit 10
SA11	28	I1	System Bus Address, Bit 11
SA12	29	I1	System Bus Address, Bit 12
SA13	30	I1	System Bus Address, Bit 13
SA14	31	I1	System Bus Address, Bit 14
SA15	32	I1	System Bus Address, Bit 15
XTAL1	82	NA	Crystal Clock Input—18.432 MHz
XTAL2	83	NA	Crystal Clock Output—18.432 MHz
IOR#	11	I1	System Bus I/O Read
IOW#	12	I1	System Bus I/O Write
RES	125	I1	System Reset
AEN	13	I1	System Bus Address Enable
ALE	14	I1	System Bus Address Latch Enable
IOCS16#	116	O8	System Bus I/O Chip Select 16
IOCHRDY	118	O8	System Bus I/O Channel Ready
SYSCLK	128	I1	System Clock—Processor Clock Divide by 2
CS4#	7	O1	Chip Select 4—Normally for External Floppy Disk Controller
CS5#	8	O1	Chip Select 5—Normally HCS0# for IDE
CS6#	9	O1	Chip Select 6—Normally for External Floppy Disk Controller
CS7#	10	O1	Chip Select 7—Normally for External Floppy Disk Controller
CDAK4#	102	I1	DMA Acknowledge Forces —CS4 Active
XDDIR	120	I1	X Data Bus Transceiver Direction
XDIRS	5	O1	Modified X Data Bus Transceiver Direction Control Signal—Excludes Real Time Clock and Keyboard Controller Decodes
XDIRX	6	O1	X Data Bus Transceiver Control Signal—Includes All CS Decodes Generated On Chip

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
COMMON BUS I/O (Continued)			
XDEN#	4	O1	X Data Bus Transceiver Enable
TRI#	126	I4	Three-State Control Input—For All Outputs to Isolate Chip for Board Tests
ICT#	127	I4	In Circuit Test Mode Control
POWER, GROUND, AND UNCOMMITTED			
VDD	15, 16, 50, 81, 113		System Power: +5V
VSS	1, 38, 55, 60, 80, 84, 107, 112, 117		System Ground

I/O LEGEND

Pin Type	mA	Type	Comment
01	2	TTL	
02	24	TTL	
04	12	TTL-OD	Open Drain, Weak Pull-Up, No VDD Diode
06	4	TTL-TS	Three-State
07	24	TTL-TS	Three-State
08	24	TTL-OD	Open Drain, Fast Active Pull-Up
I1	—	TTL	
I2	—	CMOS	
I4	—	TTL	30 kΩ Pull-Up
I5	—	TTL	Schmitt-Trigger
I02	24	TTL-TS	Three-State
I04	12	TTL-OD	Open Drain, Slow Turn-On
I05	12	TTL-TS	Three-State
I06	24	TTL-TS	Three-State, 30 kΩ Pull-Up

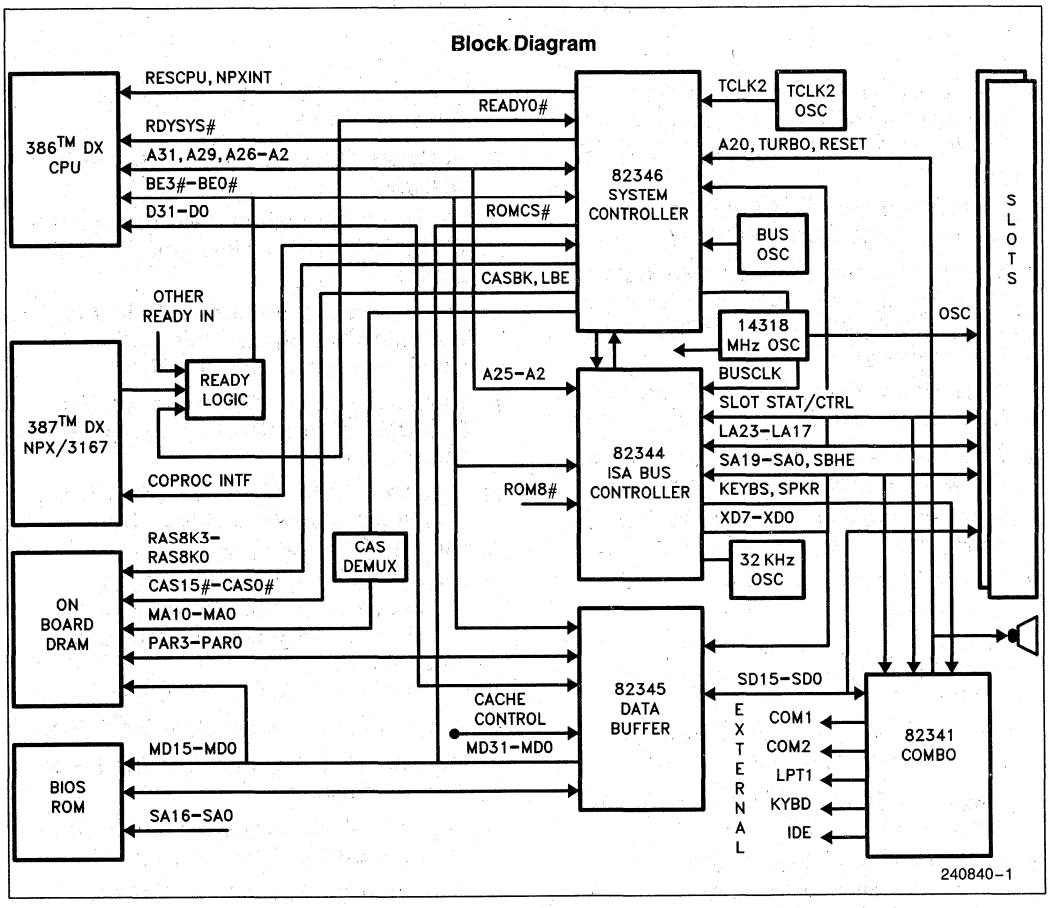
82340DX ISA CHIP SET

- Three Chip ISA (Industry Standard Architecture) Chip Set Capable of Use in 386™ DX-Based System up to 33 MHz
  - — 82346 System Controller,
  - — 82344 ISA Bus Controller,
  - — 82345 Data Buffer
- Two 128-Pin and One 160-Pin (82344) Quad Flatpacks, 1.0- and 1.5-Micron CMOS
- Memory Control of One to Four Banks of 32-Bit DRAM Using 256k, 1M, or 4M Components Allowing 64 Mbytes on System Board
- Page Mode DRAM Operation on Any Number of Banks
- Two/Four-Way Interleaving or Direct Access on System Board Memory
- Programmable Option for Block or Word Interleave
- Programmable DRAM Timing Parameters
- Remap Option Allows Logical Reordering of System Board DRAM Banks
- System Board Refresh Optionally Decoupled from Slot Bus Refresh
- Staggered Refresh Minimizes Power Supply Load Variations
- Built-In "Sleep" Mode Features, Including Use of Slow Refresh DRAMs in Power Critical Operations
- Hardware Supports Full LIM EMS 4.0 Spec over Entire 64 Mbyte Memory Map
- DMA Expanded to Allow Transfers over 64M range
- Shadow RAM Support in 16k Increments over Entire 64k to 1M range
- Support for 387™ DX
- Coprocessor Software Reset Can Be Disabled
- Internal Switching and Programmable CLK2 Support for *PC/AT-Compatible and "Turbo" Modes
- Programmable Drive Reduces the Need for External Buffering on DRAM and Slot Bus Interface Signals
- ISA Bus Control of 386 DX-Based PC/AT-Compatibles. Capable of Asynchronous or Synchronous Bus Operation to 16 MHz
- Compatible with *Lotus 1-2-3 Version 3.0 in 1M Systems
- Bus "Quiet" Mode Assures That Slot Bus Signal Lines Are Driven Only During Slot Accesses
- Integrated Peripheral Functions:
  - — Two 82C37A DMA Controllers with Extended 74LS612 Page Register
  - — Two 82C59A Interrupt Controllers
  - — One 82C54 Timer
  - — One 146818 Real Time Clock
- Additional 64 Bytes of Battery Backed RAM in RTC Provides for Non-Volatile Storage of 82340DX Chip Set Configuration Data and User Specific Information
- Supports 8- or 16-Bit Wide BIOS ROMs
- Cache Support for Posted Writes
- System Memory on MD or D Bus in Non-Cached Systems
- Separate Parity Generation/Checkers for High Speed Operation
- Internal I/O Programmable for 10- or 16-Bit Decode
- Three-State Control Pins Added for Board Level Testability

386 and 387 are trademarks of Intel Corporation.
PC/AT™ is a trademark of International Business Machines Corporation.
*Lotus 1-2-3® is a registered trademark of Lotus Development Corporation.

October 1990
Order Number: 240840-001

Block Diagram



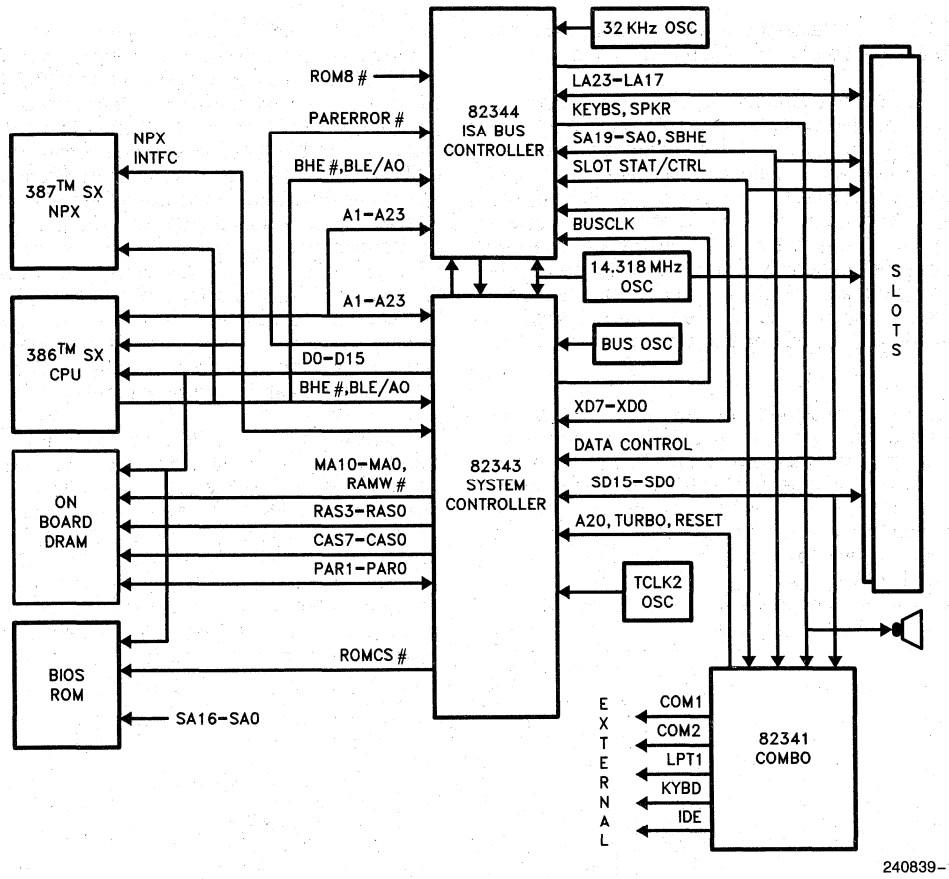
82340SX ISA CHIP SET

- Two Chip ISA (Industry Standard Architecture) Chip Set Capable of Use in 386™ SX-Based Systems Up to 20 MHz
- Both Chips are 160 Quad Flatpacks, 1.0- and 1.5-Micron CMOS
- Memory Control of One to Four Banks of 16-Bit DRAM Using 256k, 1M, or 4M Components Allowing 32 Mbytes on System Board
- Page Mode DRAM Operation on Any Number of Banks
- Two/Four-Way Interleaving or Direct Access on System Board Memory
- Programmable Option for Block or Word Interleave
- Programmable DRAM Timing Parameters
- Remap Option Allows Logical Reordering of System Board DRAM Banks
- System Board Refresh Optionally Decoupled from Slot Bus Refresh
- Staggered Refresh Minimizes Power Supply Load Variations
- Built-In "Sleep" Mode Features, Including Use of Slow Refresh DRAMs in Power Critical Operations
- EMS Hardware Supports Full LIM EMS 4.0 Spec over Entire 32 Mbyte Memory Map with Backfill to 256k—Includes Two Sets of 36 Mapping Registers Each
- Shadow RAM Support in 16k Increments over Entire 640k to 1M range
- Support for 387™ SX Numerical Coprocessors
- Software Coprocessor Reset can be Disabled
- Internal Switching and Programmable CLK2 Support for Slow and "Turbo" Modes
- Programmable Drive on DRAM and Slot Bus Interface Signals Allows Direct Drive Tailored to System Size
- Asynchronous or Synchronous Slot Bus Operation with Programmable bus Clock Divider
- Bus "Quiet" Mode Assures that Slot Bus Signal Lines are Driven Only During Slot Accesses
- Integrated Peripheral Functions:
  - Two 82C37A DMA Controllers
  - Two 82C59A Interrupt Controllers
  - One 82C54 Timer
  - One 146818 Real Time Clock
- Supports 8- or 16-Bit Wide BIOS ROMs
- I/O Decode Programmable for 10- or 16-Bit Addresses
- Separate Parity Generators/Checkers for High Speed Operation
- Designed for Systems with Up to 12 MHz Backplane Operation
- Three-State Control Pins Added for Board Level Testability
- Compatible with Lotus 1-2-3* Version 3.0 in 1M Systems

386™ and 387™ are registered trademarks of Intel Corporation.
*Lotus 1-2-3® is a registered trademark of Lotus Development Corporation.

October 1990
Order Number: 240839-001

Block Diagram



8-bit ROM is on XD bus

82343
SYSTEM CONTROLLER/DATA BUFFER

The 82343 contains the system control and data buffering functions in a 160-pin quad flatpack.

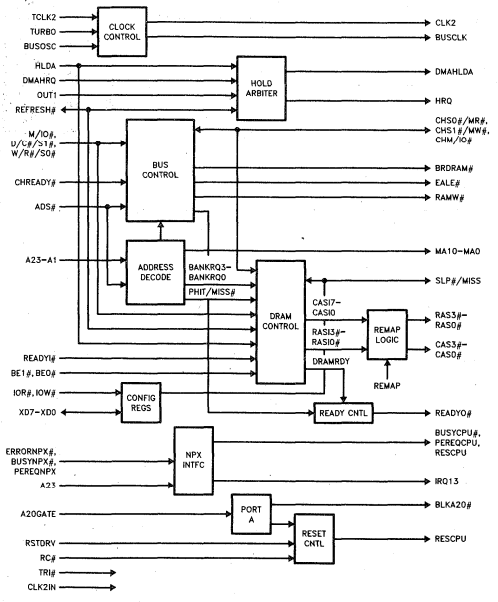
The 82343's functions are highly programmable via a set of internal configuration registers. Defaults on reset for the configuration registers mimic the compatibility requirements of the original IBM PC/AT* as closely as possible. The power-up defaults allow any possible configuration of the system to boot at the CPU's rated speed. However, operational capabilities may be temporarily reduced until the configuration registers are set to mirror the true system configuration. This normally occurs during BIOS power-on self-test in a manner completely transparent to the user.

The 82343 is designed to perform in systems running up to 20 MHz. Built-in page-mode operation, two- or four-way interleaving, and fully programmable memory timing allow the PC designer to maximize system performance using low cost DRAMs. Programmable memory timing allows the system to be setup to perfectly match the requirements of the chosen DRAMs; standard or custom. These adjustments can often be made without incurring the penalty of additional wait states.

The system controller handles system board refresh directly and also controls the timing of slot bus refresh which is actually performed by the 82344 ISA Bus Controller. Refresh may be performed in coupled or decoupled mode. The former method is the standard PC/AT-compatible mode where on- and off-board refreshes are performed synchronously. In decoupled mode, the timing of on- and off-board refreshes is independent. Both may be programmed for independent, slower than normal rates. This allows use of low power, slow refresh DRAMs. The 82343 controls all timing in both modes. In all cases, refreshes are staggered to minimize power supply loading and attendant noise on the V_{DD} and ground pins. In sleep mode, refresh switches to CAS# before RAS# refresh for maximum power savings.

The physical banks of DRAM can be logically reordered through one of the indexed configuration registers. This DRAM remap option is useful in order to map out bad DRAM banks allowing continued use of a system until repairs are convenient. It also allows DRAM bank combinations not in the supported memory maps to be logically moved into a supported configuration without physically moving memory components. This unique, programmable function

Block Diagram



240841-1

386™ is a trademark of Intel Corporation.
*IBM PC/AT® is a registered trademark of International Business Machines Corporation.

October 1990
Order Number: 240841-001

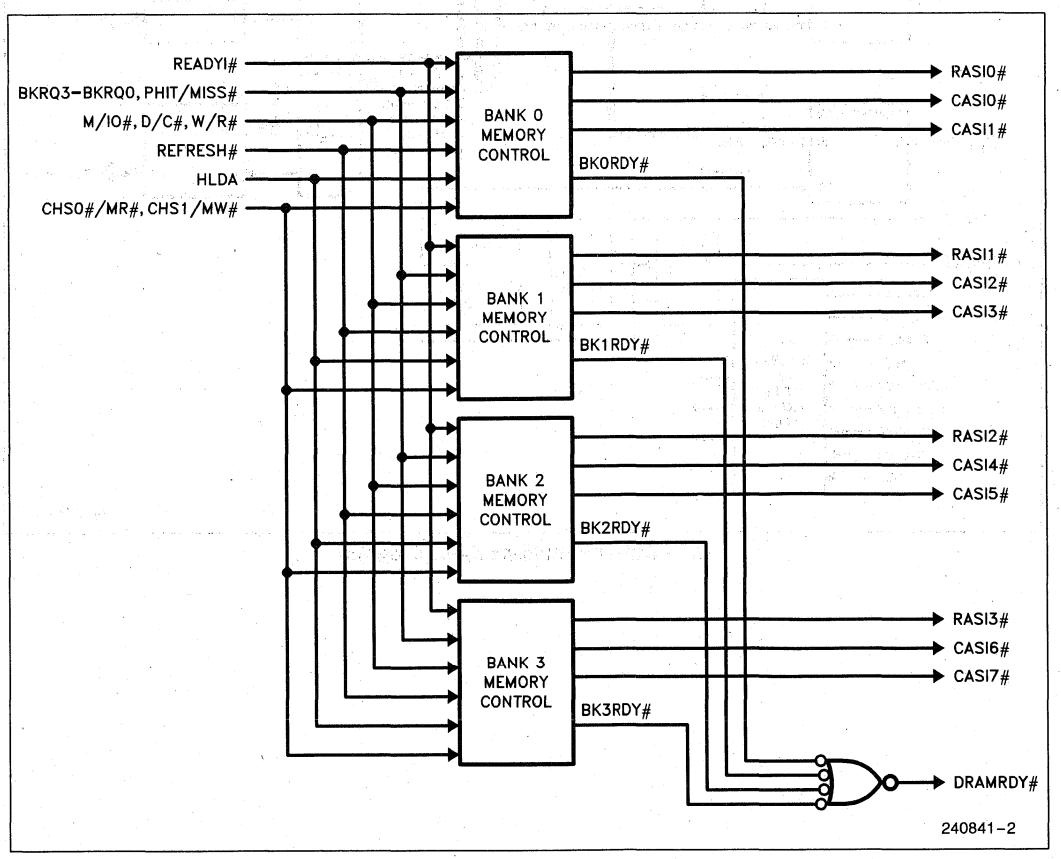
performs this task by switching the internal RAS# and CAS# signals between the external RAS# and CAS# pins. This allows internal row and column addresses generated for DRAM bank 0, for example, to be routed to any one of the four on-board DRAM banks.

Full EEMS support is provided in hardware for the complete LIM EMS 4.0 standard. Seventy-two mapping registers provide a standard and an alternate set of 36 registers each. The system allows backfill down to 256k for EEMS support and provides 24 mapping registers covering this space. Twelve of the 36 are page registers which cover the EMS space from C0000h to E0000h. These twelve registers can alternatively be mapped in the A0000–BFFFFh and D0000–DFFFFh range by changing a configuration bit in the 82343. All registers are capable of translating over the complete 32 Mbyte range of on-board DRAM. Users preferring an alternate plug-in EMS solution, can disable the on-board EMS system as well as system board DRAM, as required, down to 256k.

Shadowing features are supported on all 16k boundaries between 640k and 1M. Simultaneous EMS use, shadowed ROM, and direct system board access is possible in non-overlapping fashion throughout this memory space. Control over four access options is provided.

1. Access ROM or slot bus for reads and writes.
2. Access system board DRAM for reads and writes.
3. Access system board DRAM for reads and slot bus for writes.
4. Shadow setup mode. Read ROM or slot bus, write system board DRAM.

These controls are overridden by EMS in segments for which it is enabled.

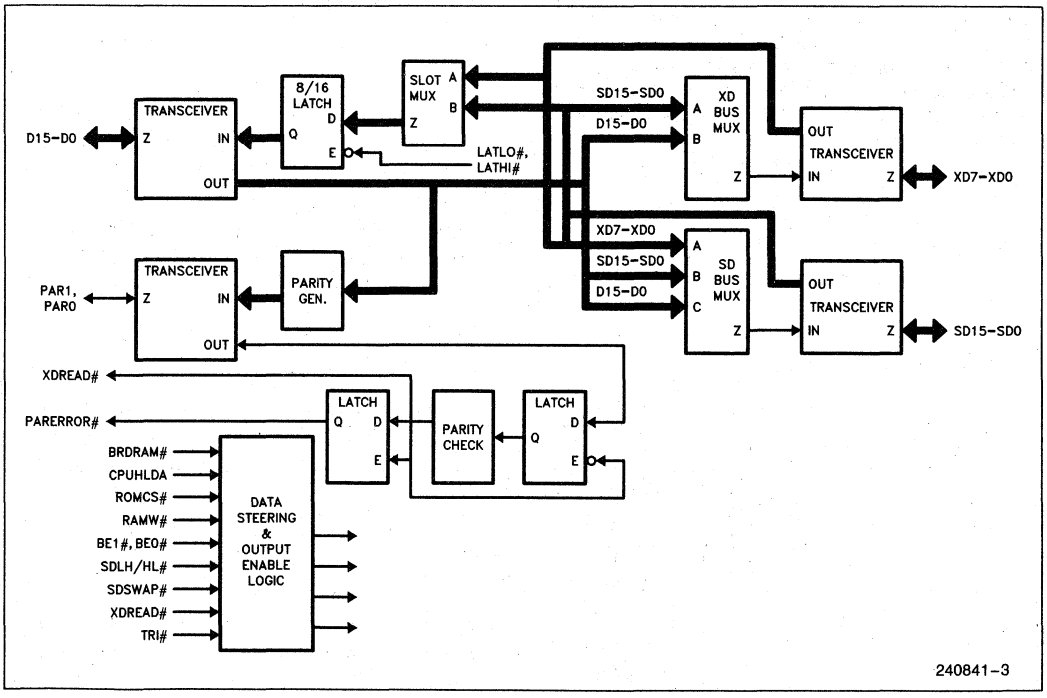


Memory Controller Block Diagram

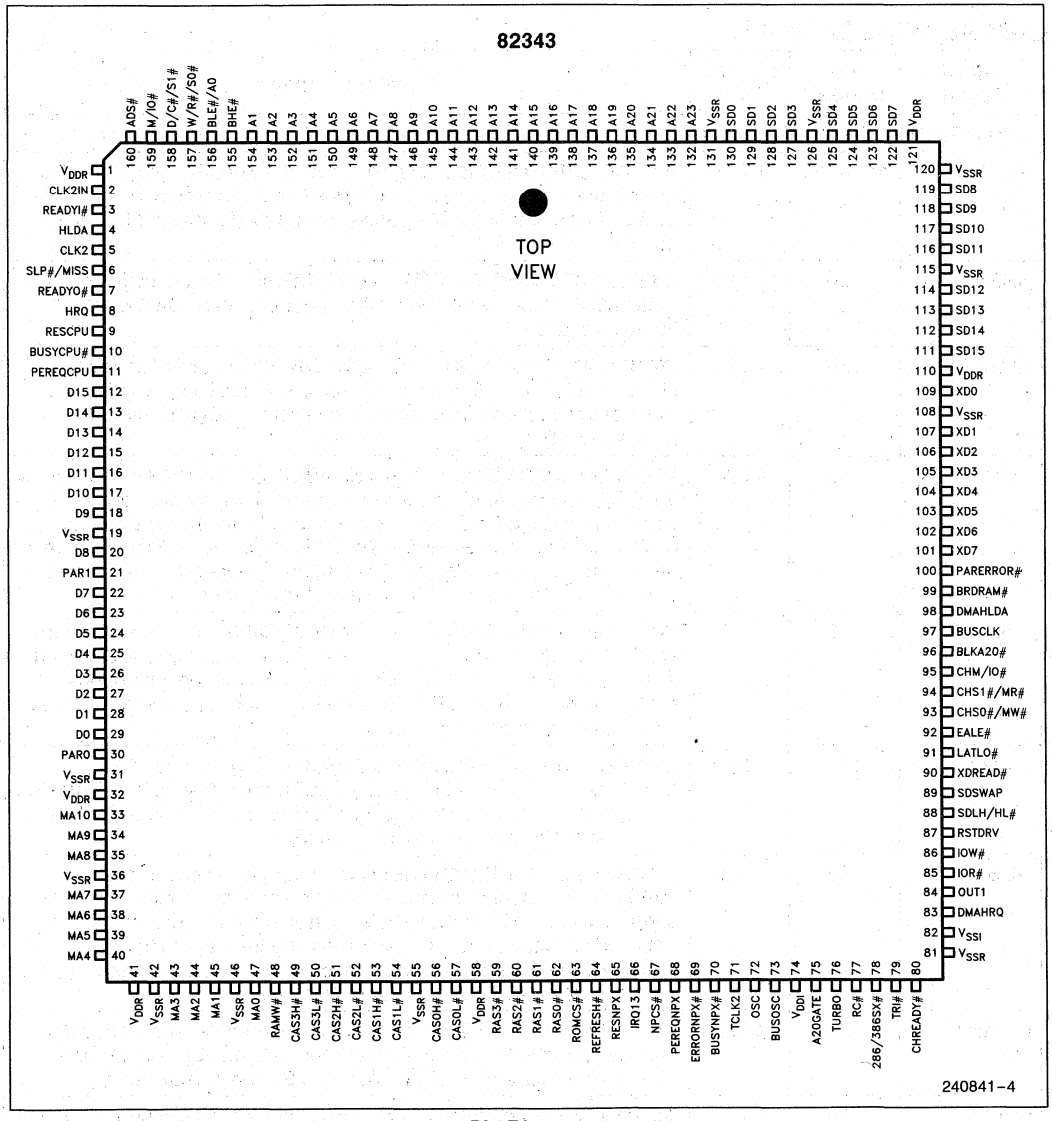
The System Controller is used to program the desired operational mode of the AT bus. Based on this programming, it provides the bus clock and signalling interface to the Bus Controller. The bus may run synchronously with the CPU's CLK2 or asynchronously via an external oscillator. A programmable divider conditions the selected BUSCLK source providing divide by 1, 2, 3 or 4.

The 82343 also performs all of the data buffering functions required for a 386™SX-based PC/AT-compatible system. Under the control of the CPU, the data buffer chip routes data to and from the CPU's D bus, XD bus, and slots (SD bus). The parity is checked for D bus DRAM read operations. When reading from ROM, the XD bus or the SD bus, the data can be converted from 8 bits wide to 16. The data is latched for synchronization with the CPU. Parity is generated for all data written to the D bus. The 82343 provides the data conversion necessary for 16-bit writes to 8-bit devices on the XD or SD buses.

Under the control of DMA or a bus master, the 82343 allows 8- or 16-bit data to be routed to and from the XD bus. The chip also is capable of performing high to low and low to high byte swaps on the SD bus. For transfers between two peripherals on the slot bus the outputs of the 82343 are disabled. The chip also provides the feature of a single input, TRI#, to disable all of its outputs for board level testability.



Data Buffer Functions Block Diagram


82343
TOP
VIEW
1 VDDR
2 CLK2IN
3 READYI#
4 HLDA
5 CLK2
6 SLP#/MISS
7 READYO#
8 HRQ
9 RESCPU
10 BUSYCPU#
11 PEREQCPU
12 D15
13 D14
14 D13
15 D12
16 D11
17 D10
18 D9
19 VSSR
20 D8
21 PAR1
22 D7
23 D6
24 D5
25 D4
26 D3
27 D2
28 D1
29 D0
30 PAR0
31 VSSR
32 VDDR
33 MA10
34 MA9
35 MA8
36 VSSR
37 MA7
38 MA6
39 MA5
40 MA4
41 VDDR
42 VSSR
43 MA3
44 MA2
45 MA1
46 VSSR
47 MA0
48 RAMW#
49 CAS3H#
50 CAS3L#
51 CAS2H#
52 CAS2L#
53 CAS1H#
54 CAS1L#
55 VSSR
56 CAS0H#
57 CAS0L#
58 VDDR
59 RAS3#
60 RAS2#
61 RAS1#
62 RAS0#
63 ROMCS#
64 REFRESH#
65 RESNPX
66 IRQ13
67 NPCS#
68 PEREQNPX
69 ERRORNPX#
70 BUSYNPX#
71 TCLK2
72 OSC
73 BUSOSC
74 VDDI
75 A20GATE
76 TURBO
77 RC#
78 286/386SX#
79 TRI#
80 CHREADY#
81 VSSR
82 VSSI
83 DMAHRQ
84 OUT1
85 IOR#
86 IOW#
87 RSTDRV
88 SDLH/HL#
89 SDSWAP
90 XDREAD#
91 LATLO#
92 EALE#
93 CHS0#/MW#
94 CHS1#/MR#
95 CHM/IO#
96 BLKA20#
97 BUSCLK
98 DMAHLDA
99 BRDRAM#
100 PARERROR#
101 XD7
102 XD6
103 XD5
104 XD4
105 XD3
106 XD2
107 XD1
108 VSSR
109 XD0
110 VDDR
111 SD15
112 SD14
113 SD13
114 SD12
115 VSSR
116 SD11
117 SD10
118 SD9
119 SD8
120 VSSR
121 VDDR
122 SD7
123 SD6
124 SD5
125 SD4
126 VSSR
127 SD3
128 SD2
129 SD1
130 SD0
131 VSSR
132 A23
133 A22
134 A21
135 A20
136 A19
137 A18
138 A17
139 A16
140 A15
141 A14
142 A13
143 A12
144 A11
145 A10
146 A9
147 A8
148 A7
149 A6
150 A5
151 A4
152 A3
153 A2
154 A1
155 BHE#
156 BLE#/A0
157 W/R#/S0#
158 D/C#/S1#
159 M/IO#
160 ADS#
240841-4

Pin Diagram

SIGNAL DESCRIPTIONS

Signal Name	Pin Number	Signal Type	Signal Description
CPU INTERFACE SIGNALS			
A23–1	132–154	I-TTL	**ADDRESS BITS 23 THROUGH 1:** Driven by the CPU when the CPU is bus master. They are driven by the Bus Controller whenever HLDA is active. These bits allow direct access of up to 16 Mbytes of local memory.
BHE#	155	I-TTL	**BYTE HIGH ENABLE, ACTIVE LOW:** This signal is driven by the CPU or the 82344. It is used to select the upper byte of a 16-bit wide memory location.
BLE#/A0	156	I-TTL	**BYTE LOW ENABLE, ACTIVE LOW, OR A0:** In 386SX mode this signal is BLE#, in 286 mode it is A0 and is driven by the CPU or the Bus Controller. It is used to select the lower byte of a 16-bit wide memory location.
W/R#/S0#	157	I-TPU	**WRITE OR ACTIVE LOW READ ENABLE, OR S0:** W/R# is driven by the CPU in 386SX mode, S0# in 286 mode. This signal is decoded with the remaining CPU control signals to indicate the type of bus cycle requested. The bus cycle types include interrupt acknowledge, halt shutdown, I/O reads and writes, memory data reads and writes, and memory code reads. This pin is pulled up internally.
D/C#/S1#	158	I-TPU	**DATA OR ACTIVE LOW CODE ENABLE, OR S1:** D/C# is driven by the CPU in 386SX mode or S1# in 286 mode. This signal is decoded with the remaining CPU control signals to indicate the type of bus cycle requested. See W/R#/S0# definition for bus cycle types. This pin is pulled up internally.
M/I#/O	159	I-TPU	**MEMORY OR ACTIVE LOW I/O ENABLE:** Driven by the CPU, M/IO# is decoded with the remaining CPU control signals to indicate the type of bus cycle requested. See W/R#/S0# definition for bus cycle types. This pin is pulled up internally.
ADS#	160	I-TPU	**ADDRESS STROBE, ACTIVE LOW:** This signal is driven by the 386SX as an indicator that the address and control signals currently supplied by the CPU are valid. It is used internally to indicate that the data and command are valid and determine the beginning of a memory cycle. This pin is a no connect in 286 mode. An internal pull up resistor keeps it inactive during 286 mode.
CLK2IN	2	I-CMOS	This is the main clock input to the System Controller and it should be connected to the CLK2 signal that is output by the System Controller. This signal is used internally to clock the System Controller logic.
TCLK2	71	I-TTL	This input is connected to a crystal oscillator whose frequency is equal to two times the system frequency. The TTL level oscillator output is converted internally to CMOS levels and sent to the CLK2 output.
CLK2	5	O	This output signal is CMOS level converted TCLK2 signal. It is output to the CPU and other on-board logic for synchronization.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
CPU INTERFACE SIGNALS (Continued)			
SLP#/MISS	6	IO – od	As a "power on reset" default this bit is an output that reflects the inverse state of the SLEEP[7] configuration register bit. It is active low when sleep mode is active. Sleep mode is activated by setting SLEEP[7] = 1. When configuration register CTRL1[0] = 1 this pin becomes a MISS output for use with a future product.
READYO#	7	O	**READY OUT, ACTIVE LOW:** This signal is an indication that the current memory or I/O bus cycle is complete. It is generated from the internal DRAM controller or the synchronized version of CHREADY# for slot bus accesses. Outside the chip it is ORed with any other local bus I/O or master. The culmination of these ORed READY signals is sent to the CPU and is also connected to the System Controller's READYI# input.
READYI#	3	I-TTL	**READY INPUT, ACTIVE LOW:** This signal is the ORed READY signals from the coprocessor or other optional add-in devices and from the 82343's READY# input.
HLDA	4	I-TTL	**HOLD ACKNOWLEDGE, ACTIVE HIGH:** This signal is issued by the CPU in response to the HRQ driven by the System Controller. It indicates that the CPU is floating its outputs to the high impedance state, so that another master can take control of the bus. When HLDA is active, the memory control is generated from CHS1#/MR# and CHS0#/MW# rather than CPU status signals.
HRQ	8	O	**HOLD REQUEST, ACTIVE HIGH:** This output is driven by the System Controller to the CPU. It indicates that a master, such as a DMA or AT channel master, is requesting control of the bus. HRQ is a result of the DMAHRQ input or a coupled refresh cycle. It is synched to CLK2.
RESCPU	9	O	**RESET CPU, ACTIVE HIGH:** An output signal that is sent to the CPU by the System Controller. It is issued in response to the control bit for software reset located in the Port A register or a dummy read to IO port EFh. It is also issued in response to signals on the RSTDRV or RC inputs and in response to System Controller's detection of a shutdown command. In all cases it is synched to CLK2.
BUSYCPU#	10	O	**BUSY CPU, ACTIVE LOW:** An output signal that is sent to the CPU. The state of BUSYNPX# is always passed through to BUSYCPU# indicating that the NPX is processing a command. ON occurrence of an ERRORNPX# signal, it is latched and held active until occurrence of a write to ports F0h, F1h, or RESNPX. The former case is the normal mechanism used to reset the active latched signal. The latter two are resets. Since ERRORNPX# generates IRQ13 for PC/AT-compatibility, BUSYCPU# is held active to prevent software access of the coprocessor until the interrupt service routine writes F0h.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
CPU INTERFACE SIGNALS (Continued)			
PEREQCPU	11	O	**PROCESSOR EXTENSION REQUEST, ACTIVE HIGH:** An output signal sent to the CPU in response to a PEREQNPX which is issued by the coprocessor to the System Controller. It indicates to the CPU that the coprocessor is requesting a data operand to be sent to or from memory by the CPU. For PC/AT-compatibility PEREQCPU is returned active on occurrence of ERRORNPX# after BUSYNPX# has gone inactive. A write to F0h by the interrupt 13 handler returns control of the PEREQCPU signal to directly follow the PEREQNPX input.
ON-BOARD MEMORY SYSTEM INTERFACE SIGNALS			
RAMW#	48	O	**RAM ACTIVE LOW WRITE/ ACTIVE HIGH READ:** This output to the DRAM memory is to control the direction of data flow of the on-board memory. It is a result of the address and bus control decode. It is active during memory write cycles and high at all other times.
MA10–MA0	33–35 37–40, 43–45, 47	O	**MEMORY ADDRESSES 10 THROUGH 0:** These address bits are the row and column addresses sent to on-board memory. They are buffered and multiplexed versions of the CPU bus addresses. They allow addressing of up to 8 Mbytes per DRAM bank.
RASBK3# – RASBK0#	59–62	O	**ROW ADDRESS BANK 0 THROUGH 3, ACTIVE LOW:** These signals are sent to their respective RAM banks to strobe in the row address during on-board memory bus cycles. The active period for this signal is fully programmable.
CAS7# – CAS0#	49–54 56, 57	O	**COLUMN ADDRESS, STROBE, ACTIVE LOW:** These signals are sent to their respective RAM banks to strobe in the column address during on-board memory bus cycles. There is a CAS# signal for upper and lower bytes of each of the four 16-bit DRAM memory banks. The active period for this signal is completely programmable. For clarity, alternate names may also be used for these signals as shown in the following table where the digit in the "Alternate Name" indicates the DRAM bank the signal drives, L indicates it drives the low byte, and H indicates it drives the high byte. **Standard Name** / **Alternate Name** CAS0# / CAS0L# CAS1# / CAS0H# CAS2# / CAS1L# CAS3# / CAS1H# CAS4# / CAS2L# CAS5# / CAS2H# CAS6# / CAS3L# CAS7# / CAS3H#

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
ON-BOARD MEMORY SYSTEM INTERFACE SIGNALS (Continued)			
REFRESH#	64	I-CMOS/ O-OD	**REFRESH SIGNAL, ACTIVE LOW:** This output is used by the System Controller to initiate an off-board DRAM refresh operation in coupled refresh mode. In decoupled mode, the Bus Controller drives refresh active to indicate to the System Controller that it has decoded a refresh request command and is initiating an off-board refresh cycle.
ROMCS#	63	O	**ROM CHIP SELECT:** This output is active in CPU mode only (CPUHLDA is negated). It is active anytime the address on the A bus selects the address range between FF0000h–FFFFFFh during a memory read cycle. It is also active 0E0000h–0FFFFFh and between FE0000–FEFFFF when RAMMAP[7] = 1, the default condition. When RAMMAP[7] = 0, accesses in these two regions are directed to the slot bus. "Holes" can be opened in this lower decode range by activation of shadow or EMS windows in this region; i.e., when 16k shadow or EMS windows are activated in this region, ROMCS# is not generated.
COPROCESSOR SIGNALS			
PEREQNPX	68	I-TPD	**COPROCESSOR EXTENSION REQUEST, ACTIVE HIGH:** This input signal is driven by the coprocessor and indicates that it needs transfer of data operands to or from memory. For PC/AT-compatibility this signal is also gated with the internal ERROR/ BUSY control logic before being output to the CPU as PEREQCPU during NPX interrupts.
ERRORNPX#	69	I-TPU	**ERROR COPROCESSOR, ACTIVE LOW:** An input signal from the coprocessor indicating that an error has occurred in the previous instruction. This signal is internally gated and latched with BUSYNPX# to produce IRQ13.
BUSYNPX#	70	I-TPU	**BUSY COPROCESSOR, ACTIVE LOW:** An input signal that is driven by the coprocessor to indicate that it is currently executing a previous instruction and is not ready to accept another. This signal is decoded internally to produce IRQ13 and to control PEREQCPU.
RESNPX	65	O	**RESET COPROCESSOR:** This output is connected to the coprocessor reset input. It is triggered through an internally generated system reset or via a write to port F1h. In the case of a system reset, the CPURESET signal is also activated. Write to port F1h only resets the coprocessor. A software FNINT signal must occur after an F1h generated reset in a 386SX system, otherwise the 387SX is not initialized to the same state that a 287 is placed in by a hardware reset alone. For compatibility, the F1 reset may be disabled by setting bit 6 of the MISCSET register to 1.
IRQ13	66	O	**INTERRUPT REQUEST, ACTIVE HIGH:** This output is driven to the Bus Controller to indicate that an error has occurred within the coprocessor. This signal is decode of the BUSYNPX# and ERRORNPX# inputs.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
COPROCESSOR SIGNALS (Continued)			
NPCS#	67	O	**COPROCESSOR CHIP SELECT:** Provides decoding of the 287 coprocessor's I/O space. This is the entire F8h to FFh region when Zenith Data Systems-Special Features are disabled. When ZDS-SF are enabled, only I/O accesses to F8h, FAh, FCh, and FEh cause NPCS# to be active. This signal is a don't care pin for 387SX operation since the 386SX provides this function using A23.
BUS CONTROL SIGNALS			
CHREADY#	80	I-CMOS	**CHANNEL READY, ACTIVE LOW:** An input issued by the Bus Controller as an indication that the current channel bus cycle is complete. This signal is synchronized internally then combined with ready signals from the coprocessor and DRAM controller to form the final version of READYO# which is sent to the CPU.
CHS0#/MW#	93	IO-TTL	**CHANNEL SELECT 0 OR MEMORY WRITE, ACTIVE LOW:** This signal is a decode of the CPU's bus control signals and is sent to the Bus Controller. When combined with CHS1# and CHM/IO# and decoded, the bus cycle type is defined for the Bus Controller. Activation of CPUHLDA reverses this signal to become an input from the Bus Controller. It is then a MEMW# signal for the DMA or bus master to access system memory.
CHS1#/MR#	94	IO-TTL	**CHANNEL SELECT 1 OR MEMORY READ, ACTIVE LOW:** This signal is a decode of the CPU's bus control signals and is sent to the Bus Controller. When combined with CHS0# and CHM/IO# and decoded, the bus cycle type is defined for the Bus Controller. Activation of CPUHLDA reverses this signal to become an input from the Bus Controller. It is then a MEMR# signal for the DMA or bus master to access system memory.
CHM/IO#	95	O	**CHANNEL MEMORY I/O:** This signal is a decode of the M/IO# signal sent by the CPU to the System Controller. It is an indicator that the current bus cycle is a channel access. When combined with CHS0#, and CHM/IO# and decoded, the bus cycle type is defined for the Bus Controller.
BLKA20#	96	O	**BLOCK A20, ACTIVE LOW:** An output driven to the Bus Controller to deactivate address bit 20. It is a decode of the A20GATE signal and Port A bit 1 indicating the dividing line of the 1 Mbyte memory boundary. Port A bit 1 may be directly written or set by a dummy read of I/O port EEh. BLKA20# is forced high when HLDA is active.
BUSOSC	73	I-TTL	**BUS OSCILLATOR:** This signal is supplied from an external oscillator. It is supplied to the Bus Controller when the System Controller's internal configuration registers are set for asynchronous slot bus mode. This signal is two times the AT bus clock speed (SYSCLK).
BUSCLK	97	O	**BUS CLOCK:** This is the source clock used by the the Bus Controller to drive the slot bus. It is two times the AT bus clock (SYSCLK). It is programmable division from CLK2 or from an external oscillator when the System Controller is set up for a synchronous mode.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
BUS CONTROL SIGNALS (Continued)			
DMAHRQ	83	I-CMOS	**DMA HOLD REQUEST, ACTIVE HIGH:** This signal is an input sent by the Bus Controller. It is internally synchronized by the System Controller before it is sent out to the CPU by the HRQ signal. It is the indicator of the DMA controller or other bus master's desire to control the bus.
DMAHLDA	98	O	**DMA HOLD ACKNOWLEDGE:** An output sent to the Bus Controller which indicates that the current hold acknowledge state is for the DMA controller or other bus master.
BRDRAM#	99	O	**BOARD DRAM, ACTIVE LOW:** An output to Bus Controller and Data Buffer to indicate that on-board DRAM is being addressed.
EALE#	92	O	**EARLY ADDRESS LATCH ENABLE, ACTIVE LOW:** This output is sent to the Bus Controller in order latch the A23–A1 and byte enable signals. In 286 mode, this signal is generated internally by decode of the CPU status signals. In 386SX mode, the 82343's ADS# input is gated directly to the EALE# output.
OUT1	84	I-CMOS	Indicates a refresh request from the Bus Controller. This signal is provided by the 8254 megacell.
PERIPHERAL INTERFACE SIGNALS			
A20GATE	75	I-TTL	**ADDRESS BIT 20 ENABLE:** An input from the keyboard controller that is used internally along with Port A bit 1 to determine if address bit 20 from the CPU is true or gated low. It also determines the state of BLKA20#.
TURBO	76	I-TTL	**TURBO, ACTIVE HIGH:** This input to the System Controller determines the speed at which the system board operates. This input signal is normally the externally ANDed signal from the keyboard controller and a turbo switch. It is internally ANDed with a software settable latch. When high, operation is at full speed. When low, CLK2 is divided by the value coded in configuration register MISCSET. A range is provided that allows slow operation at, or below 8 MHz for any valid CPU speed. Slow speed takes precedence. When any one request for slow mode is present, slow mode is active. Turbo mode is active only when all TURBO requests are active.
RC#	77	I-TTL	**RESET CONTROL, ACTIVE LOW:** The falling edge of this signal causes a RESCPU signal. RC# is generated by the keyboard controller and its inverse is ORed with Port A bit 0 to form RESCPU.
BUS INTERFACE SIGNALS			
OSC	72	I-TTL	**OSCILLATOR:** This is the buffered input of the external 14.318 MHz oscillator.
IOR#	85	I-TTL	**I/O READ, ACTIVE LOW:** Driven by the Bus Controller to indicate to the System Controller that an I/O read cycle is occurring on the bus. Whenever an I/O cycle occurs, the memory interface signals are inactive.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
BUS INTERFACE SIGNALS (Continued)			
IOW#	86	I-TTL	**I/O WRITE, ACTIVE LOW:** Driven by the Bus Controller to indicate to the System Controller than an I/O write cycle is occurring on the bus. Whenever an I/O cycle occurs, the memory interface signals are inactive.
RSTDRV	87	I-TTL	**RESET DRIVE, ACTIVE HIGH:** This signal is output by the Bus Controller. It indicates that a hardware reset signal has been activated. This is the same signal which is output to the channel. This signal is used to reset internal logic and to derive the RESCPU which is output by the System Controller.
SDLH/HL#	88	I-TTL	**SYSTEM DATA BUS LOW TO HIGH/HIGH TO LOW SWAP:** This signal is driven by the Bus Controller. It is used to establish the direction of byte swaps. (Similar to DIR245 in the existing PC/AT-type chip sets.)
SDSWAP	89	I-TTL	**SYSTEM DATA BUS BYTE SWAP ENABLE:** This signal is driven by the Bus Controller. It is the qualifying signal needed for SDLH/HL#. (Used to be called GATE245 on the existing PC/AT-type chip sets.)
XDREAD#	90	I-TTL	**PERIPHERAL DATA BUS (XD BUS) READ:** This signal is driven by the Bus Controller and it determines the direction of the XD bus data flow. (It is analogous to the XDATADIR control pin on the existing PC/AT-type chip sets). When this signal is high, the XD Bus is output enabled.
LATLO#	91	I-TTL	**SD BUS LOW BYTE LATCH:** This signal is needed to latch the SD bus low byte to the local data bus until the CPU is ready to sample the bus. This signal is driven by the Bus Controller.
D15–D0	12–18, 20, 22–29	IO-TTL	**CPU DATA BUS:** This is the data bus directly connected to the CPU. It is also referred to as the local data bus.
SD15–0	111–114, 116–119, 122–125, 127–130	IO-TTL	**SYSTEM DATA BUS:** This bus connects directly to the slots. It is used to transfer data to/from local and system devices.
XD7–XD0	101–107, 109	IO-TTL	**PERIPHERAL DATA BUS:** This bus is connected to the Bus Controller and the System Controller. These I/O's are used to read and write to on-board 8-bit peripherals.
PAR1, PAR0	21, 30	IO-TTL	**PARITY BIT BYTES 1 AND 0:** These bits are generated by the parity generation circuitry. They are written to memory along with their corresponding bytes during memory write operations. During memory read operations, these bits become inputs and are used along with their respective data bytes to determine if a parity error has occurred.
PARERROR#	100	O	**PARITY ERROR, ACTIVE LOW:** This signal is the result of a parity check on the read from on-board memory.
286/386SX#	78	I-TPU	**286 OR 386SX MODE:** Tied high or left open to allow internal logic to switch to 286/287 compatibility mode. If grounded, the 82343 System Controller/Data Buffer switches into 386SX/387SX compatibility mode.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
TEST MODE PIN			
TRI#	79	I1	**THREE-STATE:** This pin is used to drive all outputs to a high impedance state. When TRI# is low, all outputs and bidirectional pins are three-stated. TRI# is internally pulled up.
POWER AND GROUND PINS The power connections are split into an internal supply for the core-logic, and a pad-ring supply for the I/O drivers. Each supply should be individually bypassed with decoupling capacitors.			
VDDR	1, 32, 41, 110, 58, 121	PWR	Pad-ring power connection, nominally +5V. These pins along with the VSSR pins should be separately bypassed.
VSSR	19, 31, 36, 42, 46, 55, 81, 108, 115, 120, 126, 131	GND	Pad-ring ground connection, nominally 0V. These pins along with the VDDR pins should be separately bypassed.
VDDI	74	PWR	Internal core-logic power connection, nominally +5V. This pin along with the VSSI pin should be separately bypassed.
VSSI	82	GND	Internal core-logic ground connection, nominally 0V. This pin along with the VDDI pin should be separately bypassed.

SIGNAL TYPE LEGEND

Signal Code	Signal Type	Signal Code	Signal Type
I-TTL	TTL Level Input	IO-OD	Input or Open Drain, Slow Turn On
I-TPD	Input with 30 kΩ Pull-Down Resistor	O	CMOS and TTL Level Compatible Output
I-TPU	Input with 30 kΩ Pull-Up Resistor	O-TTL	TTL Level Output
I-TSPU	Schmitt-Trigger Input with 30 kΩ Pull-Up Resistor	O-TS	Three-State Level Output
I-CMOS	CMOS Level Input	I1	Input Used for Testing Purposes
IO-TTL	TTL Level Input/Output	GND	Ground
IT-OD	TTL Level Input/Open Drain Output	PWR	Power

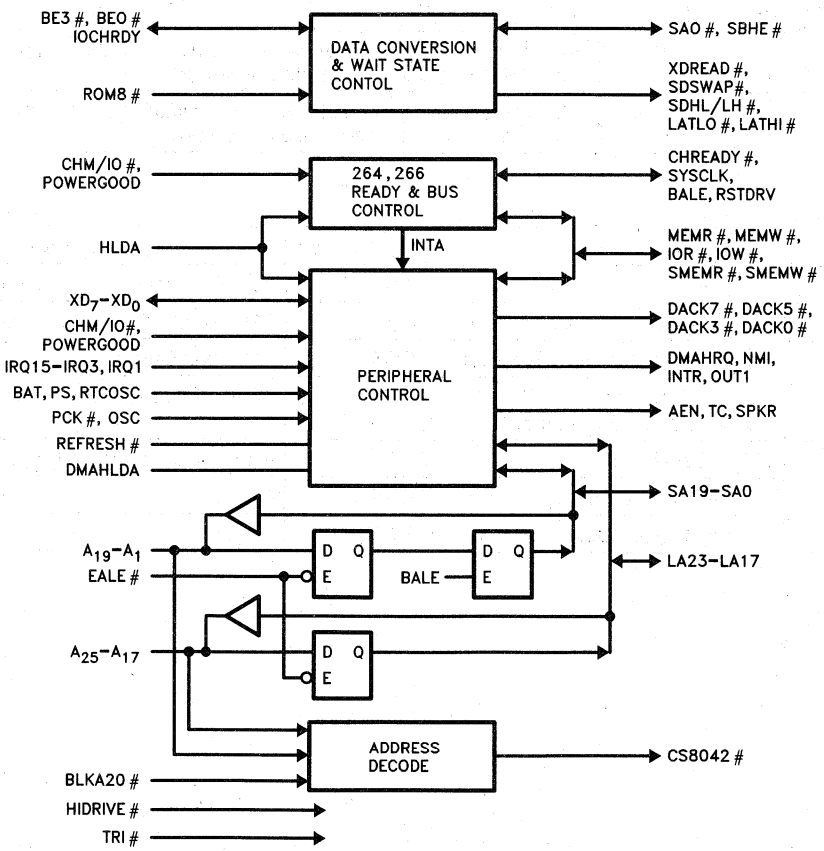
82344
ISA BUS CONTROLLER

The 82344 ISA Bus Controller replaces several of the LSI controllers used in PC/AT*-type designs with one single 160-pin quad flatpack. The Bus Controller provides the functions of DMA, page address register, timer, interrupt control, Port B logic, slot bus refresh address generation, and real time clock.

The Bus Controller directly drives the refresh addresses onto the AT slot address bus during refresh cycles in response to a refresh cycle command from the System Controller. To avoid problems with sensitive slot bus add-in cards, the Bus Controller features "bus quiet" mode. When no valid slot bus accesses are occurring, the SA bus and control lines do not change states. Rather, they retain their previous logic state.

Built-in sleep mode features work together with System Controller sleep features to provide a low power system idle state for extension of battery life in portable systems. When activated by the CPU via I/O write to an internal indexed configuration register, the DMA sybsystem clock is stopped and the AT slot bus remains in BUS QUIET state. The SYSCLK can be individually controlled. The interrupt controllers and the timers continue to operate. If an interrupt occurs due to an external source or any of the timers, the Bus Controller "wakes up" and in turn wakes the System Controller.

Block Diagram



240843-1

386 is a trademark of Intel Corporation.
*PC/AT is a registered trademark of International Business Machines Corporation.

October 1990
Order Number: 240843-001

The upgraded DMA channels provide a superset of AT functionality by allowing DMA to the entire 64 Mbyte memory range of the 82340DX chip set. Additional functionality is provided via DMA wait state, clock, and —MEMR timing programmability.

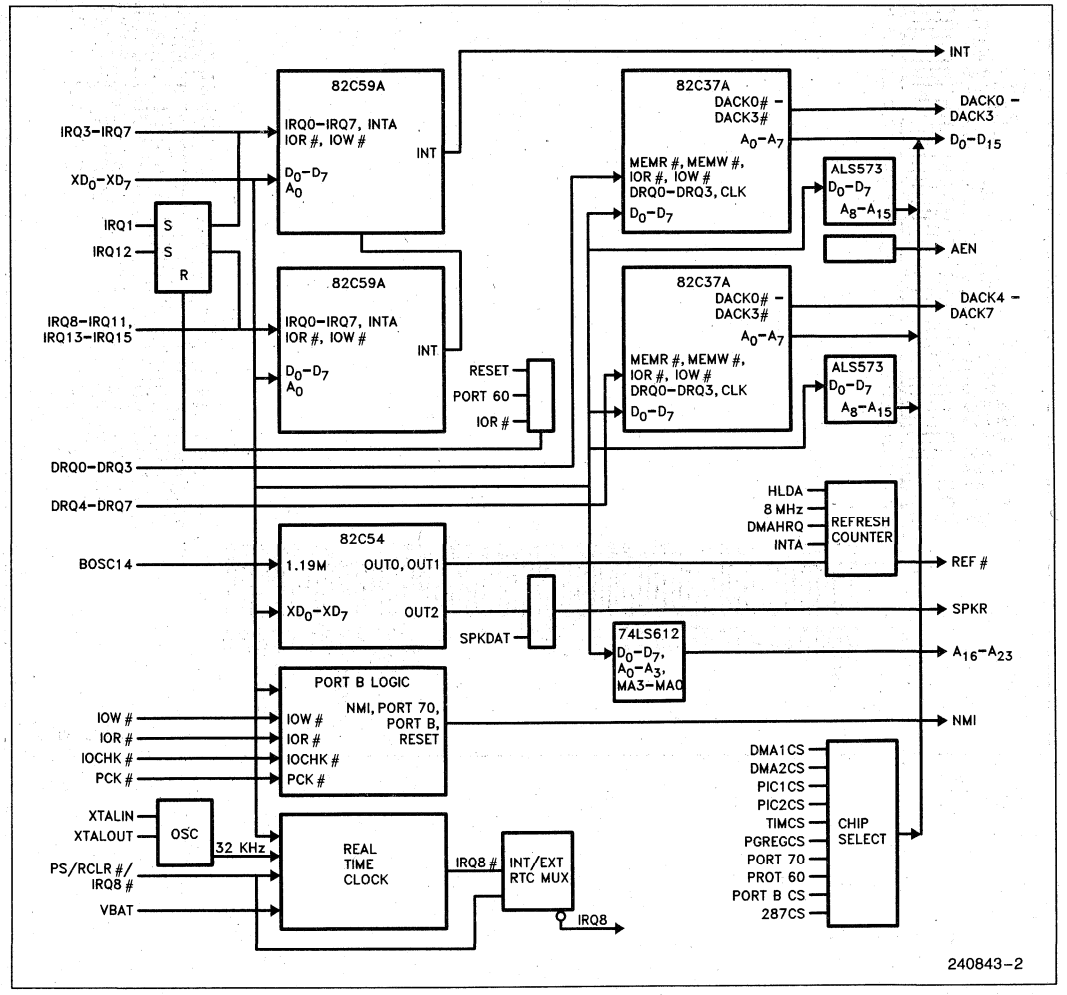
A —HIDRIVE pin can be externally strapped to provide for 12 mA or 24 mA drive to the slot bus. If left open, an internal pull-up causes the drive current to default to 24 mA. This allows systems designed with one to four slots to select a lower drive level and reduce bus ringing. A —ROM8 pin selects the bus and bus size to use for BIOS ROM accesses. The choices are 8- or 16-bit wide ROMs.

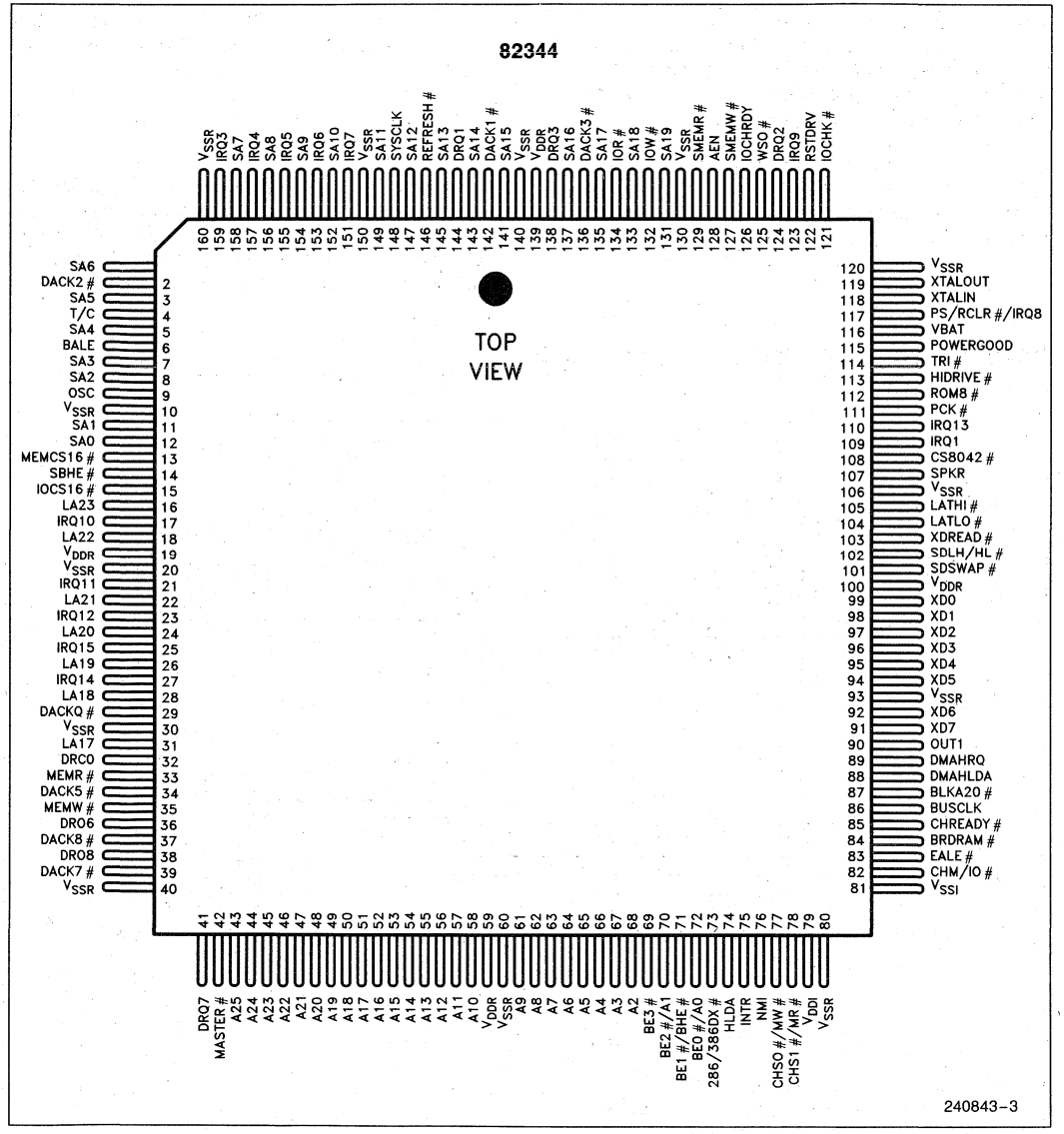
A three-state test control pin has been added for board level testability.

The Bus Controller features several megacells, implemented in 1.5-micron CMOS technology, and is intended to work in 386™ SX or 386™ DX microprocessor-based systems with CPU clock speeds up to 33 MHz and bus speeds up to 16 MHz.

1

PERIPHERAL CONTROL BLOCK DIAGRAM




82344
TOP
VIEW
1 SA6
2 DACK2 #
3 SA5
4 T/C
5 SA4
6 BALE
7 SA3
8 SA2
9 OSC
10 VSSR
11 SA1
12 SA0
13 MEMCS16 #
14 SBHE #
15 IOCS16 #
16 LA23
17 IRQ10
18 LA22
19 VDDR
20 VSSR
21 IRQ11
22 LA21
23 IRQ12
24 LA20
25 IRQ15
26 LA19
27 IRQ14
28 LA18
29 DACK0 #
30 VSSR
31 LA17
32 DRC0
33 MEMR #
34 DACK5 #
35 MEMW #
36 DR06
37 DACK8 #
38 DR08
39 DACK7 #
40 VSSR
41 DR07
42 MASTER #
43 A25
44 A24
45 A23
46 A22
47 A21
48 A20
49 A19
50 A18
51 A17
52 A16
53 A15
54 A14
55 A13
56 A12
57 A11
58 A10
59 VDDR
60 VSSR
61 A9
62 A8
63 A7
64 A6
65 A5
66 A4
67 A3
68 A2
69 BE3 #
70 BE2 #/A1
71 BE1 #/BHE #
72 BE0 #/A0
73 286/386DX #
74 HLDA
75 INTR
76 NMI
77 CHS0 #/WW #
78 CHS1 #/MR #
79 VDDI
80 VSSR
81 VSSI
82 CHM/IO #
83 EALE #
84 BRDRAM #
85 CHREADY #
86 BUSCLK
87 BLKA20 #
88 DMAHLDA
89 DMAHRQ
90 OUT1
91 XD7
92 XD6
93 VSSR
94 XD5
95 XD4
96 XD3
97 XD2
98 XD1
99 XD0
100 VDDR
101 SDSWAP #
102 SDLH/HL #
103 XDREAD #
104 LATLO #
105 LATHI #
106 VSSR
107 SPKR
108 CS8042 #
109 IRQ1
110 IRQ13
111 PCK #
112 ROM8 #
113 HIDRIVE #
114 TRI #
115 POWERGOOD
116 VBAT
117 PS/RCLR #/IRQ8
118 XTALIN
119 XTALOUT
120 VSSR
121 IOCHK #
122 RSTDRV
123 IRQ9
124 DRQ2
125 WSO #
126 IOCHRDY
127 SMEMW #
128 AEN
129 SMEMR #
130 VSSR
131 SA19
132 IOW #
133 SA18
134 IOR #
135 SA17
136 DACK3 #
137 SA16
138 DRQ3
139 VDDR
140 VSSR
141 SA15
142 DACK1 #
143 SA14
144 DRQ1
145 SA13
146 REFRESH #
147 SA12
148 SYSCLK
149 SA11
150 VSSR
151 IRQ7
152 SA10
153 IRQ6
154 SA9
155 IRQ5
156 SA8
157 IRQ4
158 SA7
159 IRQ3
160 VSSR
240843-3

SIGNAL DESCRIPTIONS

Name	Pin Number	Type	Description
CPU INTERFACE			
A25, A24	43–44	O-TS	Address Bus—These pins are outputs during DMA, master, or standard refresh modes. They are high impedance at all other times. A25 and A24 are driven from the alternate 612 registers during DMA and refresh cycles and are driven low during master cycles.
A23–A2	45–58, 61–68	IO-TTL	Address Bus—These pins are outputs during DMA, master, or standard refresh modes. They are inputs at all other times. As inputs, they are passed to the SA and LA buses and A15–A2 are used to address I/O registers internal to the bus control chip. As outputs, they are driven from different sources depending on which mode the Bus Controller is in. While in refresh mode, these pins are driven from the 612 and refresh address counter. While in DMA mode, they are driven from the 612 and DMA controller subsection. If the Bus Controller is in master mode, the pins A23–A17 are driven from the inputs LA23–LA17 and the pins A16–A2 are driven from the inputs SA16–SA2.
—BE3	69	IO-TTL	Byte Enable 3, active low—This pin is an output during DMA, master, or standard refresh modes. It is an input at all other times. As an input in 386DX mode, it is decoded along with the other byte enable signals to generate SA1, SA0 and —SBHE. As an output in 386DX mode SA1, SA0, and —SBHE are used to determine the value of —BE3. This pin should be left unconnected when using this part in 286 mode. The pin has an internal pull-up.
—BE2/A1	70	IO-TTL	Byte Enable 2, active low, or A1—This pin has a dual function depending on the state of the 286/–386DX input. If 286/–386DX is high (286 mode), then the pin is treated as address bit 1. If 286/–386DX is low (386DX mode), the pin is treated as —BE2. This pin is an output during DMA, master, or standard refresh modes. It is an input at all other times. As an input in 386DX mode, it is decoded along with the other byte enable signals to generate SA1, SA0, and —SBHE. As an output in 386DX mode. SA1, SA0, and —SBHE are used to determine the value of —BE2. When in 286 mode, it is interpreted as address A1 and passed to SA1. As an output in 286 mode it is driven from the SA1 input.
—BE1/—BHE	71	IO-TTL	Byte Enable 1 or Byte High Enable, active low—This pin has a dual function depending on the state of the 286/—386DX input. If 286/—386DX is high (286 mode), then the pin is treated as —BHE. If 286/—386 is low (386 mode), the pin is treated as —BE1. This pin is an output during DMA, master, or standard refresh modes. It is an input at all other times. As an input in 386 mode, it is decoded along with the other byte enable signals to generate SA1, SA0, and —SBHE. As an output in 386 mode, SA1, SA0, and —SBHE are used to determine the value of —BE1. When in 286 mode, it is interpreted as —BHE and passed to —SBHE. As an output in 286 mode, it is driven from the —SBHE input.

SIGNAL DESCRIPTIONS (Continued)

Name	Pin Number	Type	Description
CPU INTERFACE (Continued)			
—BE0/A0	72	IO-TTL	Byte Enable 0, active low, or A0—This pin has a dual function depending on the state of the 286/—386DX input. If 286/—386DX is high (286 mode), then the pin is treated as address bit 0. If 286/—386DX is low (386 mode), the pin is treated as —BE0. This pin is an output during DMA, master, or standard refresh modes. It is an input at all other times. As an input in 386 mode, it is decoded along with the other Byte Enable signals to generate SA1, SA0, and —SBHE. As an output in 386 mode, SA1, SA0, and —SBHE are used to determine the value of —BE0. When in 286 mode, it is interpreted as A0 and passed to SA0. As an output in 286 mode, it is driven from the SA0 input.
286/—386DX	73	I-TPU	CPU is 286 or 386DX—This pin defines the type of address bus to which the bus controller chip is interfaced. If the pin is tied high, the address bus is assumed to be emulating 286 signals. In this mode, A25, A24, and —BE3 would be left unconnected. The pins —BE2/A1, —BE1/—BHE and —BE0/A0 would take on the 286 functions. If the pin is tied low, A25, A24 can be used to generate up to 64 Mbyte addressing for DMA, and the byte enable pins will take on the normal 386DX addressing functions. This pin has an internal pull-up to cause the chip to default to 286 mode if left unconnected. This pin is a hard wiring option and must not be changed dynamically during operation. When strapped for 286 mode, the Bus Controller is assumed to be interfaced to the 82343 System Controller which in turn may be strapped for 286 or 386SX operation. The 82344 is strapped for 286 operation when used with the 82343 strapped for 386SX operation.
HLDA	74	I-TTL	Hold Acknowledge—This is the hold acknowledge pin directly from the CPU. It is used to control direction on address and command pins. When HLDA is low, the Bus Controller is defined as being in the CPU mode. In the CPU mode, the local address bus (A bus) pins are inputs. The system address bus (SA and LA) pins along with the command pins (—MEMR, —MEMW, —IOR and —IOW) are outputs. When HLDA is high, the Bus Controller can be in DMA, refresh, or master modes. In both DMA and refresh modes, the commands and all address buses (A, SA and LA) are outputs. In master mode, the commands and system address bus (SA and LA) pins are inputs and the local address bus (A bus) pins are outputs. The SA bus is passed directly to the A bus except bits 17, 18, and 19 are ignored. LA23–LA17 is passed directly to A23–A17.
INTR	75	O	Interrupt Request—INTR is used to interrupt the CPU and is generated by the 8259 megacells any time a valid interrupt request input is received.
NMI	76	O	Non-Maskable Interrupt—This output is used to drive the NMI input to the CPU. This signal is asserted by either a parity error (indicated by —PCK being asserted after the ENPARCK bit in Port B has been asserted), or an I/O channel error (indicated by —IOCHCK being asserted after the ENIOCK bit in Port B has been asserted). The NMI output is enabled by writing a 0 to bit D7 of I/O port 70h. NMI is disabled on reset.

SIGNAL DESCRIPTIONS (Continued)

Name	Pin Number	Type	Description
SYSTEM CONTROLLER INTERFACE			
—CHS0/—MW	77	IO-TTL	Channel Status 0 or active low Memory Write—This input is used along with —CHS1 and CHM/—IO to determine what type of bus cycle the Bus Controller is to perform. This input has the same meaning and timing requirements as the S0 signal for a 286 microprocessor. —CHS0 going active indicates a write cycle unless —CHS1 is also active. When both status inputs are active it indicates an interrupt acknowledge cycle. This input is synchronized to the BUSCLK input. Activation of CPUHLDA reverses this signal to become an output to the System Controller. It is then a —MEMW signal for DMA or bus master access to system memory.
—CHS1/—MR	78	IO-TTL	Channel Status 1 or active low Memory Read—This input is used along with —CHS0 and CHM/—IO to determine the bus cycle type. This input has the same meaning and timing requirements as the S1 signal for a 286 microprocessor —CHS1 going active indicates a read cycle unless —CHS0 is also active. When both status inputs are active it indicates an interrupt acknowledge cycle. This input is synchronized to the BUSCLK input. Activation of CPUHLDA reverses this signal to become an output to the System Controller. It is then a —MEMR signal for DMA or bus master access to system memory.
CHM/—IO	82	I-TTL	Channel Memory or active low I/O select—This input is used along with —CHS0 and —CHS1 to determine the bus cycle type. This input has the same meaning and timing requirements as the M/—IO signal for a 286 microprocessor. CHM/—IO is sampled anytime —CHS0 or —CHS1 is active. If sampled high, it indicates a memory read or write cycle. If sampled low, an I/O read or write cycle should be executed. This input is synchronized to the BUSCLK input.
—EALE	83	I-TTL	Early Address Latch Enable, active low—This input is used to latch the A25–A2 and Byte Enable signals. The latches are open when —EALE is low and hold their value when —EALE is high. The latched addresses are fed directly to the LA23–LA17 bus to provide more address setup time on the bus before a command goes active. The lower latched addresses are latched again with an internal ALE signal as soon as —CHS0 or —CHS1 is sampled active and fed to the SA19–SA0 and —SBHE outputs. In a 386DX system, this input is connected directly to the —ADS output from the CPU. In a 286 system, this input is connected to the —EALE output from the 82343 System Controller.
—BRDRAM	84	I-TTL	On-board DRAM, active low—An input from the System Controller indicating that the on-board DRAM is being addressed.

SIGNAL DESCRIPTIONS (Continued)

Name	Pin Number	Type	Description
SYSTEM CONTROLLER INTERFACE (Continued)			
—CHREADY	85	O	Channel Ready, active low—This output is maintained in the active state when no bus accesses are active. This indicates that the Bus Controller is ready to accept a new command. During normal bus accesses, —CHREADY is negated as soon as a valid bus requested is sampled on the —CHS0 and —CHS1 inputs. It is asserted again to indicate that the Bus Controller is ready to complete the current cycle. The bus command signals are then terminated on the next falling edge of the BUSCLK input.
BUSCLK	86	I-CMOS	Bus Clock—This is the main clock input for the Bus Controller. It runs at twice the frequency desired for the SYSCLK output. All inputs are synchronous with the falling edge of this input.
—BLKA20	87	I-TTL	Block A20, active low—This input is used while CPUHLDA is low to force the LA20 and SA20 outputs low anytime it is active. When —BLKA20 is negated LA20 and SA20 are generated from A20.
DMAHRQ	89	O	Hold Request—This output is generated by the DMA controller any time a valid DMA request is received. It is connected to the DMAHRQ pin on the System Controller.
DMAHLDA	88	I-TTL	DMA Hold Acknowledge—An input from the System Controller which indicates that the current hold acknowledge state is for the DMA controller or other bus master.
OUT1	90	O	Output 1—Indicates a refresh request to the System Controller. This is the 15 μs output of timer channel 1.
ROM INTERFACE			
—ROM8	112	I-TPU	8/16 bit ROM select—This input indicates the width of the ROM BIOS. If —ROM8 is low, the Bus Controller chip generates 8- to 16-bit conversions for ROM accesses. Data buffer controls are generated assuming the ROM is on the MD bus. If —ROM8 is high, data buffer controls are generated assuming 16-bit wide ROMs are on the MD bus.
BUS INTERFACE			
—IOR	134	IO-TTL	I/O Read, active low—This signal is an input when CPUHLDA is high and —MASTER is low. It is an output at all other times. When CPUHLDA is low, —IOR is driven from the 288 bus controller megacell. When CPUHLDA is high and —MASTER is high, it is driven by the 8237 DMA controller megacells. This pin requires an external 10 KΩ pull-up resistor.
—IOW	132	IO-TTL	I/O Write, active low—This signal is an input when CPUHLDA is high and —MASTER is low. It is an output at all other times. When CPUHLDA is low, —IOW is driven from the 288 bus controller megacell. When CPUHLDA is high and —MASTER is high, it is driven by the 8237 DMA controller megacells. This pin requires an external 10 KΩ pull-up resistor.

SIGNAL DESCRIPTIONS (Continued)

Name	Pin Number	Type	Description
BUS INTERFACE (Continued)			
—MEMR	33	IO-TTL	Memory Read, active low—This signal is an input when CPUHLDA is high and —MASTER is low. It is an output at all other times. When CPUHLDA is low, —MEMR is driven from the 288 bus controller megacell. When CPUHLDA is high and —MASTER is high, it is driven by the 8237 DMA controller megacells. This signal does not pulse low for DMA addresses above 16 Mbytes. DMA above 16 Mbytes is only performed to the system board, never to the slot bus. This pin requires an external 10 KΩ pull-up resistor.
—MEMW	35	IO-TTL	Memory Write, active low—This signal is an input when CPUHLDA is high and —MASTER is low. It is an output at all other times. When CPUHLDA is low, —MEMW is driven from the 288 bus controller megacell. When CPUHLDA is high and —MASTER is high, it is driven by the 8237 DMA controller megacells. This pin requires an external 10 KΩ pull-up resistor.
—SMEMR	129	IO-TTL	Memory Read, active low—This signal is an input when CPUHLDA is high and —MASTER is low. It is an output at all other times. When CPUHLDA is low, —MEMR is driven from the 288 bus controller megacell. When CPUHLDA is high and —MASTER is high, it is driven by the 8237 DMA controller megacells. —SMEMR is active on memory read cycles to addresses below 1 Mbyte. This pin requires an external 10 KΩ pull-up resistor.
—SMEMW	127	IO-TTL	Memory Write, active low—This signal is an input when CPUHLDA is high and —MASTER is low. It is an output at all other times. When CPUHLDA is low, —MEMW is driven from the 288 bus controller megacell. When CPUHLDA is high and —MASTER is high, it is driven by the 8237 DMA controller megacells. —SMEMW is active on memory write cycles to addresses below 1 Mbyte. This pin requires an external 10 KΩ pull-up resistor.
LA23–LA17	16, 18, 22, 24, 26, 28, 31	IO-TTL	Latchable Address bus—This bus in an input when CPUHLDA is high and —MASTER is low. It is an output bus at all other times. When CPUHLDA is low, the LA bus is driven by the latched values for the A bus. When CPUHLDA is high and —MASTER is high, the SA bus is driven by the 612 memory mapper for DMA cycles and normal refresh. The LA bus is latched internally with the —EALE input.
SA19–SA0	131, 133, 135, 137, 141, 143, 145, 147, 149, 152, 154, 156, 158, 1, 3, 5, 7, 8, 11, 12	IO-TTL	System Address bus—This bus is an input when CPUHLDA is high and —MASTER is low. It is an output bus at all other times. When CPUHLDA is low, the SA bus is driven by the latched values from the A bus. When CPUHLDA is high and —MASTER is high, the SA bus is driven by the 8237 DMA controller megacells or refresh address generator. The SA bus will become valid in the middle of the status cycle generated by the —CHS0 and —CHS1 inputs. They are latched with an internally generated ALE signal.

SIGNAL DESCRIPTIONS (Continued)

Name	Pin Number	Type	Description
BUS INTERFACE (Continued)			
—SBHE	14	IO-TTL	System Byte High Enable, active low—This pin is controlled the same way as the SA bus. It is generated from a decode of the —BE inputs in CPU mode. It is forced low for 16-bit DMA cycles and forced to the opposite value of SA0 for 8-bit DMA cycles.
—REFRESH	146	IT-OD	Refresh signal, active low—This I/O signal is pulled low whenever a decoupled refresh command is received from the System Controller. It is used as an input to sense refresh requests from external sources such as the System Controller for coupled refesh cycles or bus masters. It is used internally to clock the refresh address counter and select a location in the memory mapper which drives A23–A17. —REFRESH is an open drain output capable of sinking 24 mA and requires an external pull-up resistor.
SYSCLK	148	O	System Clock—This output is half the frequency of the BUSCLK input. The bus control outputs BALE and the —IOR, —IOW, —MEMR and —MEMW are synchronized to SYSCLK.
OSC	9	I-TTL	Oscillator—This is the buffered input of the external 14.318 MHz oscillator.
RSTDRV	122	O	Reset Drive, active high—This output is a system reset generated from the POWERGOOD input. RSTDRV is synchronized to the BUSCLK input.
BALE	6	O	Buffered Address Latch Enable, active high—A pulse which is generated at the beginning of any bus cycle initiated from the CPU. BALE is forced high anytime CPUHLDA is high.
AEN	128	O	Address Enable—This output goes high anytime the inputs CPUHLDA and —MASTER are both high.
T/C	4	O	Terminal Count—This output indicates that one of the DMA channels terminal count has been reached. This signal directly drives the system bus.
—DACK7- —DACK5, —DACK3- —DACK0	39, 37, 34, 136, 2, 142, 29	O	DMA Acknowledge, active low—These outputs are the acknowledge signals for the corresponding DMA requests. The active polarity of these lines is set active low on reset. Since the 8237 megacells are internally cascaded together, the polarity of the —DACK signals must not be changed. This signal directly drives the system bus.
DRQ7–DRQ5 DRQ3–DRQ0	41, 38, 36, 138, 124, 144, 32	I-TSPU	DMA Request—These asynchronous inputs are used by an external device to indicate when they need service from the internal DMA controllers. DRQ0–DRQ3 are used for transfers from 8-bit I/O adapters to/from system memory. DRQ5–DRQ7 are used for transfers from 16-bit I/O adapters to/from system memory. DRQ4 is not available externally as it is used to cascade the two DMA controllers together. All DRQ pins have internal pull-ups.

SIGNAL DESCRIPTIONS (Continued)

Name	Pin Number	Type	Description
BUS INTERFACE (Continued)			
IRQ15–IRQ9, IRQ7–IRQ3, IRQ1	25, 27, 110, 23, 21, 17, 123, 151, 153, 155, 157, 159, 109	I-TPSU	Internal Request—These are the asynchronous interrupt request inputs for the 8259 megacells. IRQ0, IRQ2, and IRQ8 are not available as external inputs to the chip, but are used internally. IRQ0 is connected to the output of the 8254 counter 0. IRQ2 is used to cascade the two 8259 megacells together. IRQ8 is output from the RTC megacell to the 8259 megacell. All IRQ input pins are active high and have internal pull-ups.
—MASTER	42	I-TTL	Master, active low—This input is used by an external device to disable the internal DMA controllers and get access to the system bus. When asserted it indicates that an external bus master has control of the bus.
—MEMCS16	13	I-TTL	Memory Chip Select 16-bit—This input is used to determine when a 16-bit to 8-bit conversion is needed for CPU accesses. A 16 to 8 conversion is done anytime the System Controller requests a 16-bit memory cycle and —MEMCS16 is sampled high.
—IOCS16	15	I-TTL	I/O Chip Select 16-bit—This input is used to determine when a 16-bit to 8-bit conversion is needed for CPU accesses. A 16 to 8 conversion is done anytime the System Controller requests a 16-bit I/O cycle and —IOCS16 is sampled high.
—IOCHK	121	I-TTL	I/O Channel Check, active low—This input is used to indicate that an error has taken place on the I/O bus. If I/O checking is enabled, an —IOCHK assertion by a peripheral device generates an NMI to the processor. The state of the —IOCHK signal is read as data bit D6 of the Port B register.
IOCHRDY	126	I-TTL	I/O Channel Ready—This input is pulled low in order to extend the read or write cycles of any bus access when required. The cycle can be initiated by the CPU, DMA controllers or refresh controller. The default number of wait states for cycles intiated by the CPU are four wait states for 8-bit peripherals, one wait state for 16-bit peripherals and three wait states for ROM cycles. One DMA wait state is inserted as the default for all DMA cycles. Any peripheral that cannot present read data, or strobe-in write data in this amount of time must use —IOCHRDY to extend these cycles.
—WS0	125	I-TTL	Wait State 0, active low—This input is pulled low by a peripheral on the S bus to terminate a CPU controlled bus cycle earlier than the default values defined internally on the chip.
POWERGOOD	115	I-TSPU	System power on reset—This input signals that power to the board is stable. A Schmitt-trigger input is used. This allows the input to be connected directly to an RC network.
PERIPHERAL INTERFACE			
—CS8042	108	O	Chip select for 8042. This output is active any time an SA address is decoded at 60h or 64h. It is intended to be connected to the chip select of the keyboard controller. If BUSCTL[6] = 1, this pin is also active for RTC accesses at 70h and 71h. This is for use when the internal RTC is disabled and an external RTC is used.

SIGNAL DESCRIPTIONS (Continued)

Name	Pin Number	Type	Description
PERIPHERAL INTERFACE (Continued)			
XTALIN	118	I-CMOS	Crystal Input—An internal oscillator input for the real time clock crystal. It requires a 32.768 KHz external crystal or stand-alone oscillator.
XTALOUT	119	O	Crystal Output—An internal oscillator output for the real time clock crystal. See XTALIN. This pin is a no connect when an external oscillator is used.
PS/—RCLR/ IRQ8	117	I-TSPU	The Power Sense input (active high) is used to reset the status of the Valid RAM and Time (VRT) bit. This bit is used to indicate that the power has failed, and that the contents of the RTC may not be valid. This pin is connected to an external RC network. When BUSCTL[6] = 1, this pin becomes —IRQ8 input for use with an external RTC.
VBAT	116	I	Voltage Battery—Connected to the RTC hold-up battery between 2.4 and 5V.
SPKR	107	O	Speaker—This output drives an externally buffered speaker. This signal is created by gating the output of timer 2. Bit 1 of Port B, 61H, is used to enable the speaker output, and bit 0 is used to gate the output timer.
DATA BUFFER INTERFACE			
XD7–XD0	91, 92, 94–99	IO-TTL	Peripheral data bus—The bidirectional X data bus outputs data on an INTA cycle or I/O read cycle to any valid address within the Bus Controller. It is configured as an input at all other times.
—SDSWAP	101	O	System Data Swap, active low during some 8-bit accesses—It indicates that the data on the SD bus must be swapped from low byte to high byte or vice versa depending on the state of the SDLH/—HL pin. —SDSWAP is active for 8-bit DMA cycles when an odd address access occurs for data more than one byte wide. For non-DMA accesses. —SDSWAP is active for any bus cycle to an 8-bit peripheral that is addressing the odd byte.
SDLH/—HL	102	O	System Data Low to High, or High to Low—This signal is used to determine which direction data bytes must be swapped when —SDSWAP is active. When SDLH/—HL is high, it indicates that data on the low byte must be transferred to the high byte. When SDLH/—HL is low, it indicates that data on the high byte must be transferred to the low byte. SDLH/—HL is low for 8-bit DMA memory read cycles. For non-DMA accesses, SDLH/—HL is low for any memory write or I/O write when —SBHE is low. SDLH/—HL is high at all other times.
—XDREAD	103	O	Peripheral Data Read—This output is active low any time an INTA cycle occurs or an I/O read occurs to the address space from 0000h to 00FFh, which is defined as being resident on the peripheral bus.

SIGNAL DESCRIPTIONS (Continued)

Name	Pin Number	Type	Description
DATA BUFFER INTERFACE (Continued)			
—LATLO	104	O	Latch Low byte—This output is generated for all I/O read and memory read bus accesses to the low byte. It is active with the same timing as the read command and returns high at the same time as the read command. This signal latches the data into the data buffer chip so that it can be presented to the CPU at a later time. This step is required due to the asynchronous interface between the System Controller and Bus Controller.
—LATHI	105	O	Latch High byte—This output is generated for all I/O read and memory read bus accesses to the high byte. It is active with the same timing as the read command and returns high at the same time as the read command. This signal latches the data into the data buffer chip so that it can be presented to the CPU at a later time. This step is required due to the asynchronous interface between the System Controller and Bus Controller.
—PCK	111	I-TPU	Party Check input, active low with pull-up—Indicates that a parity error has occurred in the on-board memory array. Assertion of this signal (if enabled) generates an NMI to the processor. The state of the —PCK signal is read as data bit D7 of the Port B register.
—HIDRIVE	113	I-TPU	High Drive Enable—This pin is a wire strap option. When this input is low, all bus drivers defined with an IOL spec of 24 mA will sink the full 24 mA of current. When this input is high, all pins defined as 24 mA have the output low drive capability cut in half to 12 mA. Note that all AC specifications are done with the outputs in the high drive mode and a 200 pF capacitive load —HIDRIVE has an internal pull-up and can be left unconnected if 12 mA drive is desired. It is tied low if 24 mA drive is desired.
TEST MODE PIN			
—TRI	114	I-TPU	Three-state—This pin is used to control the three-state drive of all outputs and bidirectional pins on the chip. If this pin is pulled low, all pins on the chip except XTALOUT are in a high impedance mode. This is useful during system test when test equipment or other chips drive the signals or for hardware fault tolerant applications. —TRI has an internal pull-up.
POWER AND GROUND PINS The power connections are split into an internal supply for the core-logic, and a pad-ring supply for the I/O drivers. Each supply should be individually bypassed wih decoupling capacitors.			
VDDR	19, 59, 100, 139	PWR	Pad-ring power connection, nominally +5V. These pins along with the VSSR pins should be separately bypassed.
VSSR	10, 20, 30, 40, 60, 80, 93, 106, 120, 130, 140, 150, 160	GND	Pad-ring ground connection, nominally 0V. These pins along with the VDDR pins should be separately bypassed.
VDDI	79	PWR	Internal core-logic power connection, nominally +5V. This pin along with the VSSI pin should be separately bypassed.
VSSI	81	GND	Internal core-logic ground connection, nominally 0V. This pin along with the VDDI pin should be separately bypassed.

Signal Type Legend

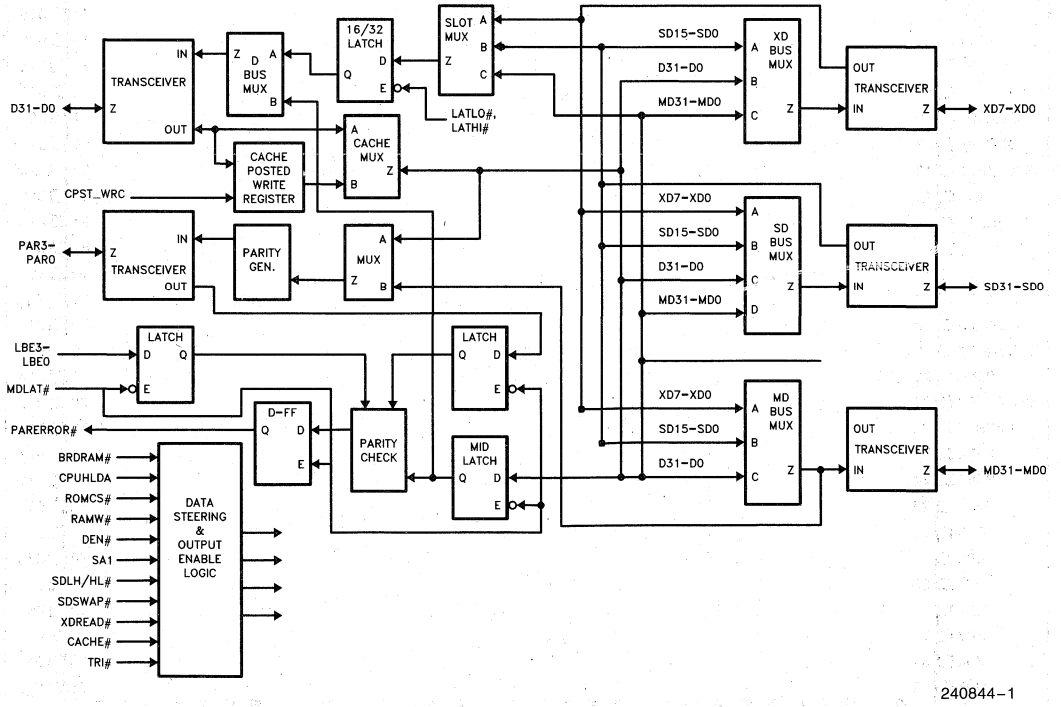
Signal Code	Signal Type
I-TTL	TTL Level Input
I-TPD	Input with 30 KΩ Pull-Down Resistor
I-TPU	Input with 30 KΩ Pull-Up Resistor
I-TSPU	Schmitt-Trigger Input with 30 KΩ Pull-Up Resistor
I-CMOS	CMOS Level Input
IO-TTL	TTL Level Input/Output
IT-OD	TTL Level Input/Open Drain Output
IO-OD	Input or Open Drain, Slow Turn On
O	CMOS and TTL Level Compatible Output
O-TTL	TTL Level Output
O-TS	Three-State Level Output
I1	Input used for Testing Purposes
GND	Ground
PWR	Power

82345
DATA BUFFER

The 82345 Data Buffer is part of a custom, three chip set which allows extremely high performance and integration in 386™ DX processor based, *PC/AT-compatible, personal computer designs. When used with the 82346 System Controller and the 82344 ISA Bus Controller, the set is called the 82340DX chip set.

The 82345 performs all of the data buffering functions required for a 386™ DX-based PC/AT-type system. Under the control of the CPU, the data buffer chip routes data to and from the CPU bus, the MD bus, the XD bus, and the slots (SD bus). For an on-board DRAM read, the data is latched in the MD latch allowing the 82346 System Controller to be programmed for early CAS terminations. The parity is checked for MD bus read operations and any errors are reported during the next read cycle. When reading from ROM, the XD bus or the SD bus, the data can be converted from 8-bits wide to 16-, 24- or 32-bits wide or from 16 bits to 32 bits at the 16/32 latch. The data is latched with LATLO# and LATHI# for synchronization with the CPU. The data conversion is accomplished without the use of the bus size 16 (BS16#) input to the 386DX allowing it to remain in pipelined mode.

CPU writes to any of the three buses are accomplished in several different ways. The 82345 supports posted writes from a cache controller or non-posted writes to the MD bus. Parity is generated for all data written to the MD bus. The 82345 provides the data conversion necessary for 32- or 16-bit writes to 16- or 8-bit devices on the XD or SD buses.



240844-1

Block Diagram

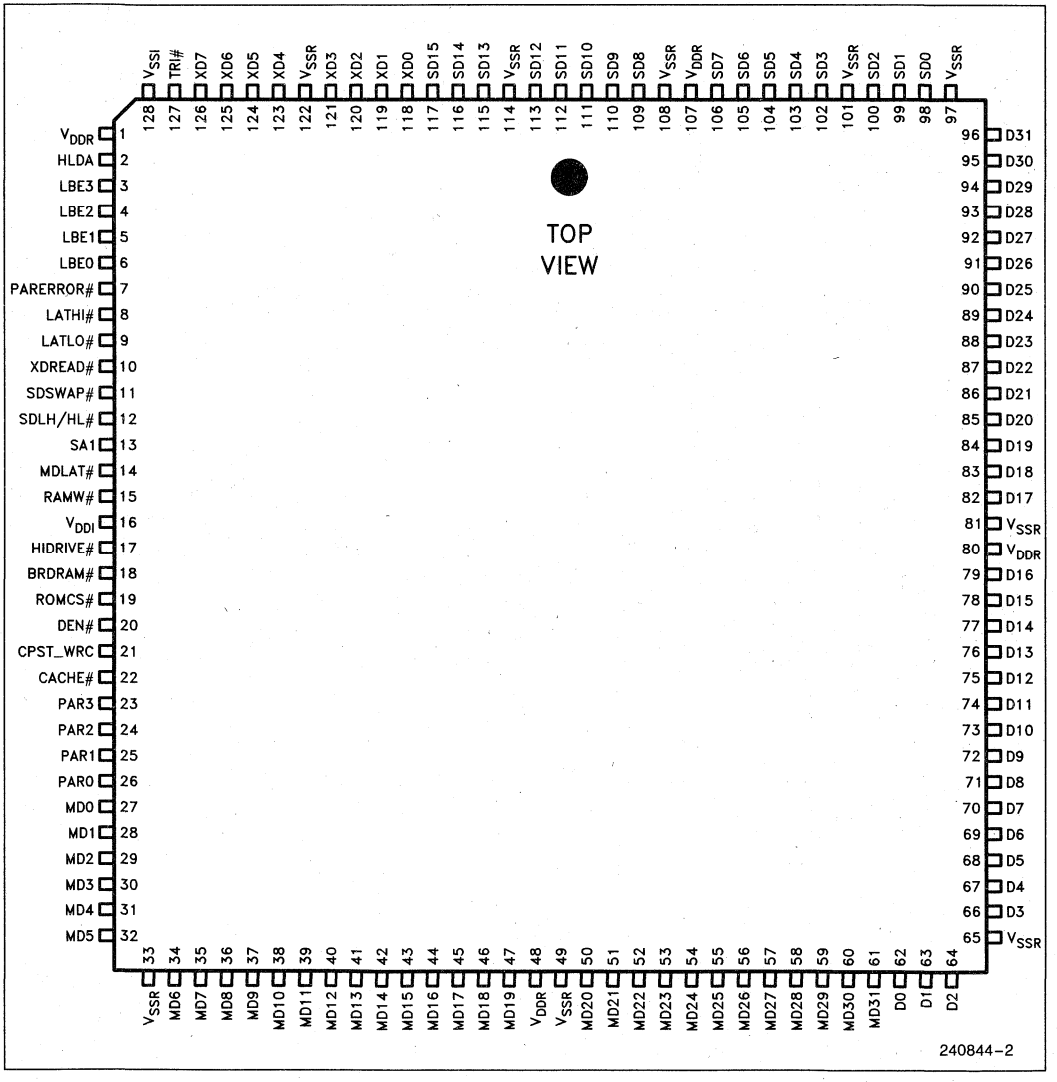
386™ is a trademark of Intel Corporation.
*PC AT is a trademark of International Business Machines Corporation.

October 1990
Order Number: 240844-001

In non-cached systems, system board DRAM can be placed on either the MD bus or the CPU's D bus. In slower systems (≤ 16 MHz) true zero wait state operation is possible with available 60 ns DRAMs when the D bus is used. This is due to the extra timing margin available when the MD bus delay through the 82345 is removed from the critical path. Faster non-cached systems can come close to zero wait state performance using 80 ns to 100 ns DRAMs and page mode interleaving. This requires an even number of DRAM banks.

Under the control of DMA or a bus master, the 82345 will allow 8- or 16-bit data to be routed to and from the XD and the MD buses. The chip also is capable of performing high to low and low to high byte swaps on the SD bus. For transfers between two peripherals on the slot bus, the outputs of the 82345 will be disabled. The chip also provides the feature of a single input, TRI#, to disable all of its outputs for board level testability.

PIN DIAGRAM



240844-2

SIGNAL DESCRIPTIONS

Signal Name	Pin Number	Signal Type	Signal Description
CPU INTERFACE			
HLDA	2	I-TTL	**CPU HOLD ACKNOWLEDGE, ACTIVE HIGH:** This is the hold acknowledge pin directly from the CPU. It indicates the CPU has given up the bus for either a DMA master or a slot bus master. It is used in the steering logic to determine data routing.
D31–D0	96–82, 79–66, 64–62	I/O-TTL	**CPU DATA BUS:** This is the data bus directly connected to the CPU. It is also referred to as the local data bus. This bus is output enabled by the DEN# signal.
CACHE INTERFACE			
CPST_WRC	21	I-TPU	**POSTED CACHE WRITE CLOCK:** This clock signal is driven by the cache controller and is needed to latch the write data during a posted cache write cycle. The data is latched on the rising edge of this signal. The latch inside of the Data Buffer is bypassed if the CACHE# input is high. Also, when CACHE# is high, the state of CPST_WRC determines on which bus (D or MD) system DRAM is accessed. When high, DRAM is accessed on the D bus. When low, DRAM is accessed on the MD bus. This pin is pulled up internally.
CACHE#	22	I-TPU	**CACHE ENABLE, ACTIVE LOW:** This signal is used to enable the cache posted write register. When there is not a cache in the system, data bypasses the register. When CACHE# is inactive (high) the state of the CPST_WRC pin determines whether the system DRAM is on the CPU's D bus or on the MD bus. This pin is pulled up internally.
SYSTEM CONTROLLER INTERFACE			
MDLAT#	14	I-TTL	**MEMORY DATA LATCH:** This latching signal serves two purposes simultaneously and is only activated during on-board memory read and write cycles. As a memory data latch, this transparent low signal allows read data to flow through to the CPU's local bus. It follows CAS# on early CAS# high read cycles and on the positive going edge, latches the memory data and holds it for the CPU to sample. As a parity clock, it clocks out PARERROR# on its falling edge an on the rising edge it latches the parity bits (PAR3–PAR0), the byte enables (LBE3–LBE0) and the memory data for parity error processing. Any parity errors will be reported on the next read cycle. It is the negative NOR of all CAS# signals gated by W/R#.
RAMW#	15	I-TTL	**RAM WRITE, ACTIVE LOW:** This signal is supplied by the System Controller to indicate to the Bus Controller that an on-board memory write cycle is occurring. It is used internally to direct the parity logic and to enable the MD bus outputs.
ROMCS#	19	I-TTL	**ROM CHIP SELECT:** This signal tells the Data Buffer when the ROM is to be accessed so that it can latch the data and convert it from 16 or 8 bits to 32 bits. This signal is driven by the System Controller.
BRDRAM#	18	I-TTL	**BOARD MEMORY SELECTED, ACTIVE LOW:** This signal is driven by the System Controller and indicates when on-board DRAM is being accessed.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
SYSTEM CONTROLLER INTERFACE (Continued)			
DEN#	20	I-TTL	**DATA ENABLE, ACTIVE LOW:** This is a control signal generated by the System Controller. It is used to enable data transfers on the local data bus and as an output enable for the D bus.
LBE3–LBE0	3–6	I-TTL	**LATCH BYTE ENABLES 3 THROUGH 0:** These signals are driven by the System Controller. They are used internally to enable the appropriate bytes (in a 4 byte wide memory configuration) for parity generation and checking.
PARERROR#	7	O	**PARITY ERROR, ACTIVE LOW:** This signal is the result of a parity check on the appropriate bytes being read from memory. It is generated on the falling edge of MDLAT#.
BUS CONTROLLER INTERFACE			
SA1	13	I-TTL	**SYSTEM ADDRESS BUS BIT 1:** This input will be driven by the Bus Controller or by the Controlling DMA or bus master. This signal is used for 16- to 32-bit conversion. When low, this signal indicates the low word is to be used.
SDLH/HL#	12	I-TTL	**SYSTEM DATA BUS LOW TO HIGH/HIGH TO LOW SWAP:** This signal is driven by the Bus Controller. It is used to establish the direction of byte swaps. (Similar to DIR245 in the existing PC/AT-type chip sets).
SDSWAP#	11	I-TTL	**SYSTEM DATA BUS BYTE SWAP ENABLE, ACTIVE LOW:** This signal is driven by the Bus Controller. It is the qualifying signal needed for SDLH/HL#. (It was formerly named GATE245 on the existing PC/AT-type chip sets).
XDREAD#	10	I-TTL	**PERIPHERAL DATA BUS (XD BUS) READ, ACTIVE LOW:** This signal is driven by the Bus Controller and it determines the direction of the XD bus data flow. (It is analogous to the XDATADIR control pin on the existing PC/AT-type chip sets). When this signal is high, the XD Bus is output enabled.
LATHI#	8	I-TTL	**SD BUS HIGH BYTE LATCH:** This signal is needed to latch the SD bus' high byte to the local data bus until the CPU is ready to sample the bus. When SA1 is low, the high byte is latched into both the one byte and the three byte of the 16/32 latch. When SA1 is high, the high byte is only latched into the three byte. This signal is driven by the Bus Controller.
LATLO#	9	I-TTL	**SD BUS LOW BYTE LATCH:** This signal is needed to latch the SD bus' low byte to the local data bus until the CPU is ready to sample the bus. When SA1 is low, the low byte is latched into both the zero byte and the two byte of the 16/32 latch. When SA1 is high, the high byte is only latched into the two byte. This signal is driven by the Bus Controller.
BUFFER INTERFACE			
MD31–MD0	61–50, 47–34, 32–27	I/O-TTL	**MEMORY DATA BUS:** This bus connects to the on-board DRAM and BIOS ROM. It is used to transfer data to/from memory during memory write/read bus cycles.
SD15–SD0	117–115, 103–109, 106–102, 100–98	I/O-TTL	**SYSTEM DATA BUS:** This bus connects directly to the slots. It is used to transfer data to/from local and system devices.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
BUFFER INTERFACE (Continued)			
XD7–XD0	126–123 121–118	I/O-TTL	**PERIPHERAL DATA BUS:** This bus is connected to the Bus Controller and the System Controller. These I/O's are used to read and write to on-board 8-bit peripherals.
PAR3–PAR0	23–26	I/O-TTL	**PARITY BIT BYTES 3 THROUGH 0:** These bits are generated by the parity generation circuitry located on the Data Buffer chip. They are written to memory along with their corresponding bytes during memory write operations. During memory read operations, these bits become inputs and are used along with their respective data bytes to determine if a parity error has occurred. The generation and check of each bit is enabled only when their respective LBE3–LBE0 bits are active.
HIDRIVE#	17	I-TPU	**HIGH DRIVE ENABLE:** This pin is intended to be a wire option. When this pin is low, all bus drivers defined with an I_{OL} of 24 mA will sink the full 24 mA of current. When the input is high, all pins defined as 24 mA will have the output low drive capability cut in half to 12 mA. Note that al A.C. specifications are done with the outputs in the high drive mode and a 200 pF capacitive load. HIDRIVE# has an internal pull-up and can be left unconnected in 12 mA drive if desired. It should be tied low if 24 mA drive is desired.
TEST MODE PIN			
TRI#	127	I-TPU	**THREE-STATE:** This pin is used to drive all outputs to a high impedance state. When TRI# is low, all outputs and bidirectional pins are three-stated. This pin should be pulled up via a 10 kΩ pull-up resistor in a standard system configuration.
POWER BUS CONNECTION The power connections are split into an internal supply for the core-logic, and a pad-ring supply for the I/O drivers. Each supply should be individually bypassed with decoupling capacitors.			
V_{DDR}	1, 48, 80, 107	PWR	**PAD-RING POWER CONNECTION NOMINALLY +5V:** These pins along with the V_{SSR} pins should be separately bypassed.
V_{SSR}	33, 49, 65, 81, 97, 101, 108, 114, 122	GND	**PAD-RING GROUND CONNECTION, NOMINALLY 0V:** These pins along with the V_{DDR} pins should be separately bypassed.
V_{DDI}	16	PWR	**INTERNAL CORE-LOGIC POWER CONNECTION, NOMINALLY +5V:** This pin along with the V_{SSI} pin should be separately bypassed.
V_{SSI}	128	GND	**INTERNAL CORE-LOGIC GROUND CONNECTION, NOMINALLY 0V:** This pin along with the V_{DDI} pin should be separately bypassed.

SIGNAL TYPE LEGEND

Signal Code	Signal Type
I-TTL	TTL Level Input
I-TPD	Input with 30 kΩ Pull-Down Resistor
I-TPU	Input with 30 kΩ Pull-Up Resistor
I-TSPU	Schmitt-Trigger Input with 30 kΩ Pull-Up Resistor
I-CMOS	CMOS Level Input
I/O-TTL	TTL Level Input/Output
IT-OD	TTL Level Input/Open Drain Output
I/O-OD	Input or Open Drain, Slow Turn On
O	CMOS and TTL Level Compatible Output
O-TTL	TTL Level Output
O-TS	Three-State Level Output
I1	Input Used for Testing Purposes
GND	Ground
PWR	Power

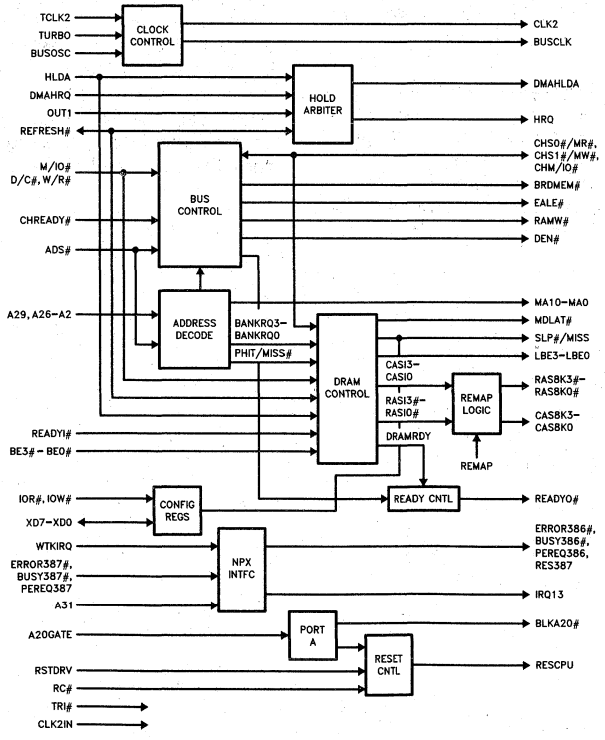
82346
SYSTEM CONTROLLER

The 82346 System Controller is highly configurable via software. No hardware jumpers are required. Defaults on reset for the configuration registers mimic the compatibility requirements of the original IBM PC/AT® as closely as possible. These power-up defaults allow any possible configuration of the system to boot at the CPU's rated speed. However, operational capabilities are reduced until the configuration registers are set to mirror the true system configuration. This normally occurs during BIOS power-on self-test in a manner completely transparent to the user.

The System Controller is designed to perform in systems running up to 33 MHz. Built-in page mode operation, two- or four-way interleaving and fully programmable memory timing allow the PC designer to maximize system performance using low cost DRAMs. Programmable memory timing allows the system to be setup to perfectly match the requirements of the chosen DRAMs; standard or custom. These adjustments can often be made without incurring the penalty of additional wait states.

The System Controller handles system board refresh directly and also controls the timing of slot bus refresh which is actually performed by the 82344 ISA Bus Controller. Refresh may be performed in coupled or decoupled mode. The former method is the standard PC/AT-compatible mode where on- and off-board refreshes are performed synchronously. In decoupled mode, the timing of on- and off-board refreshes are independent. Both may be programmed for independent, slower than normal rates. This allows use of low power, slow refresh DRAMs. The 82346 controls all timing in both modes. In all cases, refreshes are staggered to minmize power supply loading and attendant noise on the VDD and ground pins.

IBM PC/AT® is a registered trademark of IBM Corporation.



240842–1

Block Diagram

October 1990
Order Number: 240842-001

The physical banks of DRAM can be logically reordered through one of the indexed configuration registers. This DRAM remap option is useful in order to map out bad DRAM banks allowing continued use of a system until repairs are possible. It also allows DRAM bank combinations not in the supported memory maps to be logically moved into a supported configuration without physically moving memory components. This unique, programmable function performs this task by switching the internal RASI# and CASI# signals between the external RAS# and CAS pins. This allows internal addresses generated for DRAM bank 0, for example, to be routed to any one of the four on-board DRAM banks.

Active low RASBK# signals are generated to directly drive DRAM banks. Active high CASBK and LBE signals are externally decoded with NAND gates to provide 16 active low CAS# signals. This scheme provides extra timing margin and lower cost since NAND gates are cheaper and faster than equivalent OR gates.

To maintain use of low cost DRAMs through the full 33 MHz range of the system, special cache support is added. This minimizes the external glue logic required by other systems. The chip set is easily interfaced to the Intel 385DX cache controller.

Full EEMS support is provided in hardware for the complete full LIM EMS 4.0® standard. Seventy-two mapping registers provide a standard and an alternate set of 36 registers each. The system allows backfill to 256k for EEMS support and provides 24 mapping registers covering this space. Twelve of the 36 are page registers which cover the EMS space from C0000h to E0000h. These twelve registers can alternatively be mapped in the A0000-BFFFFh and D0000-DFFFFh range by changing a configuration bit in the 82346. All registers are capable of translating over the complete 64 MByte range of on-board DRAM. Users preferring an alternate plug-in EMS solution can disable the on-board EMS system as well as system board DRAM, as required, down to 256k.

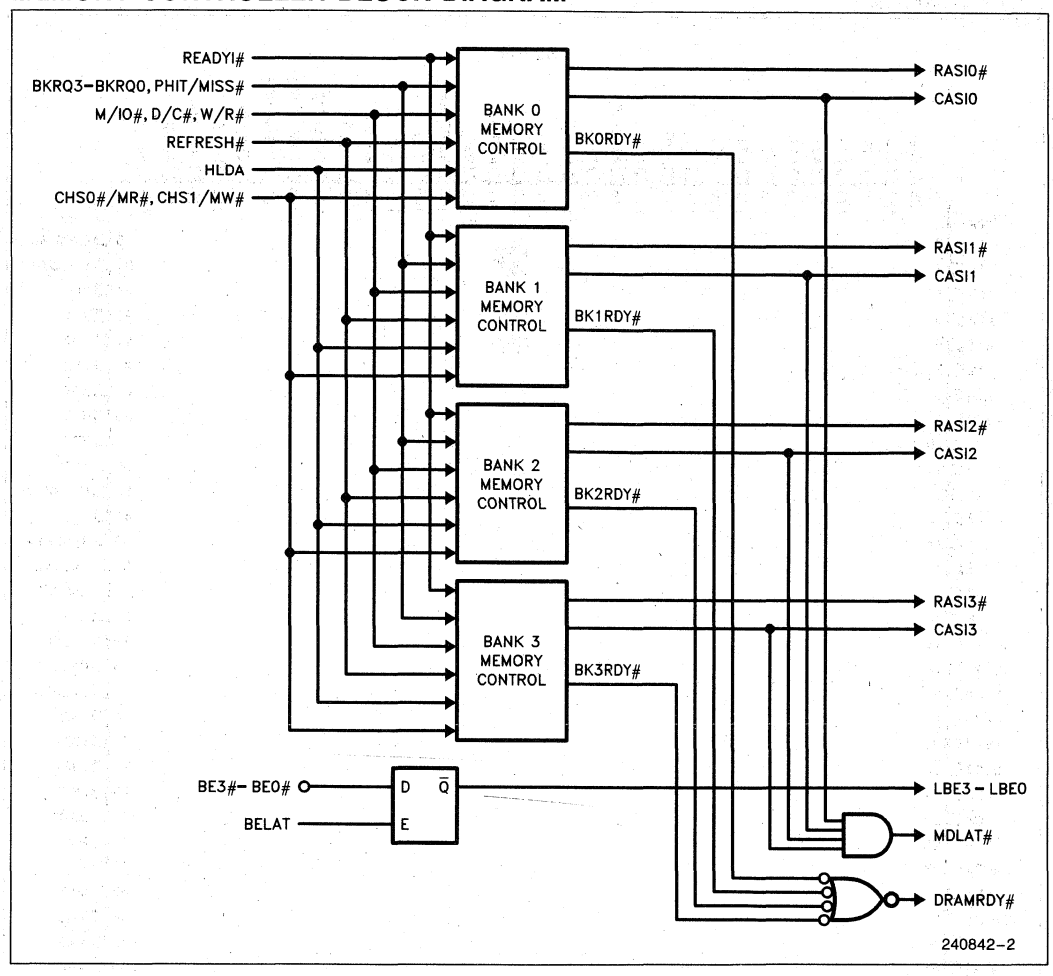
Shadowing features are supported on all 16k boundaries between 640k and 1M. EMS use, shadowed ROM, and direct system board access is possible in non-overlapping fashion throughout this memory space. Control over four access options is provided. These controls are overridden by EMS in segments for which it is enabled.

1. Access ROM or slot bus for reads and writes.
2. Access system board DRAM for reads and writes.
3. Access system board DRAM for reads and slot bus for writes.
4. Shadow setup mode. Read ROM or slot bus, write system board DRAM.

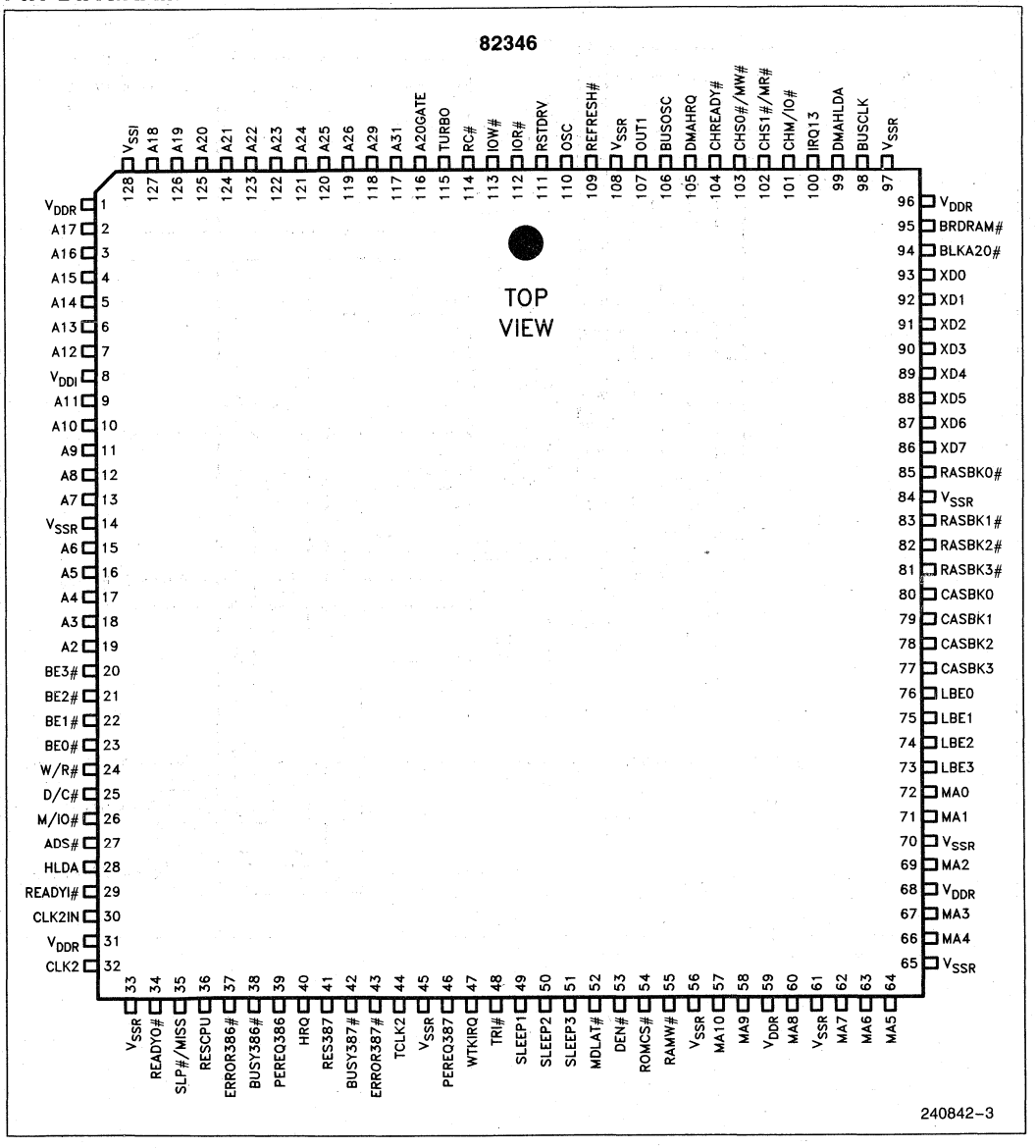
The System Controller is used to program the desired operational mode of the AT bus. Based on this programming, it provides the bus clock and signalling interface to the Bus Controller, which actually interfaces with the bus. The bus may run synchronously with the CPU's CLK2 or asynchronously via an external oscillator. A programmable divider conditions the selected BUSCLK source providing divide by 1, 2, 3 or 4.

LIM EMS 4.0® is a registered trademark of Lotus Development Corporation, Intel Corporation, and Microsoft Corporation.

MEMORY CONTROLLER BLOCK DIAGRAM



PIN DIAGRAM


82346
TOP VIEW
1 VDDR
2 A17
3 A16
4 A15
5 A14
6 A13
7 A12
8 VDDI
9 A11
10 A10
11 A9
12 A8
13 A7
14 VSSR
15 A6
16 A5
17 A4
18 A3
19 A2
20 BE3#
21 BE2#
22 BE1#
23 BE0#
24 W/R#
25 D/C#
26 M/IO#
27 ADS#
28 HLDA
29 READYI#
30 CLK2IN
31 VDDR
32 CLK2
33 VSSR
34 READYO#
35 SLP#/MISS
36 RESCPU
37 ERROR386#
38 BUSY386#
39 PEREQ386
40 HRQ
41 RES387
42 BUSY387#
43 ERROR387#
44 TCLK2
45 VSSR
46 PEREQ387
47 WTKIRQ
48 TRI#
49 SLEEP1
50 SLEEP2
51 SLEEP3
52 MDLAT#
53 DEN#
54 ROMCS#
55 RAMW#
56 VSSR
57 MA10
58 MA9
59 VDDR
60 MA8
61 VSSR
62 MA7
63 MA6
64 MA5
65 VSSR
66 MA4
67 MA3
68 VDDR
69 MA2
70 VSSR
71 MA1
72 MA0
73 LBE3
74 LBE2
75 LBE1
76 LBE0
77 CASBK3
78 CASBK2
79 CASBK1
80 CASBK0
81 RASBK3#
82 RASBK2#
83 RASBK1#
84 VSSR
85 RASBK0#
86 XD7
87 XD6
88 XD5
89 XD4
90 XD3
91 XD2
92 XD1
93 XD0
94 BLKA20#
95 BRDRAM#
96 VDDR
97 VSSR
98 BUSCLK
99 DMAHLDA
100 IRQ13
101 CHM/IO#
102 CHS1#/MR#
103 CHS0#/MW#
104 CHREADY#
105 DMAHRQ
106 BUSOSC
107 OUT1
108 VSSR
109 REFRESH#
110 OSC
111 RSTDRV
112 IOR#
113 IOW#
114 RC#
115 TURBO
116 A20GATE
117 A31
118 A29
119 A26
120 A25
121 A24
122 A23
123 A22
124 A21
125 A20
126 A19
127 A18
128 VSSI
240842-3

SIGNAL DESCRIPTIONS

Signal Name	Pin Number	Signal Type	Signal Description
CPU INTERFACE SIGNALS			
A31, A29, A26	119–117	I-TPU	A26 is used to prevent aliasing above 64 MByte. A29 is used to separate upper BIOS accesses from Weitek 3167 accesses. A 31 is used to determine 387DX accesses. These address lines are driven only by the CPU. When HLDA is active, these signals are held low internally. Since externally these pins are three-stated, this is required in order to prevent errant Bus Master and DMA accesses to on-board memory.
A25–2	120–127, 2–7, 9–13, 15–19	I-TTL	Address bits driven by the CPU when it is Bus Master. They are driven by the 82344 Bus Controller whenever HLDA is active. These bits allow direct access of up to 64 Mbytes of memory.
BE3# – BE0#	20–23	I-TTL	**BYTE ENABLES 3 THROUGH 0, ACTIVE LOW:** These signals are driven by the CPU or the 82344.
W/R#	24	I-TPU	WRITE or active low READ enable driven by the CPU. W/R# is decoded with the remaining CPU control signals to indicate the type of bus cycle requested. The bus cycle types include: Interrupt Acknowledge, Halt, Shutdown, I/O Reads and Writes, Memory Data Reads and Writes, and Memory Code reads. WR# is internally pulled up.
D/C#	25	I-TPU	DATA or active low CODE enable driven by the CPU. D/C# is decoded with the remaining CPU control signals to indicate the type of bus cycle requested. See W/R# definition for bus cycle types. D/C# is internally pulled up.
M/IO#	26	I-TPU	MEMORY or active low I/O enable driven by the CPU. M/IO# is decoded with the remaining CPU control signals to indicate the type of bus cycle requested. See W/R# definition for bus cycle types. M/IO# is internally pulled up.
ADS#	27	I-TPU	**ADDRESS STROBE, ACTIVE LOW:** Driven by the CPU as an indicator that the address and control signals currently supplied by the CPU are valid. Ths signal is used internally to indicate that the data and command are valid and to determine the beginning of a memory cycle. ADS# is internally pulled up.
CLK2IN	30	I-CMOS	This is the main clock input to the System Controller and is connected to the CLK2 signal that is output by the System Controller. This signal is used internally to clock the System Controller's logic.
TCLK2	44	I-TTL	This input is connected to a crystal ocscillator whose frequency is equal to two times the system frequency. The TTL level oscillator output is converted internally to CMOS levels and sent to the CLK2 output.
CLK2	32	O	This output signal is a CMOS level converted TCLK2 signal. It is output to the CPU and other on-board logic for synchronization.
SLP#/MISS	35	IO-od	As a "power on reset" default, this bit is an output that reflects the inverse state of the SLEEP[7] configuration register bit. It is active low when sleep mode is active. Sleep mode is activated by setting SLEEP[7] = 1. When configuration register CTRL1(0) = 1, this pin becomes a MISS input for use with a future 82340 compatible product.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
CPU INTERFACE SIGNALS (Continued)			
READYO#	34	O	**READY OUT, ACTIVE LOW:** This signal is an indication that the current memory or I/O bus cycle is complete. It is generated from the internal DRAM controller or the synchronized version of CHREADY# for slot bus accesses. Outside the chip it is ORed with any other local bus I/O or master such as a coprocessor or cache controller. The culmination of these ORed READY signals is sent to the 386DX and is also connected to the System Controller's ReadyL# input.
READYI#	29	I-TTL	**READY INPUT, ACTIVE LOW:** This signal is the ORed READY signals from the coprocessor, cache controller, or other optional add-in device. See the READYO# description for more details on how the signal is used inside the System Controller.
HLDA	28	I-TTL	**HOLD ACKNOWLEDGE, ACTIVE HIGH:** This signal is issued by the CPU in response to the HRQ driven by the System Controller. It indicates that the CPU is floating its outputs to the high impedance state so that another master can take control of the bus. When HLDA is active, the memory control is generated from CHS1#/MR# and CHS0#/MW# rather than CPU status signals.
HRQ	40	O	**HOLD REQUEST, ACTIVE HIGH:** Driven by the System Controller to the CPU, this output indicates that a bus master, such as a DMA or AT channel master, is requesting control of the bus. HRQ is a result of the DMAHRQ input or a coupled refresh cycle. It is synchronized to CLK2.
RESCPU	36	O	**RESET CPU, ACTIVE HIGH:** This signal is sent to the CPU by the System Controller. It is issued in response to the control bit for software reset located in the Port A register or a dummy read to I/O port EFh. It is also issued in response to signals on the RSTDRV or RC inputs and in response to System Controller detection of a shutdown command. In all cases, it is synchronized to CLK2.
ERROR386#	37	O	**ERROR 386, ACTIVE LOW:** This signal is sent to the 386DX. On any CPU reset it is pulled low to set the 386DX to 32-bit coprocessor interface mode.
BUSY386#	38	O	**BUSY 386, ACTIVE LOW:** This signal is sent to the 386DX. The state of BUSY387# is always passed through to BUSY387# indicating that the 387DX is processing a command. On occurrence of an ERROR387# signal, it is latched and held active until an occurrence of a write to ports F0h, F1h, or RES387. The former case is the normal mechanism used to reset the active latched signal. The latter two are resets. Since ERROR387# generates IRQ13 for PC/AT-compatibility, BUSY386# is held active to prevent software access of the 387DX until the interrupt service routine writes F0h. The System Controller also activates BUSY386# for 16 CLK2 cycles when no 387DX is connected and I/O writes to the coprocessor space are detected.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
CPU INTERFACE SIGNALS (Continued)			
PEREQ386	39	O	**PROCESSOR EXTENSION REQUEST 386, ACTIVE HIGH:** Sent to the CPU in response to a PEREQ387, which is issued by the coprocessor to the System Controller. It indicates to the CPU that the coprocessor is requesting a data operand to be sent to or from memory by the CPU. For PC/AT-compatiblity, PEREQ386 is returned active on occurrence of ERROR387# after BUSY387# has gone inactive. A write to F0h by the interrupt 13 handler returns control of the PEREQ386 signal to directly follow the PEREQ387 input.
ON-BOARD MEMORY SYSTEM INTERFACE SIGNALS			
RAMW#	55	O	**RAM ACTIVE LOW WRITE OR ACTIVE HIGH READ:** Output to the 82345 Data Buffer and DRAM memory to control the direction of data flow of the on-board memory. It is a result of the address and bus control decode. It is active during memory write cycles and is high at all other times.
MA10–MA0	57, 58, 60, 62–64, 66, 67, 69, 71, 72	O	**MEMORY ADDRESSES 10 THROUGH 0:** These address bits are the row and column addresses sent to on-board memory. They are buffered and multiplexed versions of the bus master addresses. Along with LBE3–LBE0 they allow addressing of up to 16 MBytes per bank.
RASBK3# – RASBK0#	81–83, 85	O	**ROW ADDRESS STROBE BANK 0 THROUGH 3, ACTIVE LOW:** These signals are sent to their respective RAM banks to strobe in the row address during on-board memory bus cycles. The active period for this signal is completely programmable.
CASBK3– CASBK0	77–80	O	**COLUMN ADDRESS STROBE BANK 0 THROUGH 3:** These signals are the respective column address strobes for each of the banks. These signals are externally gated (NAND) with the LBE signals to generate the CAS# strobes for each byte of a DRAM memory bank.
LBE3–LBE0	73–76	O	**LATCHED BYTE ENABLE 0 THROUGH 3, ACTIVE HIGH:** These signals select one of four banks to access memory data from when an on-board memory access is activated. They are the latched version of the CPU's BE3# –BE0# signals when the CPU is bus master or is the latched version of SA1, SA0, and BHE# when the master or DMA is in control.
REFRESH#	109	I-CMOS/ O-OD	**REFRESH SIGNAL, ACTIVE LOW:** This output is used by the System Controller to initiate an off-board DRAM refresh operation in coupled refresh mode. In decoupled mode, the Bus Controller drives refresh active to indicate to the System Controller that it has decoded a refresh request command and is initiating an off-board refresh cycle.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
ON-BOARD MEMORY SYSTEM INTERFCE SIGNALS (Continued)			
ROMCS#	54	O	**ROM CHIP SELECT:** This output is active in CPU mode only (CPUHLDA is negated). It is active anytime the address on the A bus selects the address range between AFFE0000–AFFFFFFF, BFFE0000-BFFFFFFF, EFFE0000–EFFFFFFF, or FFFE0000h–FFFFFFFFh. It is also active during a memory read of 000E0000h–000FFFFFh when RAM-MAP[7] = 1. On reset, it also decodes the middle BIOS space between 00FE0000h-00FFFFFFh. However, this decode space can be changed via internal configuration register to System Board DRAM space after RESET if desired. NOTE: The lower ROM are from 00FE0000h–00FFFFFFh is impacted by shadow and/or EMS. Any 16k segments for which EMS is active (00EXXXXh only) or for which the shadow code had been changed from its 00b default are mapped out of the -ROMCS space.
COPROCESSOR SIGNALS			
PEREQ387	46	I-TPD	**COPROCESSOR EXTENSION REQUEST, ACTIVE HIGH:** this input signal is driven by the coprocessor and indicates that it needs transfer of data operands to or from memory. For PC/AT-compatibility, this signal is gated with the internal ERROR/BUSY control logic before being output to the CPU as PEREQ386.
ERROR387#	43	I-TPU	This is an active low numerics signal which is driven by the coprocessor to indicate that an error has occurred in the previous instruction. This signal is decoded internally with BUSY387# to produce IRQ13.
BUSY387#	42	I-TPU	This is an active low numerics input signal which is driven by the coprocessor to indicate that it is currently executing a previous instruction and is not ready to accept another. This signal is decoded internally to produce IRQ13 and to control PEREQ386 and BUSY386#.
RES387	41	O	**RESET 387, ACTIVE HIGH:** This output is connected to the 387DX reset input. It is triggered through an internally generated system reset or via a write to port F1h. In the case of a system reset, the CPURESET signal is also activated. A write to port F1h only resets the coprocessor. A software FNINT signal must occur after an F1h generated reset before the coprocessor is reset to the same internal state that a 287 is put into by a hardware reset alone. For, compatibility, the F1h reset may be disabled by setting bit 6 of MISCSET to 1.
WTKIRQ	47	I-TPD	**WEITEK 3167 INTERRUPT REQUEST, ACTIVE HIGH:** An input from the Weitek 3167 coprocessor.
IRQ13	100	O	**INTERRUPT REQUEST 13, ACTIVE HIGH:** This signal is driven to the Bus Controller to indicate than an error has occurred within the coprocessor. This signal is a decode of the BUSY387# and ERROR387# inputs ORed with the WTKIRQ input.
BUS CONTROL SIGNALS			
CHREADY#	104	I-CMOS	**CHANNEL READY, ACTIVE LOW:** This signal is issued by the Bus Controller as an indication that the current channel bus cycle is complete. This signal is synchronized internally then combined with ready signals from the coprocessor and DRAM controller to form the final version of READYO# which is sent to the CPU.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
BUS CONTROL SIGNALS (Continued)			
CHS0#/MW#	103	IO-TTL	**CHANNEL SELECT 0/MEMORY WRITE, ACTIVE LOW:** This signal is a decode of the 386DX's bus control signals and is sent to the Bus Controller. When combined with CHS1# and CHM/IO# and decoded, the bus cycle type is defined for the Bus Controller. Activation of CPUHLDA reverses this signal to become an input from the Bus Controller. It is then a MEMW# signal for DMA or bus master access to system memory.
CHS1#/MR#	102	IO-TTL	**CHANNEL SELECT 1/MEMORY READ, ACTIVE LOW:** This signal is a decode of the 386DX's bus control signals and is sent to the Bus Controller. When combined with CHS0# and CHM/IO# and decoded, the bus cycle type is defined for the Bus Controller. Activation of CPUHLDA reverses this signal to become an input from the Bus Controller. It is the a MEMR# signal for DMA or bus master access to system memory.
CHM/-IO	101	O	**CHANNEL MEMORY/ACTIVE LOW IO:** A decode of the M/IO# signal sent by the CPU to the System Controller. It is an indicator that the current bus cycle is a channel access. When combined with CHS0#, and CHM/IO# and decoded, the bus cycle type is defined for the Bus Controller.
BLKA20#	94	O	**BLOCK A20, ACTIVE LOW:** Driven to the Bus Controller to deactivate address bit 20. It is a decode of the A20GATE signal and Port A bit 1 indicating the dividing line of the 1 MByte memory boundary. Port A bit 1 may be directly written or set by a dummy read of I/O port EEh. BLKA20# is forced high when HLDA is active. (Refer to the "Sleep Mode Control Subsystem" section.)
BUSOSC	106	I-TTL	**BUS OSCILLATOR:** This signal is supplied from an external oscillator. It is supplied to the Bus Controller when the System Controller's internal configuration registers are set for asynchronous slot bus mode. This signal is two times the AT bus clock speed (SYSCLK).
BUSCLK	98	O-TTL	**BUS CLOCK:** This is the source clock used by the Bus Controller to drive the slot bus. It is two times the AT bus clock (SYSCLK). It is a programmable division from CLK2 or BUSOSC when in a synchronous bus mode.
DMAHRQ	105	I-CMOS	**DMA HOLD REQUEST, ACTIVE HIGH:** This input is sent by the Bus Controller, it is internally synchronized by the System Controller before it is sent out to the CPU as the HRQ signal. It is the indicator of the DMA controller or an other bus masters' desire to control the bus.
DMAHLDA	99	O	**DMA HOLD ACKNOWLEDGE, ACTIVE HIGH:** This output to the Bus controller indicates that the current hold acknowledge state is for the DMA controller or an other bus master.
BRDRAM#	95	O	**BOARD DRAM, ACTIVE LOW:** An output to Bus Controller and Data Buffer to indicate that on-board DRAM is being addressed.
OUT1	107	I-CMOS	Indicate a refresh request from the Bus Controller.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
PERIPHERAL INTERFACE SIGNALS			
A20GATE	116	I-TTL	**ADDRESS BIT 20 ENABLE:** This is an input from the keyboard controller and is used internally along with Port A bit 1 to determine if address bit 20 from the CPU is true or gated low. It also determines the state of BLKA20#.
TURBO	115	I-TTL	**TURBO, ACTIVE HIGH:** This input to the System Controller determines the speed at which the system board operates. It is normally the externally ANDed signal from the keyboard controller and a turbo switch. It is internally ANDed with a software settable latch. When high, operation is full speed. When low, CLK2 is divided by the value coded in configuration register MISCSET. A range is provided that allows slow operation at or below 8 MHz for any valid CPU speed. Slow speed takes precedence. When any one request for slow mode is present, slow mode is active. Turbo mode is active only when all TURBO requests are active.
RC#	114	I-TTL	**RESET CONTROL, ACTIVE LOW:** The falling edge of this signal causes a RESCPU signal. RC# is generated by the keyboard controller and its inverse is ORed with Port A bit 0 to form RESCPU.
SLEEP1	49	O-OD	**SLEEP SIGNAL 1, ACTIVE HIGH:** This pin is the logical OR of the enable and external control bits (bits 1 and 7) of the sleep indexed configuration register. It can be used with external interface logic to control external devices. The pin is always active while in sleep mode but can also be controlled via software when sleep mode is inactive. It is pulled low when inactive and three-states when active. An external pull-up is required. This allows an external interface to control logic operation at voltages different than VDD.
SLEEP2	50	O-OD	**SLEEP SIGNAL 2, ACTIVE HIGH:** This pin is the logical OR of the enable and external control bits (bits 2 and 7) of the sleep indexed configuration register. It can be used with external interface logic to control external devices. The pin is always active while in sleep mode but can also be controlled via software when sleep mode is inactive. It is pulled low when inactive and three-states when active. An external pull-up is required. This allows an external interface to control logic operation at voltages different than VDD.
SLEEP3	51	O-OD	SLEEP SIGNAL 3, ACTIVE HIGH: This pin is the logical OR of the enable and external control bits (bits 3 and 7) of the Sleep indexed configuration register. It can be used with external interface logic to control external devices. The pin is always active while in sleep mode but can also be controlled via software when sleep mode is inactive. It is pulled low when inactive and three-states when active. An external pull-up is required. This allows an external interface to control logic operation at voltages different than VDD.

SIGNAL DESCRIPTIONS (Continued)

Signal Name	Pin Number	Signal Type	Signal Description
BUS INTERFACE SIGNALS			
XD7–XD0	86–93	IO–TTL	**PERIPHERAL DATA BUS:** This bus is used to read and write the internal configuration registers.
DEN#	53	O	**DATA ENABLE, ACTIVE LOW:** This signal is an output to the 82345 Data Buffer to enable data transfers on the local bus. This signal is low during any CPU read cycles or INTA cycles.
IOR#	112	I-TTL	**I/O READ CYCLE, ACTIVE LOW:** Driven by the Bus Controller to indicate to the 82346 that an I/O read cycle is occurring on the bus. Whenever an I/O cycle occurs, the memory interface signals are inactive.
IOW#	113	I-TTL	**I/O WRITE CYCLE, ACTIVE LOW:** Driven by the Bus Controller to indicate to the 82346 that an I/O write cycle is occurring on the bus. Whenever an I/O cycle occurs, the memory interface signals are inactive.
RSTDRV	111	I-TTL	**RESET DRIVE, ACTIVE HIGH:** This reset signal is output by the Bus Controller. It indicates that a hardware reset signal has been activated. This is the same signal which is output to the channel. This signal is used to reset internal logic and to derive the RESCPU which is output by the System Controller.
MDLAT#	52	O	**MEMORY DATA BUS LATCH:** This is an output signal to the Data Buffer. On the rising edge, the Data Buffer latches the memory data bus. MDLAT# is low anytime one of the CASBK signals is high. When low, the Data Buffer latches are transparent.
OSC	110	I-TTL	**OSCILLATOR:** This is the buffered input of the external 14.318 MHz oscillator.
TEST MODE PIN			
TRI#	48	I1	**THREE-STATE:** This pin is used to drive all outputs to a high impedance state. When TRI# is low, all outputs and bidirectional pins are three-stated.
POWER AND GROUND PINS			
The power connections are split into an internal supply for the core-logic, and a pad-ring supply for the I/O drivers. Each supply should be individually bypassed with decoupling capacitors.			
VDDR	1, 31, 59, 68, 96	PWR	Pad-ring power connection, nominally +5V. These pins along with the VSSR pins should be separately bypassed.
VSSR	14, 33, 45, 56, 61, 65, 70, 84, 97 108	GND	Pad-ring ground connection, nominally 0V. These pins along with the VDDR pins should be separately bypassed.
VDDI	8	PWR	Internal core-logic power connection, nominally +5V. This pin along with the VSSI pin should be separately bypassed.
VSSI	128	GND	Internal core-logic ground connection, nominally 0V. This pin along with the VDDI pin should be separately bypassed.

Signal Type Legend

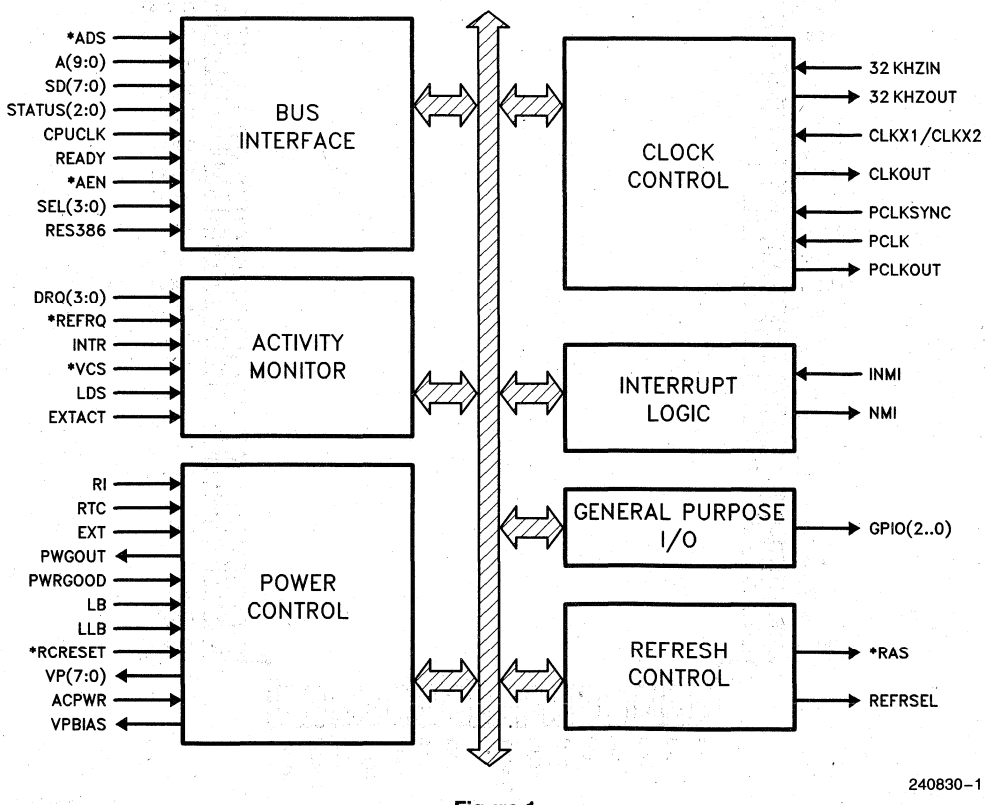
Signal Code	Signal Type
I-TTL	TTL Level Input
I-TPD	Input with 30 kΩ Pull-Down Resistor
I-TPU	Input with 30 kΩ Pull-Up Resistor
I-TSPU	Schmitt-Trigger Input with 30 kΩ Pull-Up Resistor
I-CMOS	CMOS Level Input
IO-TTL	TTL Level Input/Output
IT-OD	TTL Level Input/Open

Signal Code	Signal Type
IO-OD	Input or Open Drain, Slow Turn On
O	CMOS and TTL Level Compatible Output
O-TTL	TTL Level Output
O-TS	Three-State Level Output
I1	Input used for Testing Purposes
GND	Ground
PWR	Power

82347
POWER MANAGEMENT PC PERIPHERAL

- Supports Multiple Power Savings Modes
  - — DOZE Mode
    Entered after 0.125 to 14 Seconds of Inactivity
  - — SLEEP Mode
    Entered after 1 to 15 Minutes of Inactivity
  - — SUSPEND Mode
    Entered into by User Command or 5 to 75 Minutes of Inactivity
- Multiple Independent Device Power Shutdown Support in Each Mode through 9 Power Control Outputs
- CPU and Peripheral Clock Control
- Independent Programmable Timeouts for All Modes
- Power-On Control via Pushbutton, RTC Clock Alarm or Modem Ring-In
- Power UP/DOWN Power Sequencing for LCD Panel
- Multiple Low Battery Monitoring
- Refresh Support for Normal and Slow REFRESH DRAM during SUSPEND Mode. Supports Static and Pseudo Static RAMs
- Supports 386™ DX and 386™ SX Processors
- Packaged in 80 Pin Plastic Flat Pack

82347 Internal Block Diagram



240830-1

Figure 1

October 1990
Order Number: 240830-001

INTRODUCTION

The Intel 82347 is a power management unit designed to provide power management for PC compatible systems. The 82347 provides a system activity monitor and timers to determine when the system is idle. When idle, the 82347 can remove power from unused devices. The 82347 is also capable of reducing the CPU oscillator frequency to achieve additional power savings. The 82347 supports DRAM refresh to maintain system memory during system power down. The 82347 is packaged in an 80-pin Flat Pack and is fabricated with 1.5 micron CMOS technology.

VIP DESIGN

The capabilities of the 82347 are fully supported by the VIP design. For information on supported features see VIP8000SX Technical Specification.

NOTE:
The 82347 implements proprietary power management technology for which Vadem has pending patents.

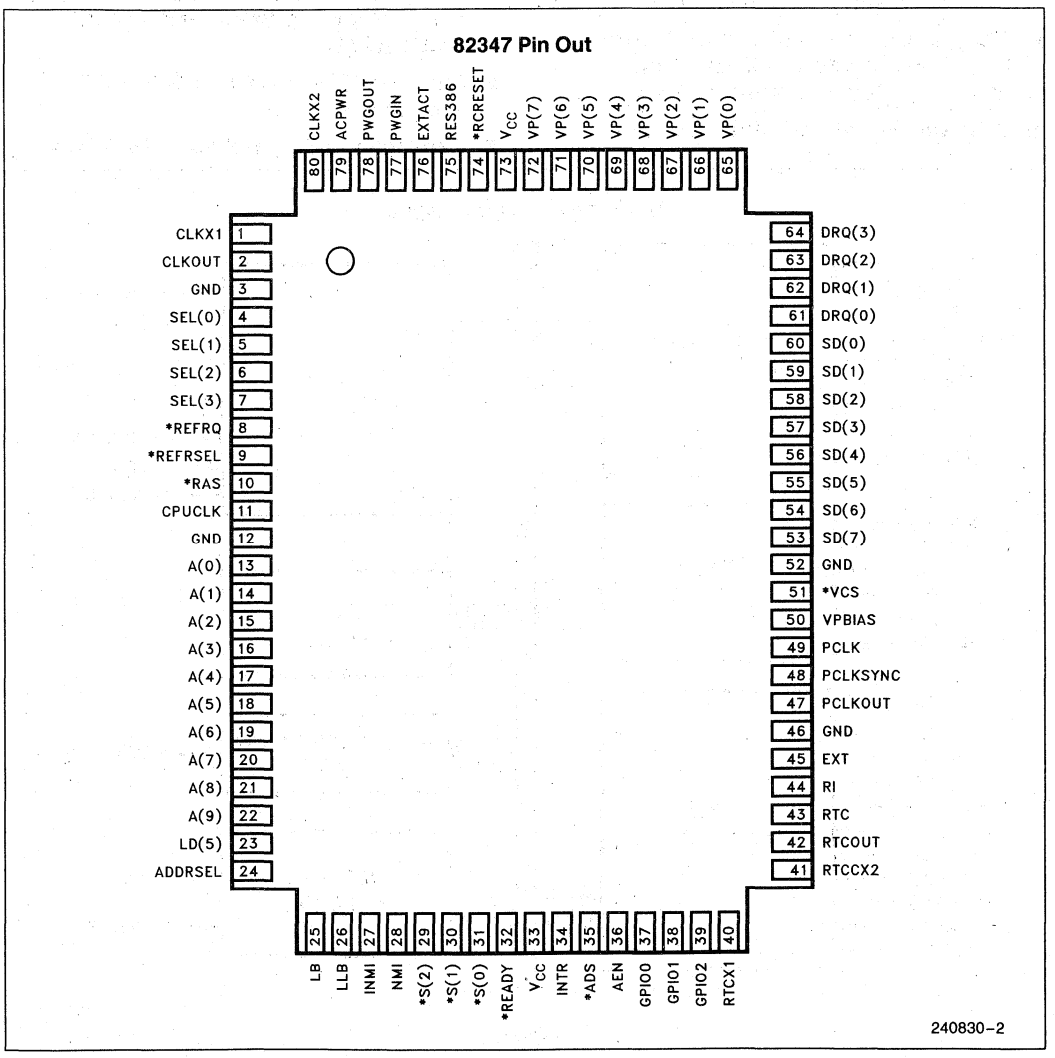


Figure 2

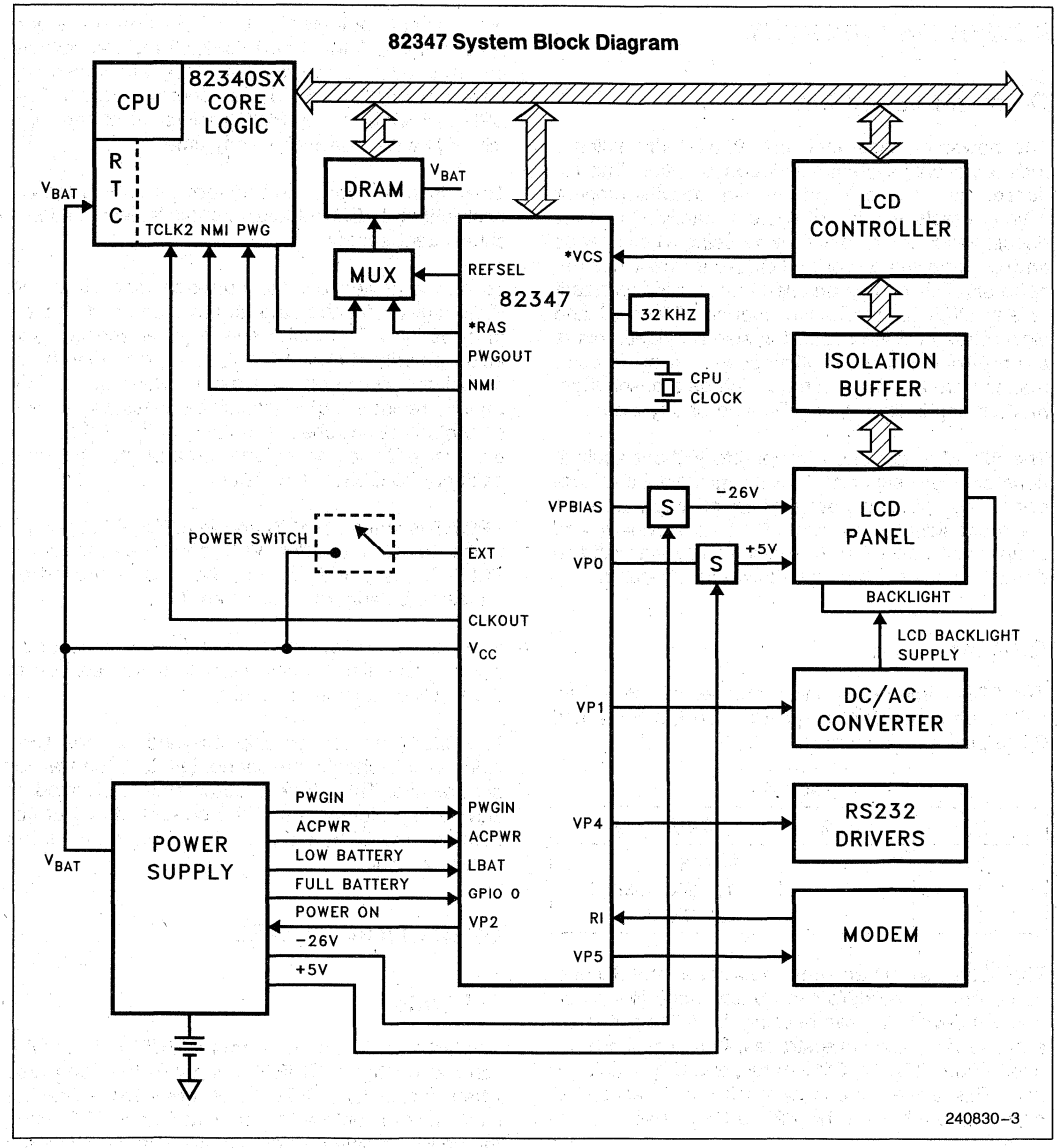

82347 System Block Diagram
CPU
82340SX CORE LOGIC
R T C
TCLK2 NMI PWG
V BAT
DRAM
V BAT
MUX
REFSEL
82347
*RAS
PWGOUT
NMI
*VCS
32 KHZ
CPU CLOCK
LCD CONTROLLER
ISOLATION BUFFER
LCD PANEL
BACKLIGHT
LCD BACKLIGHT SUPPLY
DC/AC CONVERTER
VPBIAS
S
-26V
VP0
S
+5V
POWER SWITCH
EXT
CLKOUT
V CC
VP1
PWGIN
ACPWR
LBAT
GPIO 0
VP2
VP4
RI
VP5
POWER SUPPLY
PWGIN
ACPWR
LOW BATTERY
FULL BATTERY
POWER ON
-26V
+5V
V BAT
RS232 DRIVERS
MODEM
240830-3

Figure 3

POWER MANAGEMENT

General Description

The power management unit (82347) dramatically reduces overall system power consumption. This reduction is accomplished via an activity monitor which detects periods of system inactivity, and reduces power consumption by reducing clock speeds and/or removing power and clocks from various peripherals. This can be enabled to occur independent of any DOS or application support. Standard and slow refresh DRAM support is also provided. Inputs are provided which will allow power on or off commands from external sources such as a pushbutton, modem ring indicator, or RTC time of day alarm.

The 82347 is designed to operate without applications software support and with minimal BIOS support. It has its own power on reset, separate from the reset signal to the rest of the user system, and upon initial power on its registers are initialized to default values to provide baseline functionality.

CPU Interface

The 82347 interfaces with 4 classes of CPUs. The CPU type is selected by jumpers on the SEL2 and SEL3 pins, per the table:

Table 1

SEL3	SEL2	CPU Type
0	1	80286
1	0	386 SX, 386 DX
1	1	386 SX LP

The 82347 is connected to the Addres and SD buses so that its registers can be accessed like a normal I/O device. Instead of using the AT bus memory and I/O read and write strobes, bus cycles are decoded from READY, CPU status, and CPU clock signals. This allows the activity monitor to detect all I/O cycles, even those within 82340 SX chipset.

Some pins change function for different processor types. This is described in Table 18.

Functional Description

The 82347 has five operating states: ON, DOZE, SLEEP, SUSPEND, and OFF. Transition between these states can occur due to system inactivity, external input pins (suspend switch), or by direct CPU command. Each of the four states has a PWR register associated with it which indicates which of 8 power control outputs VP[0 . . . 7] will be active during that state. A MUX selects the 8 register outputs corresponding to the current state, and these will drive the VP[0 . . . 7] power control outputs. The CPU can write to any PWR register to change the mix of powered devices at any time.

There is an exception to the above involving the EL backlight and LCD outputs, and this is described in a subsequent section.

To turn a device on, the corresponding bit in the PWR register for the state(s) in which it is to be on must be high. The POLARITY register specifies the actual polarity of each VP output required to turn the associated switch on. The default value of the POLARITY register is 0FFh, which indicates a logic high to turn on the switches. The value of the VP[0 . . . 7] bits just prior to the polarity control may be read back on the OUTPUT register.

VPBIAS is slaved to VP0 and PCLKOUT. It goes on 8 ms to 16 ms after VP0 and goes off when PCLKOUT is gated off. Its polarity is controlled by the same polarity register bit as VP0.

In addition to switching power to devices, a peripheral clock can also be switched. This is described in the PCLK Switching section.

The 82347 has an on-chip oscillator for the CPU clock which should be connected to a crystal for best results. This clock is automatically stopped or slowed in various states. The controlled output appears on the CLKOUT pin, which in turn drives the core logic chip.

OPERATING STATES

ON State

The ON state is entered from the SUSPEND or OFF state when the *RCRESET input is low, and also when one of EXT, RTC or RI goes high. The ON state may be entered from DOZE or SLEEP when the activity monitor detects activity. In the ON state all power control outputs will be controlled by the PWRON register. Upon entering the ON state the DOZE timeout will be triggered. The EL and LCD timeouts will be retriggered when entering the ON state from SUSPEND or OFF.

The CCLK[0, 1] bits in the Control register can be used to slow the processor clock in the ON state. This allows the user to save power when running non-computationally intensive applications such as word processing.

When ACPWR is high, the power saving features of the 82347 are disabled. The CPU will run at full clock speed and CCLK[0, 1] will have no effect. The DOZE and LCD timers will be continuously retriggered and will never time out, thus the 82347 will remain in the ON state and the LCD will stay on. However, because EL backlighting is subject to wearout, the EL timeout will operate in the normal fashion.

DOZE State

The DOZE state is entered from the ON state when the activity monitor has not detected activity within the time specified by the DOZE register. In the DOZE state the power control outputs are controlled by the PWRDOZE register.

The clock will slow down or stop as described in the CPU clock control section which follows.

SLEEP State

The SLEEP state is entered when the 82347 has been in the DOZE state for the time specified by the SLEEP register. In the SLEEP state the CLKOUT behavior is the same as in DOZE. The power control outputs are controlled by the PWRSLEEP register.

Alternatively, the 82347 can be programmed to generate NMI and remain in DOZE instead of automatically entering SLEEP.

SUSPEND State

If BIOS software support is provided for SUSPEND, the 82347 can be programmed to generate NMI after the SUSPEND timeout elapses, or in response to activity on the EXT pin. The CPU can then write the code for SUSPEND to the STATUS register, and the 82347 will enter SUSPEND.

In no software support is provided, 0.5 seconds after the EXT input goes high the 82347 will automatically enter SUSPEND. Alternatively, the 82347 will enter SUSPEND after a falling edge on the PWGIN input. CLKOUT goes low asynchronously. The power control outputs are controlled by the PWRSUSPEND register. Only activity on the EXT, RI or RTC inputs can cause an exit from SUSPEND, and into the ON state. When power is restored, the 82347 will reset the CPU, which will then execute a warm start. DRAM refresh may be enabled in SUSPEND, as described in the Refresh section.

OFF State

The OFF state is entered when the CPU writes the code for OFF (0FFh) to the STATUS register.

The OFF state is meaningful only when the 82347 is powered from a battery while the rest of the system is turned off. This type of power connection is necessary only if the 82347 must awaken the system from the OFF state by activating VP outputs in response to transitions on the EXT input. If this function is not required, then the 82347 may be powered off when the system is powered off, and the OFF state as described below does not exist.

In this state all except battery backed devices in the computer are powered off. Only activity on the EXT, RI or RTC inputs can cause an exit from OFF, and the new state will be ON. The 82347 bus interface is inhibited.

ACTIVITY MONITOR

The activity monitor has three internal outputs. ELTRIG is a function of keyboard activity only and is used to retrigger the EL backlight timer. LCDTRIG is true for keyboard activity or video memory writes, and retriggers the LCD timer. ACTIVITY is a logical OR function of the EXTACT pin and a programmable selection of different activities. When active, this output returns the 82347 to the ON state and retriggers the DOZE timeout. The activity monitor does not respond to accesses to the 82347's own registers.

The activity monitor can monitor access to the following PC I/O devices:

Parallel I/O ports—R/W access to LPT1 and LPT2

LPT1—378H-37FH
LPT2—278H-27FH

Keyboard Port—Reads to 60H

Real Time Clock—R/W access to 70H and 71H

Serial I/O Ports—R/W access to COM1 through COM4

COM1—3F8H-3FFH
COM2—2F8H-2FFH
COM3—3E8H-3EFH
COM4—2E8H-2EFH

Access to Floppy Disk—R/W access to address 3F5H

Hard Disk Controller—R/W access to 320H–323H and 1F0H–1F8H

Video Memory Writes—*VCS input

Programmable I/O—Programmed by IORNG register

All of the above activity sources can be masked through the ACTMASK register (C3H). See Table 11 for bit assignments for this register.

A programmable input to the activity monitor allows the designer to monitor a non-standard I/O device for activity. This I/O address is specified through the IORNG register at offset C5H. See Table 13 for programming details.

CLOCKS

An external CMOS level 32.768 kHz crystal oscillator drives the RTCX1 input pin to the 82347. RTCX2 should be left open. This signal generates the timing for all timers in the 82347, and clocks for some of the synchronous circuits. It is expected to be running at all times while the 82347 is powered up. The 82347 provides an on-chip oscillator for use with an external 24 MHz to 40 MHz crystal for the CPU clock. This clock runs in ON, DOZE, and SLEEP.

CPU CLOCK CONTROL

The CCLK[0, 1] bits in the Control register can be used to slow CLKOUT in the ON state.

CLKOUT can be automatically slowed or stopped in DOZE and SLEEP. To control this feature, four jumper inputs are used to select the processor type and crystal frequency. It is necessary to specify these parameters in order for the 82347 to satisfy minimum clock speeds for non-static CPU's and provide adequate response time to DRAM refresh requests. The following table shows the nominal crystal frequency, the divisor applied to that frequency in DOZE and SLEEP, and the resulting CLKOUT frequency. The 82347 supports non-static versions of the 386, and only fully static versions of the 80C286. For non-static CPUs, CLKOUT is scaled as a function of processor type to meet the processor's minimum clock frequency. When static processors are operated with DRAM the minimum CPU clock rate is a function of the length of time required for the system to resond to refresh requests, and CLKOUT is scaled as a function of processor type to produce a minimum 4 μs CPU bus cycle. If both static RAM and a static CPU are used, the CPU may set the STATIC bit in the Control register. The 82347 will then stop the clock in DOZE and SLEEP.

In DOZE and SLEEP mode, CLKOUT is reduced to the value specified in Table 2. Interrupts or DMA requests will temporarily speed up the clock to normal speed. The enable request functions for interrupts and DMA are separate and CLKOUT is enabled when any event requests it, so that an interrupt handler will run to completion even if it is interrupted by a DMA request. If CLKOUT was enabled in response to NMI, eventually an EOI will occur in response to a timer interrupt, and CLKOUT will slow or stop at that time. These enable request functions are independent of the activity monitor and the ACTMASK register. Enabling CLKOUT does not cause the 82347 to leave DOZE or SLEEP, unless the activity monitor is subsequently triggered. If this occurs, the 82347 will enter the ON state and the enable request logic will be cleared. CLKOUT is controlled synchronously without glitches.

In addition, if the MSK_VIDM bit is set, video activity will speed up or enable CLKOUT in the same way as NMI.

TIMERS

There are 6 timers in the 82347. Each except the LB timer has a 4-bit register associated with it. Setting a timer register to 0 disables it; setting it to a nonzero value enables it. If enabled, certain timers are triggered by the transition to the ON state. Individual timers are also triggered by events specific to their functions. Some timers are retriggerable, timing out at a programmable time following the last trigger. One runs continuously, retriggering itself.

The DOZE timer is programmable from 1/8 to 1 second with a resolution of 1/8 second, and from 2 seconds to 14 seconds with a resolution of 2 seconds. The SLEEP timer is programmable from 1 minute to 15 minutes with a resolution of 1 minute. The SUSPEND timer is programmable from 5 to 75 minutes with a resolution of 5 minutes. The LCD and EL timers are programmable from 1 to 15 minutes with a resolution of one minute. The resolution of the timers will be no more than 1/8 of their programmed value.

The LCD timer and the EL timer are retriggerable and are triggered as described in the Activity Monitor section. The timer outputs are logically ANDed with the power control bits selected by the power control MUX. This provides the flexibility to turn these outputs off when the associated timers time out, or in any 82347 state.

The DOZE timer is retriggerable and is triggered by the activity monitor ACTIVITY output in the ON state, and triggers the transition to DOZE state when it times out.

The SLEEP timer is triggered when DOZE state is entered and is cleared when DOZE is exited. It either generates NMI or triggers the transition to SLEEP state when it times out.

The SUSPEND timer is triggered when the SLEEP state is entered and is cleared when SLEEP is exited. If unmasked, an NMI will be generated when it times out.

The LB timer is enabled in ON, DOZE and SLEEP. It is triggered within 15 seconds of when LB is first detected. If not masked, NMI is generated by the LB timer once per minute when it times out, until a period of one minute elapses during which LB remains continuously false. The NMI cause will be identified as an LB or LLB interrupt. Software can maintain a counter and display a message once per X interrupts. It can also monitor LLB and shut the computer down after Y interrupts with LLB true.

NMI

The 82347 logically OR's together a number of internally generated NMI requests to produce the NMI output. These requests can be masked by bits in the NMIMASK register. The INMI input comes from external NMI-generating logic such as the parity detector, and can be OR'ed with the internal NMI requests to generate NMI when not masked by the OS2 bit in the NMIMASK register. The NMI output generally goes to the CPU NMI input, except on systems where it must go to an IRQ. The NMI CAUSE code bits in the Status register indicate the cause of the NMI. An internally generated NMI is cleared by reading the NMIMASK register.

NMI may be generated to indicate a low battery. This is described in the Timers section.

If the MSKSLEEP bit is cleared, the 82347 will generate NMI when the SLEEP timer times out and remain in DOZE instead of entering SLEEP, and will also generate NMI when the activity monitor causes the 82347 to go back to ON.

NMI is also generated when the SUSPEND timer times out. Software can then save status and go to SUSPEND or OFF state.

A high on the EXT input while not in the OFF or SUSPEND state will generate NMI. Software can then save status and go to SUSPEND or OFF state. If the NMI is not serviced within ½ second, the 82347 assumes there is no software support for SUSPEND, and will enter the SUSPEND state anyway.

REFRESH IN SUSPEND STATE

Refresh pulses are generated automatically in SUSPEND if the system is not fully static (STATIC bit is low). When an event occurs that would put the 82347 in SUSPEND, the 82347 drives PWGOUT and $\overline{\text{REFRSEL}}$ low. $\overline{\text{REFRSEL}}$ controls a MUX which takes control of the RAM away from the CPU and gives it to the 82347, driving all $\overline{\text{CAS}}$ lines low and allowing the 82347 to drive all $\overline{\text{RAS}}$ lines. The 82347 enters SUSPEND and begins generating single $\overline{\text{RAS}}$ pulses. If the SLWREF bit in the Control register is low these pulses occur on each edge of the 32 kHz clock. If SLWREF is high, the pulses occur at ⅛ this rate.

Table 2

SEL[3 . . . 0]	XTAL Freq (MHz)	Divisor	CLKOUT (MHz)	Notes
0100	24	24	1	80C286
0101	32	32	1	80C286
0110	40	40	1	80C286
0111				Not Allowed
1000	24	3	8	386
1001	32	4	8	386-16
1010	40	5	8	386-20
1011				Not Allowed
1100	24	6	4	386LP
1101	32	8	4	386LP-16
1110	40	10	4	386LP-20
1111				Factory Test

The 82347 will continue to generate single $\overline{RAS}$ pulse until VP0 goes active. The BIOS must not attempt to access DRAM until it is certain that the 82347 has released control of the DRAM. After the first $\overline{REFRQ}$ pulse, which indicates the core logic is not controlling refresh, the 82347 will stop the $\overline{RAS}$ and $\overline{REFRSEL}$ will go high. The CPU can then access DRAM.

Pseudo static RAM refresh is also supported. When *REFRSEL goes low, *RAS can drive *RFSH low for auto refresh mode.

EXTERNAL WAKEUP INPUTS

RI is a rising edge sensitive input, intended for use with a modem ring indicator output. The number of rising edges required for this input to be recognized is specified in bits D[4 . . . 6] of the Control register. The default is one transition. If these bits are zero, the input is disabled. If enabled, a rising transition on this input will force the 82347 to the ON state.

RTC is an edge sensitive input, intended for use with a real time clock wakeup alarm. A rising or falling transition on this input will force the 82347 to the ON state.

EXT is a rising edge sensitive input, intended for use with an external pushbutton. A rising transition on this input while the 82347 is in OFF or SUSPEND will force the ON state. A transition in ON, DOZE or SLEEP will generate NMI.

EXT is internally debounced. A rising edge immediately generates NMI but only if EXT has been sampled low at least twice by a 32-Hz debounce clock prior to the rising edge. The 82347 will not respond to any activity on any wakeup input until after EXT has been sampled low twice by the debounce clock.

RESUME AND POWER ON

If the BIOS supports SUSPEND and RESUME, at power on the CPU must read the RESUME bit in the Status register. RESUME will be cleared if the startup is a cold start from OFF and will be set to indicate a warm start (RESUME) from SUSPEND. If RESUME is cleared the wakeup bits WU[0 . . . 1] in the Status register will be zero, otherwise they will indicate which external input caused the RESUME. The RESUME bit will be cleared after the Status register is read, if write access to the 82347 is enabled (see Register Description).

PCLK CLOCK SWITCHING

A circuit is provided to switch PCLK for peripherals. The unbuffered PCLK signal is applied to PCLKIN and is passed on to the PCLKOUT output when VP0 is on. When VP0 is off, PCLKOUT will go low synchronously as described below. In SUSPEND and OFF, PCLKOUT will go low asynchronously.

If it is necessary to stop PCLKOUT only when the peripheral is in a certain state, for example not accessing any memory, PCLKSYNC must be tied to a signal synchronous with PCLK which is high when PCLKOUT is allowed to stop, such as an active low chip select generated by that peripheral. When VP0 is off, PCLKSYNC is sampled on each falling edge of PCLK, and PCLKOUT will stop synchronously in the low state on the falling edge when PCLKSYNC is sampled high. PCLKOUT will start again if PCLKSYNC goes low or the VP0 is on. Note that if logic that is powered down by VP0 causes PCLKSYNC to go low again, PCLKOUT will begin following PCLK again. Generally this would happen if the circuit that receives PCLKOUT and generates PCLKSYNC is powered down in SLEEP.

PCLKSYNC must be tied to the 82347's V_{CC} if the peripheral is powered down by VP0 or if there is no restriction as to the time when PCLKOUT is allowed to stop. PCLKSYNC has no effect in SUSPEND and OFF. PCLK and PCLKSYNC should be connected to ground if not used.

LOW BATTERY DETECTION

The LB and LLB inputs indicate low battery and low low battery. The status of LB and LLB can be read in the SUPPLY register (both low if battery is good). A low battery indication can generate NMI as described in the timer section. If only one indication is available the other input must be grounded.

RESET AND POWER ON/OFF

The 82347 has its own private *RCRESET signal, typically from an external RC network. This signal resets only the 82347 when power is first applied to it.

The PWRGOOD signal from the power supply drives the PWGIN pin on the 82347. This is used to generate the PWGOUT signal, which drives the PWRGOOD input of the core logic chip. If the power supply does not provide PWRGOOD, PWGIN should be tied to the core logic V_{CC}.

When PWGIN goes low or the CPU commands the 82347 to enter OFF or SUSPEND, PWGOUT goes

low. If the system uses DRAM, before entering SUSPEND the 82347 will begin refreshing when PWGOUT goes low, and will enter SUSPEND within 500 μs.

When waking up from OFF or SUSPEND, the 82347 can power on all devices immediately except for the LCD display (VP0 and VPBIAS) and EL backlight (VP1). PWGOUT and PCLKOUT will also remain low. After PWGIN and *RCRESET are high, the 82347 starts a delay. After 0.5 to 1 second, VP0 turns on, and PCLK is enabled. PWGOUT then goes high. VPBIAS turns on 8 ms to 16 ms after VP0 turned on, and VP1 turns on 500 ms after VP0 turned on.

LEAKAGE CONTROL DURING SUSPEND AND OFF

When CPU power is turned off, many inputs to the 82347 could float, depending on system design. This could result in additional power consumption in the 82347 input buffers. Although the pin list shows these as inputs, in actuality some inputs are bidirectional. Under normal operation these are inputs only, but following a delay of 0.5 to 1 second after entering SUSPEND or OFF the 82347 will drive these input pins low to prevent them from floating. This delay allows the switched power to the devices that normally drive these pins to approach zero, thus preventing contention. These output drivers turn off immediately when entering the ON state to allow the system to function normally.

The following inputs become low driven outputs after entering SUSPEND or OFF:

A [0 . . . 9], SD[0 . . . 7], LD5, $\overline{\text{READY}}$, $\overline{\text{S}}$[0 . . . 2], CPUCLK, AEN, INMI, INTR, DRQ[0 . . . 3], $\overline{\text{REFRQ}}$, PCLK, RES386, $\overline{\text{ADS}}$ and EXTACT.

Some inputs are connected to external user circuits, and are not practical for the 82347 to drive. It is the responsibility of the system designer to verify that all inputs to powered CMOS devices be driven during all modes.

$\overline{\text{VCS}}$, PCLKSYNC, EXT, RI, RTC, PWGIN, ACPWR, GPIO[2 . . . 0], LB and LLB.

Other inputs do not present a problem. ADDRSEL and SEL [0 . . . 3] are hardwired to V_{CC} or ground. $\overline{\text{RCRESET}}$ is maintained at V_{CC} by an external pull-up resistor. RTCX1 and RTCX2 are biased properly internal to the 82347. In SUSPEND, the 82347 outputs VP[0 . . . 7] $\overline{\text{RAS}}$ and $\overline{\text{REFRSEL}}$ are in a logic state determined by their respective register programming. NMI, CLKOUT, and PCLKOUT are low. In OFF, all 82347 outputs are low, except for VP outputs that are programmed as high.

GENERAL PURPOSE I/O PINS

The 82347 has 3 general purpose pins that may be individually programmed as inputs, outputs, or bidirectionals. They default to inputs at power on. If not used they must be tied high or low to prevent leakage, alternatively they may be programmed as outputs to prevent leakage. These could be used for monitoring power supply status, EEPROM interfaces or as software controlled VP outputs, for example.

REGISTER DESCRIPTION

The 82347 is accessed via an indexed address and data register scheme. The index value is first written to the INDEX register and the data is read or written at the DATA register port. The ADDRSEL pin selects the port addresses per Table 3:

Table 3

ADDRSEL	Index Register	Data Register
0	26H	27H
1	178H	179H

The INDEX register will autoincrement after each write, to point to the next register. The counter only wraps around in the C0–DF range. After use, the index register should be left at a value outside this address range to prevent inadvertant access of these registers. The INDEX register is a write-only register. All data registers may be read and written, except as noted in the descriptions below.

Write access to the data registers is locked out when entering SUSPEND or OFF, and when power is first applied to the 82347. Bit D0 in the SUPPLY register will be set to indicate that write access is not allowed. To unlock write access, the CPU must first write to the INDEX register (C1H) and read the SUPPLY register.

THE DATA REGISTERS ARE:

Table 4

Register	Index
STATUS	C0H
SUPPLY	C1H
CONTROL	C2H
ACTMASK	C3H
NMIMASK	C4H
IORNG	C5H
PWRON	C6H
PWRDOZE	C7H
PWRSLEEP	C8H
PWRSUSPEND	C9H
POLARITY	CAH
OUTPUT	CBH
DOZE	CCH
SLEEP	CDH
SUSPEND	CEH
LCD	CFH
EL	D0H

Table 5. Status Register (COH)

Bit	Name	Function
D7	RESUME	Resuming from SUSPEND (Warmstart)
D6	WU1	Wakeup Code MSB
D5	WU0	Wakeup Code LSB
D4	NMI2	NMI cause Code
D3	NMI1	
D2	NMI0	
D1	STATE1	State MSB
D0	STATE0	State LSB

Only D0 and D1 are affected by a write. The CPU can write the state code to this register to put the 82347 in another state. Writing 0FFh puts the 82347 in the OFF state. The NMI cause, state and wakeup codes are decoded as follows:

Table 6

NMI		State		Wakeup	
Code	Cause	Code	Name	Code	Cause
000	None, or INMI	00	ON	00	
001	EXT Input	01	DOZE	01	EXT Input
010	LB	10	SLEEP	10	RTC Input
011	LLB Timeout	11	SUSPEND	11	RI Input
100	SLEEP Timeout				
101	SUSPEND Timeout				
110	SLEEP to ON (Activity)				

$\overline{\text{RCRESET}}$ clears all bits. D[2 . . . 4] are cleared when the NMIMASK register is read. D7 is cleared after the STATUS register is read.

Supply Register (CH1)

This register has different functions for read and write.

Table 7. READ to (C1H)

Bit	Name	Function
D7	ACPWR	
D6	GPIN2	
D5	GPIN1	
D4	GPIN0	
D3	ACTIVITY	System Activity Present
D2	LLB	Low Battery 2 (Second Warning)
D1	LB	Low Battery 1 (First Warning)
D0	LOCKOUT	82347 Register Write-Protected

Table 8. WRITE to (C1H)

Bit	Name	Default	Function
D7	Reserved	0	
D6	GPOUT2	0	General Purpose Outputs
D5	GPOUT1	0	
D4	GPOUT0	0	
D3	Reserved	0	
D2	GPEN2	0	General Purpose I/O Direction Control Bits, 1 = Output, 0 = Input
D1	GPEN1	0	
D0	GPEN0	0	

The GPIO[0 . . . 2] pins are programmed as inputs or outputs by D[0 . . . 2] and are read and written on D[4 . . . 6]. For read operations, D[0 . . . 2, 4 . . . 6] are driven directly by the input pins. D3 is set when system activity is detected and is cleared when this register is read.

Table 9. Control Register (C2H)

Bit	Name	Default	Function
D7	Reserved	0	
D6	RING2	0	RI Pulses Required for Turn On
D5	RING1	0	
D4	RING0	1	Default = 1
D3	STATIC	0	Static CPU and RAM, Clock Stops in DOZE and SLEEP
D2	SLWREF	0	1 = Slow Refresh DRAM
D1	CCLK1	0	CPUCLK Divisor to Slow Clock in ON Mode
D0	CCLK0	0	Default = Full Speed

The RING[0 . . . 2] bits are used to set the number of RI pulses required for turn on. The default value is 1 so that only one pulse is required for turn on. If set to 0, RI is disabled. The CCLK[0 . . . 1] bits select the clock divisor for CLKOUT in the ON mode, according to the table. Setting of CCLK[0 . . . 1] has no effect on the DOZE/SLEEP mode frequency.

Table 10

CCLK[0 . . . 1]	Divisor
0	1 (Full Speed)
1	2
2	4
3	8

Table 11. ACTMASK Register (C3H)

Bit	Name	Default	Function
D7	MSK_IORNG	1	Mask Access to 16 Ports at IORNG[0 . . . 5]
D6	MSK_VIDM	0	Mask Access to Video Memory
D5	MSK_HD	0	Mask Hard Disk Activity
D4	MSK_FLP	0	Mask Access to Port 3F5
D3	MSK_SIO	0	Mask Access to Com 1–4
D2	MSK_RTC	1	Mask Access to Port 70h, 71h
D1	MSK_KBD	0	Mask Keyboard Port 60H Reads
D0	MSK_PIO	0	Mask Access to LPT 1, 2

The activity monitor ACTIVITY output is the logical OR of all unmasked activity sources. This register affects only the ACTIVITY output.

NMIMASK Register (C4H)

This register masks the various NMI sources. In the default state only the INMI input can generate NMI.

Table 12

Bit	Name	Function	Default
D7	Reserved		
D6	OS2	Mask INMI Input	0
D5	MSK_SUSPEND	Mask SUSPEND Timeout	1
D4	MSK_SLEEP	Mask SLEEP Timeout	1
D3	MSK_LLB	Mask LLB Input	1
D2	MSK_LB	Mask LB Input	1
D1	MSK_EXT	Mask EXT Input	1

Table 13. IORNG Register (C5H)

Bit	Name	Default	Function
D7	RNGSIZE	0	1 = 8 Bytes, 0 = 16 Bytes
D6	IORNG6	0	
D5	IORNG5	0	
D4	IORNG4	0	
D3	IORNG3	0	Maskable I/O Range Base Address (Default = 0)
D2	IORNG2	0	
D1	IORNG1	0	
D0	IORNG0	0	

IORNG[0 . . . 6] are the base address bits A[3 . . . 9] for the maskable I/O port range in the activity monitor. RNGSIZE is the size of the range. IORNG0 is ignored when RNGSIZE is high.

PWR Registers

The bits in these registers D[0 . . . 7] coresond directly with the power control outputs VP[0 . . . 7]. In a particular state, the corresponding PWR register outputs control the VP pins. The exception is VP0 and VP1 which are LCD and EL power respectively. These outputs are logically ANDed with the LCD and EL timer outputs prior to driving the pins. All bits are then logically exclusive NOR'ed with the POLARITY register, and the result drives the pins. VPBIAS turns on 8 ms–16 ms after VP0 and turns off at the same time as VP0. The default values for these registers are as follows, where 1 indicates that the controlled device is on:

Table 14

Register	Default Value	Index
PWRON	FFh	C6H
PWRDOZE	FFh	C7H
PWRSLEEP	0Ch	C8H
PWRSUSPEND	00h	C9H

POLARITY Register (CAH)

This register controls the polarity of the VP outputs. If a logic low is required on a VP pin to turn the external device on, the corresponding bit in the POLARITY register must be low. If a high is required, set the bit high. The default value is 0FFH. The polarity of VPBIAS is the same as VP0.

OUTPUT Register (CBH)

The OUTPUT register is a read only register. For each VP[0 . . . 7] output that is on, the corresponding bit in the OUTPUT register will be equal to 1. The content of this register is sampled before the polarity register.

Timer Registers (CCH–D0H)

Loading a value into a timer register enables the timer and selects the timeout. All timer registers have 4 valid bits [0 . . . 3]. Data written to the upper bits [4 . . . 7] has no effect. The upper bits are 0 when read back. Except for the DOZE timer, all timer registers can be set for a timeout from 1 to 15 time units, where a unit is the resolution of the timer. A zero disables the timeout. Reading a timer register returns the value that was last written to it, not the actual time remaining. The default values are tabulated below:

Table 15

Timer	Range	Default	Index
DOZE	1/8–14 Sec.	4 Sec.	CCH
SLEEP	1–15 Min.	2 Min.	CDH
SUSPEND	5–75 Min.	0 (Disabled)	CEH
LCD	1–15 Min.	2 Min.	CFH
EL	1–15 Min.	2 Min.	D0H

The DOZE timer programming for different timeouts is tabulated below:

Table 16

D[3 . . . 0]	Time	D[3 . . . 0]	Time
0000	Disabled	1000	1 Sec.
0001	1/8 Sec.	1001	2 Sec.
0010	1/4 Sec.	1010	4 Sec.
0011	3/8 Sec.	1011	6 Sec.
0100	1/2 Sec.	1100	8 Sec.
0101	5/8 Sec.	1101	10 Sec.
0110	3/4 Sec.	1110	12 Sec.
0111	7/8 Sec.	1111	14 Sec.

SLEEP Register—Offset CDH

The SLEEP register is a 4-bit R/W register which contains the SLEEP timeout. The default value is 2 minutes (0010)

Table 17

D[0 . . . 3]	TIme	D[0 . . . 3]	TIme
0000	Disabled	1000	8 minute
0001	1 minute	1001	9 minute
0010	2 minute	1010	10 minute
0011	3 minute	1011	11 minute
0100	4 minute	1100	12 minute
0101	5 minute	1101	13 minute
0110	6 minute	1110	14 minute
0111	7 minute	1111	15 minute

SUSPEND Register—Offset CEH

The SUSPEND register is a 4-bit R/W register which contains the SUSPEND timeout. The default value is Disabled (0000)

Table 18

D[0 . . . 3]	Time	D[0 . . . 3]	Time
0000	Disabled	1000	40 minute
0001	5 minute	1001	45 minute
0010	10 minute	1010	50 minute
0011	15 minute	1011	55 minute
0100	20 minute	1100	60 minute
0101	25 minute	1101	65 minute
0110	30 minute	1110	70 minute
0111	35 minute	1111	75 minute

LCD Register—Offset CFH

The LCD register is a 4-bit READ/WRITE register which contains the LCD timeout. The default value is 2 minutes (0010)

Table 19

D[0 . . . 3]	Time	D[0 . . . 3]	Time
0000	Disabled	1000	8 minute
0001	1 minute	1001	9 minute
0010	2 minute	1010	10 minute
0011	3 minute	1011	11 minute
0100	4 minute	1100	12 minute
0101	5 minute	1101	13 minute
0110	6 minute	1110	14 minute
0111	7 minute	1111	15 minute

EL Register—Offset D0H

The EL register is a 4-bit R/W register which contains the EL timeout. The default value is 2 minutes (0010)

Table 20

D[0 . . . 3]	Time	D[0 . . . 3]	Time
0000	Disabled	1000	8 minute
0001	1 minute	1001	9 minute
0010	2 minute	1010	10 minute
0011	3 minute	1011	11 minute
0100	4 minute	1100	12 minute
0101	5 minute	1101	13 minute
0110	6 minute	1110	14 minute
0111	7 minute	1111	15 minute

Table 21. Pin List

Name	Type	Pin No.	Function
A[0 . . . 9]	I/O	13–22	Address Bus, connected to the processor or system address bus, depending on the type of CPU used. See Table 18 for assignments. These pins driven low during SUSPEND.
SD[0 . . . 7]	I/O	60–53	Bidirectional Tri-State data bus lines connected to the system data bus. These pins driven low during SUSPEND.
LD[5]	I/O	23	Local CPU data bus bit 5. Connected to the CPU data bus. Driven Low during SUSPEND.
$\overline{\text{READY}}$	I/O	32	CPU Ready Input. This pin driven low during SUSPEND.
$\overline{\text{S[0 . . . 2]}}$	I/O	31–29	CPU Status Inputs. See Table 18 for assignments. These pins are driven low during SUSPEND.
CPUCLK	I/O	11	CPU clock output from core chip set. This is the same clock going to the CPU clock input. This pin driven low during SUSPEND.
SEL[0 . . . 3]	I	4–7	CPU type and frequency select jumper. Used to select the type and operating frequency of the CPU (see Table 2).
ADDRSEL	I	24	Selects the I/O address for the 82347 index and data registers. A low on this line selects I/O address 26H and 27H. A high selects I/O address 178H and 179H.
VP0	O	65	LCD Power Control. Used to control the LCD panel logic power. Defaults to High = On.
VP1	O	66	LCD EL Backlight Power Control. Defaults to High = On.
VP[2 . . . 7]	O	67–72	General Purpose Peripheral Power Control. Defaults to High = On.
VPBIAS	O	59	Power Control for LDB Bias Supply. Defaults to High = On.
$\overline{\text{RAS}}$	O	10	Row address strobe signal for DRAM backup during SUSPEND mode.
$\overline{\text{VCS}}$	I	51	Video RAM chip select, connected to video RAM chip select to trigger activity monitor.
AEN	I/O	36	AEN from Core Logic. This pin driven low during SUSPEND mode.
NMI	O	28	Non Maskable Interrupt Output to CPU. This pin driven low during SUSPEND mode.
INTR	I/O	34	Interrupt Request Output of 8259. Connect to CPU IRQ input. Driven low during SUSPEND.
DRQ[0 . . . 3]	I/O	61–64	System DMA Requests. These pins driven low during SUSPEND.
$\overline{\text{REFRQ}}$	I/O	8	Indicates Refresh DMA Cycle. This pin driven low during SUSPEND.
EXT	I	45	External Command Input. Typically a pushbutton switch. See text for more description.
RI	I	44	Ring Indicator from External Modem. Can be used to put the system into the ON state from OFF or SUSPEND state.
RTC	I	43	Real Time Clock Alarm. Can be used to put the system into the ON state from OFF or SUSPEND state.
CLKX1	I	1	CPU Oscillator or Crystal Input. Driven high during SUSPEND.
CLKX2	O	80	CPU Crystal Feedback Loop.
CLKOUT	O	2	Clock Out to Core Chip Set. CLKOUT is the clock controlled by the internal clock divider. This pin is driven low during SUSPEND.

Table 21. Pin List (Continued)

Name	Type	Pin No.	Function
LB	I	25	Low Battery Detect—First Warning.
LLB	I	26	Low Battery Detect—Second Warning.
$\overline{\text{RCRESET}}$	I	74	Used to initialize the 82347 on initial power up. Connected an external RC reset network. A 1 μF and 100k resistor should be used and connected to the backup power.
$\overline{\text{REFRSEL}}$	O	9	Signal used to gate 82347 *RAS to DRAMs during SUSPEND mode.
RTCX1	I	40	External oscillator input for 32.768 kHz.
RTCX2	O	41	
RTCOUT	O	42	General purpose 32.768 kHz output. Can be used to drive system Real Time clock.
PCLK	I/O	49	Input for 82347 controlled peripheral clock. This clock is controlled internally by VP0. See description in the text for more information. In the clock switching feature is not used, this input should be tied low. This pin driven low during SUSPEND.
PCLKSYNC	I	48	Sync signal for switched peripheral clock. See description in the text for more information. If the clock switching feature is not used, this input should be timed low.
PCLKOUT	O	47	Output from 82347 controlled peripheral clock.
RES386	I/O	75	Connected to the 386 DX/386 SX reset input from core chip. This pin is driven low during SUSPEND.
$\overline{\text{ADS}}$	I/O	35	Address strobe from 386 DX/386 SX. This pin is driven low during SUSPEND.
PWGOUT	O	78	Power good signal to core chip logic. This pin is driven low during SUSPEND
PWGIN	I	77	Power good signal from power supply.
EXTACT	I/O	76	General purpose external activity signal input to the activity monitor. This pin is driven low during SUSPEND. If not used this pin should be tied low.
ACPWR	I	79	Indicates the A.C. power is connected. This signal will disable the internal activity timers. However this will not disable the LCD backlight timer. If not used this pin should be tied low.
GPIO[0 . . . 2]	I/O	37–39	General purpose I/O pins. These pins can be used as additional power controls or used as general purpose I/O controls. Default is set to input. If not used this pin should be tied low.
INMI	I	27	NMI input from core logic.
V_{CC}		33, 73	Power (2 Pins)
GND		3, 12, 46, 52	Ground (4 Pins)

Table of Alternate Pin Functions

Table 22

Pin	80286	386™
$\overline{S0}$	$\overline{S0}$	D/$\overline{C}$
$\overline{S1}$	$\overline{S1}$	W/$\overline{R}$
$\overline{S2}$	M/$\overline{IO}$	M/IO
$\overline{READY}$	$\overline{READY}$	$\overline{READY}$
A0	SA0	BLE
A1–A9	SA1–SA9	A1–A9
RES386	Ground	RESET
$\overline{ADS}$	Ground	$\overline{ADS}$

ABSOLUTE MAXIMUM RATINGS*

(T_A = 25°C)

Power Supply Voltage (V_{CC}) ..... −0.3V to +7.0V

Input Voltage (V_I) ..... −0.3V to V_{CC} + 0.3V

Input Current (I_{IN}) ..... ±10 mA

Operating Temperature Range (T_{OP}) ..... −25°C to +85°C

Storage Temperature Range (T_{stg}) ..... −40°C to +125°C

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

RECOMMENDED OPERATING CONDITIONS

Table 23

Symbol	Parameter	Limits		Units
		Interfacing with TTL		
		Min	Max	
V_{CC}	Power Supply Voltage	4.75	5.25	V
T_{OPT}	Operating Temperature	0	70	°C
V_{ILT}	Low-Level Input Voltage (TTL)		0.8	V
V_{ILC}	Low-Level Input Voltage (CMOS)		1.5	V
V_{IHT}	High-Level Input Voltage (TTL)	2.0		V
V_{IHC}	High-Level Input Voltage (CMOS)	3.5		V
V_P	Positive Schmitt Trigger Voltage	1.4	2.2	V
V_N	Negative Schmitt Trigger Voltage	0.8		V
V_H	Hysteresis	0.4	0.5	V

D.C. CHARACTERISTICS

Table 24

Symbol	Parameter	Limits		Units	Test Conditions
		Min	Max		
V_{OL}	Low-Level Output Voltage		0.4	V	TTL and CMOS
V_{OH}	High-Level Output Voltage	2.4		V	TTL
I_{IL}	Low-Level Input Current	−10	10	μA	$V_{IN} = V_{SS}$
I_{IH1}	High-Level Input Current		10	μA	$V_{IN} = V_{CC}$
I_{DS}	Output Short Circuit Current		−100	mA	$V_O = 0V$
V_{IC}	Input Clamp Voltage		−1.5	V	$I_I = -20$ mA, $V_{CC} = 4.5V$
I_{OLZ1}	Output Leakage Current	−10	10	μA	Hi-Z
V_{ESD}	Electro-Static Voltage		2000	V	
$I_{STANDBY}$	Quiescent		100	μA	SUSPEND Mode
I_{OP}	Operating Current		40	mA	

82347 OUTPUT CURRENT DRIVE CAPABILITIES

Table 25

Pin Name	Pin Number	Type	I_{OL}	I_{OH}	C_{LOAD}
CLKX1	1	OSC-I			
CPUCLK	2	C/C	1 mA	1 mA	<50 pF
CLKOUT	3	C	12 mA	12 mA	<50 pF
SEL[0 . . . 3]	5–8	C			
$\overline{\text{REFRQ}}$	9	T/C	1 mA	1 mA	<50 pF
$\overline{\text{REFRSEL}}$	10	C	2 mA	2 mA	<50 pF
$\overline{\text{RAS}}$	11	C	6 mA	6 mA	<50 pF
A[0 . . . 9]	13–22	T/C	1 mA	1 mA	<50 pF
LD5	23	T/C	2 mA	2 mA	<50 pF
ADDRSEL	24	T			
LB	25	C			
LLB	26	C			
INMI	27	T/C	1 mA	1 mA	<50 pF
NMI	28	C	2 mA	2 mA	<50 pF
$\overline{\text{S[0 . . . 2]}}$	29–31	T/C	1 mA	1 mA	<50 pF
$\overline{\text{READY}}$	32	T/C	1 mA	1 mA	<50 pF

NOTES:
CS CMOS Schmitt Trigger Input
C/C CMOS Input/CMOS Output
T/C TTL Input/CMOS Output
C CMOS Level
T TTL Level
TS TTL Schmitt Trigger Input

82347 OUTPUT CURRENT DRIVE CAPABILITIES (Continued)

Table 25 (Continued)

Pin Name	Pin Number	Type	I_{OL}	I_{OH}	C_{LOAD}
INTR	34	T/C	1 mA	1 mA	<50 pF
$\overline{ADS}$	35	T/C	1 mA	1 mA	<50 pF
AEN	36	T/C	1 mA	1 mA	<50 pF
GPIO[0 . . . 2]	37–39	T/C	6 mA	6 mA	<50 pF
RTCX1	40	OSC-I			
RTCX2	41	OSC-O			
RTCOUT	42	C	3 mA	3 mA	<20 pF
RTC	43	C			
RI	44	TS			
EXT	45	C			
PCLKOUT	47	C	12 mA	12 mA	<50 pF
PCLKSYNC	48	T			
PCLK	49	T/C	1 mA	1 mA	<50 pF
VPBIAS	50	C	6 mA	6 mA	<50 pF
$\overline{VCS}$	51	T			
SD[7 . . . 0]	53–60	T/C	12 mA	12 mA	<200 pF
DRQ[0 . . . 3]	61–64	C	1 mA	1 mA	<50 pF
VP[0 . . . 7]	65–72	C	6 mA	6 mA	<50 pF
$\overline{RCRESET}$	74	CS			
RES386	75	T/C	1 mA	1 mA	<50 pF
EXTACT	76	T/C	1 mA	1 mA	<50 pF
PWGIN	77	CS			
PWGOUT	78	C	3 mA	3 mA	<50 pF
ACPWR	79	C			
CLKX2	80	OSC-O			<20 pF

NOTES:
CS CMOS Schmitt Trigger Input
C/C CMOS Input/CMOS Output
T/C TTL Input/CMOS Output
C CMOS Level
T TTL Level
TS TTL Schmitt Trigger Input

TIMING CHARACTERISTICS

Table 26

No.	Parameter	Symbol	Limits		Units
			Min	Max	
t98	CLKX1 Rise Time	tCR		5	ns
t99	CLKX1 Fall Time	tCF		5	ns
t100	Input CLKX1 High Time	tCH	10		ns
t101	Input CLKX1 Low Time	tCL	10		ns
t102	CLKOUT to CPUCLK Delay Time	tCCL	9 (20 MHz) 14 (16 MHz) 24 (12 MHz)		ns
t104	System Address Set-Up Time	tAS	9		ns
t105	System Address Hold Time	tAH	6		ns
t106	Write Data Set-Up Time	tWDS	9		ns
t107	Write Data Hold Time	tWDH	6		ns
t108	Valid Read Data Time	tRD	14	40	ns
t109	Read Data Hold Time	tRDH	6	23	ns
t110	System Address Set-Up Time	tAS	4		ns
t111	System Address Hold Time	tAH	20		ns
t112	Write Data Set-UpTime	tWDS	4		ns
t113	Write Data Hold Time	tWDH	12		ns
t114	Valid Read Data Time	tRD	14	40	ns
t115	Read Data Hold Time	tRDH	9	23	ns
t116	System Address Set-Up Time	tAS	5		ns
t117	System Address Hold Time	tAH	20		ns
t118	Write Data Set-Up Time	tWDS	5		ns
t119	Write Data Hold Time	tWDH	12		ns
t120	Valid Read Data Time	tRD	14	49	ns

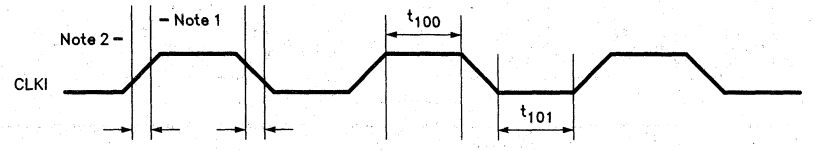
TIMING CHARACTERISTICS (Continued)

Table 26 (Continued)

No.	Parameter	Symbol	Limits		Units
			Min	Max	
t120	Valid Read Data Time	tRD	14	49	ns
t121	Read Data Hold Time	tRDH	9	23	ns
t122	EXT to VP[2 . . . 7] ON Delay	tEV	30	60	ms
t125	PWGIN ON to VP0 ON Delay	tPIV	0.5	1	s
t126	VP0 ON to PWGOUT ON Delay	tVPO	7.8 Typ.		ms
t127	VP0 ON to VPBIAS ON Delay	tVB	15.6 Typ.		ms
t128	VP0 ON to VP1 ON Delay	tVV	500 Typ.		ms
t129	PWGOUT OFF to VP1 OFF Delay	tPV	384 Typ.		μs
t130	$\overline{\text{REFRQ OFF}}$ to PWGOUT OFF Delay	tRPO	1125 Typ.		ns
t131	$\overline{\text{RAS}}$ Precharge Time (Burst Refresh)	tRP	125 Typ.		ns
t132	$\overline{\text{RAS}}$ ON Time (Burst Refresh)	tRASB	250 Typ.		ns
t133	Burst Refresh to SUSPEND Refresh Delay	tBS	15	31	μs
t134	$\overline{\text{RAS}}$ ON Time (SUSPEND Refresh)	tRASS	170	610	ns
t135A	SUSPEND Refresh Cycle Time (Regular DRAM)	tSRC	15 Typ.		μs
t135B	SUSPEND Refresh Cycle Time (Slow Refresh DRAM)	tSRC	124 Typ.		μs
t136	$\overline{\text{REFRSEL}}$ OFF to $\overline{\text{RAS}}$ (SUSPEND) OFF Delay	tRRAS	0 Typ.		ns
t137	NMI In to NMI Out Delay	tIN	4	14	ns
t138	VP0 ON to PCLKOUT High Delay	tVAS	1 PCLK + 4	1 PCLK + 16	ns
t139	VP0 OFF to PCLKOUT Low Delay	tVIS	See Note		
t140	PCLKSYNC Set Up Time	tPSU	3		ns
t141	PCLKSYNC Hold Time	tPH	0		ns

NOTES:
1. If VP0 turned off caused by SLEEP mode, the delay is 1 PCLK + 6 ns (Min.), 2 PCLK + 16 ns (Max.).
2. If VP0 turned off caused by SUSPEND or OFF mode, the delay is 4 ns (Min.), 16 ns. (Max).

82347 TIMING WAVEFORMS



NOTES:
1. (V_{CC} – 0.8V)
2. 0.8V

Figure 4. Input CLKX1

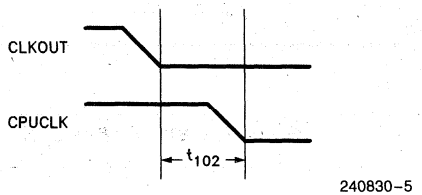


Figure 5. CLKOUT to CPUCLK Timing (286 Fully Static Mode)

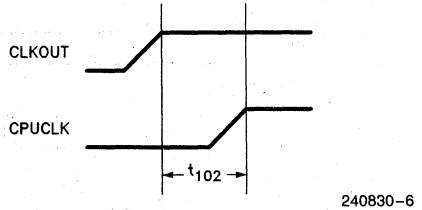


Figure 6. CLKOUT to CPUCLK Timing (8086 Fully Static Mode)

82347 TIMING WAVEFORMS (Continued)

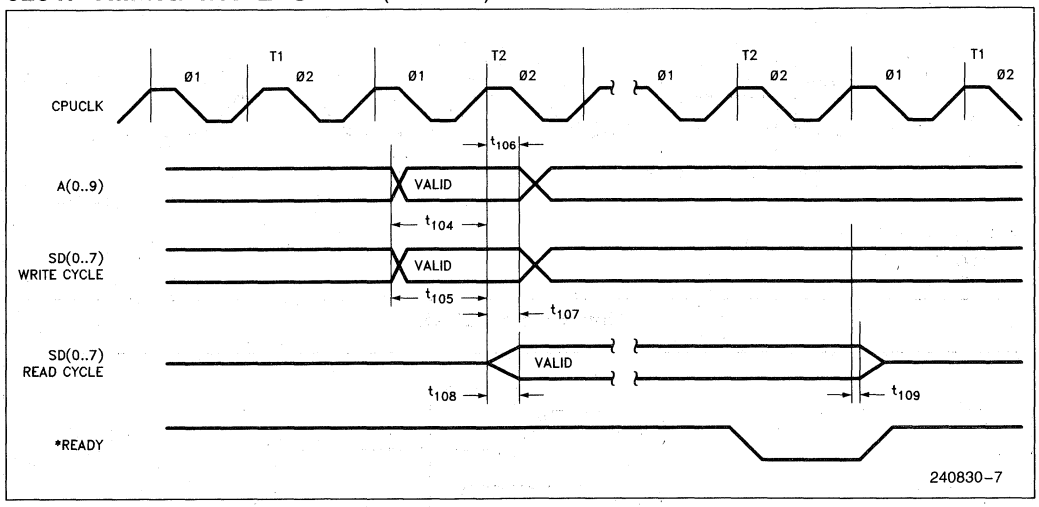


Figure 7. 386™ I/O Bus Cycle

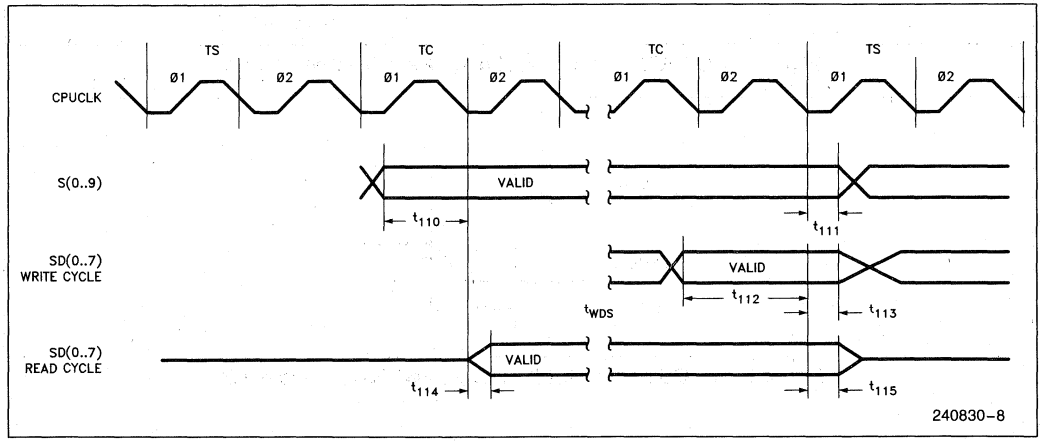


Figure 8. 80286 I/O Bus Cycles

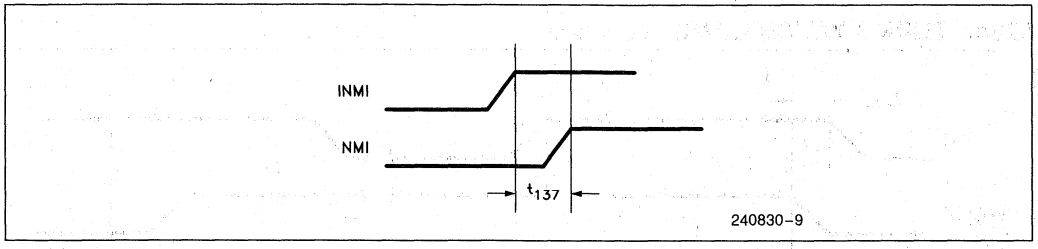
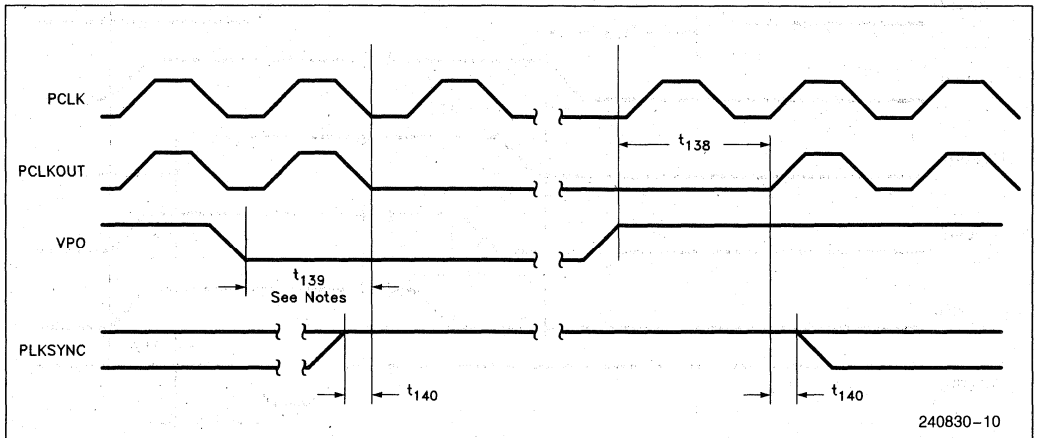


Figure 9. NMI Timing



NOTES:
1. If VP0 turned off caused by SLEEP mode, the delay is 1 PCLK + 6 ns (Min.), 2 PCLK + 16 ns (Max.).
2. If VP0 turned off caused by SUSPEND or OFF mode, the delay is 4 ns (Min.), 16 ns. (Max).

Figure 10. PCLK Timing

82347 TIMING WAVEFORMS (Continued)

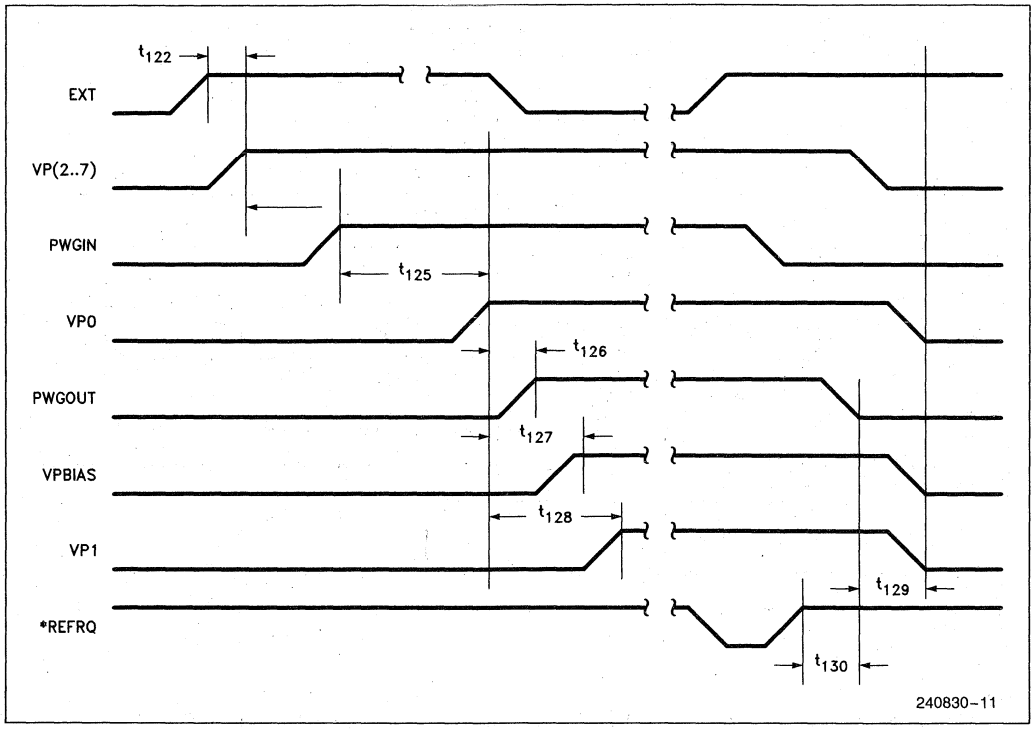


Figure 11. VP Timing

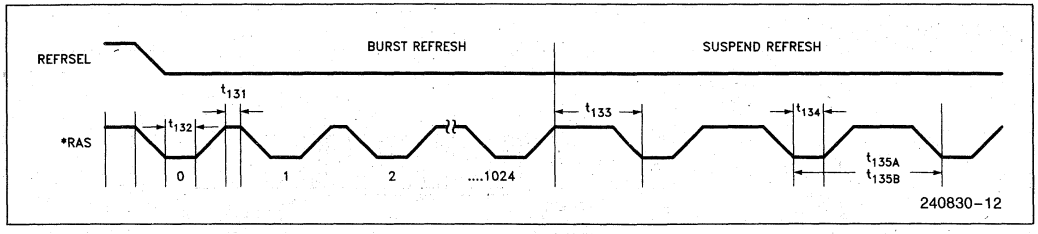


Figure 12. RAS Timing—Entering SUSPEND RAS Timing—Entering SUSPEND

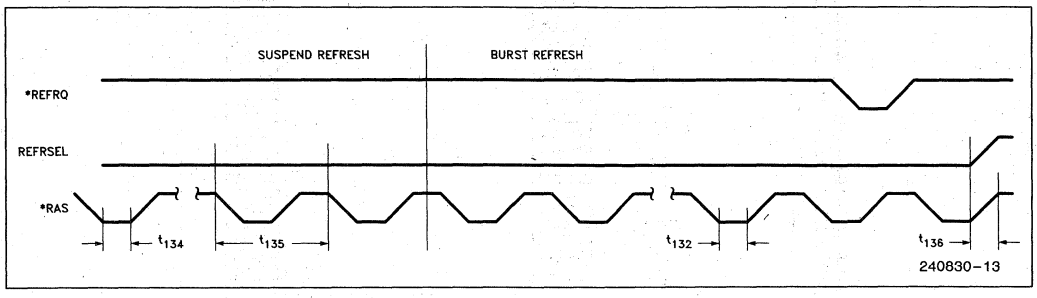


Figure 13. RAS Timing—Leaving SUSPEND

82347 PACKAGE

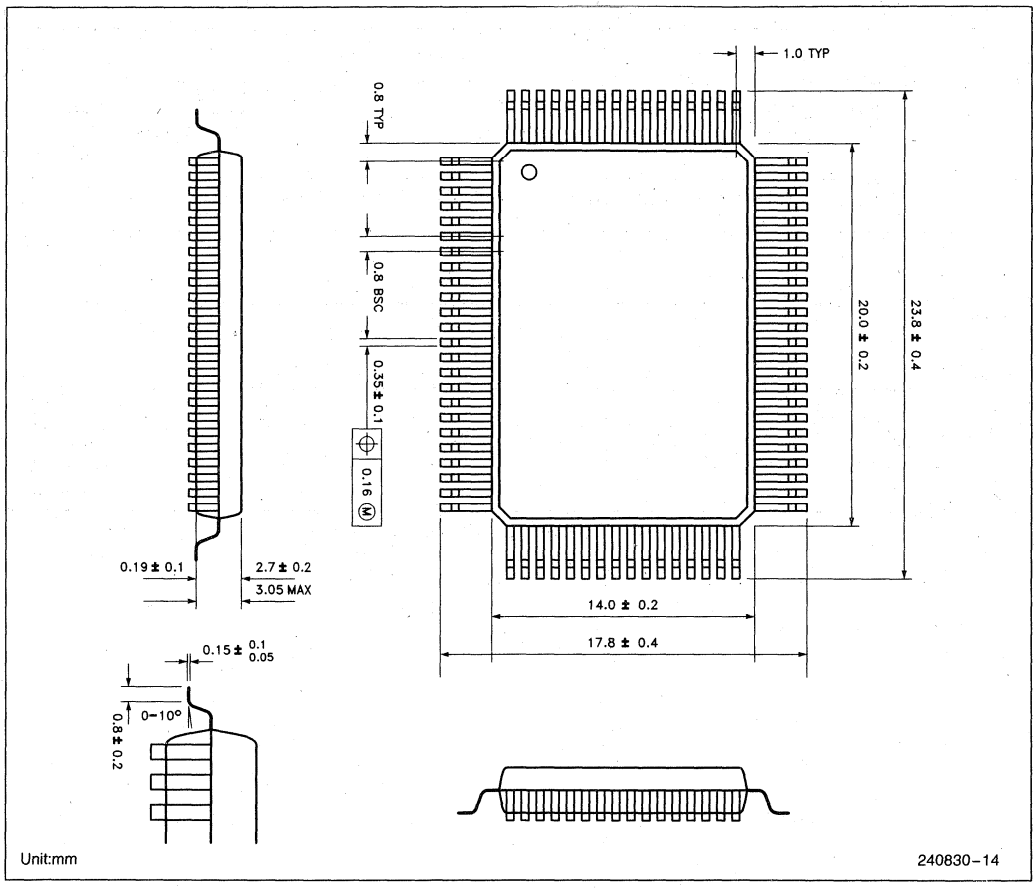


Figure 14. 80-Pin Plastic Rectangular Flat Package

82335 SX
HIGH-INTEGRATION INTERFACE DEVICE FOR 386™ SX MICROPROCESSOR BASED PC-AT SYSTEMS

- **Operates with the 82230 and 82231 to Provide 100% IBM AT™ Compatibility**
- **Optimized for 16 MHz and 20 MHz 386™ SX Microprocessor Based PC-AT Systems**
- **Page Mode, Interleaved DRAM Controller**
- **Address Mapping/Shadow ROM Support**
- **387™ SX Numeric Coprocessor Synchronization Interface**
- **Parity Generation and Checking**
- **Low Power, High Speed CHMOS IV Technology**
- **Available in 132 Lead Plastic Quad Flat Pack**

1

The Intel 82335 SX is a high integration interface device used together with the 82230/82231 to provide the most cost-effective and highest performance system design solution for AT-compatible 386 SX microprocessor based systems. The 82335 SX is plug compatible with the 82335.

The 82335 SX DRAM control feature is designed and optimized for the 16 MHz and 20 MHz 386 SX microprocessor bus architecture. The page mode, interleaved memory design allows 0 wait state performance on most memory accesses with 100 ns DRAM at 16 MHz or 80 ns DRAM at 20 MHz.

Several address mapping options are available to provide flexibility in the system memory size and configuration.

The 82335 SX also provides the necessary interface signals to allow the 387 SX numeric coprocessor to run in a PC/AT system. The 82335 SX with its intergrated parity generation and checking provides system designers with data integrity and reliability.

IBM AT is a trademark of International Business Machines Corporation.
386, 387 are trademarks of Intel Corporation.

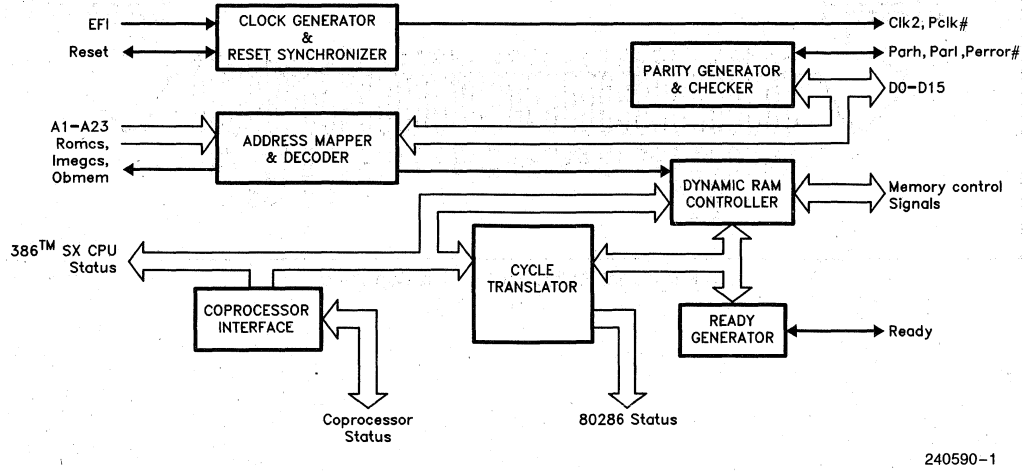


Figure 1.1. 82335 SX Internal Block Diagram

October 1990
Order Number: 240590-002

1.0 PIN DESCRIPTION

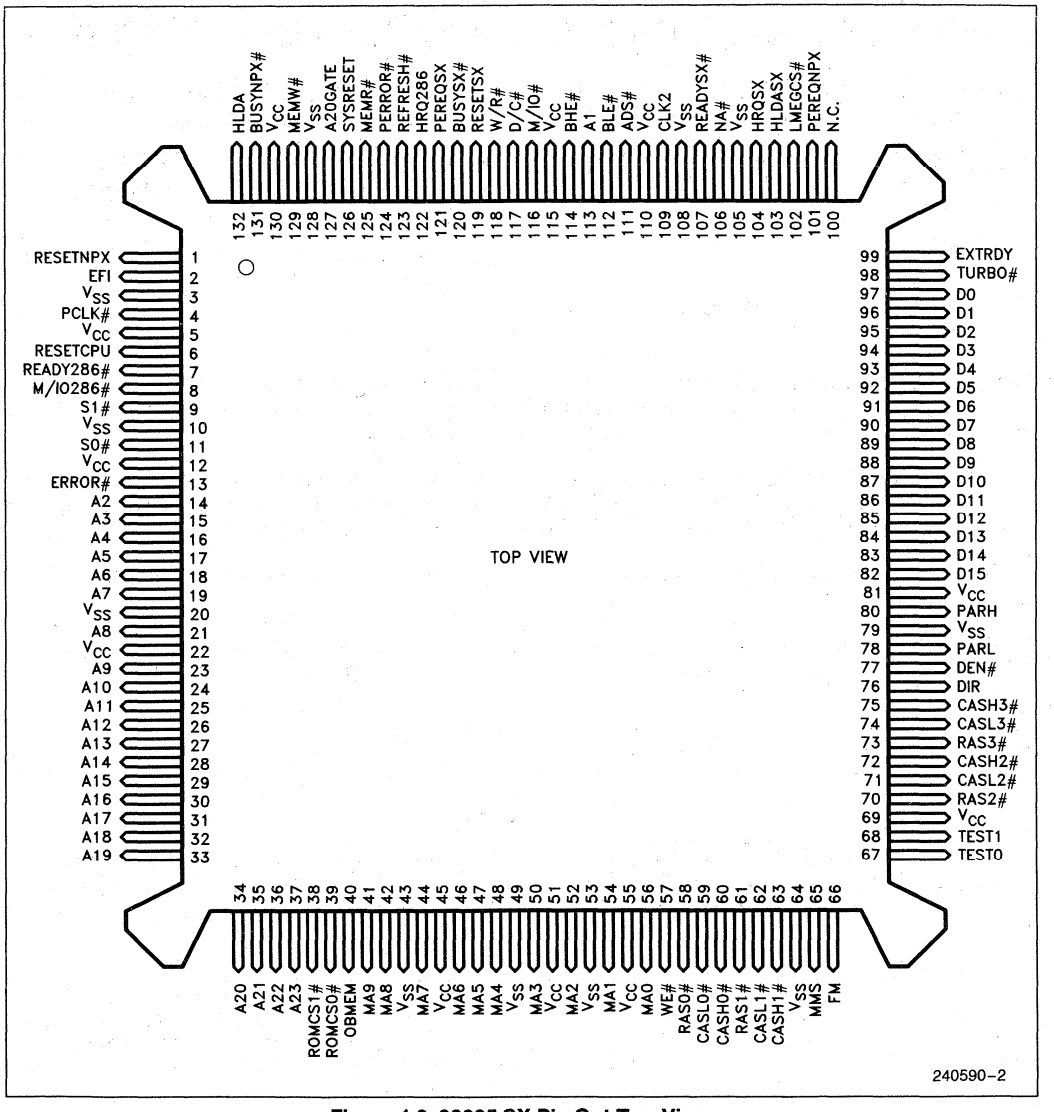


Figure 1.2. 82335 SX Pin Out Top View

Table 1.1. Alphabetical Pin Assignment

Address		Data		Control		Control		V_{CC}	V_{SS}
A1	113	D0	97	A20GATE	127	M/IO#	116	5	3
A2	14	D1	96	ADS#	111	M/IO286#	8	12	10
A3	15	D2	95	BHE#	114	NA#	106	22	20
A4	16	D3	94	BLE#	112	OBMEM	40	45	43
A5	17	D4	93	BUSYNPX#	131	PARH	80	51	49
A6	18	D5	92	BUSYSX#	120	PARL	78	55	53
A7	19	D6	91	CASH0	60	PCLK#	4	69	64
A8	21	D7	90	CASH1	63	PEREQNPX	101	81	79
A9	23	D8	89	CASH2	72	PEREQSX	121	110	105
A10	24	D9	88	CASH3	75	PERROR#	124	115	108
A11	25	D10	87	CASL0	59	RAS0#	58	130	128
A12	26	D11	86	CASL1	62	RAS1#	61		
A13	27	D12	85	CASL2	71	RAS2#	70		
A14	28	D13	84	CASL3	74	RAS3#	73		
A15	29	D14	83	CLK2	109	READY286#	7		
A16	30	D15	82	D/C#	117	N.C.	100		
A17	31			DEN#	77	READYSX#	107		
A18	32			DIR	76	REFRESH#	123		
A19	33			EFI	2	RESETCPU	6		
A20	34			ERROR#	13	RESETNPX	1		
A21	35			EXTRDY	99	RESETSX	119		
A22	36			FM	66	ROMCS0#	39		
A23	37			HLDA	132	ROMCS1#	38		
MA0	56			HLDASX	103	S0#	11		
MA1	54			HRQ286	122	S1#	9		
MA2	52			HRQSX	104	SYSRESET	126		
MA3	50			LMEGCS#	102	TEST0	67		
MA4	48			MEMR#	125	TEST1	68		
MA5	47			MEMW#	129	TURBO#	98		
MA6	46			MMS	65	WE#	57		
MA7	44					W/R#	118		
MA8	42								
MA9	41								

The 82335 SX is implemented in a 132-pin plastic flatpack package designed for direct surface mounting on component boards. The following is a description of the physical pin connections.

Table 1.2. Pin Description

Symbol	Pin No.	Type	Name and Function
A1–A23	14–19, 21, 23–37, 113	I*	**ADDRESS INPUTS:** These inputs are used to select the dynamic RAM address for a memory read or write operation.
A20GATE	127	I	**ADDRESS 20 GATE:** This active high input is used by the keyboard controller to force A20 low. When A20GATE is low, A20 is forced low internal to the 82335 SX during CPU memory cycles (not DMA or master). When A20GATE is high, A20 follows the CPU address input from the A20 pin.

NOTES:
*The following conventions are used in this table:
I = Input O = Output I/O = Input or Output
The symbol # following a signal name indicates that the signal is low active.

Table 1.2. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
ADS#	111	I	**ADDRESS STATUS:** This active low input indicates that a valid bus cycle definition and address (W/R#, M/IO#, D/C#, BLE#, BHE#, and A1–A23) is being driven by the 386™ SX microprocessor.
BHE#	114	I	**BYTE HIGH ENABLE:** This active low input indicates when data is being transferred on D8–D15.
BLE#	112	I	**BYTE LOW ENABLE:** This active low input indicates when data is being transferred on D0–D7.
BUSYNPX#	131	I	**BUSY NP:** This active low input is used by the 387™ SX coprocessor to indicate that it is busy. It is designed to be connected directly to BUSY# of the 387 SX. BUSYNPX# has a weak internal pullup resistor.
BUSYSX#	120	O	**BUSY SX:** This active low output indicates to the 386 SX CPU that the 387 SX is busy. It is designed to be connected directly to BUSY# of the 386 SX CPU.
CASH0# – CASH3#	60, 63, 72, 75	O	**COLUMN ADDRESS STROBE (HIGH BYTE):** These outputs are used by the high byte of the dynamic RAM array to latch the column address present on the MA0–MA9 pins. They can drive the dynamic RAM array directly and need no external drivers.
CASL0# – CASL3#	59, 62, 71, 74	O	**COLUMN ADDRESS STROBE (LOW BYTE):** These outputs are used by the low byte of the dynamic RAM array to latch the column address present on the MA0–MA9 pins. They can drive the dynamic RAM array directly and need no external drivers.
CLK2	109	O	**CLOCK2:** This output drives the 386 SX and 387 SX input clocks. It is generated by the External Frequency Input (EFI) and outputs the same frequency as EFI (32 MHz or 40 MHz).
D/C#	117	I	**DATA/CONTROL SELECT:** This input from the 386 SX is used to distinguish between data and control bus cycles.
DEN#	77	O	**DATA ENABLE:** This active low output is used by the data transceivers to enable the transfer of data between the dynamic RAM array and the local data bus.
DIR	76	O	**DIRECTION:** This signal is used to control the direction input of the data transceivers which connect the dynamic RAM array to the local data bus. When DIR is high, data is being written into memory.
D15–D0	82–97	I/O	**DATA BUS:** These inputs are used by the 82335 SX for parity generation and checking of data which is transferred between the local bus and the DRAM array. During initialization of the 82335 SX, they are used to write/read control words to/from the internal memory configuration registers.
EFI	2	I	**EXTERNAL FREQUENCY IN:** This input is driven by an external oscillator. It is used by the 82335 SX to generate the CLK2 and PCLK# output clocks. All internal 82335 SX logic is also driven by EFI. The EFI frequency is the same as the CLK2 (32 MHz or 40 MHz).
ERROR#	13	I	**ERROR:** This input indicates when a numeric coprocessor error has occurred. ERROR# is designed to be directly connected to the ERROR# output of the 387 SX and ERROR# input of the 82230. ERROR# has a weak pull-up resistor inside the 82335 SX.

Table 1.2. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
EXTRDY	99	I	**EXTERNAL READY:** This is an active high, level triggered input used to insert additional wait states into local memory bus cycles. Deactivation of EXTRDY during a local memory or numerics coprocessor access delays activation of the READYSX# output until EXTRDY is sampled active. Deactivation of EXTRDY will **block** (not delay) recognition of the READY286# input.
FM MMS	66 65	I I	FM and MMS are used to select DRAM operating modes. Refer to the DRAM controller section for available modes. These pins should be static after system reset.
HLDA	132	O	**HOLD ACKNOWLEDGE:** This output indicates the 386 SX has relinquished control of the local bus. It is asserted in response to activation of the HLDASX input from the 386 SX. It is designed to be connected to the 82231 HLDA input.
HLDASX	103	I	**HOLD ACKNOWLEDGE SX:** This input is asserted by the 386 SX in response to assertion of the 386 SX HOLD pin. It indicates that the processor has relinquished control of the local bus. It is designed to be connected to the 386 SX HLDA output.
HRQ286	122	I	**CPU HOLD REQUEST INPUT:** This active high input receives hold request signals for the 386 SX. It is designed to be driven by CPUHRQ from the 82231.
HRQSX	104	O	**CPU HOLD REQUEST OUTPUT:** This active high output drives the 386 SX HOLD input. It is the HRQ286 input with the trailing edge delayed.
LMEGCS#	102	O	**LOWER MEG CHIP SELECT:** This output is held low during local DRAM accesses made to the first megabyte of memory (000000H–0FFFFFH).
MA9–MA0	41, 42, 44, 46–48, 50, 52, 54, 56	O	**MULTIPLEXED ADDRESS:** These outputs are designed to provide the row and column addresses for CPU or DMA access, and row addresses for refresh access to the dynamic RAM array.
MEMR#	125	I	**MEMORY READ COMMAND:** This active low input is used to indicate when a DMA memory read cycle is being performed. It is designed to be connected directly to the —MEMR output of the 82230. MEMR# can be an asynchronous input.
MEMW#	129	I	**MEMORY WRITE COMMAND:** This active low input is used to indicate when a DMA memory write cycle is being performed. It is designed to be connected directly to the —MEMW output of the 82230. MEMW# can be an asynchronous input.
M/IO#	116	I	**MEMORY/IO SELECT:** This input from the 386 SX is used to distinguish between memory and I/O accesses.
M/IO286#	8	O	**MEMORY I/O SELECT 286:** This output emulates the M/IO# output of the 80286. It is used by the 82230/82231 and other system peripherals to distinguish memory access from I/O access. It is also used with the status output (S0, S1) to indicate halt/shutdown and interrupt acknowledge cycles.
NA#	106	O	**NEXT ADDRESS:** The NA# output is used to control the address pipelining of the 386 SX. It must be connected to the 386 SX NA# input. Consecutive local memory accesses are always pipelined after the first access.

Table 1.2. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
OBMEM	40	O	**ON-BOARD MEMORY:** This active high output indicates a local DRAM access is in progress.
PARH	80	I/O	**PARITY HIGH BYTE:** This three state input/output is used for the upper byte parity bit of data on the local bus (D8–D15). For memory write cycles, the 82335 SX outputs the internally generated parity bit to the DRAM array via the PARH pin. During a memory read, the 82335 SX uses the data received at PARH to validate the upper byte of data from the DRAM array.
PARL	78	I/O	**PARITY LOW BYTE:** This 3-state input/output is used for the lower byte parity bit of data on the local bus (D0–D7). Its function is identical to the PARH pin described above.
PCLK#	4	O	**PERIPHERAL CLOCK:** This clock signal is generated by dividing the EFI input by 2 in 16 MHz operation or by 2.5 in 20 MHz operation. It is designed to drive the X3 input of the 82230.
PEREQNPX	101	I	**PROCESSOR EXTENSION REQUEST NP:** This input is used by the 387 SX to indicate that it requires a data transfer. It is designed to be connected directly to PEREQ of the 387 SX.
PEREQSX	121	O	**PROCESSOR EXTENSION REQUEST SX:** This output to the 386 SX processor is used to request a data transfer to or from the numeric coprocessor. It is designed to be connected directly to PEREQ of the 386 SX.
PERROR#	124	O	**PARITY ERROR:** This active low output indicates that the 82335 SX has detected a parity error in either the upper or lower byte of data from the DRAM array. It is designed to drive the 82231 DPCK# input.
RAS0# – RAS3#	58, 61, 70, 73	O	**ROW ADDRESS STROBE:** These outputs are used by the dynamic RAM array to latch the row address present on the MA0–MA9 pins. The four outputs support up to a four-way interleaved dynamic RAM configuration with page-mode access. They drive the dynamic RAM array directly and need no external drivers.
READY286#	7	I	**READY 286:** This active low input is used to indicate the completion of system I/O or memory bus cycles. It is designed to be driven by the 82230 READY# pin.
N.C. (formerly READYNPX#)	100	I	**NO CONNECT:** This pin is not internally connected. The 82335 SX will generate READYSX# after 1 wait-state on all numerics coprocessor accesses if a coprocessor is present in the system.
READYSX#	107	O	**READY SX:** This active low output indicates the completion of the current bus cycle to the processor. It is a function of the internally generated ready signal for local memory, on-chip I/O, or coprocessor accesses, and the READY286#, EXTRDY, and TURBO# inputs.
REFRESH#	123	I	**REFRESH:** This active low input is used to notify the dynamic RAM controller that the dynamic RAM array requires refresh. It is designed to be driven by the 82231 –REFRESH output.
RESETCPU	6	I	**RESET CPU:** This high active input is used to generate the RESETSX output which resets the 386 SX CPU. It must be activated with SYSRESET during power-on reset. After power-on, activation of RESETCPU without SYSRESET will activate RESETSX without RESETNPX. RESETCPU is designed to be driven by the 82230 RES CPU output.

Table 1.2. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
RESETNPX	1	O	**RESET NPX:** This output is designed to drive the RESETIN pin of the 387 SX. It is activated only when SYSRESET and RESETCPU are both active.
RESETSX	119	O	**RESET SX:** This output is designed to drive the RESET pin of the 386 SX processor. It is a function of the SYSRESET and RESETCPU inputs.
ROMCS0# ROMCS1#	39 38	O	**ROM CHIP SELECT:** These outputs are used to support shadow RAM. They select the ROMs or EPROMs during system initialization. If ROM shadowing is selected, the ROMCS0–1 outputs are disabled and the ROM (EPROM) addresses are mapped into the DRAM physical address space by the 82335 SX.
S0# S1#	11 9	O	**BUS CYCLE STATUS:** The S0# and S1# outputs indicate the initiation of a system (non-local) bus cycle and, along with M/IO286#, define the type of bus cycle.
SYSRESET	126	I	**SYSTEM RESET:** This high active input is combined with RESETCPU to generate the RESETSX and RESETNPX outputs as well as synchronize the clock outputs. SYSRESET is designed to be driven by the 82230 +RESET output.
TEST0 TEST1	67 68	I	**TEST MODE:** These inputs are used for special test modes, and must be connected to V_{SS} during normal operation. TEST0 is a high active, level triggered input that disables all 82335 SX output buffers when active. The TEST1 input is reserved and must be connected to V_{SS}.
TURBO#	98	I	**TURBO MODE SELECT:** This active low input, when asserted, allows the 386 SX local bus to run with maximum performance. Deactivating the TURBO# input causes the 82335 SX READY generation logic to insert additional wait states into each bus cycle. In the non-turbo mode, 386 SX performance approximates 80286 bus efficiency.
V_{CC}	5, 12, 22, 45, 51, 55, 69, 81, 110, 115, 130	—	**POWER SUPPLY:** 11 V_{CC} pins total.
V_{SS}	3, 10, 20, 43, 49, 53, 64, 79, 105, 108, 128	—	**GROUND:** 11 V_{SS} pins total.
WE#	57	O	**WRITE ENABLE:** This output is used by the dynamic RAM array to enable input for a write operation. It is designed to drive two banks of DRAM with no additional buffering.
W/R#	118	I	**WRITE/READ SELECT:** This input from the 386 SX is used to distinguish between read and write cycles.

2.0 FUNCTIONAL DESCRIPTION

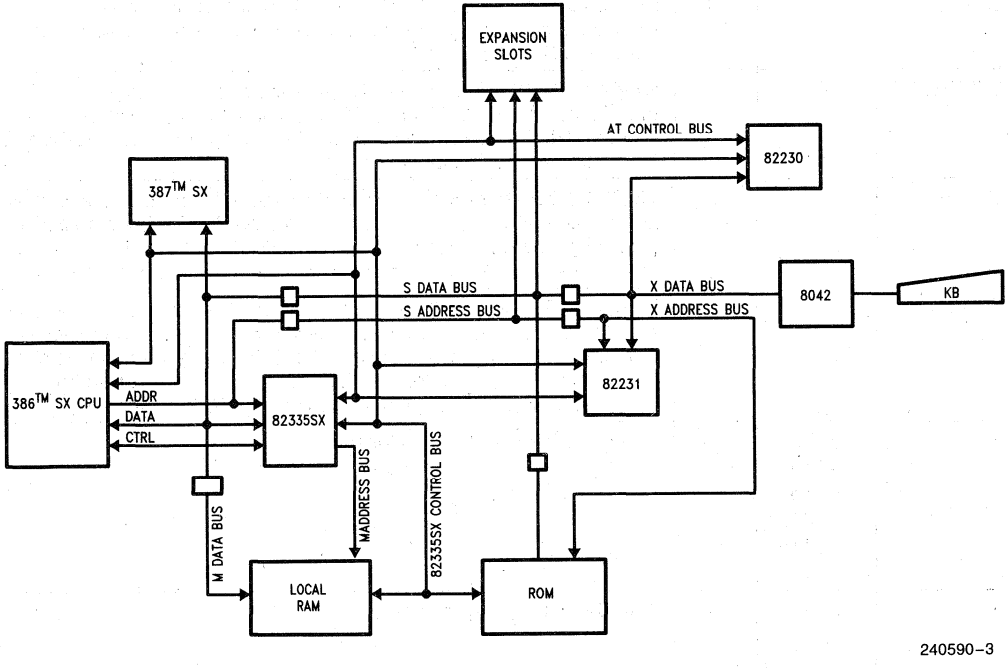
2.1 Introduction

The 82335 SX is a high-integration VLSI companion chip for the Intel 386 SX 32-bit microprocessor. It interfaces the 386 SX microprocessor to the 387 SX numeric coprocessor and to the 82230/82231 highly integrated peripherals to form an AT compatible system. The 82335 SX accomplishes this by converting 386 SX processor bus cycles to 80286 compatible cycles, generating necessary clock signals, and providing local dynamic memory control. Figure 2.1 shows a block diagram of this system.

The 82335 SX is composed of seven functional blocks:

1. DRAM Controller
2. Address Mapper/Decoder
3. Ready Generator
4. Bus Cycle Translator
5. Math Coprocessor Interface
6. Clock Generator/Reset Synchronizer
7. Parity Generator/Checker

Each functional block is described in the following sections.



NOTE:
To satisfy 82335 SX AC timing requirements, this configuration must use the 12 MHz 82230/82231.

Figure 2.1. 386™ SX CPU with 82335 SX System Block Diagram

2.2 DRAM Controller

2.2.1 INTRODUCTION

The 82335 SX dynamic RAM (DRAM) controller is designed and optimized for the Intel 386 SX Architecture. It keeps track of 386 SX CPU bus states and provides the necessary signals to address and refresh up to four 16-bit banks of 256K or 1M dynamic RAMs.

To optimize memory performance and flexibility, the DRAM controller has built in support for both paging and bank interleaving, and can be configured for several different modes of operation. These include four different memory modes to accommodate DRAM with different levels of performance: F1 and F4 are for either 100 ns (16 MHz) or 80 ns (20 MHz) fast page-mode DRAMs. WO1 and WO2 are for slower DRAMs. These modes are described in more detail in the DRAM mode configuration section.

In addition to the four memory modes available, the DRAM controller can be configured to operate in either turbo or non-turbo mode. The turbo mode allows the 386 SX microprocessor based system to run at peak efficiency, while the non-turbo mode allows it to approximate 80286 bus cycle timing for timing dependent software.

2.2.2 DRAM BANK CONFIGURATION

The local Dynamic RAM for the 386 SX/82335 SX system can be configured into one, two, or four 16-bit banks. Each 16-bit bank of memory is further divided into two 8-bit banks, low and high. Each 8-bit bank may contain one extra bit for parity.

Either 1 Mbyte or 256 Kbyte DRAM may be used. The same type of DRAM must be used in all banks installed. See Figure 2.2 for a block diagram of the 82335 SX to DRAM interface.

The exact memory configuration (DRAM size and number of banks installed) can be determined during system initialization through execution of a memory autoscan routine in the BIOS. Memory configuration information can then be programmed into the memory configuration register, roll compare registers, and address compare registers. See the address mapping/decoding section for details on register programming.

For local memory accesses the 82335 SX generates the DRAM control signals Row address strobe (RAS#), Column address strobe low and high (CASL# and CASH#), and Write enable (WE#), as well as the multiplexed row and column addresses (MA). Each bank has its own separate RAS#,

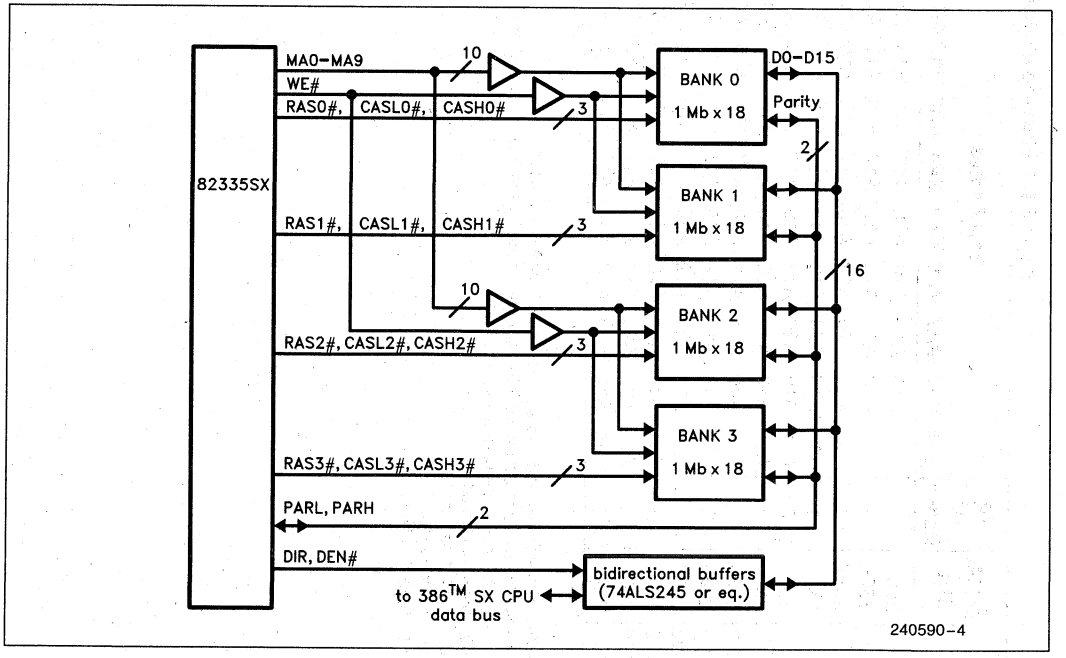


Figure 2.2. 82335 SX to DRAM Interface

Table 2.1. Address to Multiplexed Address Translation

Mode	DRAM Size	No. of Banks	Row Address										Column Address										Bank Select	
			MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0	B1	B0
F1/F4	256K	1		A10	A11	A12	A18	A17	A16	A15	A14	A13		A9	A8	A7	A6	A5	A4	A3	A2	A1	0	0
F1/F4	256K	2		A19	A11	A12	A18	A17	A16	A15	A14	A13		A9	A8	A7	A6	A5	A4	A3	A2	A1	0	A10
F1/F4	256K	4		A19	A20	A12	A18	A17	A16	A15	A14	A13		A9	A8	A7	A6	A5	A4	A3	A2	A1	A11	A10
F1/F4	1M	1	A11	A19	A20	A12	A18	A17	A16	A15	A14	A13	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	0	0
F1/F4	1M	2	A21	A19	A20	A12	A18	A17	A16	A15	A14	A13	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	0	A11
F1/F4	1M	4	A21	A19	A20	A12	A18	A17	A16	A15	A14	A13	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A12	A11
WO1/WO2	256K	1		A10	A11	A12	A18	A17	A16	A15	A14	A13		A9	A8	A7	A6	A5	A4	A3	A2	A1	0	0
WO1/WO2	256K	2		A19	A11	A12	A18	A17	A16	A15	A14	A13		A9	A8	A7	A6	A5	A4	A3	A2	A10	0	A1
WO1/WO2	256K	4		A19	A20	A12	A18	A17	A16	A15	A14	A13		A9	A8	A7	A6	A5	A4	A3	A11	A10	A2	A1
WO1/WO2	1M	1	A11	A19	A20	A12	A18	A17	A16	A15	A14	A13	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	0	0
WO1/WO2	1M	2	A21	A19	A20	A12	A18	A17	A16	A15	A14	A13	A11	A9	A8	A7	A6	A5	A4	A3	A2	A10	0	A1
WO1/WO2	1M	4	A21	A19	A20	A22	A18	A17	A16	A15	A14	A13	A10	A9	A8	A7	A6	A5	A4	A3	A11	A10	A2	A1

NOTES:
1. When 256K DRAM is installed, the address output MA9 is not used.
2. During refresh the row addresses are internally generated by the 82335 SX and are not affected by the A1–A23 address inputs.
3. The bank select bits are interpreted as follows:

B1 B0 = 00 bank 0 selected
01 bank 1 selected
10 bank 2 selected
11 bank 3 selected

CASL#, and CASH# signals. The RAS#, CASL#, and CASH# outputs can directly drive the DRAM.

The WE# and Multiplexed Address lines are common to all banks. These outputs can directly drive approximately two banks of memory. If the capacitive loading on these outputs exceeds the maximum loadings defined in the AC timing specifications, then they will need to be buffered.

The multiplexed address outputs (MA0–MA9) are derived from the address inputs A1–A23 during local memory read or write operations. The way the 82335 SX translates the address inputs into row address, column address, and bank select bits depends upon several factors. These factors include DRAM size, number of banks installed, and mode of operation. Table 2.1 shows how the input addresses are translated.

2.2.3 PAGE-MODE DRAM OPERATION

Every DRAM access requires that the DRAM be provided with both a row address and a column address. A DRAM access requiring both a new row address and new column address has a long cycle time which requires at least one wait state. Figure 2.3 shows examples of DRAM cycles that require both new row and new column addresses.

The 82335 SX significantly improves bus performance by "paging" row addresses. Memory locations sharing the same row address are in the same "memory page". When successive memory accesses are in the same page (a page-hit memory access), only a new column address is required. This way, the row address strobe, RAS#, can be kept active, and each page-hit memory cycle only requires a new column address. This reduces the memory access time and allows most memory cycles to run at zero wait states. Figure 2.3 shows examples of both page-hit and page-miss cycles.

The effectiveness of page-mode operation in reducing wait states is dependent upon several factors. The most important factors are page location, page size, and page-mode cycle time. Page location and size are discussed below and page-mode cycle time is discussed in Section 2.2.4.

Page location is determined by the selection of address bits used for DRAM row addresses. To increase page-hits (zero wait state cycles), pages should be located such that successive memory accesses are in the same page. The 82335 SX generates row addresses from the higher order address bits. Since these bits are less likely to change than the lower order address bits, this increases the page-hit rate.

The page-hit rate can also be increased by increasing the page size. The page size for a single bank is 1 KByte if 256K DRAM chips are installed and 2 KByte if 1M DRAM chips are installed. The 82335 SX can keep a page of memory active in each bank. This can double the effective page size in a two bank configuration. A four bank configuration can effectively increase the page size by a factor of four. This increase in the active page size can significantly improve system performance. A memory access to an active page in a different bank (a page-hit-bank-miss) only requires a new column address, and therefore runs at zero wait states.

The 82335 SX keeps a memory page active for a particular bank by holding the RAS# signal active for that bank. The RAS# signal for a bank is activated by either an access to that particular bank or by a memory refresh.

RAS# deactivation can be caused by a number of events. In non-turbo mode RAS# is always deactivated at the end of each memory cycle. In turbo mode RAS# is deactivated by a page-miss memory cycle, RAS# timeout, or ending successive local memory accesses.

In addition to the RAS# deactivation events in turbo mode, F1 mode will also deactivate the RAS# signal for the previously accessed bank when a bank switch occurs.

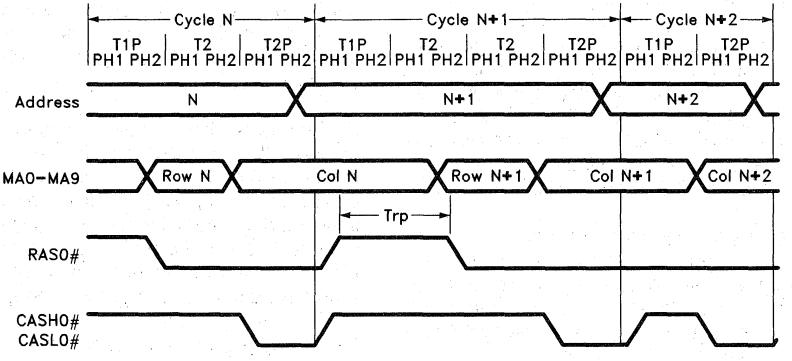
A page-miss memory cycle in modes F4, WO1, and WO2 will deactivate the RAS# signal only for the bank currently being accessed. Wait states are then inserted into the memory cycle to insure that the minimum RAS# precharge time required by the DRAM is satisfied.

RAS# deactivation can also be caused by a "RAS# timeout". Dynamic RAM chips have a limit on the maximum time that RAS# can be held active. The 82335 SX controls RAS# active time with four built-in watch-dog timers. Once timed-out, a RAS# signal will deactivate at the end of the current memory cycle. Each timer operates independently and guarantees a maximum RAS# active time of 10 μs.

RAS# signals that have been activated in mode F4, WO1, or WO2 and have not been timed-out, will remain active while successive local memory accesses are made. If the CPU executes a bus cycle other than a local memory cycle (i.e., I/O cycle, system memory cycle, etc.) or the CPU bus enters a hold or idle state, then all RAS# signals will deactivate. Table 2.2 summarizes RAS# activation.

Table 2.2. RAS# Activation

Mode	When	What
RAS# Activation		
All Modes	• A memory access is made to a bank. • A memory refresh is initiated.	RAS# for that bank is activated. All RAS# signals are activated.
RAS# De-Activation		
Non-Turbo	• End of each local memory cycle.	All RAS# are de-activated.
Turbo F4, Turbo WO1, Turbo WO2	• A page-miss occurs. • A RAS# timeout occurs. • Any cycle other than a local memory access.	RAS# is de-activated for that bank. RAS# is de-activated for that bank. All RAS# signals are de-activated.
Turbo F1	• A bank switch occurs. • A page-miss occurs. • A RAS# timeout occurs. • Any cycle other than a local memory access.	RAS# de-activates for the previous bank. RAS# is de-activated for that bank. RAS# is de-activated for that bank. All RAS# signals are de-activated.



240590-5

NOTES:
1. New RAS# in cycle N requires one wait-state for row address latching.
2. Page-miss-bank-hit in cycle N+1. Two wait-states are required for RAS# precharge (T_{rp}).
3. Page-hit-bank-hit in cycle N+2 requires no wait-states.

Figure 2.3. 82335 SX Page Mode DRAM Cycles with New RAS#, Page Miss, and Page Hit Cycles in F1/F4 Modes

2.2.4 PAGE-MODE BANK INTERLEAVE OPERATION

Most fast page-mode DRAM have short cycle times which allow zero wait-state bus cycles on successive DRAM accesses. Slower DRAM, however, have longer cycle times which require wait-states to be inserted between successive accesses to the same memory bank. The 82335 SX allows most memory accesses to occur at zero wait-states, even for slow DRAM, by "interleaving" memory banks.

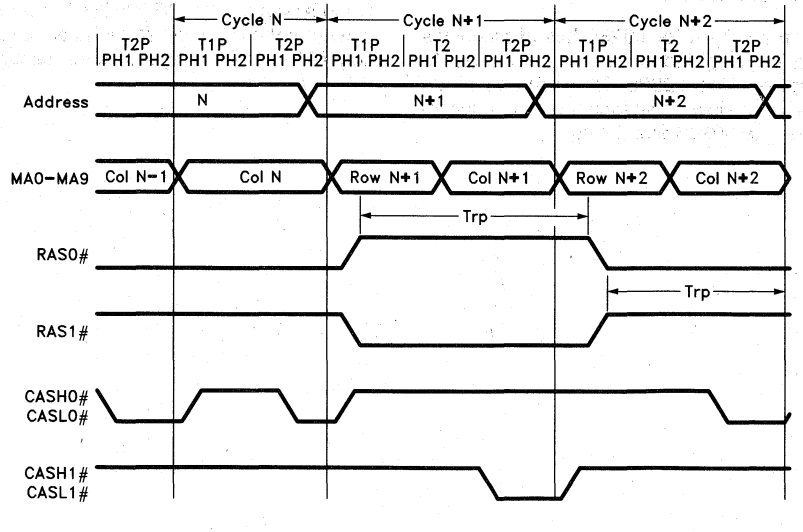
Interleaving refers to alternating bank accesses. Multi-bank configurations are always interleaved. In fast mode (F4/F1), banks are interleaved in pages. In slow mode (WO1/WO2), the lower address bit(s) are used to interleave banks by words (two bytes). Word interleaving alternates bank-hits for consecutive memory accesses thereby increasing the cycle time available to slow DRAM. This increases the number of zero wait-state page-hit-bank-miss cycles and improves system performance. Table 2.1 shows the address bits used for bank selection. An example of two bank interleaving in F1 mode is shown in Figure 2.4.

2.2.5 DRAM MODE CONFIGURATION

The 82335 SX can be configured to run in four different modes to operate with DRAM of various performance levels. This allows the system designer considerable flexibility. There are two modes (F1 and F4) for fast page mode DRAM and two modes (W01 and W02) for slower DRAM. The mode of operation is selectable by setting the input pins FM and MMS to the values shown in Table 2.3. Tables 2.3A and 2.3B show a summary of the different DRAM modes. A brief description of each mode follows.

F4: This is a high performance mode for fast 100 ns DRAM at 16 MHz or fast 80 ns DRAM at 20 MHz. The critical timing specifications that determine which DRAM can use this mode are listed in Tables 2.3A and 2.3B. Mode F4 allows up to four memory pages to be active simultaneously. This increases the page hit rate and allows more memory cycles to run at zero wait-states.

F1: This mode works with the same fast DRAM as mode F4. It differs from mode F4 in that only



NOTES:
1. DRAM access alternates between banks 0 and 1.
2. RAS precharge for bank 0 is hidden behind memory access of bank 1 and vice versa.

Figure 2.4. 82335 SX DRAM Cycle with Interleaved Memory in F1 Mode

one memory page can be kept active at a time. Activating only one page at a time reduces power consumption. Since only one page is kept active at a time, any bank-miss is also a page-miss which requires at least one wait-state.

WO1: Mode WO1 can be used with all 100 ns DRAM at 16 MHz or all 80 ns DRAM at 20 MHz. Slow page-mode DRAM that don't meet the critical specifications required for modes F4 and F1 can be used in mode WO1. This mode inserts one wait-state in page-hit-bank-hit cycles to allow the DRAM sufficient cycle time. At 20 MHz this mode also inserts one wait-state in page miss cycles to correspond to DRAM specifications. Up to four memory pages can be kept active simultaneously in this mode.

WO2: This mode allows inexpensive 120 ns DRAM to be used with the 82335 SX. It inserts additional wait-states in all cycles except page-hit-bank-miss cycles which run at zero wait-states. At 20 MHz there is no WO2 mode. At 20 MHz, if FM = 0, the 82335 SX will operate in WO1 mode regardless of the state of MMS. Up to four memory pages can be kept active simultaneously in this mode.

2.2.6 NON-TURBO MODE

Some programs written for 80286 based machines have software timing loops that are sensitive to processor speed. The 82335 SX non-turbo mode feature allows it to slow down local memory accesses to approximate 80286 bus timing.

The 82335 SX is put into non-turbo mode by driving the TURBO# input high (TURBO# > V_{IH}). The TURBO# input is then latched internally to synchronize it with DRAM accesses. It is latched on the falling edge of the 82335 SX HLDA output. This allows TURBO# to be an asynchronous input. Once the TURBO# input has been sampled high, a fixed number of bus states (six per memory cycle at 16 MHz or eight per memory cycle at 20 MHz) will be used for local memory accesses. The TURBO# input does not affect DMA cycle time. Figures 4.7A and 4.7B in the A.C. timing diagram section show examples of non-turbo mode read and write cycles.

2.2.7 REFRESH CYCLE

The 82335 SX controls DRAM refreshing in addition to controlling DRAM accesses. A DRAM refresh is initiated by asserting a hold request (HRQ286), receiving a hold acknowledge (HLDA), and then pulsing the REFRESH# input low. The 82335 SX then performs a "RAS only" refresh (no CAS# signals are generated). Once the refresh is completed (all banks have received a RAS# pulse), the REFRESH# and HRQ286 inputs can be deasserted. REFRESH# should not be asserted until the hold request has been acknowledged by the 82335 SX. Once asserted, the REFRESH# input must be held active for the minimum refresh period (see specification T83 in the A.C. specification section).

Table 2.3A. 16 MHz Summary of DRAM Modes

FM	MMS	Mode	Max Pages Active	Wait States				New RAS	DRAM Type	Critical DRAM Specifications
				Page Hit		Page Miss				
				Bank Hit	Bank Miss	Bank Hit	Bank Miss			
1	1	F4	4	0	0	2	2	1	100 ns Fast Page Mode	$t_{CAS} \leq 35$ ns $t_{CAC} \leq 35$ ns $t_{CP} \leq 20$ ns
1	0	F1	1	0	NA	2	1	1	100 ns Fast Page Mode	$t_{CAS} \leq 35$ ns $t_{CAC} \leq 35$ ns $t_{CP} \leq 20$ ns
0	1	W01	4	1	0	2	2	1	All 100 ns DRAM	
0	0	W02	4	2	0	3	3	2	120 ns DRAM	

Table 2.3B. 20 MHz Summary of DRAM Modes

FM	MMS	Mode	Max Pages Active	Wait States				New RAS	DRAM Type	Critical DRAM Specifications
				Page Hit		Page Miss				
				Bank Hit	Bank Miss	Bank Hit	Bank Miss			
1	1	F4	4	0	0	2	2	1	80 ns Fast Page Mode	$t_{CAC} \leq 30$ ns $t_{CP} \leq 20$ ns $t_{CAS} \leq 30$ ns $t_{RP} \leq 45$ ns $t_{AA} \leq 75$ ns $t_{RSH} \leq 30$ ns
1	0	F1	1	0	NA	2	1	1		
0	X	W01	4	1	0	3	3	2	All 80 ns DRAM and some 100 ns DRAM	$t_{RP} \leq 90$ ns $t_{CAS} \leq 50$ ns $t_{CP} \leq 40$ ns $t_{RSH} \leq 50$ ns

NOTES:
1. This table assumes the following input status: TURBO# $\leq V_{IL}$ max., EXTRDY $\geq V_{IH}$ min.
2. The first local memory access following an idle cycle or bus cycle other than a local memory access requires one extra wait-state to switch from non-pipelined to pipelined operation.
3. Definitions:

New RAS = The RAS# signal for a given bank transitions from an inactive to an active state.
Page hit = An access made to an active memory page.
Page miss = An access made to a memory page that is not currently active.
Bank hit = An access to the same memory bank accessed in the immediately preceding bus cycle.
Bank miss = An access to a memory bank that was not immediately preceded by an access to the same bank.
T_{CAS} = Column Address Strobe pulse width.
T_{CAC} = Column Address Strobe access time.
T_{CP} = Column Address Strobe precharge time.
T_{RP} = Row Address Strobe precharge time.
T_{AA} = Column Address access time.
T_{RSH} = Row Address Strobe hold time.

The 82335 SX generates its own refresh addresses. A ten bit refresh address counter within the 82335 SX is incremented by one at the beginning of every refresh. The refresh address appears on the address lines MA0–MA9 (MA0–MA8 if 256K DRAM are used) and is followed by RAS# activation. The address inputs A23:A1 are ignored during refresh.

During a refresh cycle, each bank of memory is refreshed when its RAS# signal activates. The 82335 SX staggers RAS# activation to reduce current surge during refresh. RAS# activation for each memory bank is separated by one PCLK# clock cycle and remains active for approximately two PCLK# cycles. All RAS# signals are activated during a refresh cycle if the REFRESH# input is held active long enough (see A.C. timing specification T83). If less than four memory banks are installed, the REFRESH# input may be de-activated after the installed banks have been refreshed. When the REFRESH# input is de-activated, any active RAS# signals will de-activate and the refresh cycle will be truncated. Figure 4.13 in the A.C. timing diagram section shows refresh timing.

2.3 Address Mapper/Decoder

2.3.1 INTRODUCTION

Several address mapping and decoding options are provided to improve performance and allow flexibility in the system memory size and configuration. These options include ROM/EPROM shadowing, mapping up to 512K addresses above the top of physical memory into physical addresses, and decoding input addresses to generate chip select signals. Selection of these options is done via programming of the configuration, roll compare, and address compare registers.

2.3.2 SHADOWING

Shadowing refers to copying data from slow memory devices like ROM and EPROM memories into RAM to speed up memory accesses. Since access to local RAM is much faster than ROM, this can improve BIOS performance considerably. The 82335 SX has built-in support for shadowing three different areas of memory: BIOS ROM, adapter ROM, and video RAM. Figure 2.5 shows the memory address ranges for each area. At least one megabyte of local memory must be installed to use the shadowing feature.

Shadowing can be done in two ways, each with different granularity. For 128 Kbyte granularity, shadowing can be selected for the BIOS ROM area (0E0000H–0FFFFFH), adapter ROM area (0C0000H–0DFFFFH), or video RAM area (0A0000H–0BFFFFH) by programming the configuration register bits ROMEN#, ENADP#, and ENV# respectively. If the configuration register bit for an area is cleared, then any access to an address in that area will be from ROM (or from system video RAM for the video area). If the configuration register bit for an area is set, then any access to that area will be from local memory. Each area can be shadowed independently. Note this matches 82335 shadowing exactly.

The second way has granularity of 32 Kbyte. The granularity is selected by programming the EXGRAN bit (bit 0) of the Granularity Enable register (I/O address 2CH). If EXGRAN = 0, then 128 Kbyte shadowing as discussed in the previous paragraph is used. The 82335 SX powers up with EXGRAN = 0. If EXGRAN = 1 then 32 Kbyte granularity is used.

If 32 Kbyte granularity is used, video RAM is shadowed the same way as in the 128 Kbyte granularity configuration. Shadowing of the BIOS and adaptor ROM areas is implemented in 32 Kbyte granularity by programming the Extended Granularity register (I/O address 2EH). Each bit in the register controls shadowing of one 32 Kbyte segment in the address range 0C0000H–0FFFFFH. The ENADP# and ROMEN# bits in the configuration register must be high if any 32 Kbyte segment in the respective address range needs to be shadowed. Section 2.3.4.4 and 2.3.4.5 contain more information on the Granularity Enable Register and the Extended Granularity Register respectively.

The BIOS ROM area FE0000H–FFFFFFH is addressable regardless of the shadow options selected. An access to an address in this region will always be from ROM.

When shadowing the BIOS ROM and adapter ROM, the ROM contents must be copied to the shadow RAM area. Since an access to one of these areas will access either the ROM *or* the RAM (not both), the following steps can be used to copy the ROM contents to the shadow area.

COPY BIOS ROM

a. Set configuration register to reflect correct DRAM size and number of banks installed, and clear the ROMEN# bit (enable BIOS ROM access).
b. Set address range compare registers to top of local memory (at least 1 MByte).
c. Copy contents of BIOS ROM to a temporary buffer in low memory (e.g., 040000h–05FFFFh).
d. Jump to code in temporary buffer to continue execution while enabling shadowing.
e. Set the ROMEN# bit in the configuration register to enable BIOS shadowing.

f. If 32 Kbyte granularity is desired
   1. Set EXGRAN = 1.
   2. Set the shadow bits in the Extended Granularity register that correspond to the address range desired for shadowing.
g. Copy contents of temporary buffer to BIOS shadow RAM area.
h. Jump back to high memory to execute code from BIOS shadow RAM area.
i. If 32 Kbyte granularity was selected, set the write-protect bits in the Extended Granularity register that correspond to the address range desired for write-protecting.

COPY ADAPTER ROM

a. Set configuration register to reflect correct DRAM size and number of banks installed, and clear the ENADP# bit (enable adapter ROM access).
b. Set address range compare registers to top of local memory (at least 1 MByte).
c. Copy contents of adapter ROM to a temporary buffer in low memory (e.g., 040000h–05FFFFh).
d. Set the ENADP# bit in the configuration register to enable adapter shadowing.
e. If 32 Kbyte granularity is desired
   1. Set EXGRAN = 1.
   2. Set the shadow bits in the Extended Granularity register that correspond to the address range desired for shadowing.
f. Copy contents of temporary buffer to adapter shadow RAM area.
g. If 32 Kbyte granularity was selected, set the write-protect bits in the Extended Granularity register that correspond to the address range desired for write-protecting.

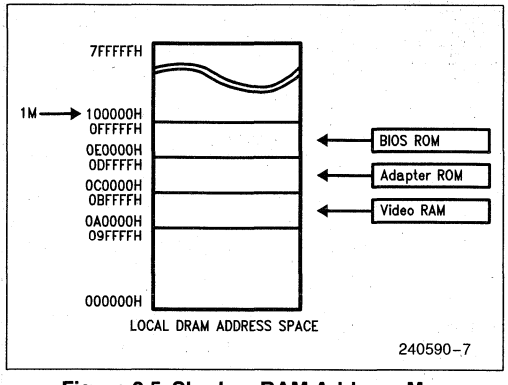


Figure 2.5. Shadow RAM Address Map

2.3.3 ROLL ADDRESS MAPPING

Roll address mapping is a method of utilizing DRAM memory space that may otherwise not be accessible. The memory space between 640K and 1M (0A0000h–0FFFFFh) is set aside for ROM and system video RAM in AT compatible systems. This space may also be associated with DRAM memory (e.g., one bank of 1M DRAM will reside at the physical address space 0K–2M). The "hidden" DRAM memory that resides in the same address space as the ROM and system video RAM can be either shadowed or it can be "rolled". Address rolling refers to re-mapping one address space to another.

The 82335 SX allows two memory address spaces to be defined; physical and logical. The physical address space corresponds to the amount of DRAM physically connected to the 82335 SX (e.g., one bank of 1M DRAM has the physical address space 0M–2M). The logical address space is the address range that the microprocessor sees as usable. The 82335 SX allows the logical address space to be larger than the physical address space. This is done by remapping logical addresses from above the physical address space (the "roll address" region) into the physical address space hidden by the ROM and video RAM areas. The physical memory between 512K and 640K may also be roll addressed when a 512K base memory is selected. Figure 2.6 illustrates address rolling from a logical to a physical address.

The amount of logical memory that can be rolled is dependent upon the amount of physical memory installed, the base memory selected and the shadow options selected. Up to 512K logical address space can be rolled into physical addresses in 128 Kbyte blocks. At least one megabyte of physical memory must be installed to utilize roll addressing.

Four physical DRAM areas can be utilized by roll addressing. These are called the BIOS ROM (0E0000h–0FFFFFh), ADAPTOR ROM (0C0000h–0DFFFFh), VIDEO RAM (0A0000h–0BFFFFh), and EXTRA (080000h–09FFFFh) areas. The BIOS ROM, ADAPTOR ROM, and VIDEO RAM areas can be selected for roll addressing if they are not selected for shadowing (configuration register bits ROMEN#, ENADP#, and ENV# = 0). The EXTRA area can be selected for roll addressing if a base memory of 512K is selected (memory configuration register bit S640 = 0). Each area can be selected independently of the others.

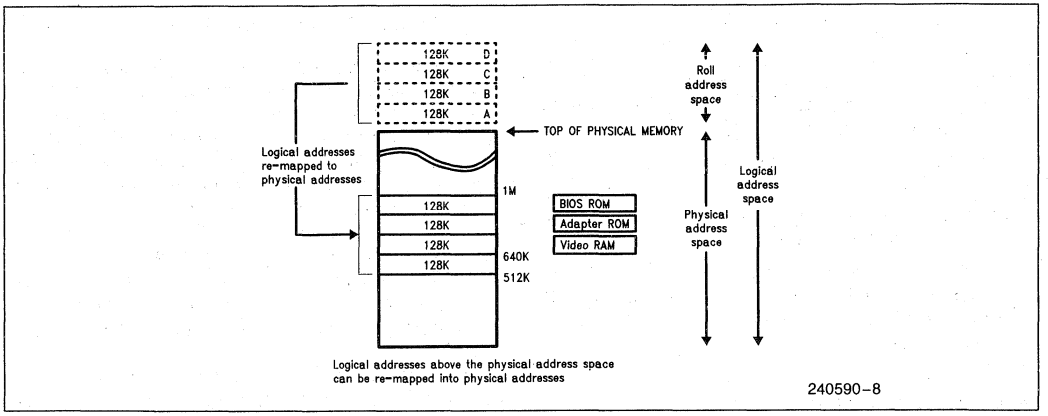
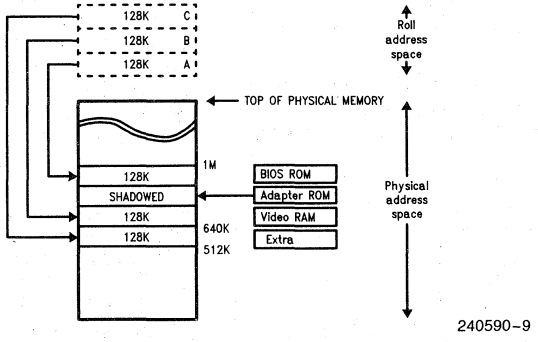


Figure 2.6. Roll Address Mapping Range

The number of DRAM areas selected for roll addressing determine the size of the roll address space. The address range for this roll address space is programmed via the roll compare registers. The roll address space must be programmed to start on a 512K boundary above the top of the physical address space. It can be programmed such that there is a gap between the physical memory connected to the 82335 SX and the roll address space (e.g., to accommodate add-in memory boards). See Section 2.3.4.2 for details on roll compare register programming.

Memory locations addressed in the roll address space do not map linearly into physical memory locations. Which roll address block maps to which physical memory block is controlled by the shadow and base memory options selected and the number of 128K byte blocks in the roll address space. Table 2.4 indicates how the roll address space maps into the physical address space for the various shadow and memory configurations available.

Figure 2.7 is an example illustrating how roll address blocks map into physical memory blocks.



EXAMPLE:
Assume:
- 512K base memory
- ADAPTOR ROM shadowed
- BIOS ROM and VIDEO RAM not shadowed
- 384K roll address space (3 blocks)

From Table 2.4, the following block mapping is found for this example:
Block A maps to BIOS area
Block B maps to VIDEO area
Block C maps to EXTRA area

Figure 2.7. Roll Address Block Mapping Example

Table 2.4. Roll Address Block Mapping

Blocks Not Shadowed	# of Blocks Rolled	Order of Logical Memory Blocks Rolled into Physical Memory Blocks			
		A	B	C	D
EXTRA	1	EXTRA			
VIDEO	1	VIDEO			
ADAPTER	1	ADAPTER			
BIOS	1	BIOS			
EXTRA, VIDEO	1	VIDEO			
	2	VIDEO	EXTRA		
EXTRA, ADAPTER	1	ADAPTER			
	2	ADAPTER	EXTRA		
EXTRA, BIOS	1	BIOS			
	2	BIOS	EXTRA		
VIDEO, ADAPTER	1	ADAPTER			
	2	ADAPTER	VIDEO		
ADAPTER, BIOS	1	BIOS			
	2	BIOS	ADAPTER		
VIDEO, BIOS	1	BIOS			
	2	BIOS	VIDEO		
EXTRA, VIDEO, ADAPTER	1	ADAPTER			
	2	ADAPTER	VIDEO		
	3	ADAPTER	VIDEO	EXTRA	
EXTRA, ADAPTER, BIOS	1	BIOS			
	2	BIOS	ADAPTER		
	3	BIOS	ADAPTER	EXTRA	
EXTRA, VIDEO, BIOS	1	BIOS			
	2	BIOS	VIDEO		
	3	BIOS	VIDEO	EXTRA	
VIDEO, ADAPTER, BIOS	1	BIOS			
	2	BIOS	VIDEO		
	3	BIOS	VIDEO	ADAPTER	
EXTRA, VIDEO, ADAPTER, BIOS	1	EXTRA			
	2	EXTRA	VIDEO		
	3	EXTRA	VIDEO	ADAPTER	
	4	EXTRA	VIDEO	ADAPTER	BIOS

LEGEND:
Physical Memory Block Addresses
BIOS AREA = 0E0000h–0FFFFFh
ADAPTER AREA = 0C0000h–0DFFFFh
VIDEO AREA = 0A0000h–0BFFFFh
EXTRA AREA = 080000h–09FFFFh

Logical Memory Block Order
Area A = lowest roll address memory block
Area D = highest roll address memory block
*Figure 2.6 illustrates logical memory block order

2.3.4 REGISTERS

There are seven registers in the 82335 SX that control the operation of the address mapping and DRAM control options. These registers are the configuration, roll compare (RC1 and RC2), address range compare (CC0 and CC1), granularity enable, and extended granularity registers. Each of these registers reside in the local I/O space of the 82335 SX and are read/writable until the LOCK bit has been set in the configuration register. The contents and purpose of each register are described in the following sections.

The 82335 SX contains a one-bit parity check enable register (PARCHEN). This register is a write only register and is not affected by the status of the LOCK bit in the configuration register. See the Parity Generator/Checker section for details on this register.

2.3.4.1 Memory Configuration Register

The memory configuration register resides at I/O location 22H upon system reset and is used to select a number of address mapping and DRAM control options. Upon reset, all bits in this register are set to zero. Figure 2.8 shows the bits used in the memory configuration register. The purpose of each bit is described in the following paragraphs.

ROMEN#: This bit is used to enable or disable shadowing of the BIOS ROM/EPROM in the address range 0E0000H–0FFFFFH. When this bit is cleared, BIOS ROM shadowing is disabled. A memory read in this range will access ROM by asserting ROMCS0# or ROMCS1# and by deactivating the OBMEM output. If BIOS ROM shadowing is disabled, this memory space can be accessed via roll addressing. See roll address mapping (Section 2.3.3) for details on address re-mapping.

Memory Configuration Register = I/O Address 22H

15	14–12	11	10	9	8	7 6	5	4	3	2 1	0
LOCK		VRO	ENV#	ENADP#	ROMSIZE	INTERL		DSIZE	S640		ROMEN#

Bit Position	Name	Function
0	ROMEN#	0 = Enable BIOS ROM/EPROM Accesses (0E0000H–0FFFFFH) 1 = Disable BIOS ROM/EPROM Accesses, Shadow Enabled
3	S640	0 = Base Memory Size is 512K 1 = Base Memory Size is 640K
4	DSIZE	0 = 256K DRAM Installed 1 = 1 Mb DRAM Installed
7, 6	INTERL	00 = 1 Mem. Bank Installed (Note 1) 01 = 2 Mem. Bank Installed (Note 1) 10 = 2 Mem. Bank Installed (Note 1) 11 = 4 Mem. Bank Installed (Note 1)
8	ROMSIZE	0 = 256K ROM/EPROM 1 = 512K ROM/EPROM
9	ENADP#	0 = Enable Adaptor ROM/EPROM Accesses (0C0000H–0DFFFFH) 1 = Disable Adaptor ROM/EPROM Accesses, Shadow Enabled
10	ENV#	0 = Enable Video RAM Accesses (0A0000H–0BFFFFH) (Note 2) 1 = Disable Video RAM Accesses, Shadow Enabled
11	VRO	Video Read Only 0 = Video Area Read-Write (0A0000H–0BFFFFH) 1 = Video Area Read-Only
15	LOCK	0 = Enable all Configuration Register Accesses 1 = Disable all Configuration Register Accesses

NOTES:
1. When more than one bank of memory is installed, banks are always interleaved.
2. For AT compatible operation, Video RAM should be enabled (ENV# = 0).

Figure 2.8. Memory Configuration Register

When this bit is set, and EXGRAN and the extended granularity register is set appropriately, BIOS ROM shadowing is enabled and memory accesses to this address range are made from local DRAM. During shadow DRAM accesses, the OBMEM signal is asserted and the ROMCS0# and ROMCS1# signals are disabled.

S640: This bit selects the base memory size. When cleared, a base memory of 512K is selected. When set, a base memory of 640K is selected. If a base memory of 512K is selected, the address range 080000H–09FFFFH can be accessed via roll addressing. See roll address mapping (Section 2.3.3) for details on address re-mapping.

DSIZE: This bit is used to indicate the type of DRAM installed. When cleared, it indicates 256K DRAM and when set, it indicates 1M DRAM installed. DRAM sizes cannot be mixed. When 256K DRAM is installed, the multiplexed address line MA9 is not used.

INTERL: These two bits indicate the number of banks of memory installed. When more than one memory bank is installed, the banks are always interleaved. If the INTERL bits are set for one memory bank, RAS and CAS signals are generated for bank 0. If set to two bank operation, RAS and CAS signals are generated for banks 0 and 1. If set to four bank operation, RAS and CAS signals are generated for banks 0 and 1. The 82335 SX does not allow 3 way interleaving. Therefore if the INTERL bits are set to three, the 82335 SX will default to 2 bank operation, i.e., RAS and CAS will be generated for banks 0 and 1 only.

ROMSIZE: This bit indicates the size of the installed ROM/EPROM. When cleared, it indicates 256K bit ROM/EPROM and when set, it indicates 512K bit ROM/EPROM is installed. This bit also affects the ROMCS0# and ROMCS1# address decode ranges when ROM shadowing is disabled. See **Chip Select Signals** (Section 2.3.5) for further information.

ENADP#: This bit is used to enable or disable shadowing of the adapter ROM area. If cleared, adapter ROM shadowing is disabled and accesses to the memory range 0C0000H–0DFFFFH will be made from ROM. When this bit is set and EXGRAN and the extended granularity register is set appropriately, adapter ROM shadowing is enabled and memory accesses in this range will be from local DRAM. If adapter ROM shadowing is disabled, this memory space can be accessed via roll addressing. See roll address mapping (Section 2.3.3) for details on address re-mapping.

ENV#: This bit is used to enable or disable shadowing of the external video RAM. If cleared, video RAM shadowing is disabled and accesses to the memory range 0A0000H–0BFFFFH will be made to/from the system video RAM. If set, video RAM shadowing is enabled and memory accesses in this range will be to/from local DRAM. For AT compatible operation, video RAM shadowing should be disabled (ENV# = 0). If video RAM shadowing is disabled, this memory space can be accessed via roll addressing. See roll address mapping (Section 2.3.3) for details on address re-mapping.

VRO: This bit selects either read/write access or read only access from the video RAM area when shadowing is selected. When video RAM shadowing is enabled and this bit is set, the local video RAM area will be read only, otherwise it will be available for both read and write access.

LOCK: This bit enables or disables external access to the configuration, roll compare, granularity enable, extended granularity, and address range compare registers. When this bit is cleared the following conditions will exist:

- The configuration, roll compare, granularity enable, extended granularity, and address range compare registers will be read/writable at even I/O addresses from 22H–2EH
- The status outputs S0# and S1# will not be generated for these I/O addresses
- If EXGRAN = 0, the shadowing DRAM area will be available for both reading and writing
- If EXGRAN = 1, the write protect bits in the extended granularity register will determine which shadowing DRAM areas are read-only and which are read-writable
- If video RAM shadowing is selected, VRO# will determine if the video shadow area is read-only. Video RAM shadowing operates the same way in the 82335 SX as it does in the 82335.

When the LOCK bit is set, the following conditions will exist:

- The configuration, roll compare, granularity enable, extended granularity, and address range compare registers will not be accessible external to the 82335 SX
- The status outputs S0# and S1# will be generated for I/O addresses between 22H–2EH
- If BIOS or adapter ROM shadowing is selected and EXGRAN = 0, then the shadowed area(s) will read-only
- If EXGRAN = 1, the write protect bits in the extended granularity register will determine which shadowing DRAM areas are read-only and which are read-writable

- If video RAM shadowing is selected, VRO# will determine if the video shadow area is read-only. Video RAM shadowing operates the same way in the 82335 SX as it does in the 82335

Once the lock bit is set, the configuration, roll compare, address compare, granularity enable, and extended granularity registers can only be accessed again by resetting the system using the SYSRESET input. It is recommended that the LOCK bit be set after the 82335 SX is properly configured.

The lock bit does not affect the operations of I/O Ports 061H or 0F0H. Writing to these ports writes to both the 82335 SX and the 82230. Since this is an I/O write to the 82230, the 82230 is required to end the cycle by activating READY286#.

When writing to the 82335 SX registers, all bits are written over. Care should be taken to insure that all bits of the data to be written are set for the intended operation.

2.3.4.2 Roll Compare Registers

Two roll compare registers, RC1 and RC2, are used to re-map logical addresses to physical addresses. The input address for a memory access is compared with the roll address ranges programmed into the roll compare registers. If the input address is a logical address within the roll address area, the 82335 SX internally translates it to a physical address and outputs the translated address to the multiplexed address bus (MA0–MA9).

The size and location of the roll address area is programmable. The size of the roll address area can be 128K, 256K, 384K, or 512K bytes depending upon the shadow and base memory options programmed into the memory configuration register. Each memory block (BIOS, adapter, or video) that is not shadowed adds 128K to the memory available for roll addressing. If a base memory of 512K is selected, then another 128K block of memory is available for roll addressing.

The roll address area can be located anywhere above the physical memory and must start on a 512K boundary. The 82335 SX allows the roll address area to be discontinuous from the local memory address space. This allows system designers to accommodate external memory cards requiring hardware switch settings to work independently of address rolling options selected.

The roll address area must be programmed to be a single contiguous area. Each roll compare register can be programmed to decode a 128K, 256K, or 512K roll address space. If a 384K roll address space is desired, then one register can be programmed to decode a 256K address space and the other to decode an adjacent 128K address space. If a 128K, 256K, or 512K roll address area is desired, it can be programmed into one register and the other register can be disabled.

The size and location of the roll address area is set by programming three sections in each roll compare register. These sections are referred to as the "compare data", "address mask", and "compare enable" sections. Figure 2.9 shows the three parts of each roll compare register. Each of these roll compare register parts are described in the following paragraphs.

The compare data bits (C23–C17) are used to specify the starting address of the roll address area covered by that register. These bits should be programmed to match the address bits A23–A17 of the first address in the address space selected for that register. If only one roll compare register is used, the roll address space must start on a 512K boundary (C17 = C18 = 0). If two roll compare registers are used, then the first register must be programmed to start on a 512K boundary and the second register must be programmed to be contiguous with the first roll compare register. The examples at the end of this section illustrate roll compare register programming.

The address mask bits (M23–M17) are used to select the size of the roll address area covered by that register. For each mask bit that is set, the corresponding input address bit and compare data bit are compared for a match. If a mask bit is cleared, the corresponding input address bit and compare data bits are ignored. If all mask bits are set, the roll address area covered by that register will be 128 Kbytes. If mask bit M17 is cleared, the register will cover a 256 Kbyte roll address area. If both M17 and M18 are cleared, the register will cover a 512K byte roll address area.

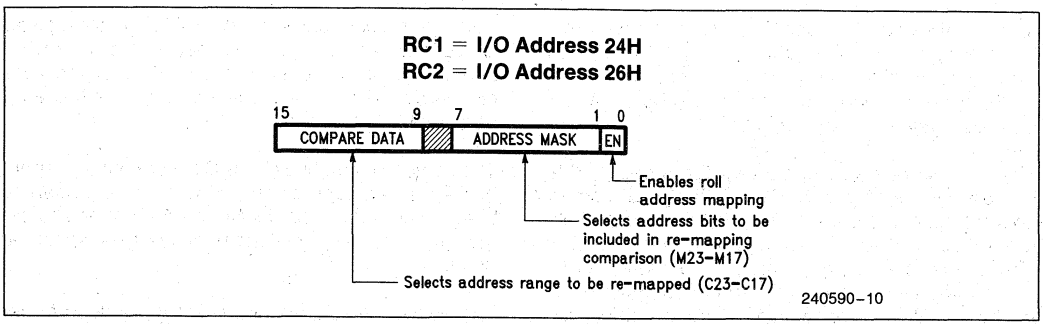


Figure 2.9. Bit Functions of the Roll Compare Registers

The compare enable bit (EN) enables roll address checking for that register. If EN is set, roll addressing is enabled for that register. If EN is cleared, roll addressing is disabled for that register and its address mask and compare data bits are ignored. Upon reset, both roll compare registers are disabled and the compare data and address mask bits are undefined.

The roll compare registers RC1 and RC2 are located at the I/O addresses 24H and 26H respectively. They can both be read and written to until the lock bit is set in the memory configuration register. Once the lock bit is set, these registers are not externally accesssible until the 82335 SX is reset by activating the SYSRESET input.

Example #1

— 4 Banks of 256K x 1 DRAM Installed

— S640 = 0, ROMEN# = 1, ENADP# = 0, ENV# = 0 (384K Roll-Over Avail.)

— Top of Physical Address Space = 1FFFFFH (2M)

— Roll Decode Range Selected = 380000-3DFFFF (384K)

Register	EN	C23-C17	M23-M17	Address Range
RC1	1	001110x	1111110	380000-3BFFFF (256K)
RC2	1	0011110	1111111	3C0000-3DFFFF (128K)

In this example, both roll compare registers are required to decode the 384K roll address area.

Example #2

— 2 Banks of 1M x 1 DRAM Installed

— S640 = 1, ROMEN# = 1, ENADP# = 0, ENV# = 0 (256K Roll-Over Avail.)

— Top of Physical Address Space = 3FFFFFH (4M)

— Roll Decode Range Selected = 600000H-63FFFFH (256K)

Register	EN	C23-C17	M23-M17	Address Range
RC1	1	011000x	1111110	600000H-63FFFFH (256K)
RC2	0	xxxxxxx	xxxxxxx	None

In this example, the 256K roll address area can be decoded with only one register. Therefore, the other register has been disabled.

2.3.4.3 Address Range Compare Registers

Two address compare registers, CC0 and CC1, are used to differentiate local memory accesses from system memory accesses (off-board memory). When a memory access is initiated, the input address is compared with the address range programmed into the address range compare registers. If the input address is within the local memory space, then the 82335 SX sets the OBMEM output high and initiates a local memory cycle. If the memory address input is outside the local memory address space, then the OBMEM output is cleared and the 286 style control signals S0#, S1#, and M/IO286# are output to initiate an off-board memory access.

If either of the roll compare registers (RC1 or RC2) are enabled, any address translation required is done before the input address is compared with the address ranges in address compare registers.

Both address range compare registers work identically. When an address comparison is made, the output of both registers is logically "ORed" together. Therefore, either register can be used for address range checking. The second address range com-

pare register was included for a feature that is not implemented. It is recommended that the unused register be disabled by clearing its compare enable bit (EN). The following paragraphs refer to the active address range compare register.

The address range compare register is designed to indicate the location and size of the local memory address space. This address space must start at address 000000H for proper DRAM addressing. The size of the local memory space is designed to be programmed into the address range compare register by the BIOS. For dynamic memory sizing, the BIOS could execute a memory autoscan routine that would determine the DRAM size being used and the number of banks installed. The size of the local memory address space could be calculated from this data and programmed into the address range compare register.

If an address space greater than 512K is programmed into the address range compare register, a memory access in the 080000H–0FFFFFH address range may not be considered a local memory access. This depends upon the base memory and shadow options selected in the memory configuration register. If a 512K base memory is selected (S640 = 0), then the address range 080000H–09FFFFH is considered non-local. If the BIOS (0E0000H–0FFFFFH), adapter (0C0000H–0DFFFFH), or video (0A0000H–0BFFFFH) areas are enabled (not shadowed), then each area not shadowed is considered non-local memory.

The size and location of the local memory address space is set by programming three sections in the address range compare register. These sections are referred to as the "compare data", "address mask", and "compare enable" sections. These sections work in a similar manner to the roll compare register sections. Figure 2.10 shows the three parts of the address compare register. Each of these address compare register parts are described in the following paragraphs.

The compare data bits (C23–C19) are used to specify the starting address of the local memory address space. These bits should all be programmed to zeroes to start the local memory address space at 000000H.

The address mask bits (M23–M19) are used to select the size of the local memory address space. For each mask bit that is set, the corresponding input address bit and compare data bit are compared for a match. If a mask bit is cleared, the corresponding input address bit and compare data bits are ignored. Table 2.5 shows how to program the compare data and address mask bits for all valid local memory configurations.

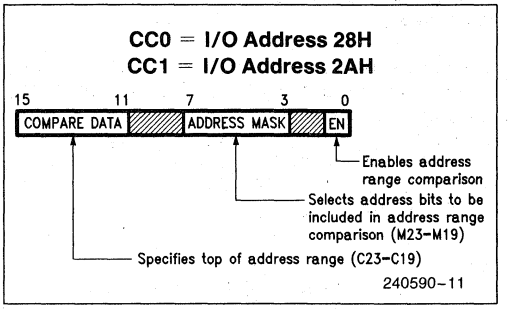


Figure 2.10. Bit Functions of the Address Range Compare Registers

Table 2.5. Address Range Compare Register Programming

Register	EN	C23–C19	M23–M19	Address Range
CC0	1	00000	11111	000000H–07FFFFH (512K)
CC0	1	0000X	11110	000000H–0FFFFFH (1M)
CC0	1	000XX	11100	000000H–1FFFFFH (2M)
CC0	1	00XXX	11000	000000H–3FFFFFH (4M)
CC0	1	0XXXX	10000	000000H–7FFFFFH (8M)

NOTES:
1. This table assumes register CC1 is disabled (EN = 0).
2. Register CC1 could be enabled over the above address ranges in place of register CC0.

The compare enable bit (EN) enables address range checking for that register. If EN is set, address range checking is enabled for that register. If EN is cleared, address range checking is disabled for that register and its address mask and compare data bits are ignored. If both address range compare registers are disabled, then all memory accesses will be non-local. Upon reset the CC0 register is enabled and programmed for a 512K local memory space, and the CC1 register is disabled.

The address compare registers CC0 and CC1 are located at the I/O addresses 28H and 2AH respectively. They are both read/writable until the lock bit is set in the memory configuration register. Once the lock bit is set, these registers are not externally accessable until the 82335 SX is reset by activating the SYSRESET input.

2.3.4.4 Granularity Enable Register

The Granularity Enable register is located at I/O location 2CH. This register is not in the 82335. It has two functions.

The EXGRAN bit (bit 0) enables extended granularity. If EXGRAN = 0, then the 82335 SX will shadow the same way the 82335 does with 128K granularity. The Extended Granularity Register will have no effect. EXGRAN = 0 upon powerup.

If EXGRAN = 1, then the 82335 will shadow and write-protect with 32 Kbyte granularity according to the Extended Granularity register.

The SPDSEL bit (bit 15) selects which speed the 82335 SX will run at. If SPDSEL = 0, the 82335 SX will assume a 32 MHz EFI (16 MHz operation). PCLK# will be a divide by 2 of EFI (see Figure 4.1A), DRAM control will correspond to Table 2.3A, and NON-TURBO mode accesses will take 6 T-states. If SPDSEL = 0 and EFI is 32 MHz, the A.C. Specifications will match the 16 MHz column in the A.C. Specifications Table. SPDSEL = 0 upon powerup.

If EFI = 40 MHz (20 MHz operation) SPDSEL should be set to 1 immediately by the BIOS. If SPDSEL = 1, PCLK# will be a divide by 2.5 of EFI (see Figure 4.1B) DRAM control will correspond to Table 2.3B, and NON-TURBO mode accesses will take 8 T-states. If SPDSEL = 1 and EFI is 40 MHz, the A.C. Specifications will match the 20 MHz column in the A.C. Specifications Table.

When writing to the SPDSEL bit, the PCLK output will change between a divide by 2 and a divide by 2.5 of EFI before the 82335 SX returns READYSX# to end the write cycle. Due to internal synchronization requirements there may be a delay of up to 20 wait states before READYSX# is activated to terminate the cycle.

When writing to the Granularity Enable register all other bits besides bits 0 and 15 should be set to zero. Once the LOCKbit is set, this register is not externally accessible until the 82335 SX is reset by activating the SYSRESET input.

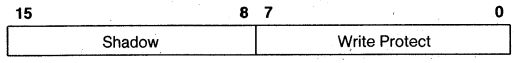
2.3.4.5 Extended Granularity Register

The Extended Granularity register is located at I/O address 2EH. This register is not in the 82335. Its function is to increase the granularity of shadowing and write-protect for the address range 0C0000H–0FFFFFH. EXGRAN must equal 1 for this register to have any effect on 82335 SX operation.

Shadowing can only occur if the corresponding bit in the configuration register is inactive. For example, ENADP# must be high (shadowing enabled) if any 32 Kbyte block in the adapter ROM space is to be shadowed. Similarly, if no 32 Kbyte block in the adapter ROM space is to be shadowed, then ENADP# should be low. ROMEN# works the same way. The function of each of the bits in the Extended granularity register is shown in Figure 2.10. Once the lock bit is set this register is not externally accessable until the 82335 SX is reset by activating the SYSRESET input. Note that the write-protect bits take effect immediately upon programming. They do not wait for the LOCK bit to be set to take effect.

When the extended granularity feature is used and BIOS is shadowed from ROM, it is recommended to shadow the entire BIOS from 0E0000h to 0FFFFFh. When only part of the BIOS ROM is shadowed with extended granularity, any unshadowed block between 0E0000h and 0FFFFFh will **not** be accessible to the CPU.

Extended Granularity Register = I/O Location 2EH



Bit Position	Name	Function
15	ROMEN7#	0 = Disable Shadow F8000H–FFFFFH 1 = Enable Shadow F8000H–FFFFFH
14	ROMEN6#	0 = Disable Shadow F0000H–F7FFFH 1 = Enable Shadow FF7FFFH
13	ROMEN5#	0 = Disable Shadow E8000H–EFFFFH 1 = Enable Shadow E8000H–EFFFFH
12	ROMEN4#	0 = Disable Shadow E0000H–E7FFFH 1 = Enable Shadow E0000H–E7FFFH
11	ROMEN3#	0 = Disable Shadow D8000H–DFFFFH 1 = Enable Shadow D8000H–DFFFFH
10	ROMEN2#	0 = Disable Shadow D0000H–D7FFFH 1 = Enable Shadow D0000H–D7FFFH
9	ROMEN1#	0 = Disable Shadow C8000H–CFFFFH 1 = Enable Shadow C8000H–CFFFFH
8	ROMEN0#	0 = Disable Shadow C0000H–C7FFFH 1 = Enable Shadow C0000H–C7FFFH
7	WRPRT7	Shadow RAM at F8000H–FFFFFH in Read/Write Mode 0 = Read/Write (Default), 1 = Read Only (Write Protected)
6	WRPRT6	Shadow RAM at F0000H–F7FFFH in Read/Write Mode 0 = Read/Write (Default), 1 = Read Only (Write Protected)
5	WRPRT5	Shadow RAM at E8000H–EFFFFH in Read/Write Mode 0 = Read/Write (Default), 1 = Read Only (Write Protected)
4	WRPRT4	Shadow RAM at E0000H–E7FFFH in Read/Write Mode 0 = Read/Write (Default), 1 = Read Only (Write Protected)
3	WRPRT3	Shadow RAM at D8000H–DFFFFH in Read/Write Mode 0 = Read/Write (Default), 1 = Read Only (Write Protected)
2	WRPRT2	Shadow RAM at D0000H–D7FFFH in Read/Write Mode 0 = Read/Write (Default), 1 = Read Only (Write Protected)
1	WRPRT1	Shadow RAM at C8000H–CFFFFH in Read/Write Mode 0 = Read/Write (Default), 1 = Read Only (Write Protected)
0	WRPRT0	Shadow RAM at C0000H–C7FFFH in Read/Write Mode 0 = Read/Write (Default), 1 = Read Only (Write Protected)

Figure 2.11. Extended Granularity Register

2.3.5 CHIP SELECT SIGNALS

The address mapper/decoder uses the configuration, roll compare, and address range compare register contents along with input addresses to generate the following output signals:

ROMCS0# — ROM 0 Chip Select

ROMCS1# — ROM 1 Chip Select

LMEGCS — Lower Meg Chip Select

OBMEM — On-Board Memory Address Range

The ROM chip select signals are functions of the ROMSIZE and ROMEN# bits in the configuration register as well as the input address. If ROM shadowing is enabled (ROMEN# = 1), then the ROM chip select outputs will be disabled for the 0E0000H–0FFFFFH address range. The ROM chip select signals for the FE0000H–FFFFFFH address range are not affected by the shadowing options selected. If ROM shadowing is disabled or the input address is between FE0000H and FFFFFFH, then the ROM chip select outputs will be activated as follows:

If ROMSIZE = 0 (256K ROM)

ROMCS0# decodes the address ranges 0E0000H–0EFFFFH and FE0000H–FEFFFFH.

ROMCS1# decodes the address ranges 0F0000H–0FFFFFH and FF0000H—-FFFFFFH.

If ROMSIZE = 1 (512K ROM)

ROMCS0# is inactive

ROMCS1# decodes the address ranges 0E0000H–OFFFFFH and FE0000H–FFFFFFH.

The lower meg chip select output (LMEGCS) is a function of both the input address and M/IO# inputs. It is activated whenever a memory address within the first megabyte of memory is decoded. It is inactive during I/O cycles.

The on-board memory output (OBMEM) is a function of the input address, M/IO#, address range comparators, roll comparators, and the bits ROMEN#, ROMSIZE, DSIZE, and S640 in the configuration register. It is used to differentiate local DRAM access from system RAM, ROM, or I/O accesses. OBMEM is active (high) during local DRAM accesses. If the M/IO# or address input signals are left floating, (during a refresh or DMA cycle for example) the OBMEM output will be undefined.

2.4 Ready Generator

Every 386 SX microprocessor bus cycle must be terminated by a ready signal to the processor. The 82335 SX generates this ready signal by translating external and internal ready inputs to a single ready output (READYSX#). The 82335 SX internally generates a ready signal for local DRAM cycles, on-chip I/O cycles, and numeric coprocessor cycles. For all other cycles (system I/O, system memory, etc.), the 82335 SX translates the ready input READY286# to the 386 SX compatible READYSX# output. The 82335 SX input EXTRDY# can be used to delay or mask the READYSX# output.

For on-chip I/O and local memory cycles, the internal ready generator in the 82335 SX determines the appropriate number of wait states to insert (if any), and activates the READYSX# output at the correct time. The READYSX# pulse can be delayed during local memory cycles by deactivating the EXTRDY input. If EXTRDY is sampled inactive, then the READYSX# output will be delayed until EXTRDY is sampled active. EXTRDY is sampled on the rising edge of CLK2 during phase 1 of each T2 and T2P state.

The external input READY286# is used to terminate all cycles other than local memory cycles, local I/O cycles, or numeric coprocessor cycles.

The READY286# input is designed to be driven by the 82230 READY# pin and is used to identify the completion of system bus cycles. If the READY286# input is sampled active during a system bus cycle and the EXTRDY input is high, the 82335 SX generates a READYSX# pulse. The READY286# input is sampled on the falling edge of PCLK# when an internal clock (half the frequency of PCLK#) is high. This synchronizes ready signals between the 82230/82231 and the 386 SX. The READY286# input is only sampled during system bus cycles.

The READYNPX# input is ignored by the 82335 SX. For Numeric Coprocessor cycles the 82335 SX will activate READYSX# after 1 wait-state. If no coprocessor is detected upon power up, the 82335 SX will not generate READYSX#, for numerics coprocessor cycles. If no coprocessor is present the 82335 SX will output S0# and S1# to the 82230/82231. The 82230/82231 will activate READY286# after completing a standard PC-AT numerics coprocessor cycle.

Deactivation of EXTRDY will delay READY SX# on numerics cycles the same way as it does on local DRAM cycles (if a coprocessor is present).

If EXTRDY is inactive (low) when the READY286# input is sampled, then no READYSX# pulse is generated. READY286# is **masked** (not delayed) if EXTRDY is low when sampled.

Setup and hold times for the READY286#, and EXTRDY inputs must be met to guarantee correct operation.

2.5 Bus Cycle Translator

The 82335 SX has a built in interface unit that translates 386 SX processor control signals to 80286 control signals. This bus cycle translator identifies the bus cycle being performed, monitors the CPU T-states, and outputs 80286-like bus control signals to the 82230/82231 and other components in a PC/AT system. It also receives 80286 control inputs and translates them into 386 SX processor compatible signals when required.

As the bus cycle translator monitors control inputs from both the 386 SX processor and the 82231, it determines what type of cycle is being requested. If one of the following cycles is being requested:

1. I/O Access
2. System Memory Access
3. Halt/Shutdown
4. Interrupt

then the bus tracker monitors the timing of the bus cycle and simultaneously outputs 80286-type bus control signals. The control signals output are S0#, S1#, and M/IO286#. The S0# and S1# outputs are not activated for local memory accesses, for numerics coprocessor accesses when a coprocessor is present or for accesses to the 82335 SX on-chip I/O (programming of the on-chip registers).

In addition to controlling status output to the 82230/82231, the bus cycle translator also controls the hold request (HRQSX) input to the 386 SX processor. A hold request signal coming from the 82230 (HRQ286) is translated into a 386 SX processor-compatible output and driven to the 386 SX processor. The 386 SX processor responds with a hold acknowledge (HLDASX) to the 82335 SX which then translates that to a HLDA output to the 82230/82231.

2.6 Math Coprocessor Interface

The 82335 SX provides interface signals between the 386 SX CPU and 387 SX numeric coprocessor to allow the 387 SX to function in a PC/AT environment with proper error handling. The 82335 SX can also identify 387 SX bus cycles and has built-in logic to automatically sense whether a 387 SX coprocessor is present in the system. Figure 2.12 shows a diagram of the coprocessor interface signals.

During coprocessor bus cycles, the 386 SX CPU initiates I/O accesses to addresses above 800000H. When a 387 SX is installed the 82335 SX recognizes these I/O accesses as coprocessor bus cycles and inhibits the status outputs S0# or S1#. If there is no 387 SX installed, the 82335 SX will activate the S0# and S1# status outputs when a coprocessor bus cycle is initiated.

The 82335 SX can detect the presence of a 387 SX coprocessor by sampling the ERROR# input during a system reset. The ERROR# input is sampled at the falling edge of the SYSRESET pulse. If ERROR# is sampled low, a 387 SX coprocessor is present. If ERROR# is sampled high, there is no coprocessor. The ERROR#, and BUSYNPX# inputs contain internal pull-up resistors to prevent false inputs when there is no coprocessor in the system.

If the 82335 SX does not detect a coprocessor in the system, the ERROR# and PEREQNPX inputs are ignored, and the PEREQSX and BUSYSX# outputs do not follow the PEREQNPX and BUSYNPX# inputs. If there is no coprocessor, the 82335 SX forces the PEREQSX output low. This insures that the 386 SX does not get any false processor extension requests. To prevent a 387 SX instruction from holding up the processor by waiting for the BUSYSX# signal to go inactive, the BUSYSX# output toggles high and low when there is no coprocessor. It toggles every time REFRESH# is activated.

If a coprocessor is present, the 82335 SX translates the BUSYNPX# and PEREQNPX inputs into BUSYSX# and PEREQSX outputs to the processor. During normal operation the BUSYNPX# and PEREQNPX signals are passed straight through to the BUSYSX# and PEREQSX outputs. There are two exceptions to this: during a system reset and when a numeric coprocessor error has occured.

During a system reset the BUSYSX# output is forced low to induce a processor self-test. The BUSYSX# output is deactivated on the falling edge of ADS# during the first bus cycle and follows the BUSYNPX# input thereafter.

When a numeric coprocessor error occurs, both the BUSYSX# and PEREQSX outputs are activated. Activation of the ERROR# input causes the current state of the BUSYSX# output to be latched and held. When the BUSYNPX# input deactivates, the PEREQSX output is activated. This holds processing on the 386 SX processor while completing the 387 SX coprocessor transfers. Both the BUSYSX# and PEREQSX outputs deactivate and return to normal

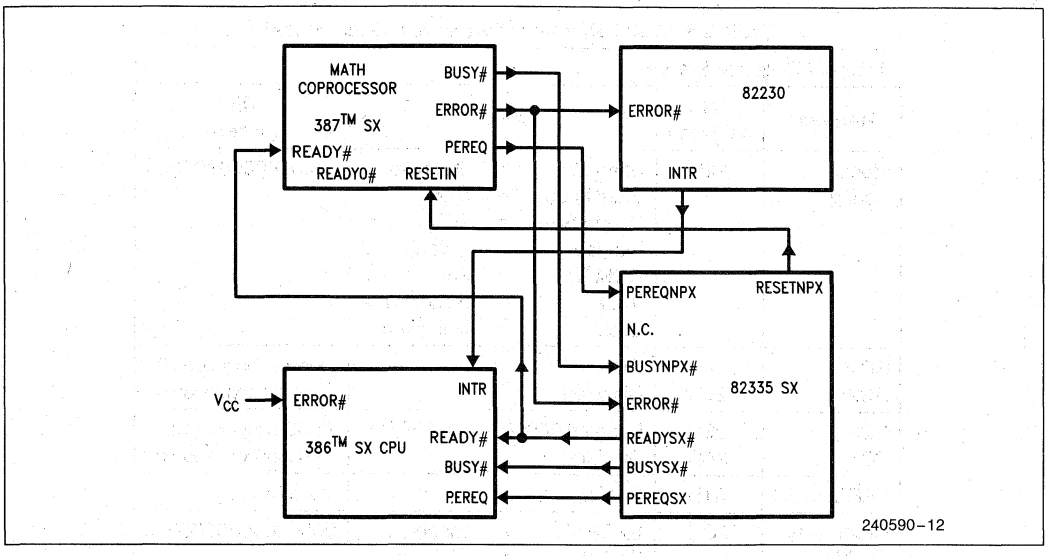


Figure 2.12. Math Coprocessor Interface

operation when an I/O write to address 0F0H is executed.

The 82335 SX handles unmasked overflow/underflow exceptions the same way PC/AT systems handle them. Both unmasked and masked overflow/underflow exceptions are treated like masked exceptions. If an error is triggered by an overflow/underflow exception, a value is written out to memory.

2.7 Clock Generator

The 82335 SX clock generator is used to synchronize the CPU, coprocessor, and peripherals by converting an input frequency into the system clock outputs CLK2 and PCLK#. The input frequency is obtained from an external oscillator connected to the 82335 EFI pin. A standard TTL oscillator can be used for this. This EFI input is internally buffered and output to the 386 SX processor and 387 SX coprocessor via the CLK2 output. It is also divided by 2 (16 MHz) or 2.5 (20 MHz) and output to the 82230/82231 using the PCLK# output.

Upon power-up both the CLK2 and PCLK# outputs are active. They are not phase synchronized, however, until after the falling edge of the RESETSX input. To synchronize the clocks, PCLK# is held high until the rising edge of CLK2. The second falling edge of PCLK# after this CLK2 edge will start a phase two clock cycle. Figure 4.2a shows the clock synchronization sequence. A CPU reset using only the RESETCPU input will not affect clock synchronization.

2.8 Reset Synchronizer

The 386 SX microprocessor requires a synchronous reset input to initialize the processor. The 82335 SX has a reset synchronizer that has two asynchronous reset inputs and generates synchronous reset signals to both the processor and numeric coprocessor. The two reset inputs to the 82335 SX (RESET CPU and SYSRESET) are used to initiate either a CPU reset or an entire system reset. The 82335 SX internal registers and output signals are initialized following a system reset. These internal registers are not affected by a processor reset.

The RESETCPU input is used to reset the processor only. This is frequently used to switch the processor from protected mode to real mode. When the RESETCPU input is activated, the 82335 SX synchronizes and activates the processor reset output (RESETSX). The RESETCPU input must be held active for at least 16 CLK2 cycles to guarantee a sufficient RESETSX pulse width to the processor. No other outputs or internal registers are affected by this reset. Figure 4.2b in the A.C. Timing Diagram Section shows processor reset timing.

The 82335 SX SYSRESET and RESETCPU inputs are used together to initiate a system wide reset and initialize the 82335 SX. When SYSRESET and RESETCPU are concurrently activated, both the processor reset (RESETSX) and coprocessor reset (RESETNPX) signals are activated. The 82335 SX internal registers and output signals are initialized to the values shown in Table 2.6.

Table 2.6. 82335 SX Initial State after System Reset

Internal Register Status			
Register	**I/O Address**	**Status**	**Bit Pattern**
Mem. Config.	22H	Shadow = Disabled Base Mem. = 512K DRAM Size = 256K Interleave = 1 Bank ROM Size = 256K VRO = Read/Write Lock = Unlocked	0000000000000000
RC1 RC2	24H 26H	Disabled Disabled	xxxxxxxxxxxxxxx0 xxxxxxxxxxxxxxx0
CC0 CC1	28H 2AH	Enabled 000000H–07FFFFH Disabled	00000xxx11111xx1 xxxxxxxxxxxxxxx0
PARCHKEN	61H	Enabled	xxxxx0xx
Granularity Enable	2CH	EXGRAN = Disabled Speed = 16 MHz	0000000000000000
Extended Granularity	2EH	Disabled	0000000000000000

Output Pin Status		
Low	**High**	**Undefined**
BUSYSX# HLDA HRQSX PEREQSX OBMEM WE#	CASx# RASx# DEN# NA# PERROR# READYSX# ROMCSx# S0# S1#	LMEGCS# MA0–MA9 M/IO286# DIR

NOTE:
x in register bit pattern denotes undefined bit state.

When a system reset occurs, the 82335 SX forces the BUSYSX# output low to initiate a processor self-test. To guarantee correct processor self-test results, the SYSRESET and RESETCPU inputs must be held active for at least 80 CLK2 cycles. If the system designer chooses not to utilize the data returned by the processor self-test, the SYSRESET and RESETCPU input signals only need to be held active for 16 CLK2 cycles to guarantee correct system reset.

During a system reset the following sequence of events occurs:

- RESETSX and RESETNPX are activated.
- The DRAM control signals RASx#, CASx#, and DEN# are deactivated.
- BUSYSX# is held low to induce processor self-testing.
- The 82335 SX internal registers and output pins are initialized to the values shown in Table 2.6.
- The ERROR# input is sampled on the falling edge of SYSRESET to determine if a numeric coprocessor is present. If the ERROR# input is sampled low a coprocessor is present, otherwise there is no coprocessor.
- The PCLK# output is synchronized with the CLK2 output.
- RESETSX and RESETNPX are deactivated.

2.9 Parity Generator/Checker

The 82335 SX has a built-in parity generator and checker to maintain data integrity for the local memory. During local memory write or DMA write cycles the 82335 SX generates two parity outputs from the data inputs D0–D15. The 82335 SX calculates even parity for both the high byte and low byte and outputs the results on the parity high (PARH) and parity low (PARL) pins. During local memory read and DMA read cycles the 82335 SX compares the data (D0–D15) and parity (PARH and PARL) inputs and generates an error signal (PERROR#) if a parity error is detected. PARH and PARL are three-state input/output pins designed to directly drive the DRAM.

If a parity error is detected, the 82335 SX drives the PERROR# output low and latches it. PERROR# is cleared by either a system reset (SYSRESET pulsed high) or by disabling parity checking in the 82335 SX. The following paragraph details how to enable/disable parity checking.

Parity checking is enabled/disabled by a one bit parity check enable register (PARCHKEN). This register is located in bit 2 of the I/O address 61H. When programmed to "0", parity checking is enabled. When programmed to "1", parity checking is disabled and the PERROR# output is deactivated (PERROR# > V_{OH} min).

The PARCHKEN register is a write-only register. In a system that contains both an 82335 SX and 82231, a write to I/O port 61H will write to the parity registers in both chips. A read from I/O port 61H will read from the 82231 register. Care must be taken when writing to this register since the 82231 utilizes several other bits at I/O location 61H register to control other system functions.

2.10 General System Considerations

1. The RAS0#–RAS3#, CASH0#–CASH3#, and CASL0#–CASL3# output buffers are designed to directly drive the heavy capacitive loads of the dynamic RAM arrays. To keep the RAM driver outputs from ringing excessively in the system environment it is necessary to match the output impedance with the RAM array by using series resistors. Each application may have different impedance characteristics and may require different series resistance values. The series resistance values should be determined for each application.
2. If the capacitive loading on the MA0–MA9 outputs exceeds the maximum capacitive loading specification (see AC DRAM Timing Specifications), then buffering of the MA0–MA9 outputs is recommended. The MA0–MA9 outputs can directly drive approximately two banks of memory.
3. The NA# pin on the 82335 SX must be connected to NA# on the 386 SX processor.
4. If there is no DRAM installed in the physical address space 080000H–0FFFFFH, then shadowing and roll address mapping must be disabled.
5. When setting the LOCK bit in the configuration register, the entire contents of the configuration register must be written. Writing to a register in the 82335 SX will overwrite all bits in that register.

3.0 MECHANICAL DATA

3.1 Package Dimensions

The 82335 SX is available in a 132 lead plastic quad flat pack (PQFP) package. Table 3.1 and Figures 3.1–3.5 show the physical dimensions of this package.

Table 3.1. Intel Case Outline Dimensions for 132 Lead Plastic Quad Flat Pack 0.025 Inch Pitch

Symbol	Description	Inch		mm	
		Min	Max	Min	Max
A	Package Height	0.160	0.170	4.06	4.32
A1	Standoff	0.020	0.030	0.51	0.76
D, E	Terminal Dimension	1.075	1.085	27.31	27.56
D1, E1	Package Body	0.947	0.953	24.05	24.21
D2, E2	Bumper Distance	1.097	1.103	27.86	28.02
D3, E3	Lead Dimension	0.800 REF		20.32 REF	
L1	Foot Length	0.020	0.030	0.51	0.76
Issue	IWS Preliminary 1/15/87				

Symbol List

Letter or Symbol	Description of Dimensions
A	Package Height: Distance from Seating Plane to Highest Point of Body
A1	Standoff: Distance from Seating Plane to Base Plane
D/E	Overall Package Dimension: Lead Tip to Lead Tip
D1/E1	Plastic Body Dimension
D2/E2	Bumper Distance
D3/E3	Footprint
L1	Foot Length

NOTES:
1. All dimensions and tolerances conform to ANSI Y14.5M-1982.
2. Datum plane H located at the mold parting line and coincident with the bottom of the lead where lead exits plastic body.
3. Datums A B and D to be determined where center leads exit plastic body at datum plane H.
4. Controlling Dimension, Inch.
5. Dimensions D1, D2, E1, and E2 are measured at the mold parting line and do not include mold protrusion. Allowable mold protrusion is 0.18 mm (0.007 in) per side.
6. Pin 1 identifier is located within one of the two zones indicated.

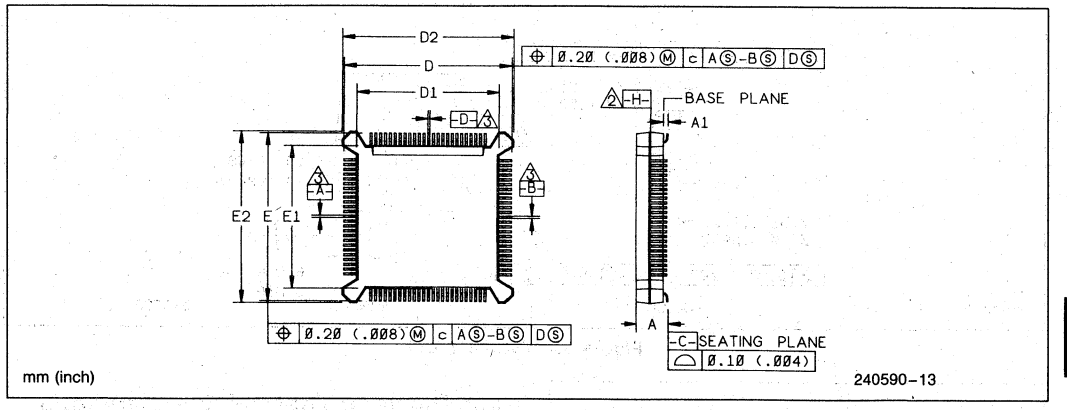


Figure 3.1. Principal Dimensions and Datums

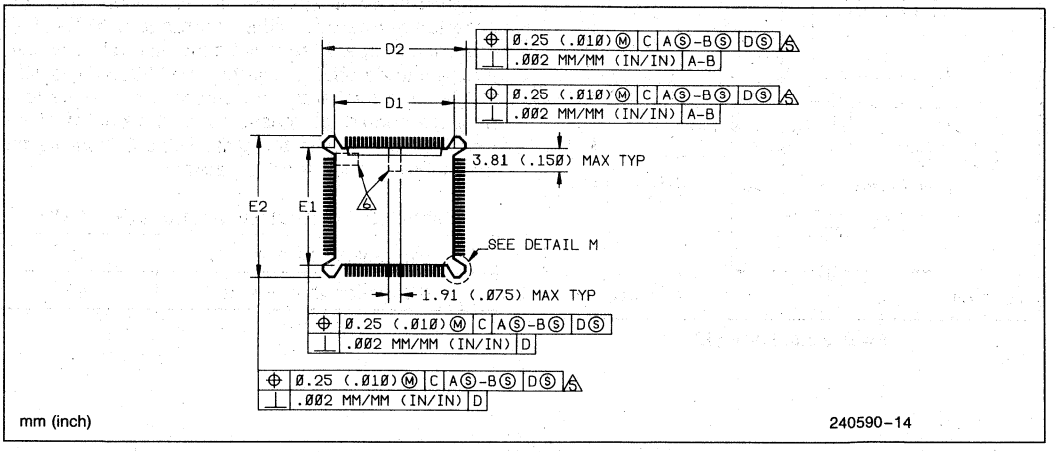


Figure 3.2. Molded Details

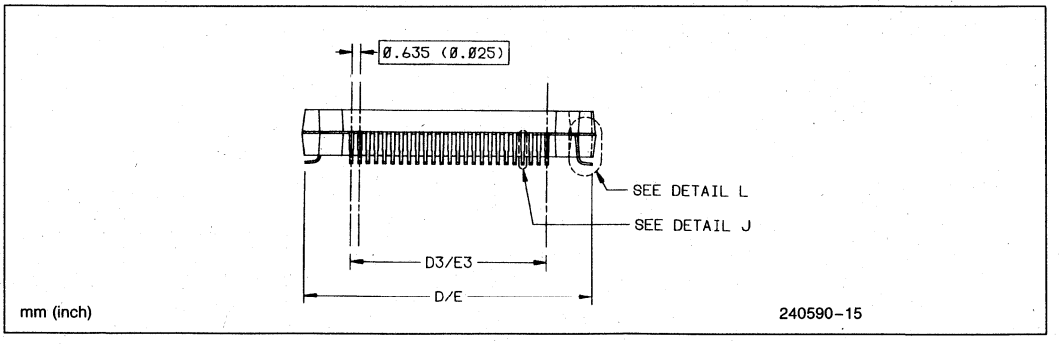


Figure 3.3. Terminal Details

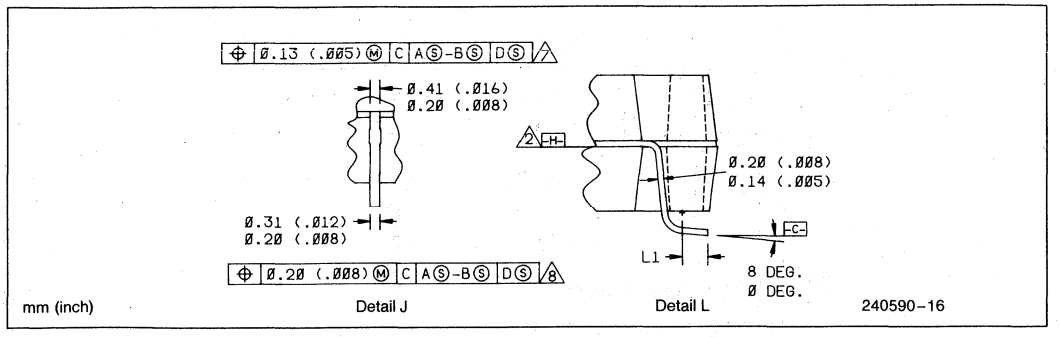


Figure 3.4. Typical Lead

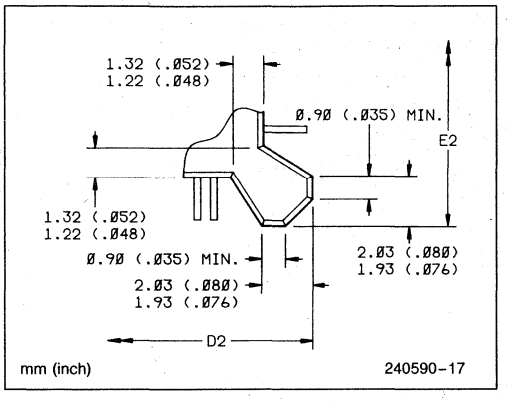


Figure 3.5. Detail M

3.2 Package Thermal Specifications

The 82335 SX is specified for operation when the case temperature is within the range of 0°C–85°C. The case temperature may be measured in any environment to determine whether the 82335 SX is within the specified range of operation. The case temperature should be measured at the center of the top surface opposite the pins. Table 3.2 shows the thermal resistance for this package.

Table 3.2. Thermal Resistances (°C/Watt)

θ Junction to Case	12
θ Junction to Ambient	40

4.0 ELECTRICAL DATA

4.1 Maximum Ratings

Table 4.1. Maximum Ratings

Parameter	Maximum Rating
Storage Temperature	−65°C to +150°C
Case Temperature under Bias	−65°C to +110°C
Supply Voltage with Respect to V_{SS}	−0.5V to +6.5V
Voltage on Other Pins	−0.5V to V_{CC} + 0.5V

Table 4.1 is a stress rating only, and functional operation at the maximums is not guaranteed. Functional operating conditions are given in Section 4.2, D.C. Specifications and Section 4.3, A.C. Specifications.

Extended exposure to the Maximum Ratings may affect device reliability. Furthermore, although the 82335 SX contains protective circuitry to resist damage from static electric discharge, always take precautions to avoid high static voltages or electric fields.

1

4.2 D.C. Electrical Specifications

Functional Operating Range: V_{CC} = 4.5V to 5.5V, T_{CASE} = 0°C to +85°C

Symbol	Parameter	Min	Max	Unit	Notes
V_{IL}	Input Low Voltage	−0.3	0.8	V	(Note 1)
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.3	V	
V_{ILC}	EFI Input Low Voltage	−0.3	0.8	V	(Note 1)
V_{IHC}	EFI Input High Voltage	2.4	V_{CC} + 0.3	V	
V_{OLC}	CLK2 Output Low Voltage		0.45	V	I_{OL} = 2 mA
V_{OHC}	CLK2 Output High Voltage	V_{CC} − 0.8	V_{CC} + 0.3	V	I_{OH} = −1 mA
V_{OLP}	PCLK# Output Low Voltage		0.45	V	I_{OL} = 2 mA
V_{OHP}	PCLK# Output High Voltage	V_{CC} − 0.5	V_{CC} + 0.3	V	I_{OH} = −1 mA
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −1 mA
I_{LI}	Input Leakage Current for All Pins except Group One		±15	μA	$0 < V_{IN} < V_{CC}$ (Note 2)
I_{IL}	Input Sustaining Current for Group One Pins		−400	μA	V_{IL} = 0.45V, (Note 2)
I_{LO}	3-State Output Leakage Current		±15	μA	$V_{OL} < V_{IN} < V_{CC}$
I_{CC}	Supply Current		150	mA	(Note 3)
C_{IN}	Input Capacitance		10	pF	(Note 4)
C_{OUT}	Output or I/O Capacitance		12	pF	(Note 4)

NOTES:
1. The min value, −0.3, is not tested.
2. Group one = BUSYNPX#, ERROR#, M/IO#, and D/C#. All these inputs have internal pullup resistors.
3. CLK2 = 32 MHz or 40 MHz, maximum loading on DRAM address and control pins.
4. Not tested. These are guaranteed by design characterization.

4.3 A.C. Specifications

Unless otherwise specified, all timings are referenced at V = 1.5V, and all timing values are in nano-seconds, all voltages are in volts, and all output loadings are 50 pF.

A.C. Timings for Clocks

Symbol	Parameter	Figure	16 MHz		20 MHz		Units	Notes
			Min	Max	Min	Max		
	Operating Frequency			16		20	MHz	
T1	EFI Period	4.1	31	33	25	27		
T2	EFI High Time	4.1	12		11		ns	at 1.5V
T3	EFI Low Time	4.1	12		11		ns	at 1.5V
T4	EFI Fall Time	4.1		6		6	ns	2.4V to 0.4V (Note 1)
T5	EFI Rise Time	4.1		6		6	ns	0.4V to 2.4V (Note 1)
T6	CLK2 Period	4.1	31	33	25	27		
T7a	CLK2 High Time	4.1	9		8		ns	at 2V
T7b	CLK2 High Time	4.1	5		5		ns	at (V_{CC} − 0.8)
T8a	CLK2 Low Time	4.1	9		8		ns	at 2V
T8b	CLK2 Low Time	4.1	7		6		ns	at 0.8V
T9	CLK2 Fall Time	4.1		8		8	ns	(V_{CC} − 0.8) to 0.8 (Note 1)
T10	CLK2 Rise Time	4.1		8		8	ns	0.8 to (V_{CC} − 0.8) (Note 1)
T11	PCLK# Delay from CLK2	4.1		10		14		

Output Loadings: CLK2: 50 pF (20 MHz), 75 pF (16 MHz) Max
PCLK#: 75 pF Max

NOTE:
1. These are not tested. They are guaranteed by design characterization.

A.C. Timings for DRAM Controller Unit

Unless otherwise specified, all A.C. timings referenced to CLK2 refer to the CLK2 rising edge in phase one.

Symbol	Parameter	Figure	16 MHz		20 MHz		Unit	Notes
			Min	Max	Min	Max		
T12a	RAS# Active Delay from CLK2	4.3		62		50	ns	(Note 1)
T12b	RAS# Active Delay from CLK2			62		30	ns	(Note 2)
T13	RAS# Inactive Delay from CLK2	4.3	4		4		ns	
T14	Row Addr Setup to RAS# Active	4.3	15		15		ns	(Note 8)
T14a	Row Addr Setup to RAS# Active	4.3	2		2		ns	(Notes 9, 10)
T15	Row Addr Hold from RAS# Active	4.3	20		15		ns	(Note 8)
T15a	Row Addr Hold from RAS# Active	4.3	15		10		ns	(Notes 10, 12)
T16	RAS# Precharge Time	4.5	91		70		ns	(Note 1)
T16a	RAS# Precharge Time		91		91			(Note 2)
T17	Col Addr Setup to CAS# Active	4.3	7		15		ns	(Note 8)
T17a	Col Addr Setup to CAS# Active		0		2			(Notes 9, 10)
T18	Col Addr Hold from CAS# Active	4.3	25		20		ns	(Note 8)
T18a	Col Addr Hold from CAS# Active	4.3	23		18		ns	(Note 11)
T19a	CAS# Active Delay from CLK2	4.3		55		45	ns	(Note 3)
T19b	CAS# Active Delay from CLK2 (Read)	4.6		43		34	ns	(Note 2)
T19c	CAS# Active Delay from CLK2 (Write)	4.4	19	39	15	34	ns	(Note 4)

Maximum Output Loadings: WE#: 270 pF MA0–MA9: 240 pF RAS#: 120 pF CAS#: 75 pF

A.C. Timings for DRAM Controller Unit (Continued)

Unless otherwise specified, all A.C. timings referenced to CLK2 refer to the CLK2 rising edge in phase one.

Symbol	Parameter	Figure	16 MHz		20 MHz		Unit	Notes
			Min	Max	Min	Max		
T20	CAS# Inactive Delay from CLK2	4.3	6		6		ns	
T21a	CAS# Active Pulse Width	4.4	37		31		ns	(Note 1)
T21b	CAS# Active Pulse Width	4.6	62		50		ns	(Note 2)
T22a	CAS# Precharge Pulse Width	4.4	22		21		ns	(Note 1)
T22b	CAS# Precharge Pulse Width	4.6	50		40		ns	(Note 2)
T23a	RAS# Hold from CAS# Active	4.5	37		31		ns	(Note 1)
T23b	RAS# Hold from CAS# Active		62		50		ns	(Note 2)
T24	Read Data Setup before CLK2	4.4	11		9		ns	(Note 5)
T25	Read Data Hold from CAS# Inactive	4.4	2		2		ns	
T26	Write Data Delay from CLK2	4.4		9		6	ns	(Note 6)
T27	Write Data Hold after CLK2	4.4	0		0		ns	
T28	PARL/PARH Setup to CAS# Active	4.4	0		0		ns	
T29	PARL/PARH Hold from CAS# Active	4.4	28		20		ns	
T30	DEN# Active from CLK2	4.4	0	20	0	20	ns	
T31	DEN# Inactive from CLK2	4.4	6		6		ns	
T32	DIR Delay from CLK2	4.4	25	55	20	45	ns	
T33	WE# Setup to CAS# Active	4.6	5		5		ns	
T34	386 SX CPU Addr Setup to CLK2	4.3	26		23		ns	
T35a	Col Address Delay from CLK2	4.4		45		26	ns	(Note 8)
T35b	Col Address Delay from CLK2	4.4		55		35	ns	(Note 9)
T36	WE# Hold from CAS# Active (Write)	4.6	30		25		ns	
T37	WE# Hold from CAS# Inactive (Read)	4.6	0		0		ns	
T79	RAS# Active Delay from MEMx#	4.10		55		55	ns	(Note 7)
T80	RAS# Pulse Width	4.10	150		115		ns	(Note 7)
T81	CAS# Inactive from Memx#	4.10	9	40	7	40	ns	(Note 7)
T82	CAS# Active Delay from RAS# Active	4.10	55	100	45	100	ns	(Note 7)
T83a	Refresh Pulse Width (1 Memory Bank)	4.13	280		280		ns	
T83b	Refresh Pulse Width (2 Memory Bank)	4.13	310		310		ns	
T83c	Refresh Pulse Width (4 Memory Bank)	4.13	370		370		ns	
T84	DEN# Active to CAS# Active Delay	4.5	7		7		ns	

NOTES:
1. F1/F4 mode.
2. WO1/WO2 mode.
3. All F1/F4 cycles except page mode write.
4. F1/F4 page mode write cycles and WO1/WO2 mode write cycles.
5. For parity checker.
6. To guarantee PARL, PARH timings.
7. DMA/MASTER mode timings.
8. With C_{Load} = 50 pF.
9. With MA0–MA9 C_{Load} = 240 pF.
10. With RAS# C_{Load} = 120 pF.
11. With CAS# C_{Load} = 75 pF.
12. If MA0–MA12 use buffers, then minimum buffer delay must be added to T15a.

Other A.C. Timings

Unless otherwise specified, all A.C. timings referenced to CLK2 refer to the CLK2 rising edge in phase one. All references to PCLK# refer to the falling edge of PCLK#.

Symbol	Parameter	Figure	16 MHz		20 MHz		Units	Notes
			Min	Max	Min	Max		
T40a	W/R#, M/IO#, D/C#, ADS# Setup to CLK2	4.3	29		22		ns	
T40b	BLE#, BHE# Setup to CLK2	4.3	29		12		ns	
T41	RESETSX Valid before CLK2	4.2	13		12		ns	C_{Load} = 30 pF
T42	RESETSX Hold after CLK2	4.2	2		2		ns	C_{Load} = 30 pF
T43	READYSX# Valid before CLK2	4.11	31		24		ns	C_{Load} = 30 pF
T44	READYSX# Hold after CLK2	4.11	6		4		ns	C_{Load} = 30 pF
T45	NA# Valid before CLK2 ↑ Phase 2	4.3	11		7		ns	C_{Load} = 30 pF
T46	NA# Hold after CLK2 ↑ Phase 2	4.3	27		17		ns	C_{Load} = 30 pF
T47	HRQSX Valid before CLK2	4.12	28		17		ns	C_{Load} = 30 pF
T48	HRQSX Hold after CLK2	4.12	5		5		ns	C_{Load} = 30 pF
T49	RESETNPX Valid before CLK2	4.2a	13		12		ns	C_{Load} = 30 pF
T50	RESETNPX Hold after CLK2	4.2a	4		3		ns	C_{Load} = 30 pF
T57	READY286# Setup to PCLK#	4.11	50	90	45	90	ns	(Note 2)
T57a	READY286# Setup to PCLK#				35	75	ns	(Notes 2, 3)
T58	READY286# Hold from PCLK#	4.11	35		10		ns	
T59	SYSRESET, RESETCPU Setup to CLK2	4.2	20		18		ns	(Note 1)
T60	SYSRESET, RESETCPU Hold from CLK2	4.2	15		12		ns	(Note 1)
T61	A20GATE Setup to CLK2	4.12	40		30		ns	(Note 1)
T62	A20GATE Hold from CLK2	4.12	15		10		ns	(Note 1)
T63	HRQ286 Setup to PCLK#		25		25		ns	(Note 1)
T64	HRQ286 Hold from PCLK#		5		5		ns	(Note 1)
T65a	TURBO# Setup to HLDASX		15		15		ns	(Note 1)
T65b	TURBO# Hold from HLDASX		15		15		ns	(Note 1)
T66	MEMR#, MEMW#, Setup to CLK2	4.12	17		17		ns	(Note 1)
T66a	REFRESH# Setup to PCLK#		25		25		ns	(Note 1)
T67	MEMR#, MEMW#, Hold from CLK2	4.12	17		17		ns	(Note 1)
T67a	REFRESH# Hold from PCLK#		0		0		ns	(Note 1)
T68	EXTRDY Setup to CLK2	4.11	5		5		ns	
T69	EXTRDY Hold from CLK2	4.11	32		30		ns	

Other A.C. Timings

Unless otherwise specified, all A.C. timings referenced to CLK2 refer to the CLK2 rising edge in phase one. All references to PCLK# refer to the falling edge of PCLK#. (Continued)

Symbol	Parameter	Figure	16 MHz		20 MHz		ns	Notes
			Min	Max	Min	Max		
T70	M/IO286#, S0# and S1# Output Delay from PCLK#	4.8	0	15	0	15	ns	$C_{Load\ max}$ = 100 pF
T71	ROMCS#, LMEGCS# Output Delay from Valid Address	4.3		40		40	ns	C_{Load} = 50 pF
T72	OBMEM Output Delay from Valid Address	4.3		40		36	ns	$C_{Load\ max}$ = 50 pF
T73a	D0–D15 Output Valid before CLK2		30		30			
T73b	D0–D15 Hold after CLK2		6		6			
T74	PERROR# Output Delay from PCLK#	4.4		50		50	ns	C_{Load} = 50 pF
T75	HLDA Active Delay from HRQ286	4.10	92		92			
T76	HLDA Inactive Delay from HRQ286	4.10	185		185			

NOTES:
1. Asynchronous parameters, provided to assure recognition at a specific clock edge.
2. READY286# is only sampled when an internal 8 MHz clock is high. (See Figure 4.11)
3. This specification is valid only when EFI = 40 MHz and PCLK = 20 MHz before programming the SPDSEL bit.

4.4 A.C. Timing Diagrams

The following diagrams illustrate A.C. timing relations.

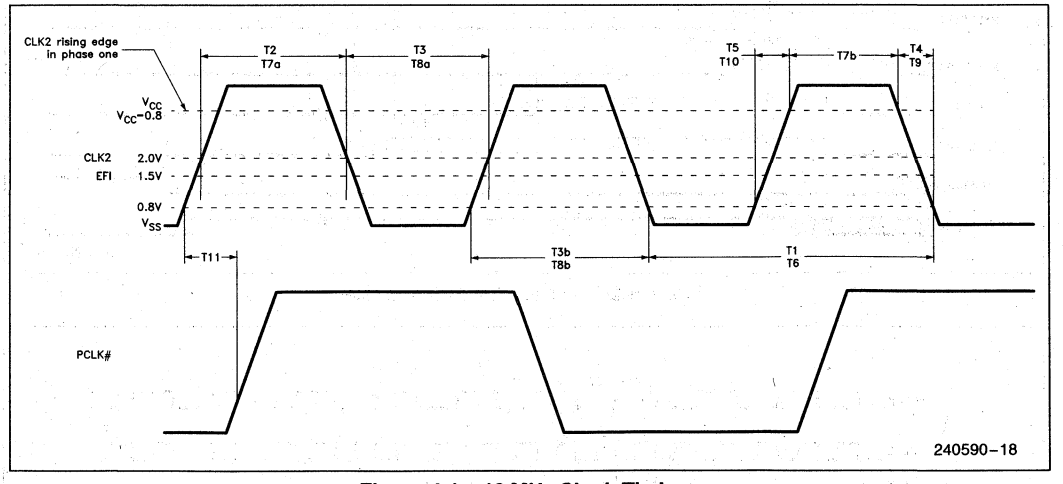


Figure 4.1a. 16 MHz Clock Timings

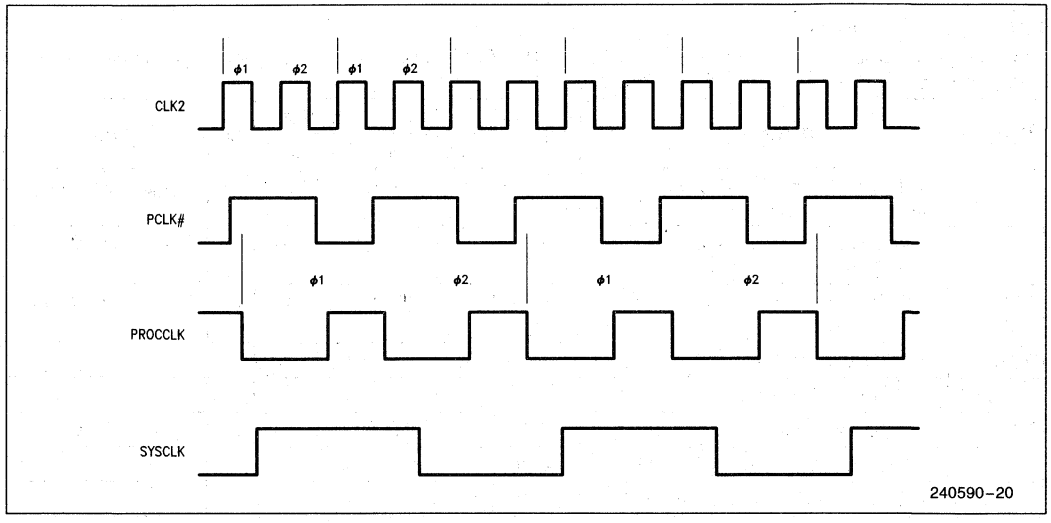


Figure 4.1b. 20 MHz Clock Timings

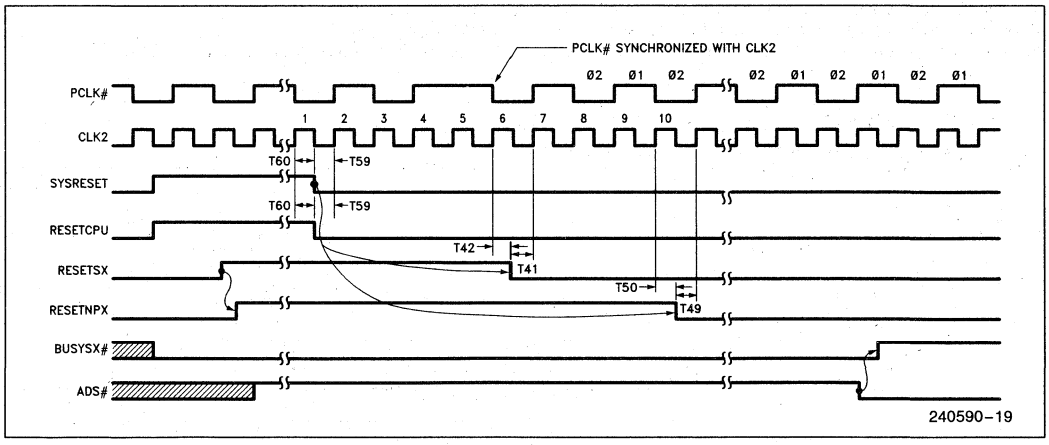
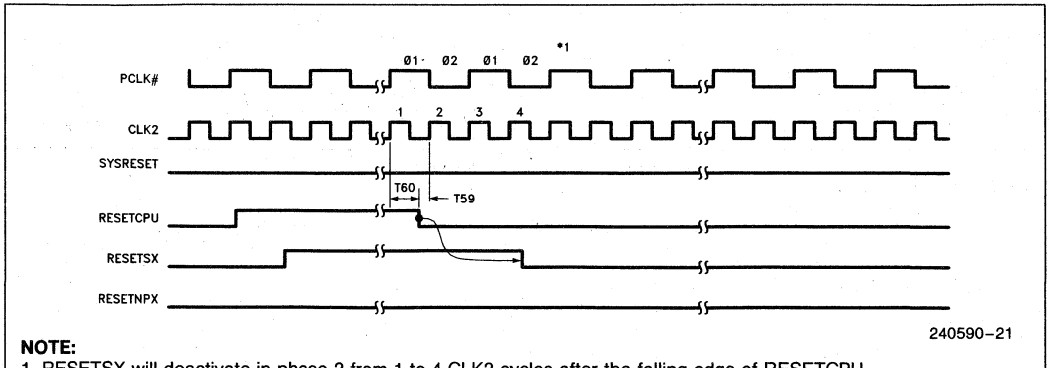


Figure 4.2a. System Reset Sequence



NOTE:
1. RESETSX will deactivate in phase 2 from 1 to 4 CLK2 cycles after the falling edge of RESETCPU.

Figure 4.2b. 16 MHz CPU Reset Sequence

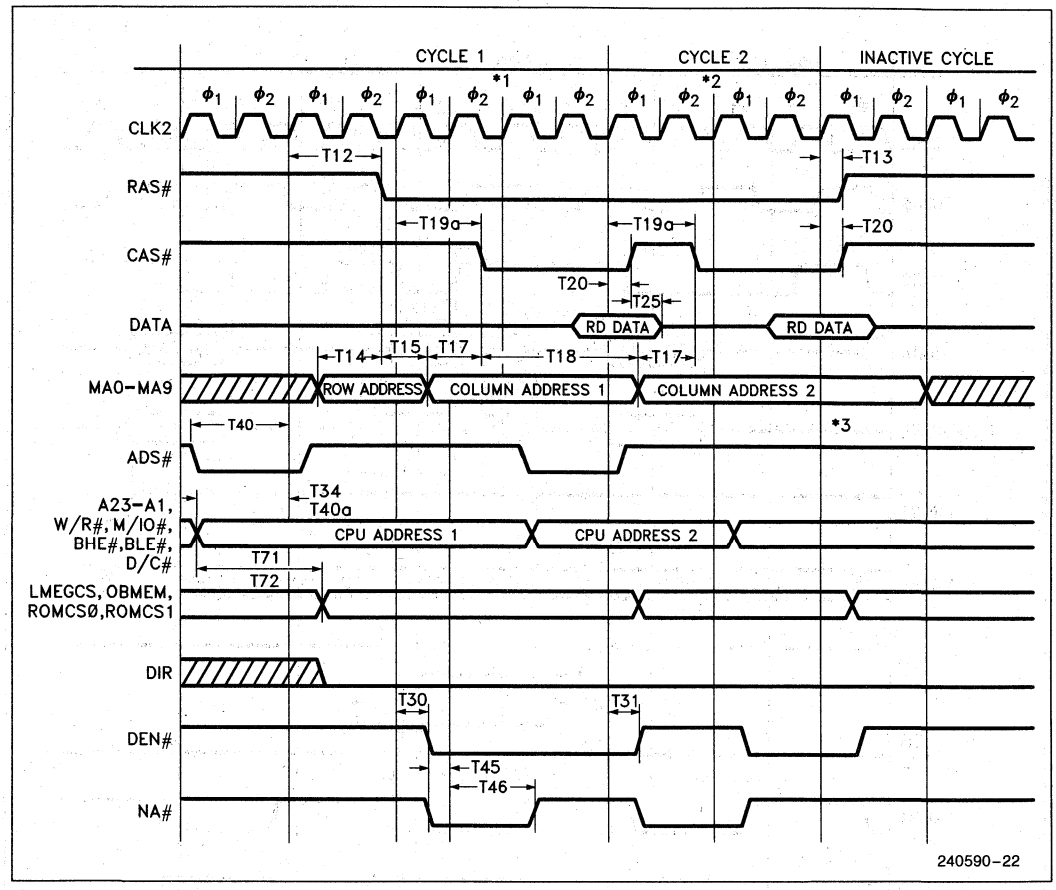


Figure 4.3. DRAM Cycles—Inactive . . . Read . . . Page Read . . . Inactive

NOTES:
1. Add 1 Wait State for W01 Mode.
2. Add 1 Wait State for W01 Mode, 2 Wait States for W02 Mode.
3. ADS# inactive at start of cycle deactivates RAS#.

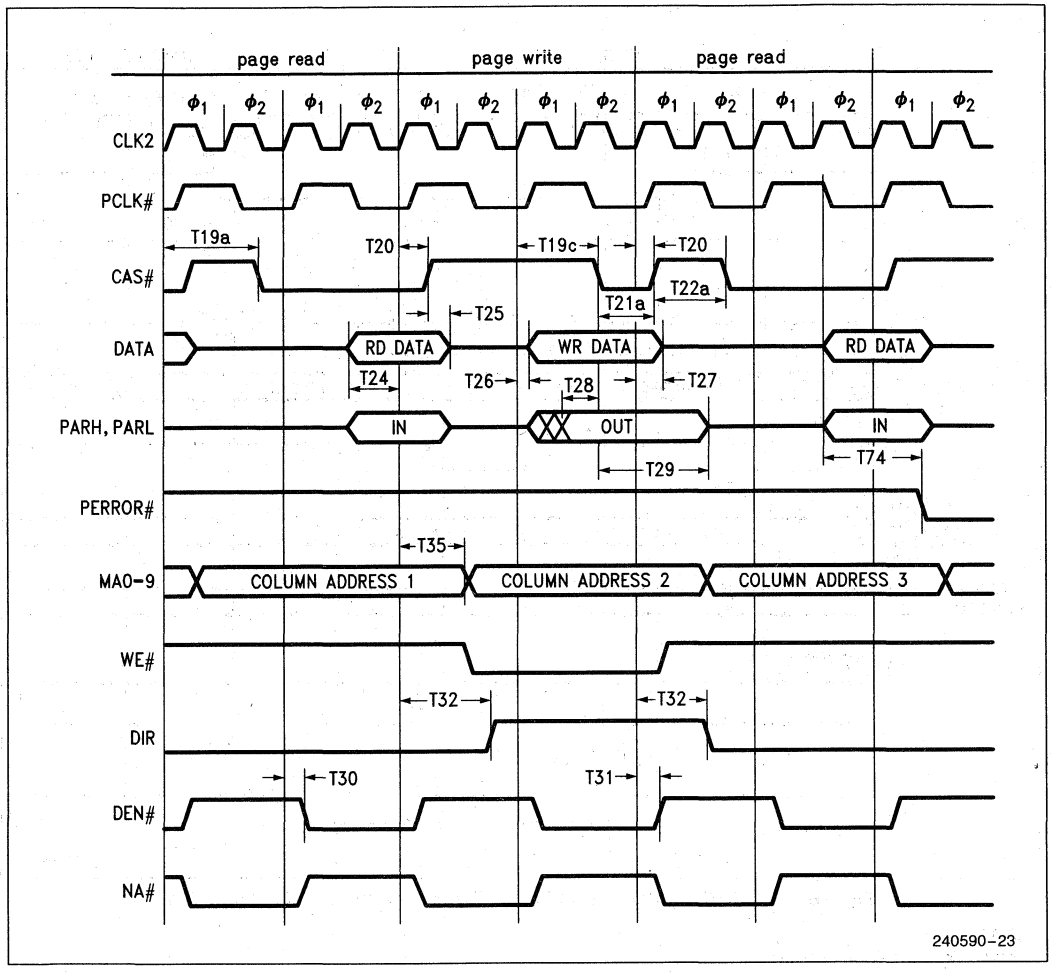


Figure 4.4. DRAM Cycles—Page Read . . . Page Write . . . Page Read F1/F4 Mode

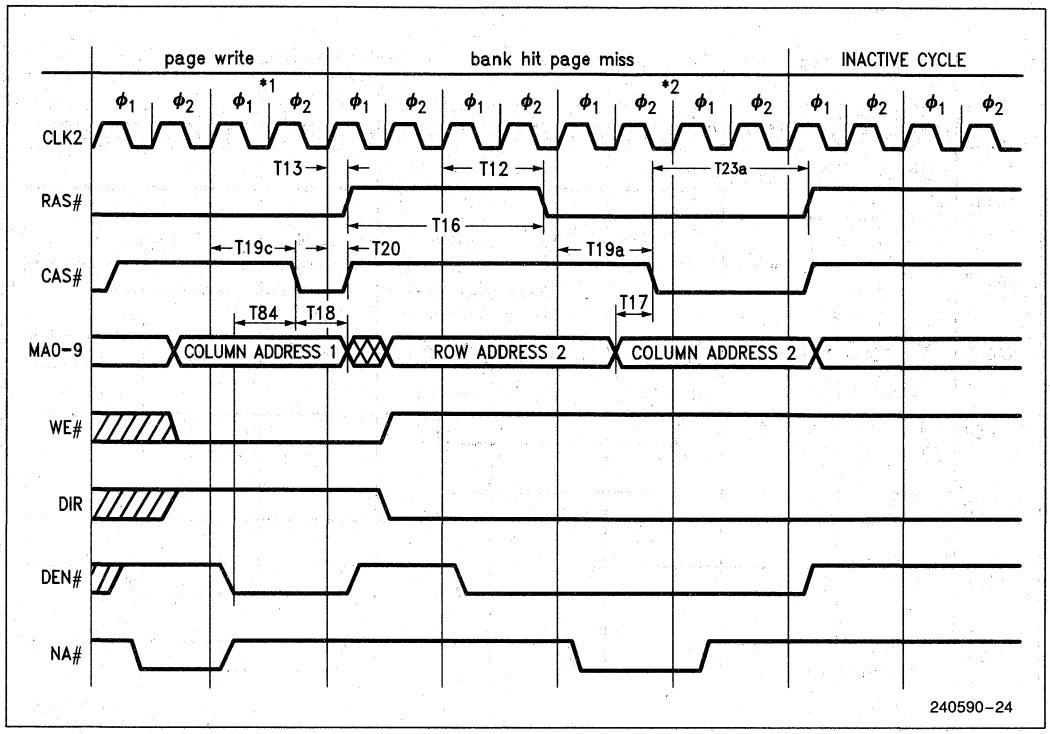


Figure 4.5. DRAM Cycles—Page Write . . . Bank-Hit-Page-Miss Read . . . Inactive

NOTES:
1. Add 1 Wait State for W01 Mode or 2 Wait States for W02 Mode.
2. Add 1 Wait State for W02 Mode.

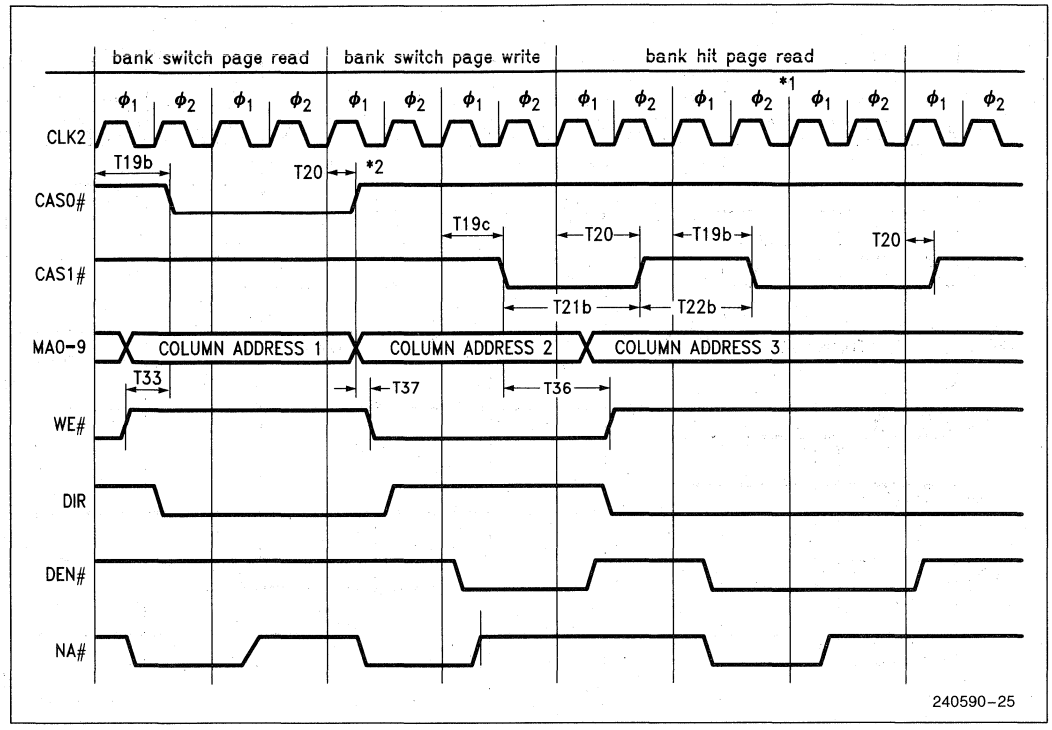


Figure 4.6. DRAM Cycles—Bank Switch Page Read . . . Bank Switch Page Write . . . Bank Hit Page Read for W01/W02 Mode

NOTES:
1. Add 1 Wait State for W02 Mode.
2. Following a read cycle, CAS# always de-activates before WE# activates. Following a write cycle, WE# may deactivate before CAS# de-activates.

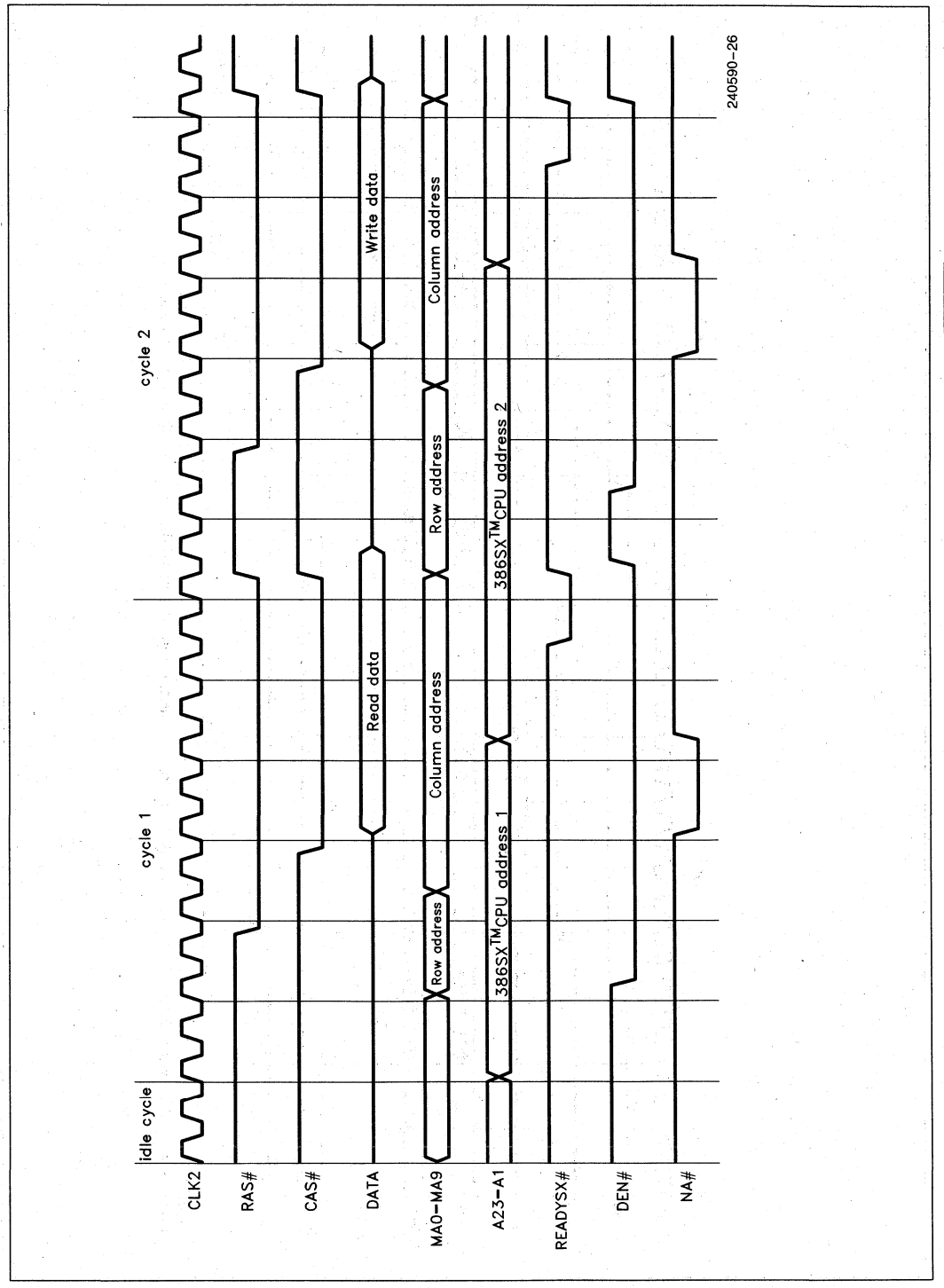
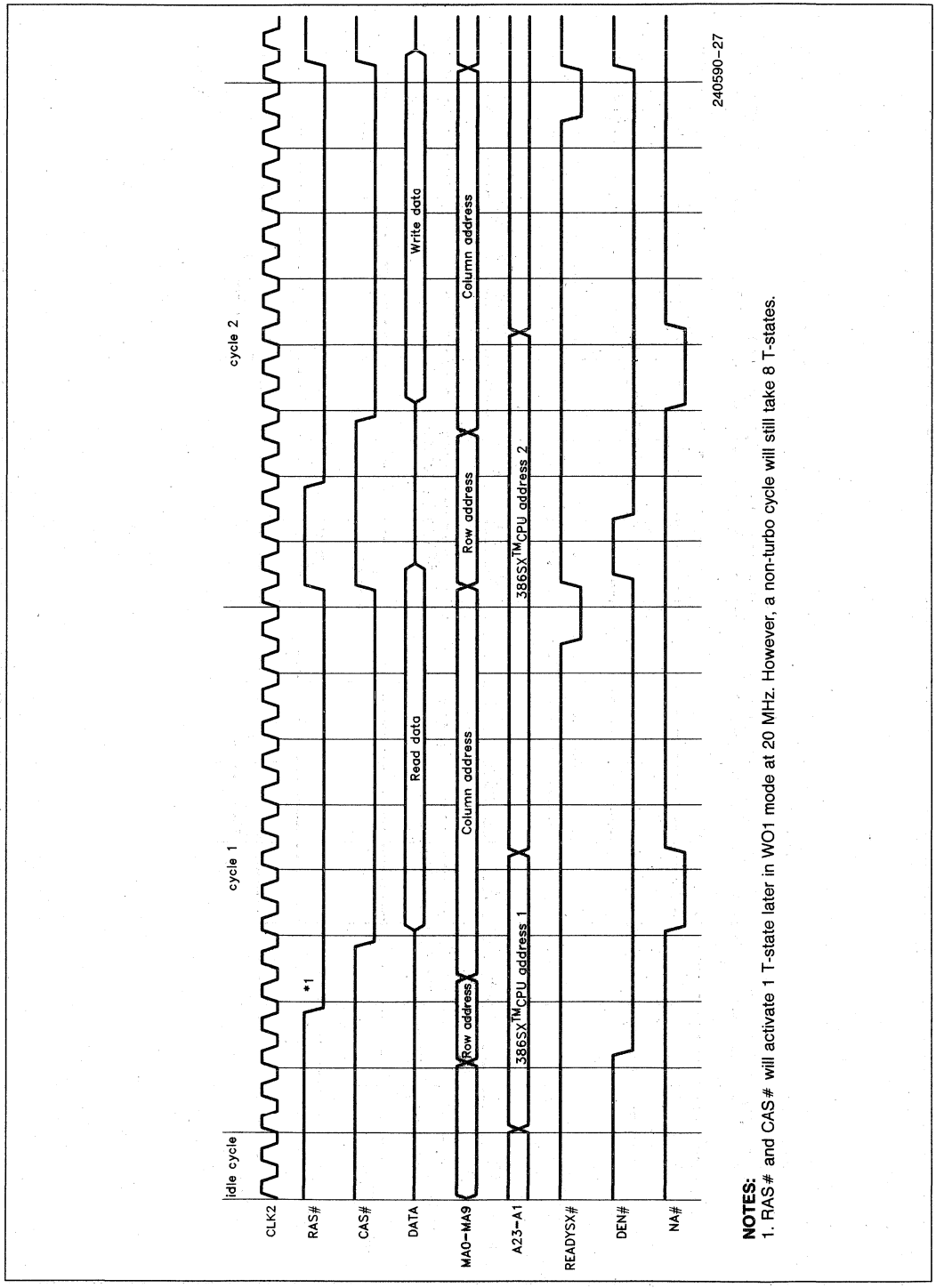


Figure 4.7a. 16 MHz DRAM Cycles—Non-Turbo Mode Read, Non-Turbo Mode Write



NOTES:
1. RAS# and CAS# will activate 1 T-state later in WO1 mode at 20 MHz. However, a non-turbo cycle will still take 8 T-states.

Figure 4.7b. 20 MHz DRAM Cycles—Non-Turbo Mode Read, Non-Turbo Mode Write

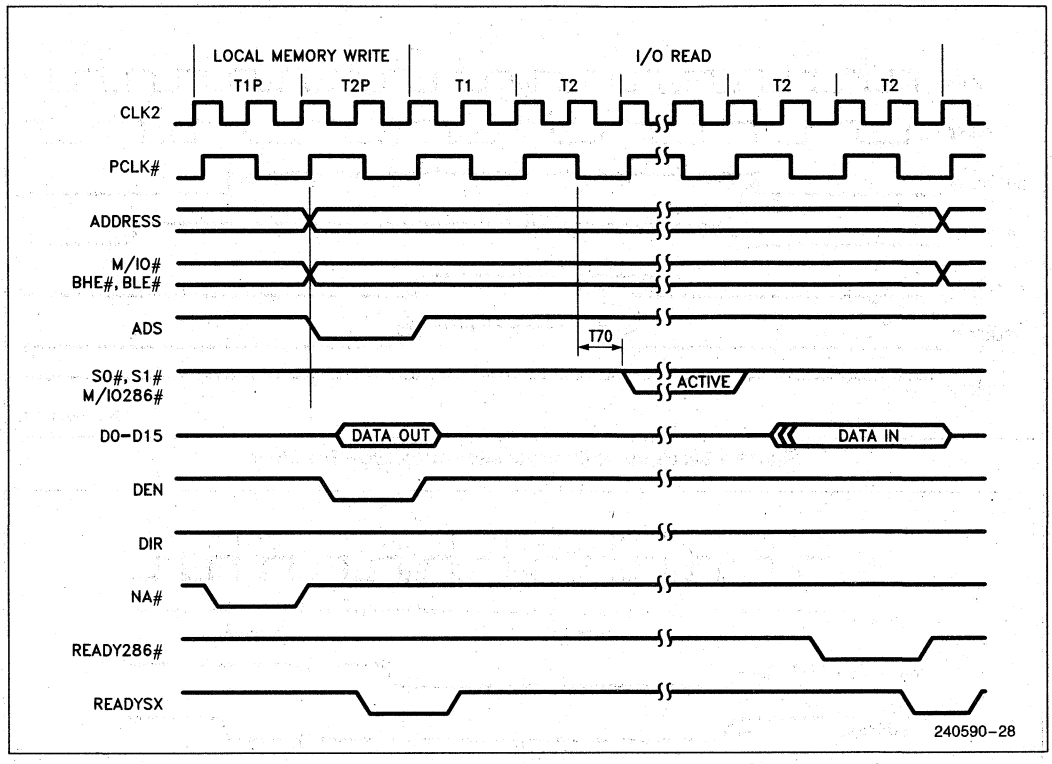


Figure 4.8. 16 MHz Local Memory Write, System Read (I/O or Memory)

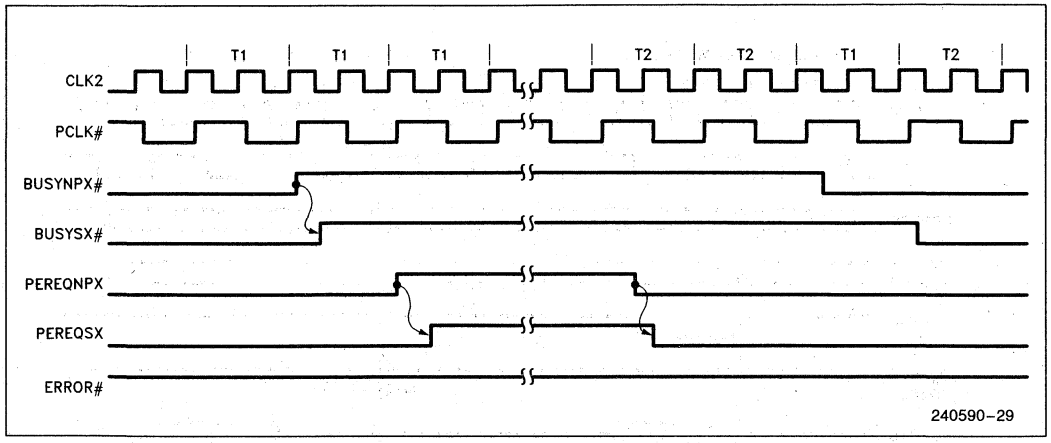


Figure 4.9a. Numeric Coprocessor Interface—No Error

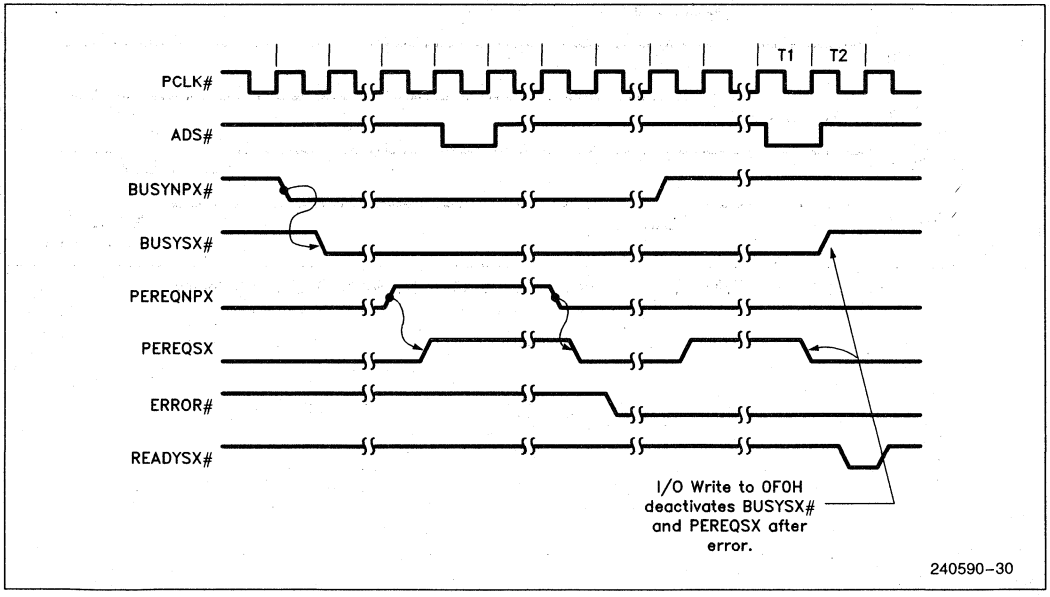


Figure 4.9b. Numeric Coprocessor Interface—with Error

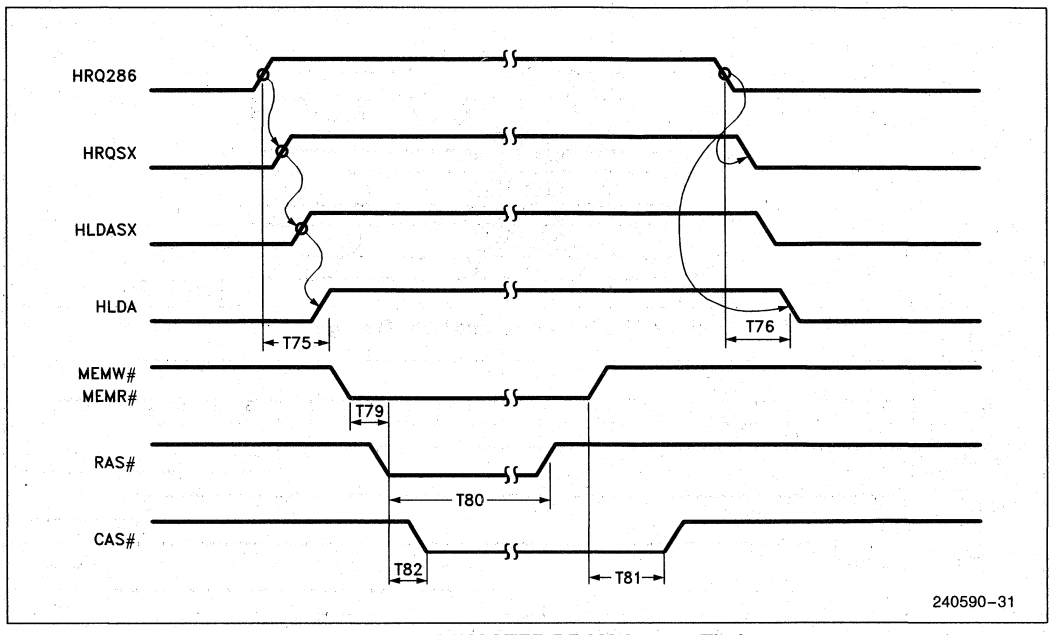
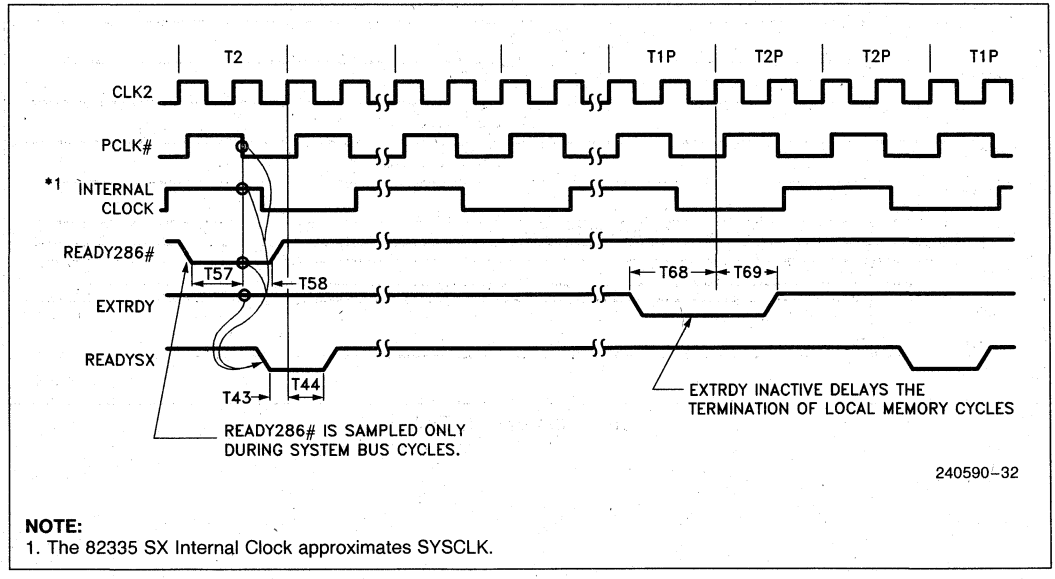


Figure 4.10. DMA/MASTER DRAM Access Timings



NOTE:
1. The 82335 SX Internal Clock approximates SYSCLK.

Figure 4.11. Ready Setup and Hold Timing

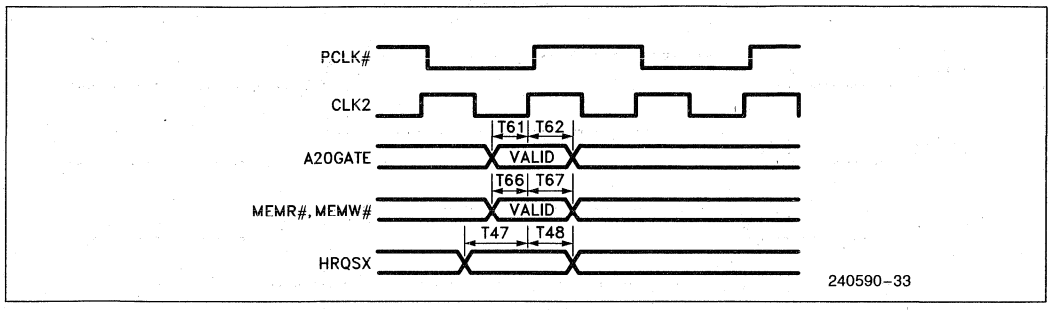


Figure 4.12. Misc. Setup and Hold Timings

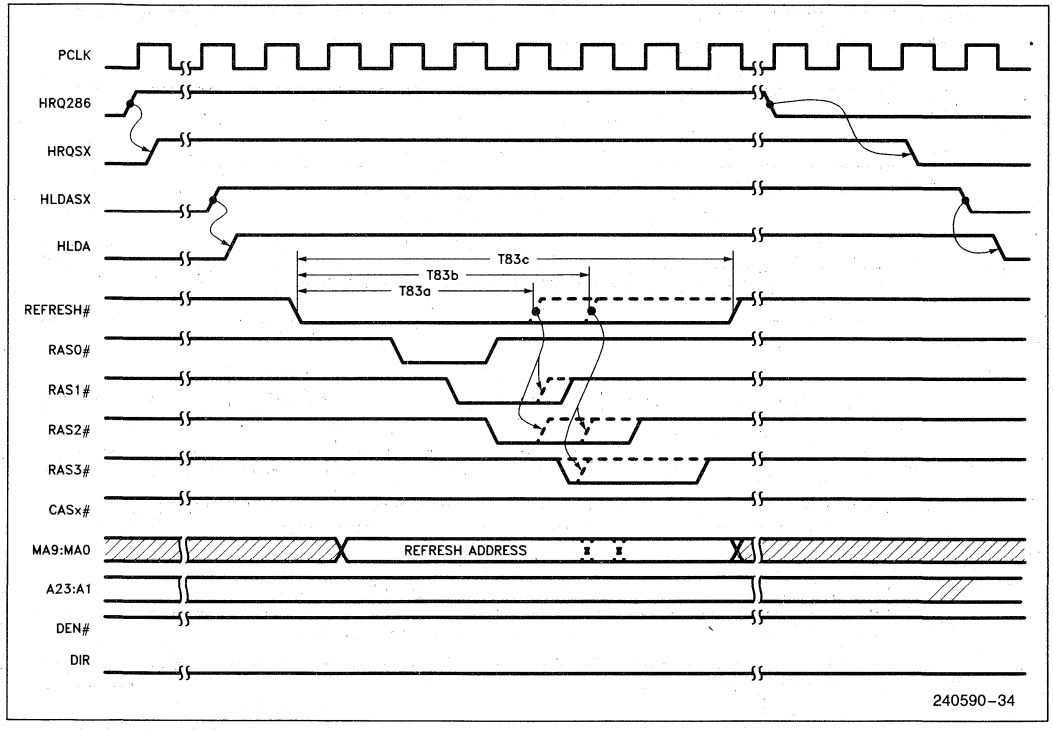


Figure 4.13. Refresh Timing

4.5 Capacitive Derating Curves

The capacitive derating curves for the various 82335 SX output buffers are shown in Figures 4.15 to 4.20. Each curve shows the output valid delay relative to a 50 pF nominal load. For A.C. Specifications that show a maximum capacitive load greater than 50 pF, the maximum value has been tested at 50 pF and derated to support the indicated capacitive load. The minimum values are all specified at 50 pF unless a capacitive load less than 50 pF is indicated. In which case the minimum value has been tested at 50 pF and derated to support the indicated capacitive load.

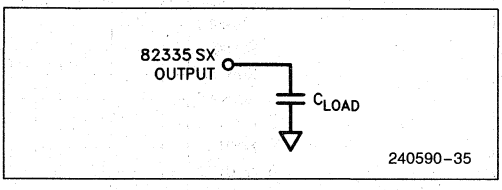


Figure 4.14. A.C. Test Loads

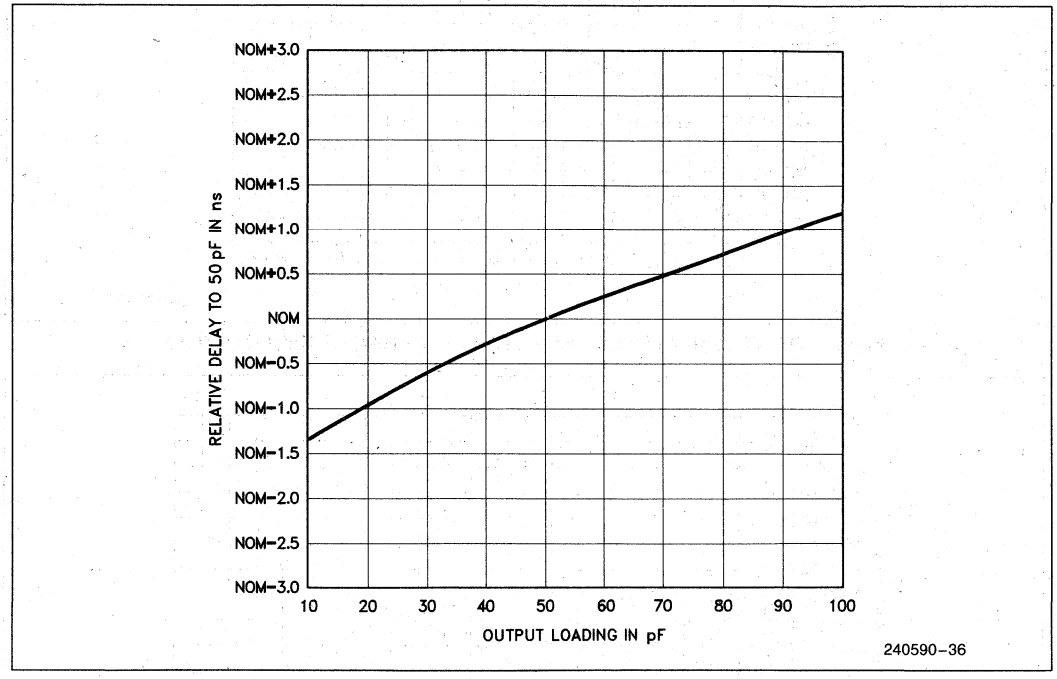


Figure 4.15. Capacitive Derating for the CASH#, CASL#, SO# and S1# Output Buffers

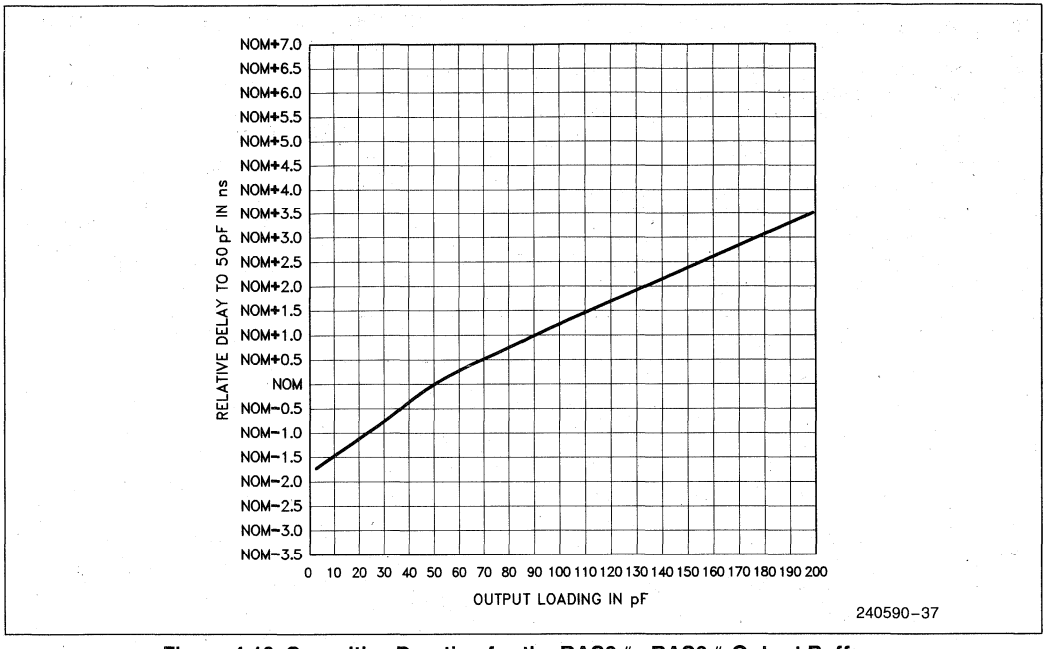


Figure 4.16. Capacitive Derating for the RAS0# – RAS3# Output Buffers

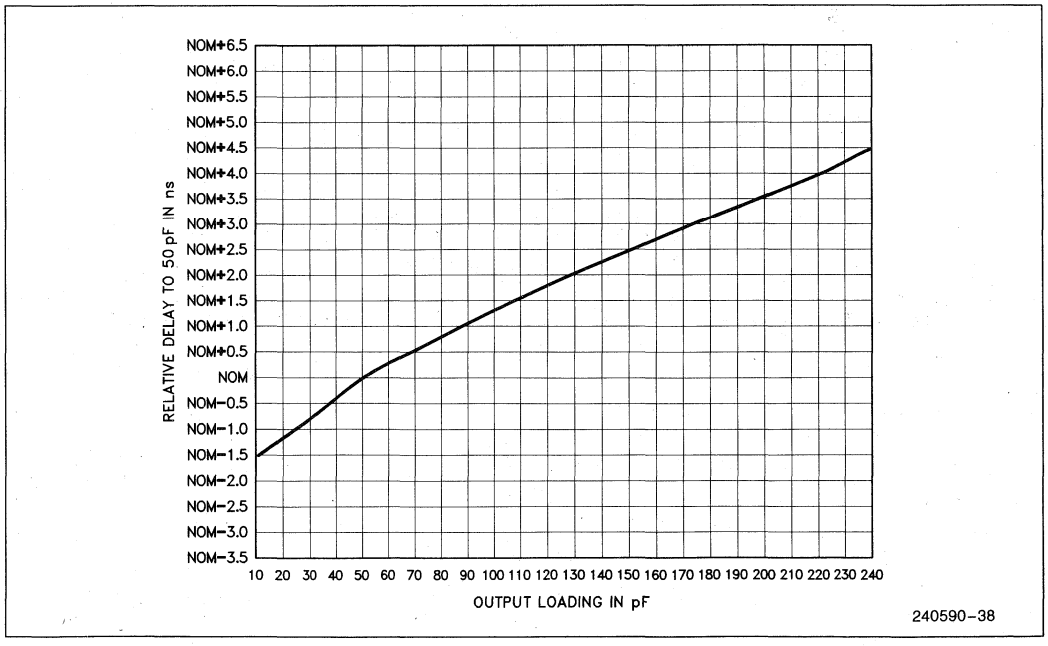


Figure 4.17. Capacitive Derating for the DEN#, DIR, MA0–MA9, WE#, OBMEM, ROMCS0#, ROMCS1#, MIO286#, HRQSX and LMEGCS# Output Buffers

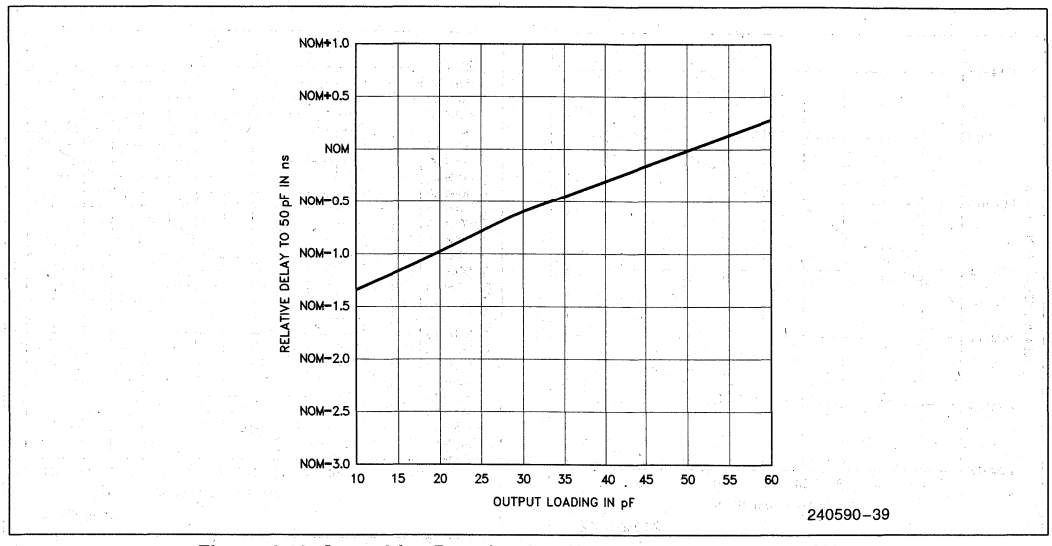


Figure 4.18. Capacitive Derating for the PARH, PARL, READYSX#, RESETSX, RESETNPX and NA# Output Buffers

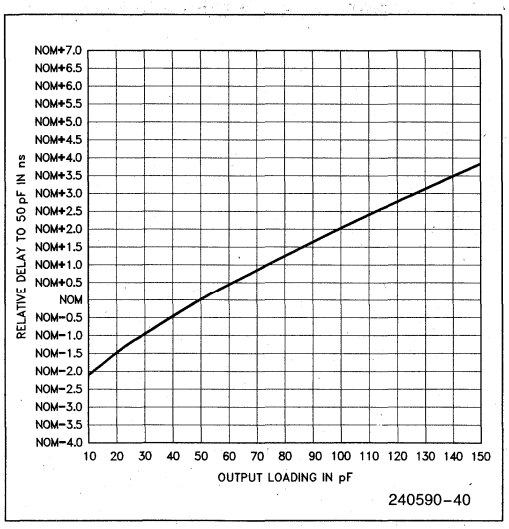


Figure 4.19. Capacitive Derating for the D0–D15 Output Buffers

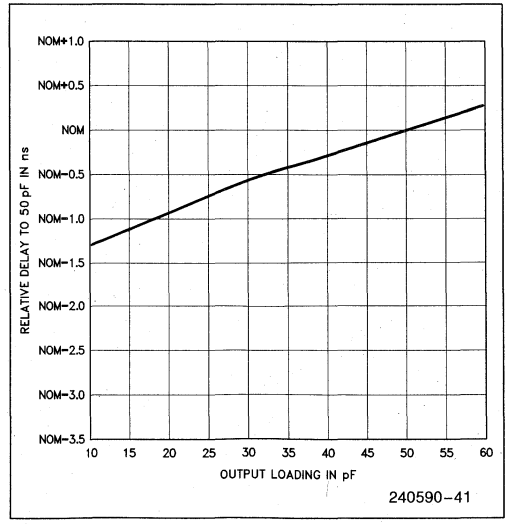


Figure 4.20. Capacitive Derating for the PCLK# Output Buffer

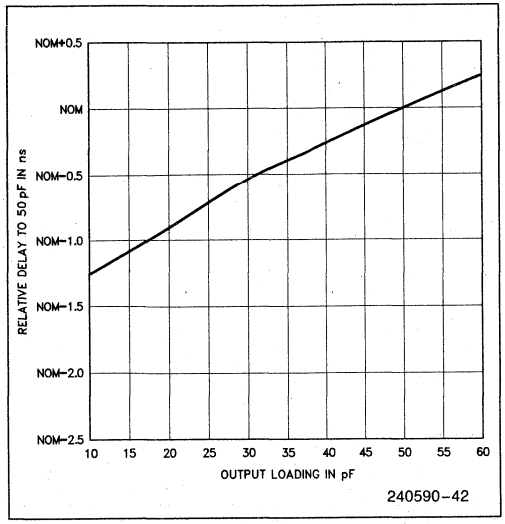


Figure 4.21. Capacitive Derating for the CLK2 Output Buffer

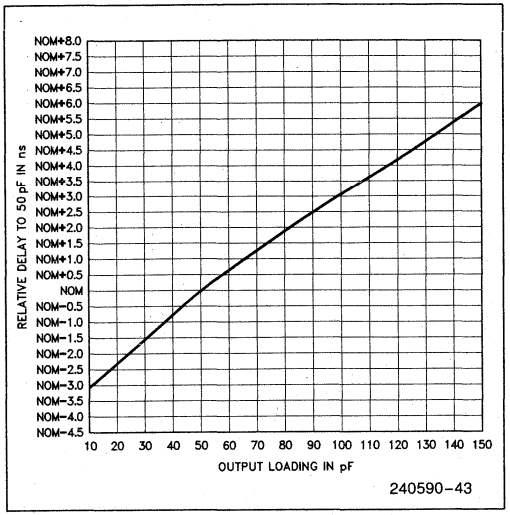


Figure 4.22. Capacitive Derating for the BUSYSX#, PEREQSX, PERROR#, HLDA Output Buffers

4.6 DIFFERENCES BETWEEN THE 82335 SX AND THE 82335

1. The 82335 SX can run at 16 MHz or 20 MHz. The 82335 can only run at 16 MHz.
   A. The 82335 SX has 16 MHz and 20 MHz A.C. Specifications. The 82335 has only 16 MHz A.C. Specifications.
   B. The 82335 SX DRAM modes are different at 20 MHz than 82335 DRAM modes. The 16 MHz 82335 SX DRAM modes are the same as the 82335.
   C. 82335 SX Non-turbo mode accesses take 8 T-states in 20 MHz operation. Non-turbo mode accesses take 6 T-states in both the 82335 and the 82335 SX in 16 MHz operation.
   D. PCLK# is a divide by 2.5 of EFI at 20 MHz, and a divide by 2 of EFI at 16 MHz in the 82335 SX. PCLK# is always a divide by 2 in the 82335 which only runs at 16 MHz.
2. The 82335 SX ignores READYNPX# (N.C.) and generates READYSX# after 1 wait-state for numerics coprocessor cycles if a coprocessor is present in the system. The 82335 uses READYNPX# to generate READYSX# on coprocessor cycles when a coprocessor is present.
3. The 82335 SX has separate output voltage specifications for PCLK#. The 82335 does not have separate output voltage specifications for PCLK#.
4. The 82335 SX has the Granularity Enable register. The 82335 does not.
5. The 82335 SX has the Extended Granularity register. The 82335 does not.
6. The 82335 SX can shadow and write-protect with 32 Kbyte or 128 Kbyte granularity. The 82335 can only shadow and write-protect in 128 Kbyte granularity.
7. The 82335 SX allows shadowing and write-protecting to be chosen independently of each other for the BIOS and adaptor ROM areas. The 82335 always write-protects shadowed areas.
8. 3-way interleaving is not possible with the 82335 SX. If the INTERLV bits in the 82335 SX are set for 3-banks, the DRAM controller will only access 2-banks. 3-way interleaving is possible with the 82335. If the INTERLV bits are set for 3-way interleaving in the 82335, the DRAM controller will access 3-banks but is not guaranteed to meet A.C. specifications. Correct DRAM operation is not guaranteed for 3-way interleaving in the 82335. Systems should not set the INTERLV bits in the 82335 for 3-banks.
9. PERROR# is latched until PARCHEN is cleared in the 82335 SX. PERROR# is not latched in the 82335.
10. TURBO is an asynchronous input in the 82335 SX. TURBO is not an asynchronous input in the 82335.
11. Only ROMCS1# is active for 512 Kbit ROM size operation in the 82335 SX. Only ROMCS0# is active for 512 Kbit ROM size operation in the 82335.
12. PCLK# is free-running during powerup in the 82335 SX. PCLK# is not active until SYSRESET is deactivated in the 82335.
13. RESET CPU will not be passed on to the 386 SX until HLDA is deactivated in the 82335 SX. RESETCPU will be passed on to the 386 SX during HLDA in the 82335. A SYSRESET combined with RESETCPU will always be passed through to the 386 SX by both the 82335 SX and the 82335.
14. The 82335 SX will only sample 386 SX bus cycle definition inputs when accompanied by ADS# active. The 82335 may sample 386 SX bus cycle definition inputs without ADS# active.
15. Some 16 MHz A.C. specifications are different in the 82335 SX than in the 82335, but the 82335 SX is upward compatible with the 82335.

The 82335 SX is plug compatible with the 82335

5.0 REVISION HISTORY

The following list represents key differences between this and the -001 version of the 82335SX data sheet. Please review this summary.

Front Sheet	CHMOS III has been changed to CHMOS IV.
Figure 2.1	This figure has been modified to include 82335SX device instead of 82335.
Table 2.3B	t_{AA} has been modified to be 75 ns maximum.
Section 2.3.4.5	A recommendation has been added to this section when using extended granularity feature. Please follow this recommendation.
Section 4.2	The V_{IHC} for the EFI has been changed to 2.4V minimum.
A.C. Timing	The following A.C. timings have been changed: T14a, T15, T16, T17, T17a, T18, T22b, T34, T42, T59 and T84. The following A.C. timings have been added: T15a, T18a. Notes 10, 11 and 12 have been added to A.C. timing for DRAM controller unit.
Figure 4.1a	The reference for EFI has been changed to 1.5V. T2 and T3 have replaced T2a for EFI high and EFI low values. T2b has been taken out.
Figure 4.3	T45 and T46 (NA# active and deactive edges) reference has been changed to the rising edge of CLK2 in phase two.
Figure 4.4	T21 label has been changed to T21a. T23a changed to T22a, and the extra T74 label has been removed.
Figure 4.6	T19a label has been changed to T19b and T23b has been changed to T22b.

82230/82231
HIGH INTEGRATION AT*-COMPATIBLE CHIP SET

- **Fully IBM PC-AT* System Compatible**
- **Two Chip Set Replaces the Major Logic Functions of the IBM PC-AT Motherboard Including the Functions of all the Microprocessor Peripherals:**
  - **— 8259A Programmable Interrupt Controller (Master)**
  - **— 8259A Programmable Interrupt Controller (Slave)**
  - **— 8254 Programmable Interval Timer**
  - **— 8284A Clock Generator**
  - **— 82284 Clock Generator & Ready Interface**
  - **— 82288 Bus Controller**
  - **— 8237 DMA Controller (2)**
  - **— 6818 Real Time Clock**
  - **— 74LS612 Memory Mapper**
- **Includes:**
  - **— Refresh Generation Logic**
  - **— Refresh/DMA Arbitration**
  - **— Address/Data Bus Control**
  - **— 16- to 8-Bit Conversion Logic**
- **Memory Refresh Controller Drives Up to 4 Mb DRAMs**
- **Numeric Processor Control Logic**
- **Up to 12 MHz System Clock Utilizing RAMs with Zero Wait States**
- **Single +5V Power Supply**
- **CHMOS Technology**
- **84 Pin PLCC Packages**
  (See Packaging Specification Order #231369-004)

1

The 82230 and 82231 are a two-chip implementation of LSI/MSI/SSI logic controlling the IBM Personal Computer AT. The devices provide a low power, highly integrated PC-AT design solution that may be applied to any 80286-based system.

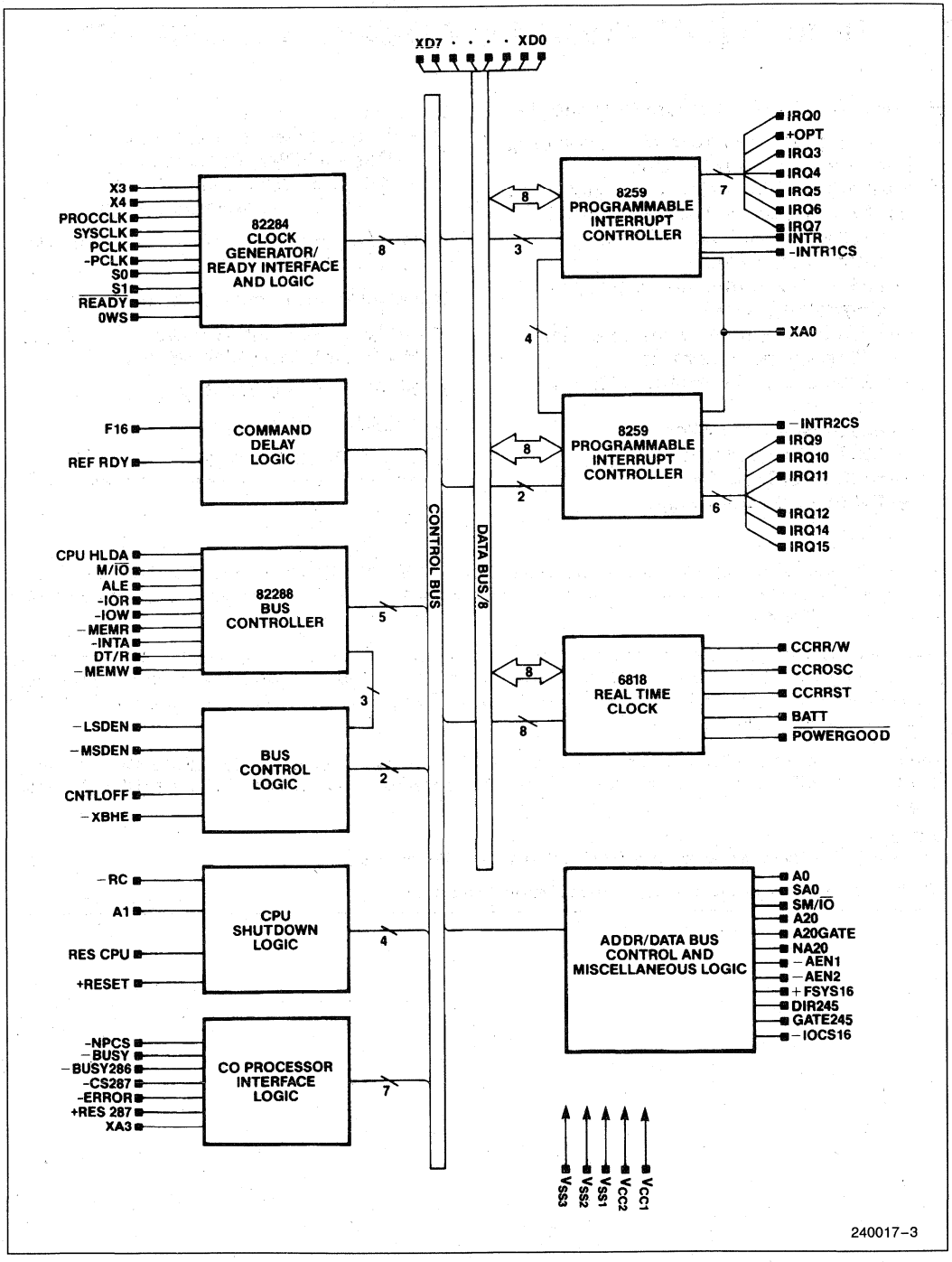
The 82230 performs the functions of the 82284 Clock Generator & Ready Interface, 82288 Bus Controller for 80286 processors, 6818 Real Time Clock/RAM, and the Master-Slave implementation of the dual 8259A Programmable Interrupt Controllers as well as Command Delay, Shut Down, Address/Data Bus Control and Ready Generation logic.

The 82231 includes the 8254 Programmable Interval Timer, 8284A Clock Generator, LS612 Memory Mapper and the dual 8237 DMA Controller functions as well as Refresh Generation and Refresh/DMA Arbitration Logic.

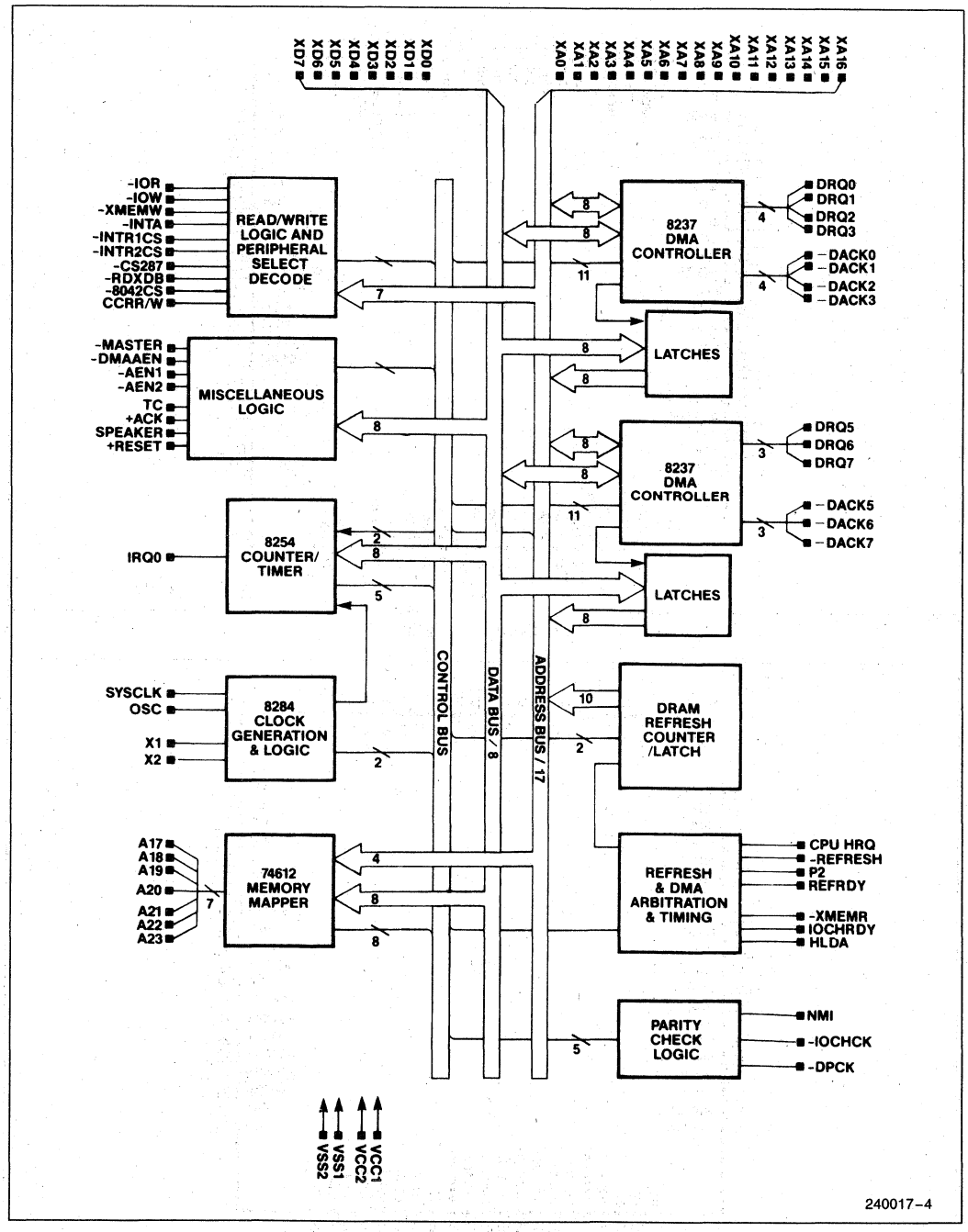
NOTE:
"+" and "−" in front of signal names is consistent with PC-AT Documentation.

*PC-AT is a Trademark of International Business Machine Corporation.

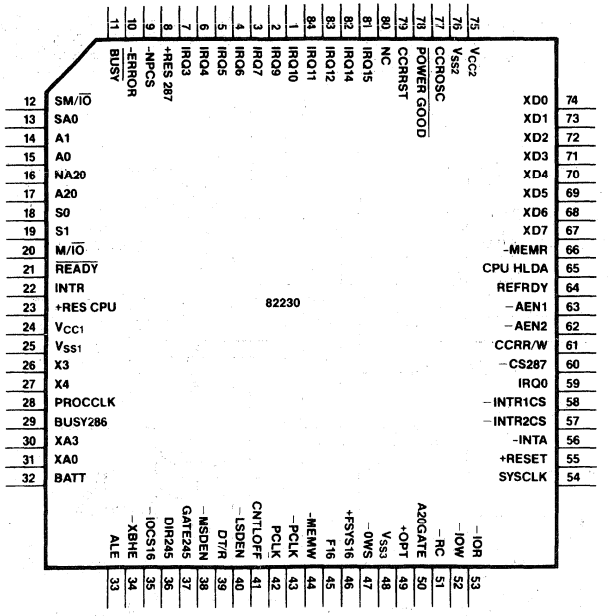
November 1989
Order Number: 240017-004



82230—Block Diagram

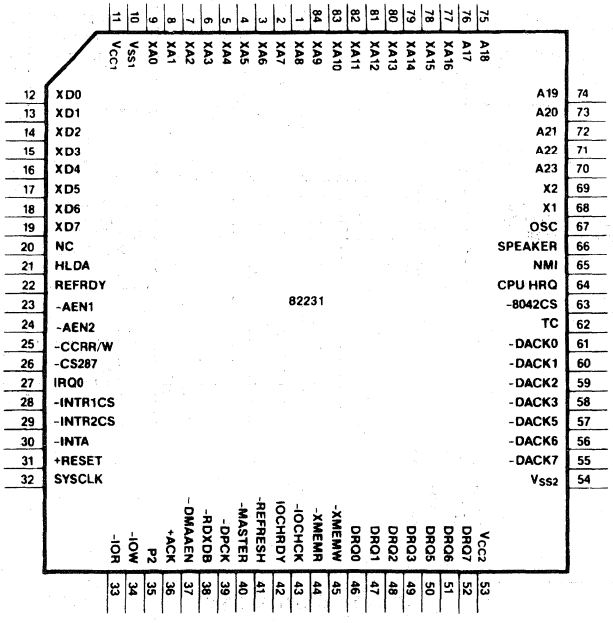


82231—Block Diagram



240017-1

Pin Diagram 82230



240017-2

Pin Diagram 82231

82230 PIN DESCRIPTION

Symbol	Pin No.	Type	Description
A0	15	I	ADDRESS 0 input from the CPU. It is used to generate SA0.
A1	14	I	ADDRESS 1 input from the CPU. It is used in conjunction with M/$\overline{IO}$, S0 and S1 to detect a CPU Shutdown condition.
A20	17	O	ADDRESS 20 is the A20 (NA20) line from the CPU after conditioning by the A20GATE signal. During a CPU Hold A20 goes to a high impedance state.
A20GATE	50	I	A20GATE from the Keyboard Controller is used to force A20 low. When A20GATE is low, A20 on the CPU Address Bus is forced low. When A20GATE is high, A20 follows the CPU Address 20. Tie directly to the P21 Pin of the Keyboard Controller.
−AEN2 −AEN1	62 63	I	ADDRESS ENABLE 1 & 2 from DMA's 1 & 2, respectively. The signal is the result of the DMA's AEN pin NAND'd with −MASTER. Tie directly from the −AEN1 and −AEN2 pins of 82231.
ALE	33	O	ADDRESS LATCH ENABLE is an active high signal that controls the address latches used to hold addresses during bus cycles. ALE is held inactive for Halt bus cycles.
BATT	32	I	BATTERY Power to the Clock Calendar and RAM.
−BUSY286	29	O	−BUSY286 is an active low output indicating the operating condition of the 80287 coprocessor to the processor. It is normally tied to the processor −BUSY pin.
−BUSY	11	I	−BUSY is an active low input from the 80287 to indicate that it is currently executing a command. It is used to generate the −BUSY286 output signal.
CCROSC	77	I	CLOCK CALENDAR OSCILLATOR; 32.768 KHz signal.
CCRRST	79	I	CLOCK CALENDAR RESET signal for the Real Time Clock. This is an active low input.
CCRR/W	61	I	CLOCK CALENDAR READ/WRITE signal for the Real Time Clock. A high enables READ/WRITE operation to the real-time clock. Tie directly from the CCRR/W Pin of 82231.
CNTLOFF	41	O	CONTROL OFF is used to enable the low byte data bus latch during byte accesses. This signal is active high.
CPU HLDA	65	I	CPU HOLD ACKNOWLEDGE is an active high input from the processor. An active condition indicates that the CPU has relinquished the bus to another bus master in the system.
−CS287	60	I	CHIP SELECT 287 is used to derive the −NPCS signal. Tie directly from the −CS287 pin of 82231.
DIR245	36	O	DIRECTION-245 controls the high to low byte and low to high byte conversion during data transfers to and from 8-bit peripherals.

82230 PIN DESCRIPTION (Continued)

Symbol	Pin No.	Type	Description
DT/$\overline{R}$	39	O	DATA TRANSMIT/RECEIVE establishes the data direction to and from the local data bus. When high, this output signals a CPU write bus cycle. A low indicates a CPU read bus cycle is being performed. This signal is always high when no bus cycle is active.
−ERROR	10	I	ERROR is a negative edge triggered input from the numeric processor indicating that an unmasked error condition exists. Tie directly from the −ERROR Pin of the 80287.
F16	45	I	F16 is an active high input indicating a word memory access. It is used to inhibit command delays for memory accesses.
+FSYS16	46	I	A latched version of F16.
−GATE245	37	O	GATE245 is an active low output. When active it enables the bus transceiver that performs the high to low byte conversion with the DIR245 signal. Conversion does not take place if A0 = 0 which indicates a word transfer.
−INTA	56	O	INTERRUPT ACKNOWLEDGE instructs an interrupting device that its interrupt request is being acknowledged. This signal is active low. −INTA is tri-stated when CPU HLDA is high and CNTLOFF is low. Tie directly to the −INTA pin of 82231.
INTR	22	O	INTERRUPT REQUEST is connected directly to the CPU's interrupt pin. INTR is active high, and is generated when a valid interrupt request has been asserted.
−INTR1CS	58	I	INTERRUPT CONTROLLER 1 (MASTER) CHIP SELECT is an active low input that is used to select the Interrupt Controller as an I/O device. This allows communication between the master interrupt controller and the CPU via the 'X' Data Bus. Tie directly from the −INTR1CS pin of 82231.
−INTR2CS	57	I	INTERRUPT CONTROLLER 2 (SLAVE) CHIP SELECT is an active low input that is used to select the Interrupt Controller as an I/O device. This allows communication between the slave interrupt controller and the CPU via the 'X' data bus. Tie directly from the −INTR2CS Pin of 82231.
−IO CS 16	35	I	I/O 16-BIT CHIP SELECT signals the system that the current data transfer is a 16-bit, one wait-state, I/O cycle. It is derived from an address decode and is an active low signal.
−IOR	53	I/O	I/O READ signal instructs a selected I/O device to drive its data onto the data bus. The −IOR signal is active low. It is tri-stated when CPU HLDA is high and CNTLOFF if low.
−IOW	52	I/O	I/O WRITE signal instructs a selected I/O device to read the data on the data bus. The −IOW signal is active low. It is tri-stated when CPU HLDA is high and CNTLOFF is low.

82230 PIN DESCRIPTION (Continued)

Symbol	Pin No.	Type	Description
IRQ0	59	I	INTERRUPT REQUEST 0 (system timer) receives interrupt requests from channel 0 of the timer/counter. Tie directly from the IRQ0 pin of 82231.
IRQ7–IRQ3 IRQ10–IRQ9 IRQ12–IRQ11 IRQ15–IRQ14	3-7 1-2 83-84 81-82	I I I I	INTERRUPT REQUESTS 3–7, 9–12, and 14–15 are used to signal the CPU that an I/O device needs attention. The interrupt requests are prioritized with IRQ9–IRQ12 and IRQ14–IRQ15 having the highest priority (IRQ9 highest) and IRQ3–IRQ7 having the lowest priority (IRQ7 lowest). IRQn signals are active high. The requesting signal is held high until the CPU acknowledges the interrupt request.
–LSDEN	40	O	LEAST SIGNIFICANT DATA ENABLE is an active low output. When active, it enables the transceiver/receiver connected to the least significant byte of the local data bus.
–MEMR	66	I/O	MEMORY READ COMMAND instructs a memory device to drive data onto the data bus. This signal is active low. –MEMR is active on all memory read cycles. It is tri-stated when CPU HLDA is high and CNTLOFF Output is low.
–MEMW	44	I/O	MEMORY WRITE COMMAND instructs a memory device to read the data on the data bus. This signal is active low. –MEMW is active on all memory write cycles. It is tri-stated when CPU HLDA is high and CNTLOFF Output is low.
–MSDEN	38	O	MOST SIGNIFICANT DATA ENABLE is an active low output. When active, it enables the transceiver connected to the most significant byte of the local data bus.
M/$\overline{IO}$	20	I	MEMORY-$\overline{INPUT\ OUTPUT}$ is the M/$\overline{IO}$ signal from the CPU. When high, it indicates a memory access. When low, it indicates an I/O access. It is used to generate the memory and I/O signals for the system.
NA20	16	I	NA20 is the CPU address 20. 82230 conditions this signal with A20GATE to produce A20. NA20 is tied directly from the CPU A20 output.
NC	80		Do Not Connect.
–NPCS	9	O	NUMERICAL PROCESSOR CHIP SELECT is an active low output used to select the 80287 Numerical Processor. It is tied directly to the $\overline{NPS1}$ pin of the 80287.
+OPT	49	I	KEYBOARD OUTPUT BUFFER FULL is an active high signal from the Keyboard Controller P24 Pin. The signal is an interrupt request (IRQ1) signaling a full keyboard buffer.
–0WS	47	I	ZERO WAIT STATE option. When pulled active (low), the current processor cycle can be terminated.

82230 PIN DESCRIPTION (Continued)

Symbol	Pin No.	Type	Description
PCLK	42	O	PERIPHERAL CLOCK is half the frequency of PROCCLK. It is used to clock peripheral controllers, specifically XTAL1 of the Keyboard Controller.
−PCLK	43	O	PERIPHERAL CLOCK INVERTED is the inverse of PCLK. It has been made available specifically for XTAL2 of the Keyboard Controller.
PROCCLK	28	O	PROCESSOR CLOCK provides the clock signal for the CPU and 80287 Numerical Processor. It is equal to the frequency of the crystal across pins X3 and X4. Tie directly to the CLK Pins of the 80286 and 80287.
$\overline{\text{POWER GOOD}}$	78	I	$\overline{\text{POWER GOOD}}$ is an active low input that indicates that system power is sufficient to maintain the integrity of the system. If high, it will force a system reset.
RC	51	I	RESET CPU from the keyboard controller P21 Pin.
$\overline{\text{READY}}$	21	O	$\overline{\text{READY}}$ is an active low output which signals that the current bus cycle is to be completed. S0, S1, $\overline{\text{POWER GOOD}}$, and 0WS control the $\overline{\text{READY}}$.
REFRDY	64	I	REFRESH/IO-CHANNEL-READY is generated by 82231. It is used to preset the READY Interface Asynchronous READY ($\overline{\text{ARDY}}$).
+RES 287	8	O	RESET 80287 is the reset signal for the 80287 Numerical Processor.
RES CPU	23	O	RESET CPU is the reset signal for the CPU. Active high, RESCPU is generated when either $\overline{\text{POWERGOOD}}$ or RC become active, or when the CPU generates a shut down status by forcing M/$\overline{\text{IO}}$ high. S0, S1 and A1 low. If this signal is initiated by RC, or by M/$\overline{\text{IO}}$, S0, S1 and A1, it will remain active for 16 PROCCLK cycles.
+RESET	55	O	RESET (SYSTEM) is an active high output derived from the $\overline{\text{POWER GOOD}}$ input. +RESET is used to force the system into an initial state. When +RESET is active, $\overline{\text{READY}}$ will also be active (Low).
S0, S1	18, 19	I	STATUS inputs from the CPU. The status signals are used by the bus controller to determine the state of the CPU.
SA0	13	O	ADDRESS 0 of the CPU bus. SA0 outputs A0 from the CPU during local CPU cycles. During a CPU Hold SA0 goes to a high impedance state so that another master on the expansion bus can take control. During an interrupt acknowledge this signal will be forced low.
SM/$\overline{\text{IO}}$	12	I	SYSTEM MEMORY-INPUT OUTPUT is the M/$\overline{\text{IO}}$ signal from the CPU, conditioned by ALE.
SYSCLK	54	O	SYSTEM CLOCK is the result of PROCCLK divided by two, thus synchronized to the processor's T-states. It may be used to clock peripheral devices that must be synchronized to the CPU.

82230 PIN DESCRIPTION (Continued)

Symbol	Pin No.	Type	Description
V_{CC1} V_{CC2}	24 75		POWER: +5V supply.
V_{SS1} V_{SS2} V_{SS3}	25 76 48		GROUND.
X3 X4	26 27	I O	CRYSTAL inputs used to generate PROCCLK and SYSCLK. The crystal frequency must be twice the processor clock frequency. Alternatively, an oscillator may be connected to X3.
XA0	31	I	ADDRESS 0 is used by the 8259A to decipher command words the CPU issues. XA0 works in conjunction with the read, write and chip select signals to the interrupt controller in determining whether the CPU wishes to issue a command or read the status of the controller.
XA3	30	I	ADDRESS 3 is used for generating the chip select and reset signals for the 80287.
−XBHE	34	I/O	BUS HIGH ENABLE is an active low signal which is used by 82230 to generate the MSDEN signal.
XD7–XD0	67–74	I/O	Data Bus 0–7 for the peripheral bus. The direction of the bus is determined by the −RDXDB signal from 82231. It is used by the 8259A to decipher command words the CPU issues.

82231 PIN DESCRIPTION

Symbol	Pin No.	Type	Description
−8042CS	63	O	8042 CHIP SELECT is an active low, chip select signal for the Keyboard Controller.
A23–A17	70–76	O	A23–A17 are the Address bits 17–23 of the CPU Address bus. They are outputs directly from the Memory Mapper Pins MO1–MO7 and supply page information during DMA transfers. These outputs are tri-stated unless HLDA and −MASTER are high.
+ACK	36	O	ACKNOWLEDGE is an active low output. When active it enables the bus transceiver between the system and peripheral (XBUS) bus. +ACK is used in conjunction with −RDXDB which controls the direction of the bus transceiver.
−AEN1 −AEN2	23 24	O O	ADDRESS ENABLE FROM DMAs 1 & 2, respectively. The signal is the result of the DMA's AEN signal NAND'd with −MASTER. Tie directly to the −AEN1 and −AEN2 pins of 82230.
CCRR/W	25	O	CLOCK CALENDAR READ/WRITE signal for the real-time clock. A high enables READ/WRITE operations to the real-time clock. Tie directly to the CCRR/W pin of 82230.
CPU HRQ	64	O	CPU HOLD REQUEST is an active high output indicating a DMA request to the CPU. It is also active during refresh cycles. CPU HRQ is normally connected to the 80286 HOLD Pin.

82231 PIN DESCRIPTION (Continued)

Symbol	Pin No.	Type	Description
−CS287	26	O	CHIP SELECT 287 is used by 82230 to derive the −NPCS signal. Tie directly to the −CS287 pin of 82230.
−DACK0−3 −DACK5−7	61−58 57−55	O O	DMA ACKNOWLEDGE 0−3 and 5−7 are used to acknowledge DMA requests (DRQ0−3 & 5−7). The output signal is an active low.
−DMAAEN	37	O	DMA ADDRESS ENABLE is an active low signal and is active when an I/O device is making a DMA access to system memory or during refresh.
−DPCK	39	I	DATA PARITY CHECK is used to generate NMI. This input is active low.
DRQ0−3 DRQ5−7	46−49 50−52	I I	DMA REQUEST 0−3 & 5−7 are synchronous channel requests used by peripheral devices and I/O processors to gain DMA service. The requests are prioritized with DRQ0 having the highest and DRQ7 having the lowest priorities. A DRQ line must be held active (high) until the corresponding DACK line goes active.
HLDA	21	I	HOLD ACKNOWLEDGE is an active high input that is equivalent to CPU HLDA. An active condition inidicates that the CPU has relinquished the bus to another bus master in the system.
−INTA	30	I	INTERRUPT ACKNOWLEDGE instructs an interrupting device that its interrupt is being acknowledged, and the device may place its interrupt vector onto the data bus. This input signal is active low. −INTA is used by 82231 in the generation of −RDXDB. Tie directly from 82230 Pin 56.
−INTR1CS	28	O	INTERRUPT CONTROLLER 1 (MASTER) CHIP SELECT is an active low output that is used by 82230 to select the Interrupt Controller as an I/O device. This allows communication between the Master Interrupt Controller and the CPU via the 'X' Data Bus. Tie directly to the −INTR1CS pin of 82230.
−INTR2CS	29	O	INTERRUPT CONTROLLER 2 (SLAVE) CHIP SELECT is an active low output that is used by 82230 to select the Interrupt Controller as an I/O device. This allows communication between the Slave Interrupt Controller and the CPU via the 'X' Data Bus. Tie directly to the −INTR2CS Pin of 82230.
−IOCHCK	43	I	I/O CHANNEL CHECK is an active low input. It is used to indicate an uncorrectable system error. It provides the system with parity error information about memory or devices on the I/O channel.
IOCHRDY	42	I	I/O CHANNEL READY is generated by an I/O device. When low it indicates a 'not ready' condition and forces the insertion of wait states in I/O or Memory accesses by the I/O device. When active (high), it will allow the completion of a memory or an I/O access by the I/O device.

82231 PIN DESCRIPTION (Continued)

Symbol	Pin No.	Type	Description
−IOR	33	I/O	I/O READ signal instructs a selected I/O device to drive its data onto the data bus. The −IOR signal is active low. It is used for data transfers between the CPU and I/O devices and by DMA transfers.
−IOW	34	I/O	I/O WRITE signal instructs a selected I/O device to read the data on the data bus. The −IOW signal is active low. It is used for data transfers between the CPU and I/O devices and by DMA transfers.
IRQ0	27	O	INTERRUPT REQUEST 0 (System Timer) from Channel 0 of the Timer/Counter. Tie directly to the IRQ0 Pin of 82230.
−MASTER	40	I	−MASTER is an active low input used in conjunction with a DRQ line to gain control of the system. A DMA controller or processor on the I/O channel may issue a DRQ to a DMA channel and receive a −DACK. The I/O processor may then activate −MASTER which will allow it to control the system address, data, and control lines.
NC	20		Do Not Connect.
NMI	65	O	NON-MASKABLE INTERRUPT is an active high output that is connected to the CPU NMI pin.
OSC	67	O	OSCILLATOR output is the clock frequency of the crystal connected across X1−X2. It is the OSC output from the Clock Generator.
P2	35	O	P2 is an active high output indicating that a valid refresh address is available on the XA bus.
−RDXDB	38	O	READ X-DATA BUS controls the direction of the bidirectional buffer between the least significant byte of the 'S' Data Bus and the 'X' Data Bus. −RDXDB is used in conjunction with +ACK to control XBUS activity. When +ACK is active (low) and −RDXDB is low, data is to be read from the peripheral bus. When +ACK is active (low) and −RDXDB is high, data is to be written to the peripheral bus.
REFRDY	22	O	REFRESH/IO-CHANNEL-READY is generated by +REFRESH OR'd with IOCHRDY. It is used by 82230 to preset the Clock Generator & Ready Interface Asynchronous Ready ($\overline{ARDY}$).
−REFRESH	41	I/O	REFRESH is an active low output used to initiate a refresh cycle for the dynamic RAMs.
+RESET	31	I	RESET (SYSTEM) is an active high input from 82230. +RESET is used to force 82231, as well as the system, into an initial state. Tie directly from 82230 Pin 55.
SPEAKER	66	O	SPEAKER DATA is an output of the Programmable interval timer tone signal used to drive the speaker.
SYSCLK	32	I	SYSTEM CLOCK input from 82230. It is used to synchronize 82231 to the system. Tie directly from 82230 SYSCLK Pin.
TC	62	O	TERMINAL COUNT provides a pulse when the terminal count for any DMA channel is reached.

82231 PIN DESCRIPTION (Continued)

Symbol	Pin No.	Type	Description
V_{CC1} V_{CC2}	11 53		POWER: +5V supply.
V_{SS1} V_{SS2}	10 54		GROUND.
X1 X2	68 69	I O	CRYSTAL inputs for the internal oscillator used to generate clocking for I/O devices. A parallel resonant fundamental frequency mode crystal is required. An alternative oscillator may be connected to X1.
XA8–XA0 XA9 XA16–XA10	1–9 84 77–83	I/O I/O O	XBUS ADDRESSES 0–16 are the peripheral addresses for the local I/O bus.
XD0–XD7	12–19	I/O	Data Bus 0–7 for the peripheral bus. The direction of the bus is determined by the −RDXDB signal from 82231.
−XMEMR	44	I/O	MEMORY READ signal indicating a DMA read operation from peripheral devices or memory.
−XMEMW	45	O	MEMORY WRITE signal indicating a DMA write operation to peripheral devices or memory. It is tri-stated except during DMA transfers.

FUNCTIONAL DESCRIPTION

Introduction

The 82230 and 82231 are a two-chip implementation of LSI/MSI/SSI logic controlling the IBM Personal Computer AT. The devices provide a low power, highly integrated PC-AT design solution that may also be applied to any 80286-based system. With the 82230 and 82231, a PC-AT system can be designed to operate at 12 MHz with zero wait state RAM accesses.

These standard cell products contain most of the logic peripheral to the microprocessors and memory on the "standard" AT motherboard. The LSI peripherals which support the AT design reside as supercells on the 82230/82231 chips. These peripherals are compatible with the products they replace and the user should refer to the standard product data sheets for additional information on the operation of these devices.

The PC-AT schematics in the IBM PC-AT Technical Reference Manual are also a good source of information about the 82230/82231's internal logic.

The 82230 performs the functions of the 82284 Clock Generator & Ready Interface, 82288 Bus Controller, 6818 Real Time Clock/RAM, and the Master-Slave implementation of the dual 8259A Programmable Interrupt Controllers as well as Command Delay, Shut Down, Address/Data Bus Control and Ready Generation logic.

The 82231 includes the 8254 Programmable Interrupt Timer, 8284A Clock Generator, LS612 Memory Mapper and the dual 8237 DMA Controller functions as well as Refresh Generation and Refresh/DMA Arbitration Logic.

1

PC-AT BLOCK DIAGRAM

The block diagram is shown below in Figure 1 which shows the 82230 and 82231 being used in a PC-AT compatible design. Note how the basic structure of the PC-AT is retained in the design. The five address busses and four data busses are present. The 82230 and 82231 'sit' on the X Address and Data Busses and monitor the status and control line outputs from the 80286 in order to operate the peripheral supercells.

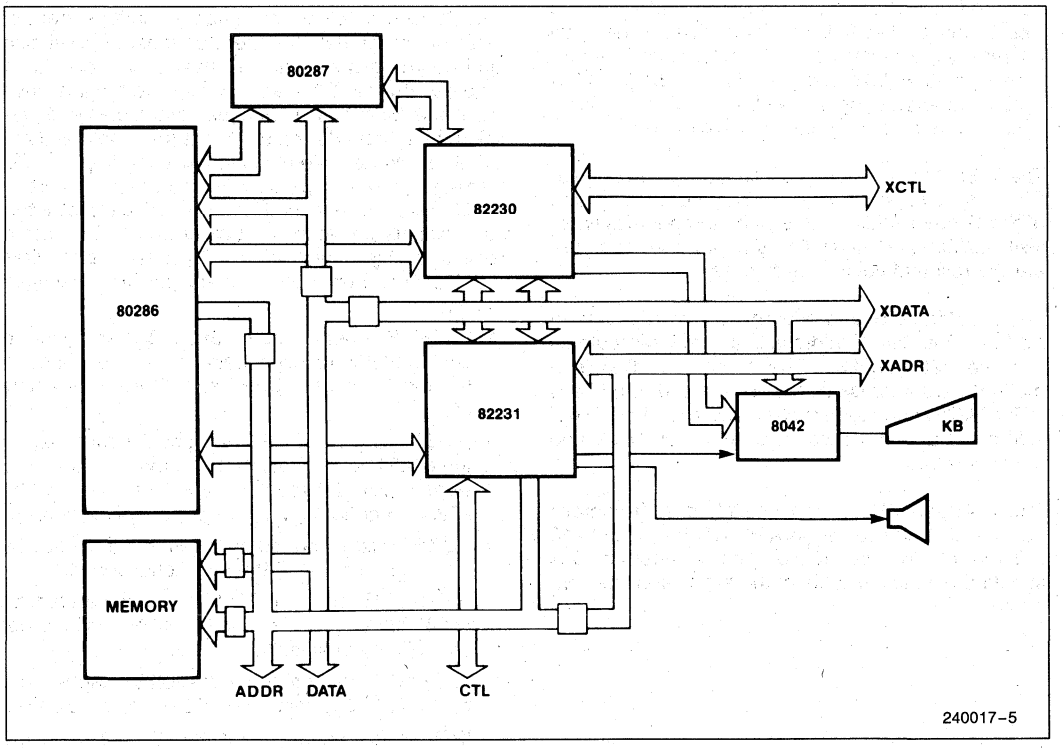


Figure 1. Block Diagram of 82230 and 82231 for PC/AT

A brief description of each of the basic PC-AT busses is included in order to show how a system design using the 82230/82231 allows for the retention of the PC-AT bus structure.

ADDRESS BUSSES

The 82230/82231 allows a straightforward PC-AT design which preserves the five basic address busses. System designers can interact with these busses, create new busses, or eliminate busses, depending upon design objectives. Each bus must be independently buffered and separated by buffers/latches. Most handshake signals are generated by the 82230/82231.

The Local Address Bus is comprised of the 24 address pins emanating from the 80286. External buffers are required in order to separate the local address bus from the system address bus. A0, however, is input directly into the 82230 and is used in conjunction with XBHE in order to enable the appropriate memory bank. The 82230 inputs CPU address bit NA20 and outputs A20.

The System Address Bus is the main AT address bus. This is a latched version of the local address bus, and is a 20-bit bus. SA0 is an output of the 82230 and SA1–SA19 are latched from the local address bus. The latch signal is ALE, which is generated by the 82288 supercell in the 82230. CPU HLDA, generated by the 80286 and active high, should be used as the output enable signal.

The Memory Address Bus applies to the RAM on the PC-AT system board only. It is a multiplexed version of the System Address Bus with nine address lines, MA0–MA8. External multiplexers are used to generate the row and column addresses.

The X Address Bus is separated from the System Address Bus. The X address lines are input/outputs into the 82230/82231. The X Address Bus is a motherboard address bus which is used to address ROM (BIOS) and motherboard I/O. In addition, the X Address Bus generates addresses for DMA and memory refresh.

The L Address Bus is an unlatched 7-bit address bus. LA17–LA23 can allow a PC-AT design up to 16 MBytes of address space. The L Address Bus should be made available at the expansion bus connector.

DATA BUSSES

The Local Data Bus is the name for the data bus lines emanating directly from the 80286. The local data bus has 16 lines, D0–D15. Because the 80286 can do word and byte transfers and because word transfers need not be aligned, it is necessary to design a bus interface which differentiates between the high bus byte and the low bus byte.

The System Data Bus is the main data bus of a PC-AT system and interfaces with all other data busses. The 82230 and 82231 are designed to control these interfaces in order to simplify system design and maximize bus flexibility.

The Memory Data Bus interfaces both DRAM and ROM. It is a 16-bit bus and connects with the System Data Bus through buffers.

The X Data Bus is the bus intended primarily for system board I/O functions. It interfaces to functions such as the DMA controllers, Interrupt controllers, Keyboard controller, and Real Time Clock.

82230/82231 Interface

The 82230 and 82231 are relatively independent of each other; the 82230 generates most of the timing and control signals and the 82231 controls the X Address Bus for DMA and refresh. Both chips have additional functions but because of the desire to partition the system design such that the 82230 and the 82231 could be assembled in low cost 84-pin PLCC packages, each chip relies on the other for certain functions. This entails introducing dedicated interface signals between the 82230 and 82231 into a system design. The 82230/82231 interface requires 14 pins on each device and these pins are described below.

- REFRDY is generated by the 82231 and used to tell the 82230 to insert wait-states in response to the 82231 input IOCHRDY. IOCHRDY is active high.
- −AEN1 and −AEN2 are signals generated by the 82231 which indicate DMA byte (−AEN1) or word (−AEN2) transfers, and are used by the 82230 to generate bus buffer control signals.
- CCRR/W is generated by the 82231 and used by the 82230 as part of the 6818 chip select.
- −INTR1CS and −INTR2CS are generated by the 82231 and used by the 82230 as the interrupt controller chip selects.
- −CS287 is generated by the 82231 and is used by the 82230 to generate 80287 control signals.
- IRQ0 is the output of the 8254 timer 0 on the 82231 and is connected to interrupt request 0 on the master interrupt controller in the 82230.

- +RESET is generated by the 82230 in response to POWERGOOD and is used by the 82231 for initialization.
- SYSCLK is PROCCLK divided by two.
- −INTA, −MEMR, −IOR, and −IOW are commands generated by the 82230 in response to the 80286 status inputs S0 and S1, and are used by the 82231 as basic system commands.

Coprocessor Interface

The 82230 contains a coprocessor interface logic block to allow interfacing with an 80287 math coprocessor. The coprocessor interface includes the IBM PC-AT compatible error handling hardware.

Memory Operations

When the 82230 and 82231 are used in a PC-AT system design, the system can be designed to operate with the full 16 MBytes of memory that the 24 address lines of the 80286 allow.

The 82230/82231 chipset normally operates with one wait state inserted for memory operations and four wait-states inserted for I/O operations. The number of wait states may be increased for slow memory or I/O devices, or decreased if use of high-speed memory or I/O devices is desired in order to provide higher system performance.

During normal operation, the number of ROM accesses are relatively few compared to RAM accesses, so ROM subsystem speed does not significantly affect system performance. When designing high performance PC-AT systems, it should be verified that the ROM subsystem is fast enough for operation. Note also that older versions of the IBM BIOS may not operate in systems faster than 8 MHz. Modifications of both RAM and ROM subsystem performance are covered in detail in the Intel 286EX Application Note.

System Clocks and Oscillators

Because of the high level of integration of the 82230/82231, several different clock frequencies are present on the chips. Figure 2 shows the relationships between the various clock signals.

PROCCLK is the main system clock which is used in a PC-AT system to generate all the basic system and bus timings. The clock is generated by the 82230 and operates at twice the system clock frequency. See Table 1 for processor clock capacitance. The tolerance of the PROCCLK oscillator is independent of the 82230/82231 and is primarily determined by the requirements of the 80286 and 80287 processors. For reliable operation at the specified V_{DD} and temperature condition, the processor timing specifications must not be exceeded.

Table 1. Recommended Fundamental Mode Crystal Characteristics and Recommended Load Capacitance for PROCCLK

Oscillator	Crystal Freq.	C_{LOAD} (pF)	C_{IN} (pF) X3	C_{OUT} (pF) X4
PROCCLK	12 MHz	20	30	10
PROCCLK	16 MHz	20	30	10
PROCCLK	20 MHz	20	27	8
PROCCLK	24 MHz	20	22	8

SYSCLK is PROCCLK divided by two, and should be used as the expansion bus clock. SYSCLK is an output of the 82230. To ensure compatibility with expansion cards, SYSCLK should not be above 8 MHz.

X1 is an input to the 82231 which is 12 times the frequency used to clock the three counters in the 8254 timer on the 82231. In order to be compatible with AT hardware and software, it should be a 14.318 MHz fundamental mode crystal with C_{LOAD} = 32 pF, and a 27 pF capacitor should be placed in series with the crystal. Alternately, a 14 MHz funda-

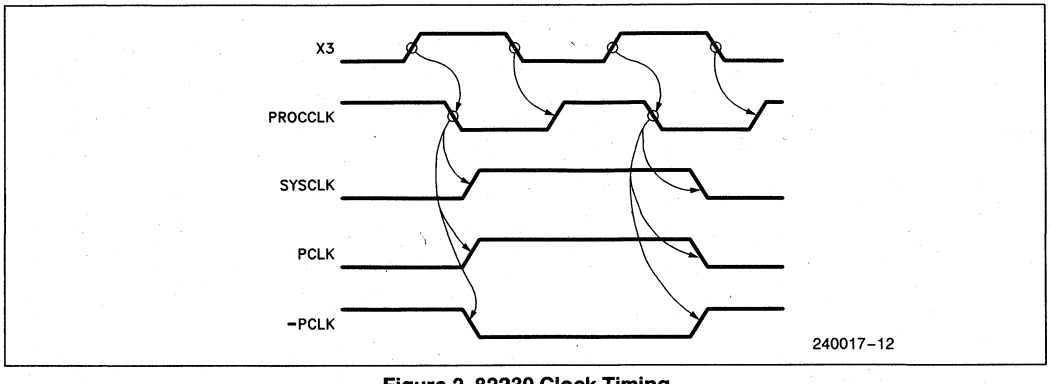


Figure 2. 82230 Clock Timing

mental mode crystal (C_{LOAD} = 32 pF) may be trimmed with a 5–50 pF trimmer capacitor in series for high accuracy applications such as NTSC or RS170 video-compatibility with chroma colorburst.

Note that the trim cap must be adjusted with a low-capacitance nylon or teflon tuning wand for accurate trimming. Tolerance for this clock is 0.1% or less for non-video or non time-critical use, 0.01% or greater if used for video color-burst or time-critical applications.

The CCR clock is the low-frequency oscillator used to clock the 6818 Clock/Calendar/RAM on the 82230. CCROSC tolerance is variable and dependent upon real-time clock accuracy requirements. See Table 2 for CCROSC clock tolerance and accuracy.

Table 2. CCROSC Tolerance/Accuracy

Tolerance	Accuracy
0.001% or 10 ppm	5 minutes/year trimming required
0.01% or 100 ppm	1 minute/week no trimming required
0.02% or 200 ppm	2 minutes/week no trimming required
0.05% or 500 ppm	5 minutes/week no trimming required

EXTERNAL OSCILLATORS

External CMOS output drive oscillators may be used for either PROCCLK or OSC. Simply connect the external oscillator outputs to PROCCLK inputs X1 or X3. TTL output oscillators may be used if the output drive V_{OH} is greater than 4.1V; pull-up resistors will generally suffice. The oscillator invertor outputs X2 and X4 may be left open, or may be used to drive one moderate rise-time CMOS load if needed.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . . 0°C to 70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on any Pin with Respect to Ground . . . . −0.5V to V_{CC} +0.5V
Power Dissipation 1W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

82230/82231 DC CHARACTERISTICS

V_{CC} = 5V ±5%, V_{BAT} = 2.8V to V_{CC}, T_A = 0°C to +70°C

Parameter	Conditions	Min	Max	Units
V_{IL}			0.5	V
V_{IH}		2.0		V
V_{IL} 82230 Pins 26, 77, 78, 79 82231 Pins 41, 42, 68			0.5	V
V_{IH} 82230 Pins 26, 77, 78, 79 82231 Pins 41, 42, 68		V_{CC} − 0.5		V
V_{IH} 82230 Pin 47		V_{CC} − 0.5(1)		V
I_{I0}	V_{IN} = 0	−100		µA
I_{I0} 82230 Pins 26, 77, 78, 79	V_{IN} = 0	−10		µA
I_{I1}	V_{IN} = V_{CC}		10	µA
I_{OH} Except for 82231 Pin 41(2)	V_{OH} = 2.4		−4	mA
I_{OL}	V_{OL} = 0.45V	4		mA
I_{OL} 82230 Pins 13, 28, 44, 52, 53 54, 56, 66 82231 Pins 33 34, 67	V_{OL} = 0.45V	16		mA
I_{OL} 82231 Pin 41	V_{OL} = 0.45V	18		mA
I_{OZ}	V_O = 0 to V_{CC}	−10	+10	µA
I_{CC} 82230	F = 10 MHz F = 12 MHz		55 60	mA mA
I_{CC} 82231	F = 10 MHz F = 12 MHz		45 50	mA mA
I_{CC} 82230 from Battery	F = 32.768 KHz V_{BAT} = 5V V_{CC} = 0V		25	µA
I_{CC} 82230 from Battery	F = 32.768 KHz V_{BAT} = 2.8V V_{CC} = 0V		20	µA

NOTES:
1. −0WS (82230 Pin 47) is driven by an open collector output. It is pulled up to CMOS voltage levels of V_{CC} − 0.5V by a pullup resistor.
2. −REFRESH (82231 Pin 41) is an open collector output.
3. CCROSC is the only signal that should switch in the battery back up mode. For back up operation with V_{BATT} < 4.75V, CCROSC, POWERGOOD and CCRRST input levels V_{IL}/V_{IH} should be 10% and 90% of V_{BATT}, respectively.

82230 AC CHARACTERISTICS (V_{DD} = 5V ±5%, T_A = 0°C to 70°C)

Symbol – Figure	Parameter	10 MHz		12 MHz		Units	Notes
		Min	Max	Min	Max		
53–15 54–15	–0WS Setup Time to PROCCLK ↓ –0WS Hold Time from PROCCLK ↓	25 0		25 0		ns ns	13 13
	–0WS Setup Time to PROCCLK ↓ –0WS Hold Time from PROCCLK ↓	36 0		36 0		ns ns	11 11
69–22	A0 Setup Time to ALE A0 Hold Time from ALE	30 0		25 0		ns ns	
26–6 27–6	A1 Setup Time to S1, S0 A1 Hold Time from S1, S0	27 0		22 0		ns ns	
43–13 44–13 45–13 46–13	A20 Delay from NA20 A20 Delay from A20GATE A20 Disable Delay from CPUHLDA ↑ A20 Enable Delay from CPUHLDA ↓		27 37 35 35		22 32 30 30	ns ns ns ns	 4
17–3 18–3	ALE Active Delay from PROCCLK ↓ ALE Inactive Delay from PROCCLK ↓		25 30		23 25	ns ns	
68–21	$\overline{\text{BUSY286}}$ Delay from $\overline{\text{BUSY}}$, –IOW		35		35	ns	
	CCROSC High Time CCROSC Low Time CCROSC Input Rise/Fall Time	25 25	 20	25 25	 20	µs µs ns	5 5 12
39–10, 11 40–10, 11	CCRR/W Setup Time to IOR/IOW ↓ CCRR/W Hold Time from IOR/IOW ↑	0 17		0 15		ns ns	
	CCRRST Pulse Width	100		83		ns	
67–20	CNTLOFF Delay from PROCCLK ↓		30		25	ns	
70–23 71–23	CPUHLDA Setup Time to PROCCLK ↓ CPUHLDA Hold Time from PROCCLK ↓	20 0		15 0		ns ns	
49–14 49–14	DIR245 Delay from –IOR ↓, –IOW ↓ DIR245 Delay from –MEMR, –MEMW DIR245 Delay from –AEN1, –AEN2		17 17 40		15 15 35	ns ns ns	
55–16 56–16	DT/R Delay High from PROCCLK ↓ DT/R Delay Low from PROCCLK ↓		45 45		40 40	ns ns	
60–16 61–16	F16 Setup Time to PROCCLK ↓ F16 Hold Time from PROCCLK ↓	30 0		30 0		ns ns	
	+FSYS16 Setup Time to PROCCLK ↓ +FSYS16 Hold Time from PROCCLK ↓	100 50		83 40		ns ns	
50–14 50–14	–GATE245 Delay from –IOR ↓, –IOW ↓ –GATE245 Delay from –MEMR, –MEMW –GATE245 Delay from –AEN1, –AEN2		22 22 45		20 20 40	ns ns ns	

82230 AC CHARACTERISTICS (V_{DD} = 5V ±5%, T_A = 0°C to 70°C) (Continued)

Symbol – Figure	Parameter	10 MHz		12 MHz		Units	Notes
		Min	Max	Min	Max		
37–9	Interrupt Request Pulse Width	100		100		ns	8
29–7, 8	–IOR, –IOW Active Delay from PROCCLK ↓	3	15	3	15	ns	
30–7, 8	–IOR, –IOW Inactive Delay from PROCCLK ↓	3	15	3	15	ns	
	–IOR, –IOW Enable/Disable Delay from CPUHLDA		40		40	ns	
	–INTA Active Delay from PROCCLK ↓	3	45	3	35	ns	
	–INTA Inactive Delay from PROCCLK ↓	3	45	3	35	ns	
	–INTA Enable/Disable Delay from CPUHLDA		40		40	ns	
38–9	INTR Delay from Interrupt		175		150	ns	
33–7, 8	–INTR1CS, – INTR2CS Setup Time to –IOR, –IOW ↓	0		0		ns	
34–7, 8	–INTR1CS, – INTR2CS Hold Time from –IOR, –IOW ↑	0		0		ns	
72–24	–IO CS 16 Setup Time to SYSCLK ↓	85		75		ns	
73–24	–IO CS 16 Hold Time from SYSCLK ↓	0		0		ns	
57–16	–LSDEN, –MSDEN Active Delay from PROCCLK ↓		45		40	ns	
58–16	–LSDEN, –MSDEN Inactive Delay from PROCCLK ↓		35		30	ns	
66–19	–LSDEN, –MSDEN Delay from –NPCS		15		15	ns	
	–LSDEN, –MSDEN Active Delay from SM/$\overline{IO}$ after –CS287 Inactive		30		30	ns	
79–16	–MEMR, –MEMW Active Delay from PROCCLK ↓	3	15	3	15	ns	
80–16	–MEMR, –MEMW Inactive Delay from PROCCLK ↓	3	15	3	15	ns	
	–MEMR, –MEMW Enable/Disable Delay from CPUHLDA		40		40	ns	
64–17	–MSDEN Delay from –XHBE		27		25	ns	
62–16	M/$\overline{IO}$ Setup Time to PROCCLK ↓	28		25		ns	
63–16	M/$\overline{IO}$ Hold Time from PROCCLK ↓	0		0		ns	
65–18	–NPCS Delay from SM/$\overline{IO}$ –CS287, XA3, –INTA		40		35	ns	
11–2	PCLK, –PCLK High Time	45		35		ns	
12–2	PCLK, –PCLK Low Time	45		35		ns	
13–2	PCLK, –PCLK Delay from PROCCLK		45		40	ns	
14–2	PCLK, –PCLK Rise/Fall Times		7.5		5	ns	12
19–4	$\overline{POWER\ GOOD}$ Setup Time to PROCCLK ↓	26		26		ns	3
	$\overline{POWER\ GOOD}$ Hold Time from PROCCLK ↓	50		41		ns	3
8–4	$\overline{POWER\ GOOD}$ Rise/Fall Times		20		20	ns	12
	$\overline{POWER\ GOOD}$ Inactive Pulse Width	1		1		μs	
5–2	PROCCLK Delay from X3	5	25	5	20	ns	
6–2	PROCCLK High Time	16		13		ns	
7–2	PROCCLK Low Time	12		11		ns	
9–2	PROCCLK Rise/Fall Time		8		8	ns	12
22-5	–RC Setup Time to SYSCLK ↑	30		30		ns	3
23-5	–RC Pulse Width	100		83		ns	
51–15	$\overline{READY}$ Active Delay from PROCCLK ↓		22		18	ns	
52–15	$\overline{READY}$ Inactive Delay from PROCCLK ↓		70		60	ns	

82230 AC CHARACTERISTICS (V_{DD} = 5V ±5%, T_A = 0°C to 70°C) (Continued)

Symbol –Figure	Parameter	10 MHz		12 MHz		Units	Notes
		Min	Max	Min	Max		
	REFRDY Pulse Width	50		40		ns	
	REFRDY Hold Time from PROCCLK ↓	34		34		ns	
	REFRDY Setup Time to PROCCLK ↓	−14		−14		ns	
74–25	RES 287 Delay from −IOW		60		50	ns	
21–4	RES CPU Delay from PROCCLK ↓		27		22	ns	
20–4	+RESET Delay from PROCCLK ↓		50		50	ns	
41–12	SA0 Enable Time from CPU HLDA		60		50	ns	
42–12	SA0 Disable Time from CPU HLDA		60		50	ns	4
15–3	S1, S0 Setup Time to PROCCLK ↓	28		15		ns	
16–3	S1, S0 Hold Time from PROCCLK ↓	0		0		ns	
10–2	SYSCLK Delay from PROCCLK ↓	5	20	5	20	ns	
1–2	X3 Period	50		41.7		ns	
2–2	X3 Low Time	17		15		ns	
3–2	X3 High Time	23		20		ns	
4–2	X3 Rise/Fall Times		5		3	ns	12
35–8, 11	XD0–XD7 Delay Time from −IOR ↓		45		40	ns	
36–8, 11	XD0–XD7 Hold Time from −IOR ↑		17		15	ns	
31–7	XD0–XD7 Setup Time to −IOW ↑	100		83		ns	
32–7	XD0–XD7 Hold Time from −IOW ↑	0		0		ns	

82231 AC CHARACTERISTICS (V_{DD} = 5V ±5%, T_A = 0°C to 70°C)

Symbol –Figure	Parameter	10 MHz		12 MHz		Units	Notes
		Min	Max	Min	Max		
96–27	−8042CS Delay from XA_X		60		48	ns	
98–28	A17–A23 Delay from SYSCLK ↑		150		125	ns	
99–28	A17–A23 Enable Delay from HLDA or −MASTER		100		83	ns	
97–28	A17–A23 Disable Delay from HLDA or −MASTER		100		83	ns	4
100–29	+ACK Delay from HLDA		45		40	ns	
101–29	+ACK Delay from −MASTER		45		40	ns	
109–30	−AEN1, −AEN2 Delay from SYSCLK ↑		130		115	ns	
94–27	CCRR/W Delay from XA_X		60		48	ns	
102–29	CCRR/W Delay from HLDA or −MASTER		50		41	ns	
108–30, 39	CPU HRQ Delay from SYSCLK ↑		80		70	ns	
95–26	−CS287 Delay from XA_X		60		48	ns	
103–29	−CS287 Delay from HLDA or −MASTER		50		41	ns	
113–30	−DACK0–3, −DACK5–7 Delay from SYSCLK ↑		110		100	ns	
110–30	−DMAAEN Delay from SYSCLK ↑		140		120	ns	

82231 AC CHARACTERISTICS (V_{DD} = 5V ±5%, T_A = 0°C to 70°C) (Continued)

Symbol –Figure	Parameter	10 MHz		12 MHz		Units	Notes
		Min	Max	Min	Max		
132–32 133–32	–DPCK Setup Time to –XMEMR ↑ –DPCK Hold Time from –XMEMR ↑	8 5		6 5		ns ns	
107–30	–DRQ0–3, –DRQ5–7 Setup Time to SYSCLK ↑	0			0	ns	3, 7
	HLDA Setup Time to SYSCLK ↑ HLDA Hold Time from SYSCLK ↑	70 0		65 0		ns ns	
28–7 104–29	–INTR1CS, –INTR2CS Delay from XA_X –INTR1CS, –INTR2CS Delay from HLDA or –MASTER		60 60		41 41	ns ns	
134–33	–IOCHCK Pulse Width		25		20	ns	
	IOCHRDY Setup Time to SYSCLK ↑ (During Refresh) IOCHRDY Hold Time from SYSCLK ↑ (During Refresh)	25 25		25 25		ns ns	
114–30	–IOR, –IOW Active Delay from SYSCLK ↑ (During DMA Transfers)		125		100	ns	
115–30	–IOR, –IOW Inactive Delay from SYSCLK ↑ (During DMA Transfers)		115		100	ns	
116–30	–IOR, –IOW Float to Inactive Delay from SYSCLK ↑ (During DMA Transfers)		120		100	ns	
117–30	–IOR, –IOW Inactive to Float Delay from SYSCLK ↑ (During DMA Transfers)		170		156	ns	4
152–37	–IOW Active Pulse Width (During CPU Transfers)	90		75		ns	
137–34	IRQ0 Delay from X1		100		100	ns	
 135–33	NMI Delay from –XMEMR ↑ NMI Delay from –IOCHCK ↓		100 100		83 83	ns ns	
143-35 144-35 145-35 146-35	OSC Low Time OSC High Time OSC Rise/Fall Times OSC Delay from X1	20 20	 15 30	20 20	 15 24	ns ns ns ns	 12
	P2 Delay from SYSCLK ↑		100		83	ns	
148–36 149–36	–RDXDB Delay from –IOR –RDXDB Delay from –INTA		100 100		83 83	ns ns	
	REFRDY Delay from IOCHRDY	10	35	10	35	ns	
157–39 158–39	–REFRESH Delay from HLDA –REFRESH Delay from SYSCLK ↑		28 100		24 83	ns ns	
	+ RESET Active Pulse Width	200		160		ns	
138–34	SPEAKER Delay from X1		100		100	ns	
91–26 92–26 93–26	SYSCLK Period SYSCLK Low Time SYSCLK High Time	100 40 40		83 30 30		ns ns ns	

82231 AC CHARACTERISTICS (V_{DD} = 5V ±5%, T_A = 0°C to 70°C) (Continued)

Symbol – Figure	Parameter	10 MHz		12 MHz		Units	Notes
		Min	Max	Min	Max		
124–30	TC Delay from SYSCLK ↑		110		100	ns	
140–35	X1 Low Time	30		30		ns	
141–35	X1 High Time	30		30		ns	
142–35	X1 Rise/Fall Times		5		5	ns	12
150–37	XA_X Input Setup Time to –IOW ↓ (During CPU Transfers to 82231)	100		83		ns	
151–37	XA_X Input Hold Time from –IOW ↑ (During CPU Transfers to 82231)	45		40		ns	
	XA_X Input Hold Time from –IOR ↑ (During CPU Transfers to 82231)	100		67		ns	
111–30	XA_X Valid Delay from SYSCLK ↑ (During DMA Transfers)		160		150	ns	
159–39	XA_X Valid Delay from SYSCLK ↑ (During Refresh)		90		75	ns	
112–30	XA_X Disable Delay from SYSCLK ↑ (During DMA Transfers)		150		130	ns	4
153–37	XD_X Input Setup Time to –IOW ↑	100		83		ns	
154–37	XD_X Input Hold Time from –IOW ↑	17		15		ns	
155–38	XD_X Output Delay from –IOR ↓		125		100	ns	
156–38	XD_X Output Hold Time from –IOR ↑	17	70	15	60	ns	
125–31	–XMEMR Active Delay from SYSCLK ↑		110		100	ns	
126–31	–XMEMR Inactive Delay from SYSCLK ↑		110		100	ns	
127–31	–XMEMR Enable/Disable Delay from SYSCLK ↑		120		120	ns	4, 12
121–30	–XMEMW Active Delay from SYSCLK ↑		110		100	ns	
122–30	–XMEMW Inactive Delay from SYSCLK ↑		110		100	ns	
123–30	–XMEMW Enable/Disable Delay from SYSCLK ↑		120		120	ns	4, 12

NOTES:
1. To provide clearly understood information, the complex timing diagrams depict operation in a standard IBM PC AT system design. Combinational logic data paths are shown with less complex timing diagrams. The signal source (82230, 82231, PROCESSOR, LOGIC, etc.) follows the signal name.
2. The direction control signals are delayed to PROCCLK ↓ on an –IOW cycle. This is done to avoid changing the direction of the byte-swapping bus transceivers while data is still on the bus.
3. This signal is an asynchronous input. The timing specification is provided for testing purposes only to assure recognition at a specific clock edge.
4. The output float or high impedance condition occurs when output current is less than I_{OZ} in magnitude.
5. The frequency of CCROSC sets the count rate for the real time clock. CCROSC frequency, accuracy and stability, should be maintained as close as possible to 32.768 KHz to insure the validity of time and data information.
6. Input rise and fall times are assumed to be less than 20 ns unless otherwise specified.
7. DRQ_X must be held active with $DACK_X$ is returned.
8. The interrupt request inputs include IRQ0, IRQ3–7, IRQ9–12, IRQ14–15 and +OPT.
9. Address XA_{0-15} are output for byte DMA operations. XA_{0-16} are output for word DMA operations, with XA0 low.
10. A minimum of 16 PROCCLK cycles must occur before $\overline{\text{POWERGOOD}}$ becomes valid.
11. At the end of TC phase 1 after TCW2 or TCW3 for 16-bit transfer to 8-bit source/destination, for the first 8 bits of transfer.
12. These are not tested. They are guaranteed by design characterization.
13. At the end of TC phase 1 for 16 bit back plane memory transfers and 8-bit transfers after TCW2 or TCW3.

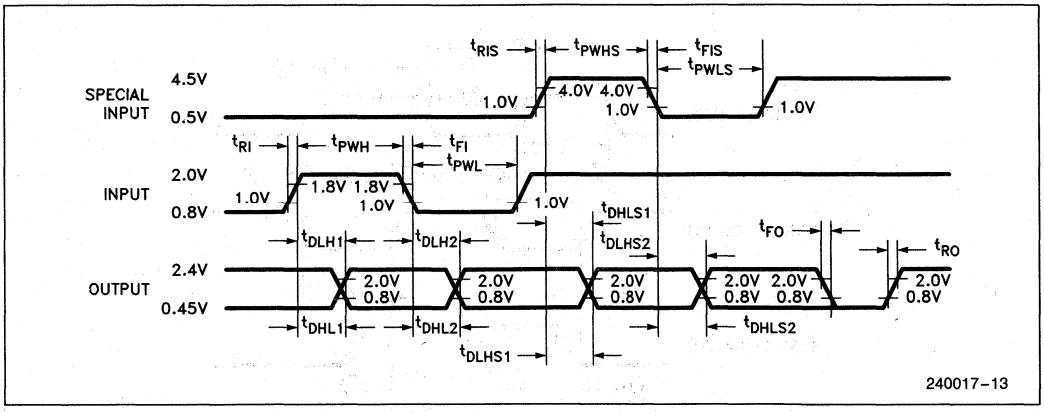


Figure 1A. Delay Time and Pulse Width Measurements

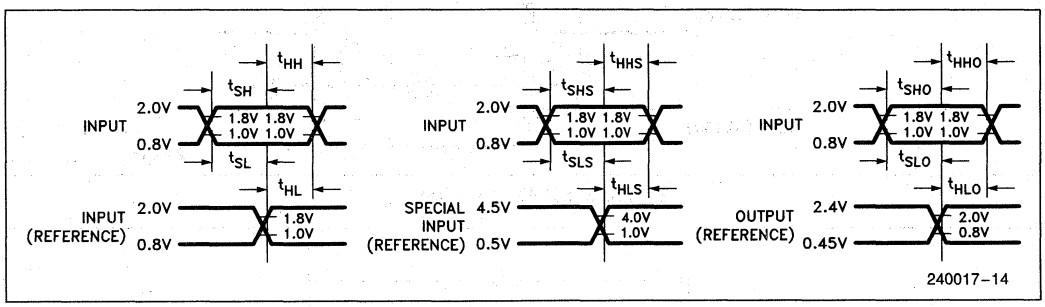


Figure 1B. Setup/Hold Time Measurements

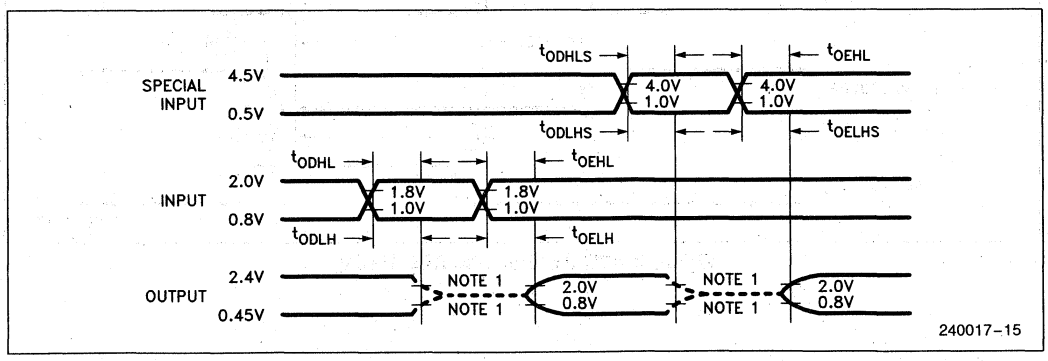


Figure 1C. Output Enable/Disable Time Measurement

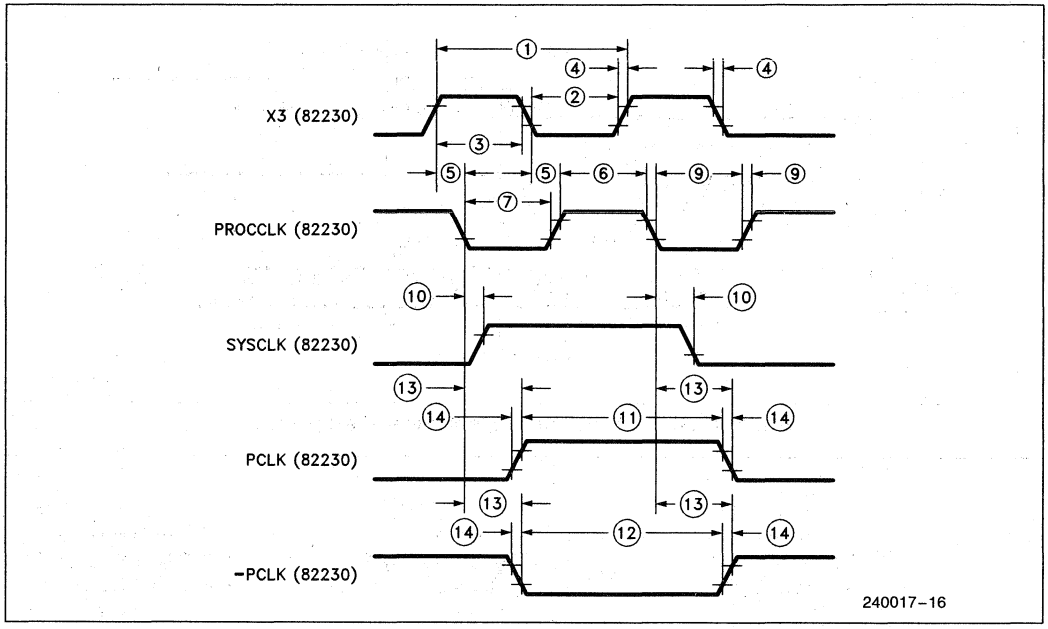


Figure 2. 82230 Clock Timing

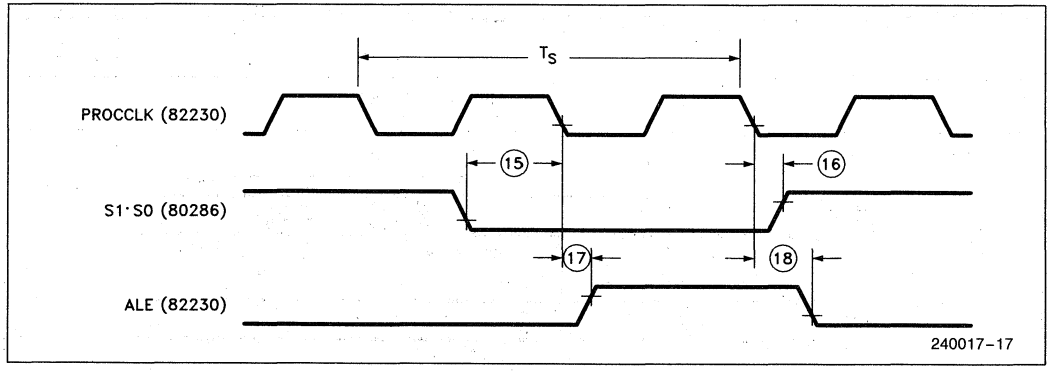


Figure 3. 82230 Status and ALE Timing

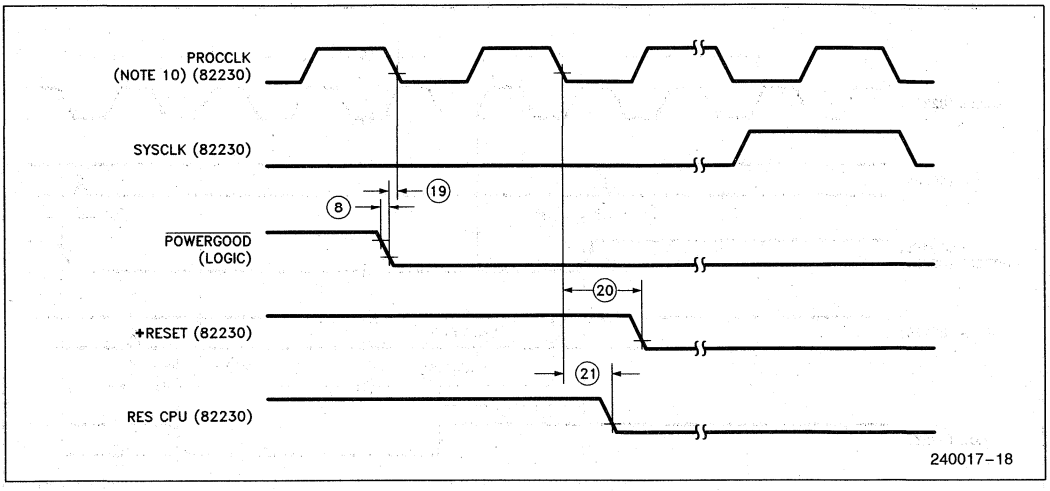


Figure 4. 82230 Power on Initiated Reset

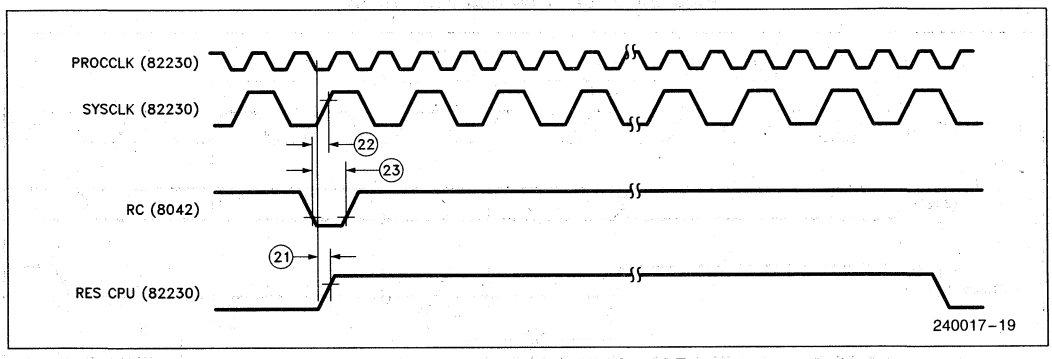


Figure 5. 82230 Keyboard Initiated Reset

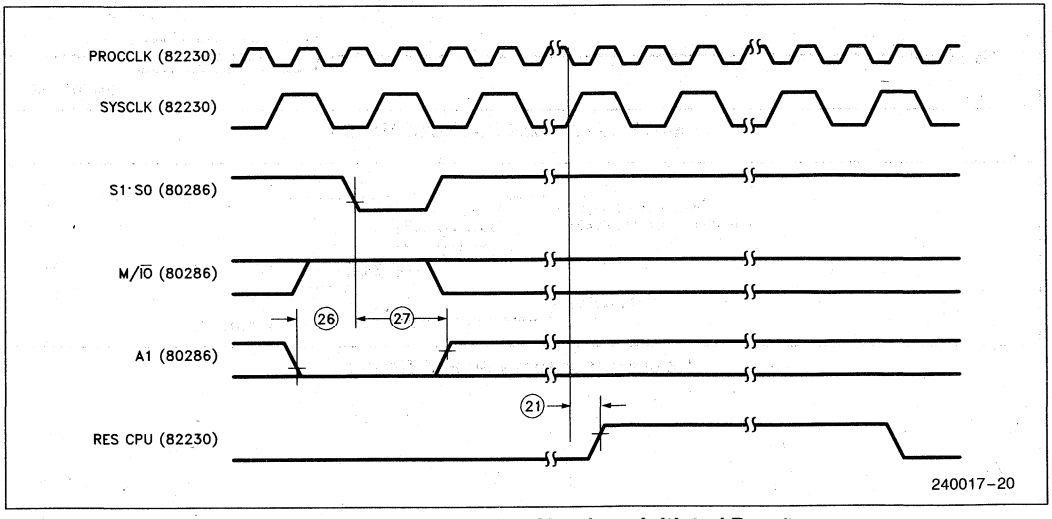


Figure 6. 82230 Processor Shutdown Initiated Reset

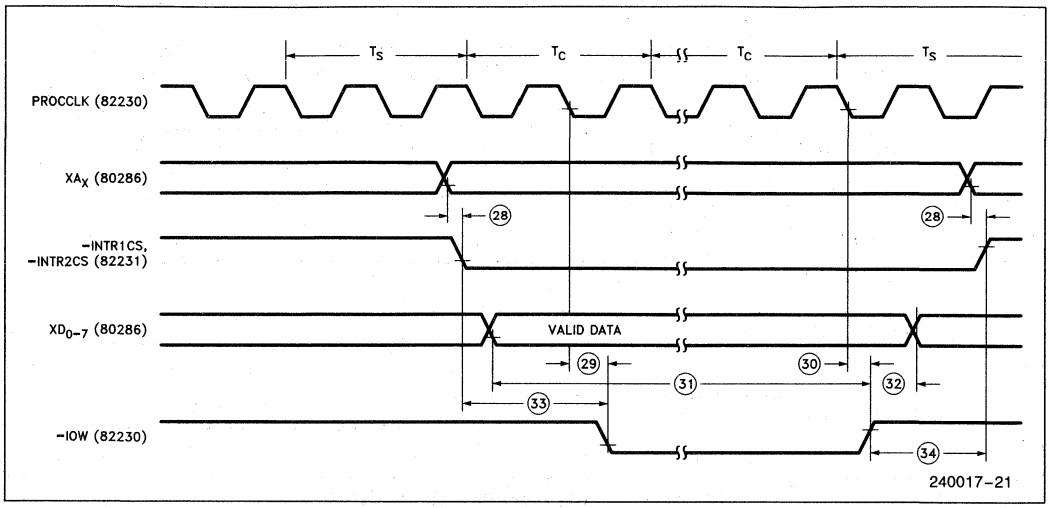


Figure 7. 82230 8254 Bus Write Timing

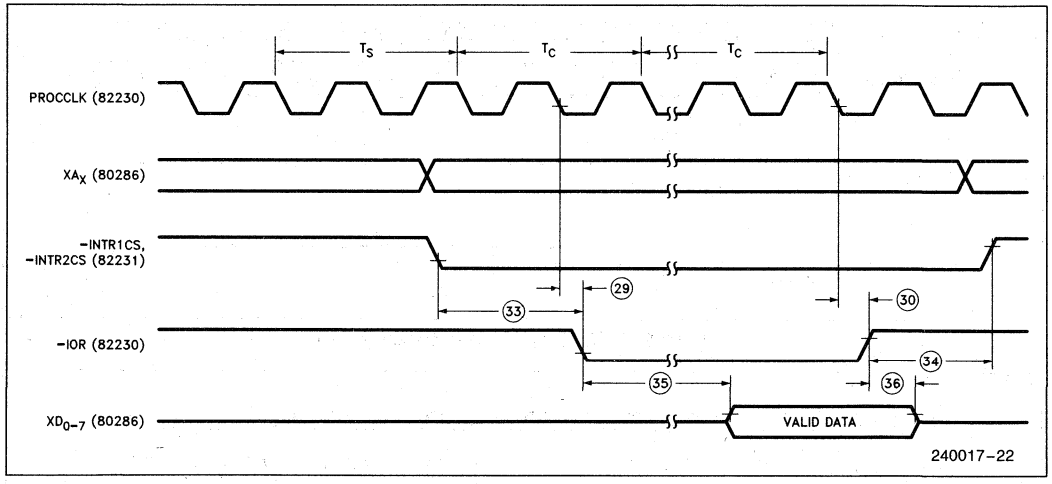


Figure 8. 82230 8254 Bus Read Timing

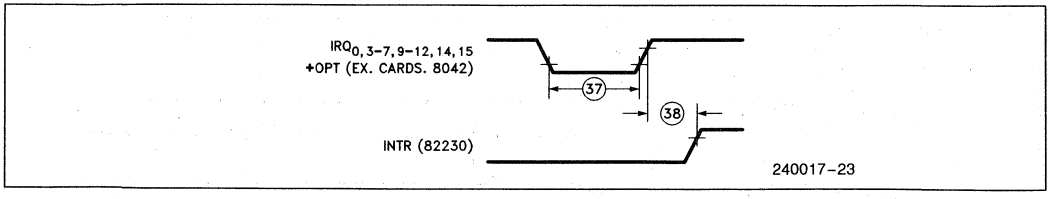


Figure 9. Interrupt Request Timing

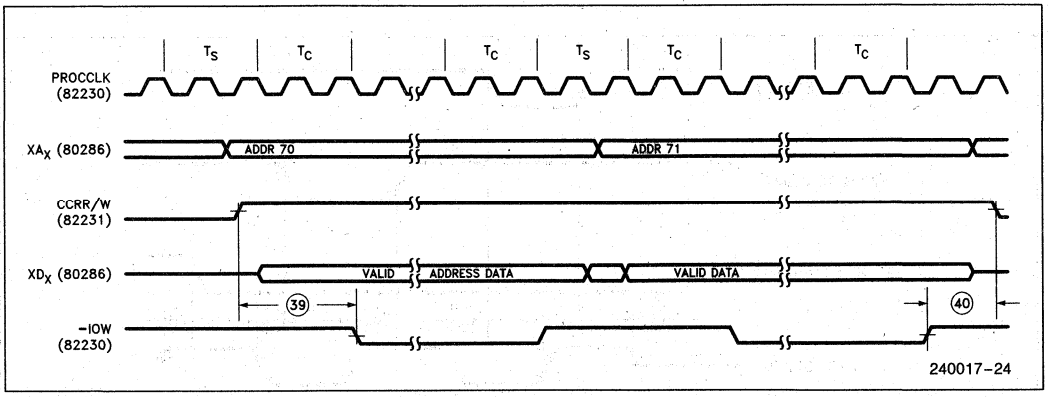


Figure 10. 82230 6818 Write Cycle

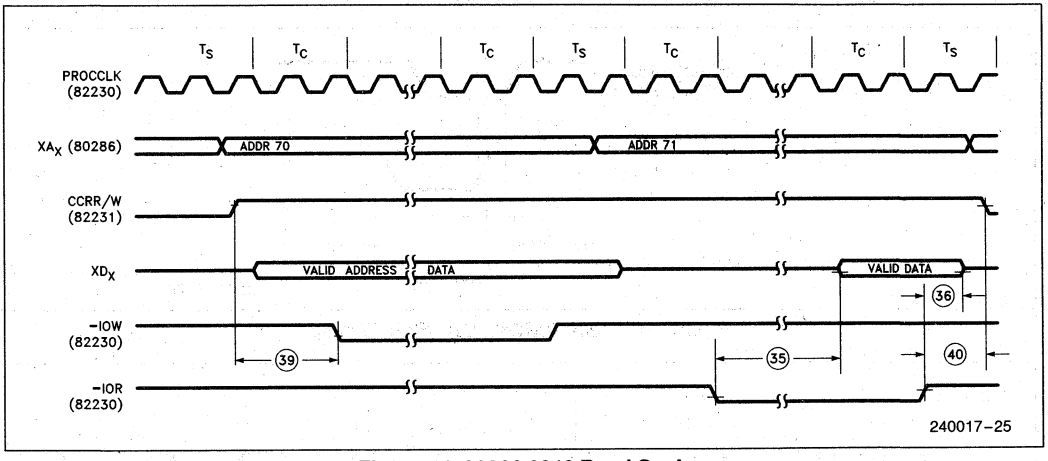


Figure 11. 82230 6818 Read Cycle

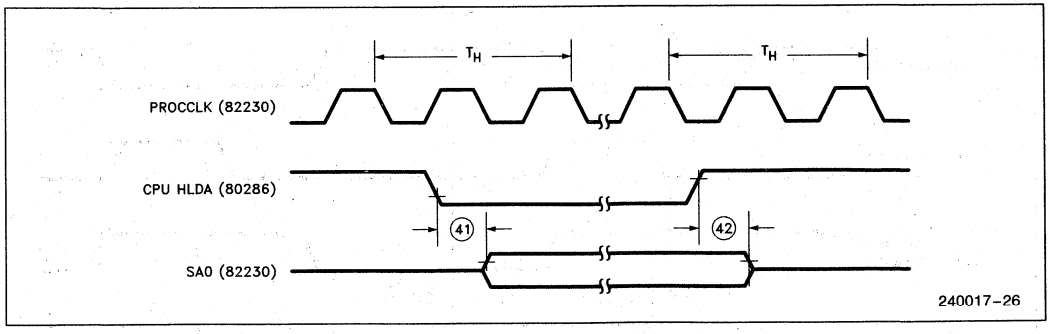


Figure 12. 82230 SA0 Timing

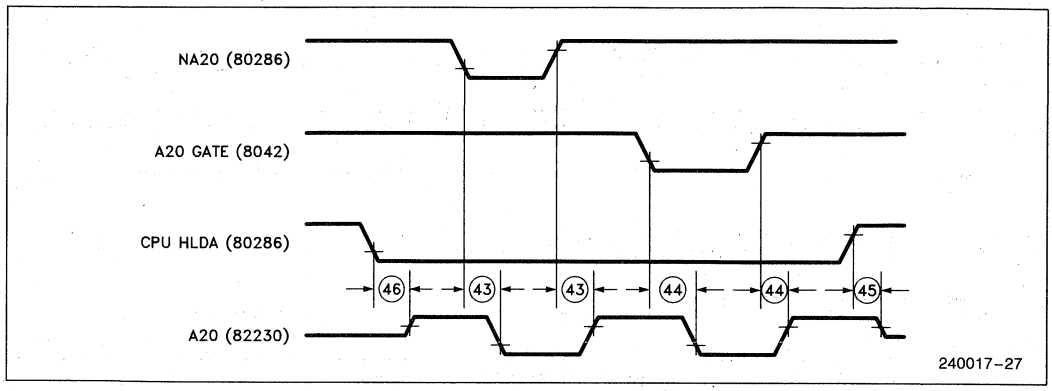


Figure 13. 82230 A20 Timing

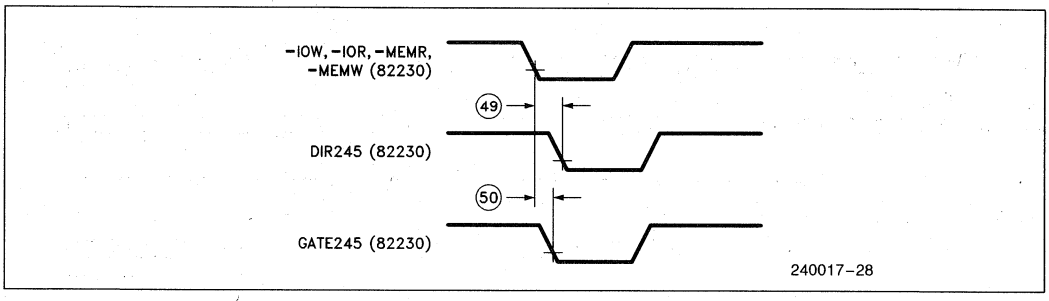


Figure 14. 82230 DIR245, GATE245 Timing

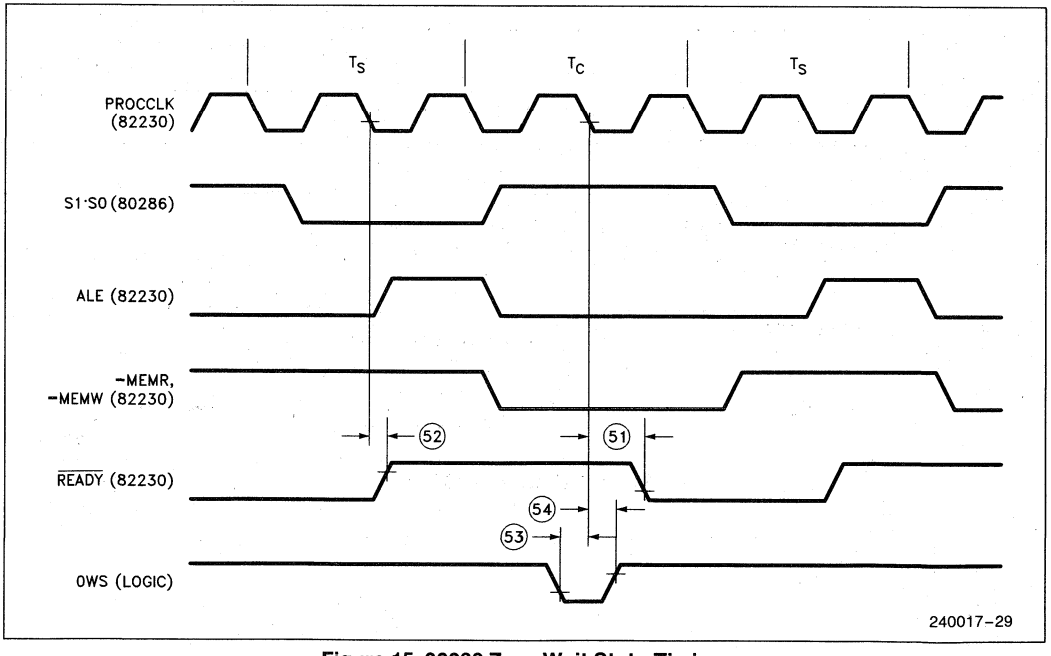


Figure 15. 82230 Zero Wait State Timing

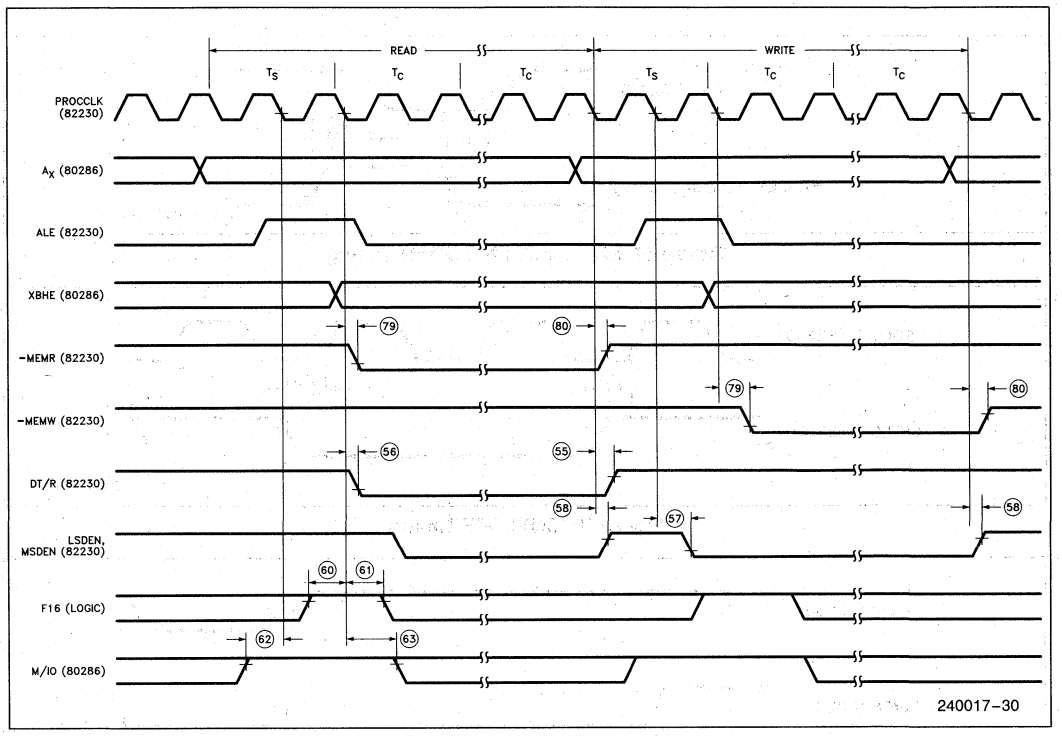


Figure 16. Memory Read/Write Cycles

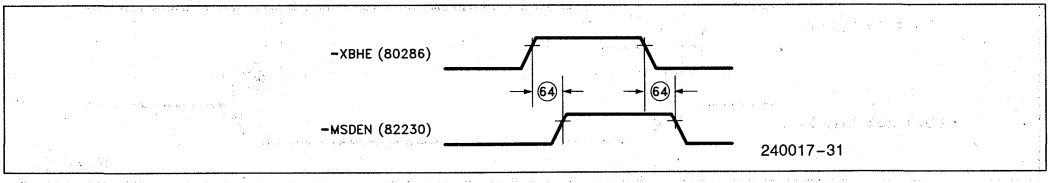


Figure 17. 82230 XHBE Timing

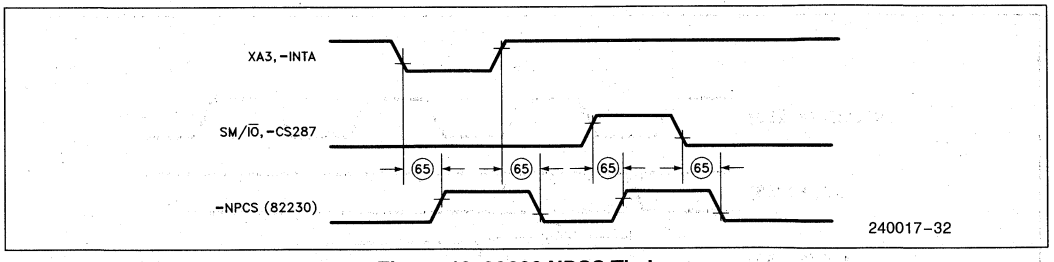


Figure 18. 82230 NPCS Timing

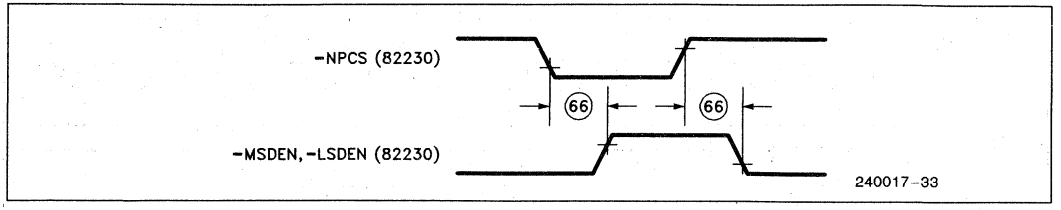


Figure 19. 82230 MSDEN, LSDEN Timing

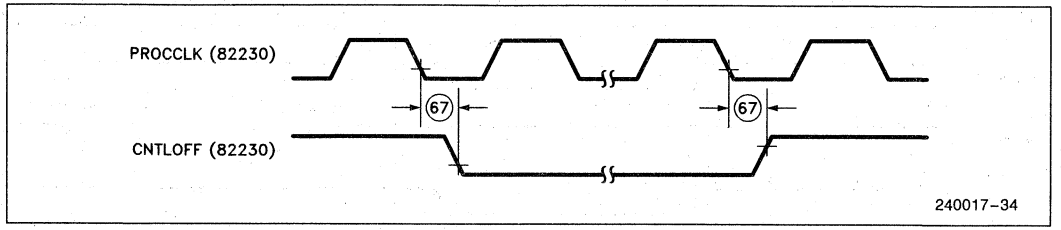


Figure 20. CNTLOFF Timing

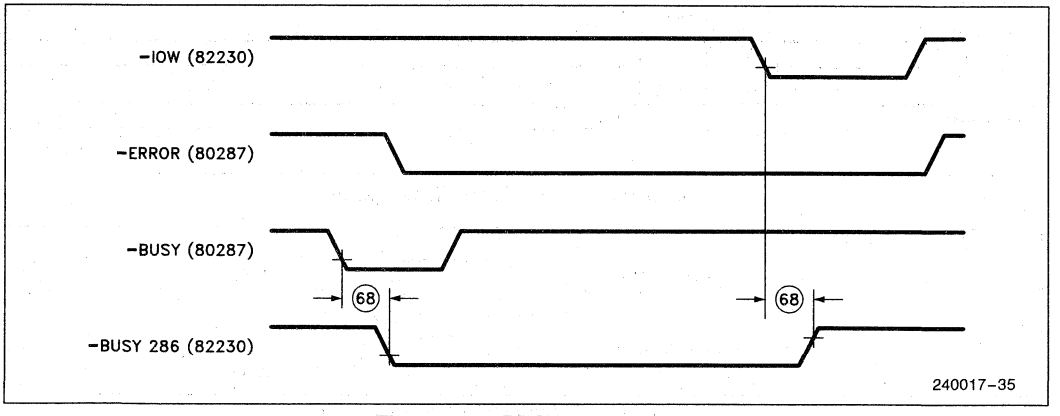


Figure 21. −BUSY286 Timing

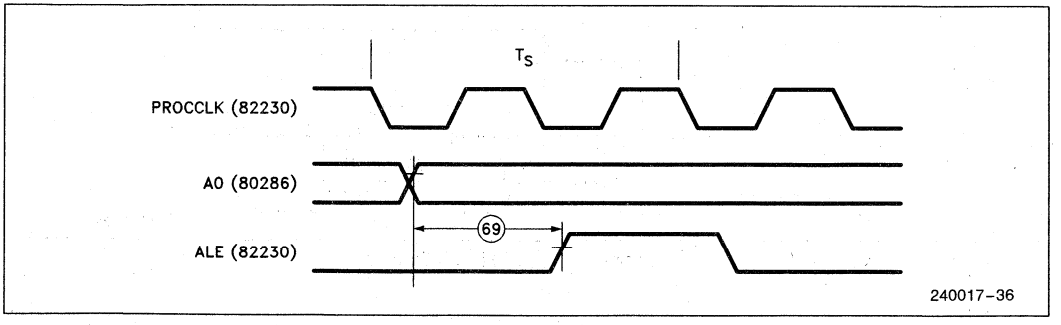


Figure 22. 82230 A0 Timing

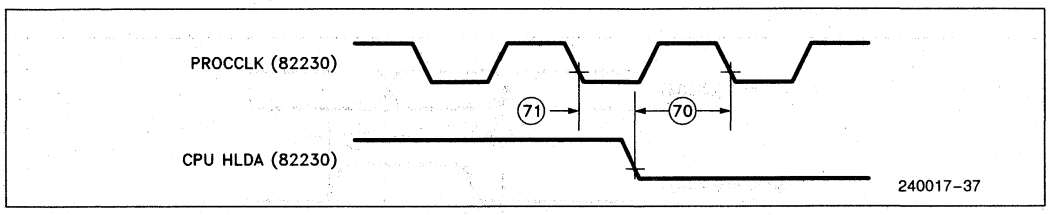


Figure 23. 82230 CPU HLDA Timing

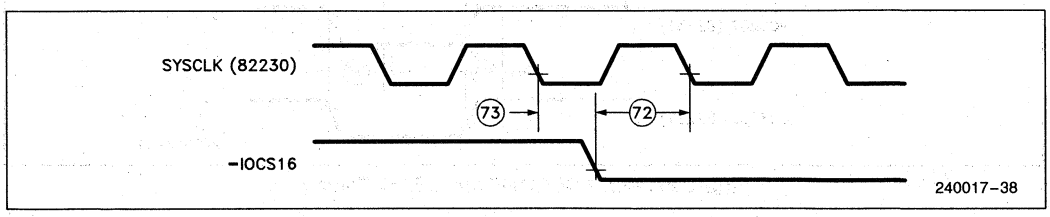


Figure 24. Bus Control Signal Timing

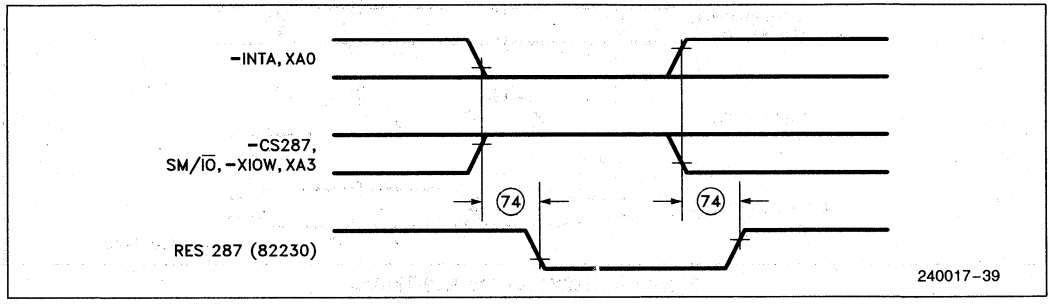


Figure 25. RES 287 Timing

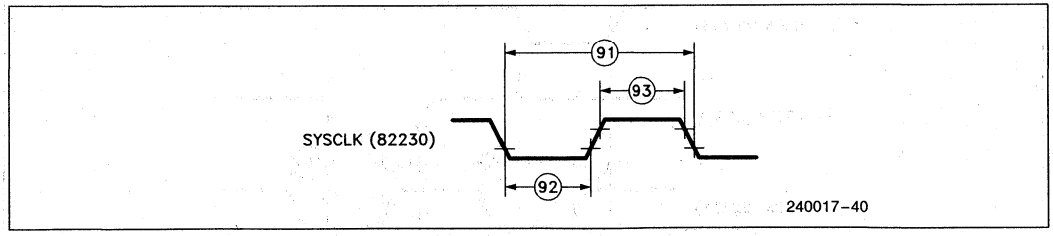


Figure 26. 82231 SYSCLK Timing

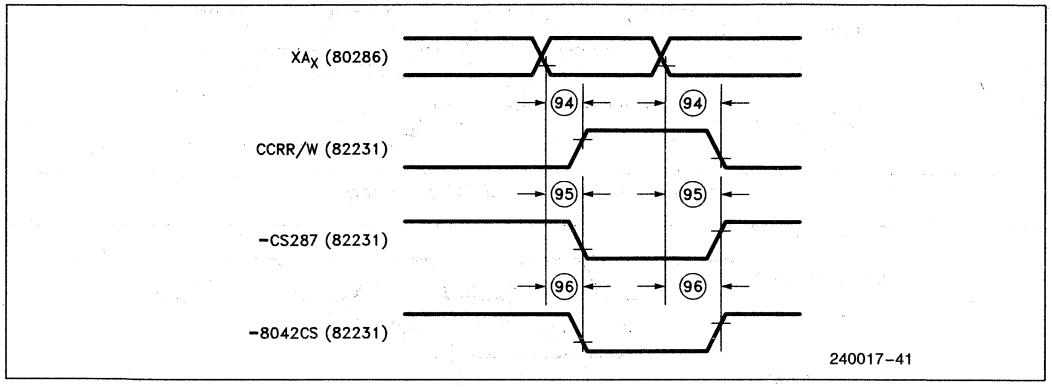


Figure 27. 82231 CCRR/W and CS287 Timing

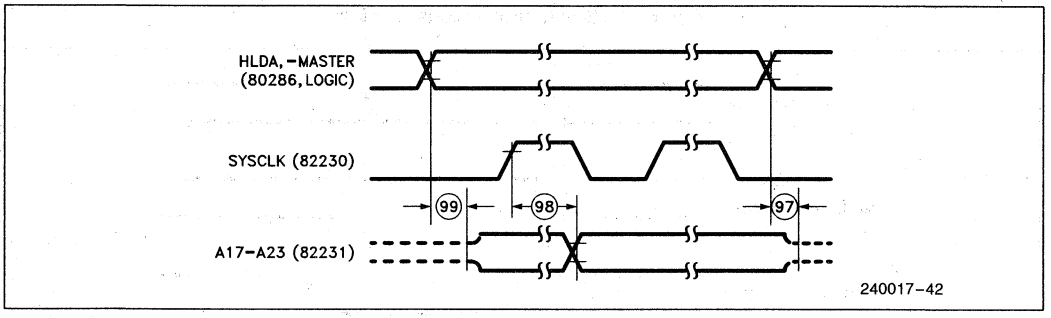


Figure 28. 82231 A17 to A23 Timing

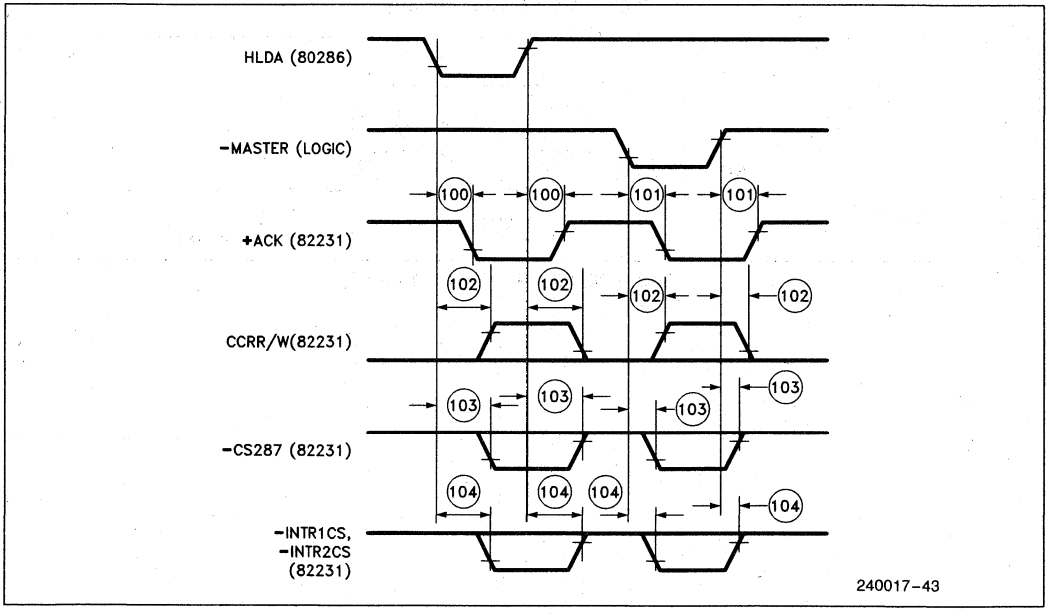


Figure 29. 82231 HLDA & −MASTER Timing

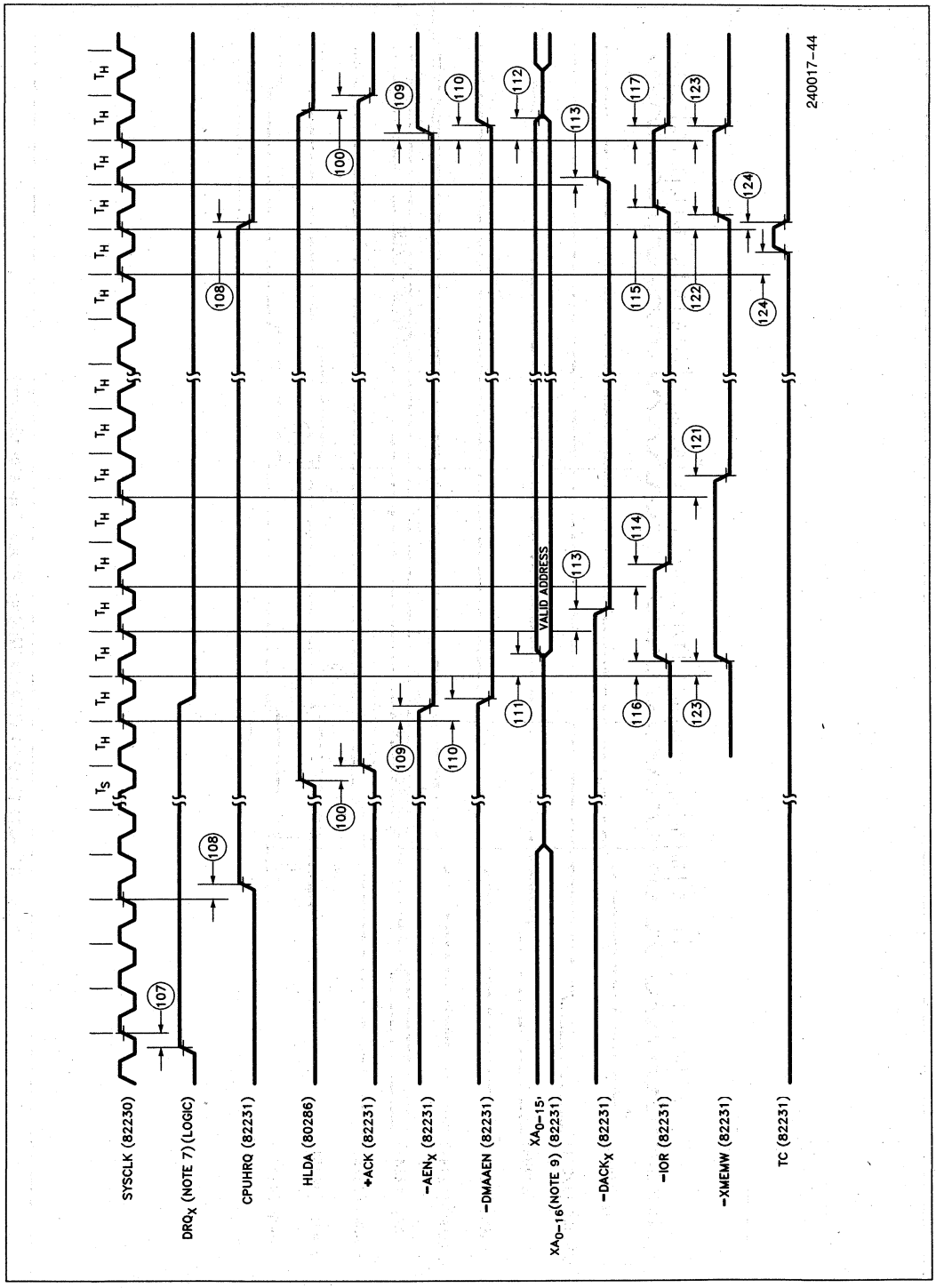


Figure 30. 82231 DMA I/O Read Timing (Single Transfer Shown)

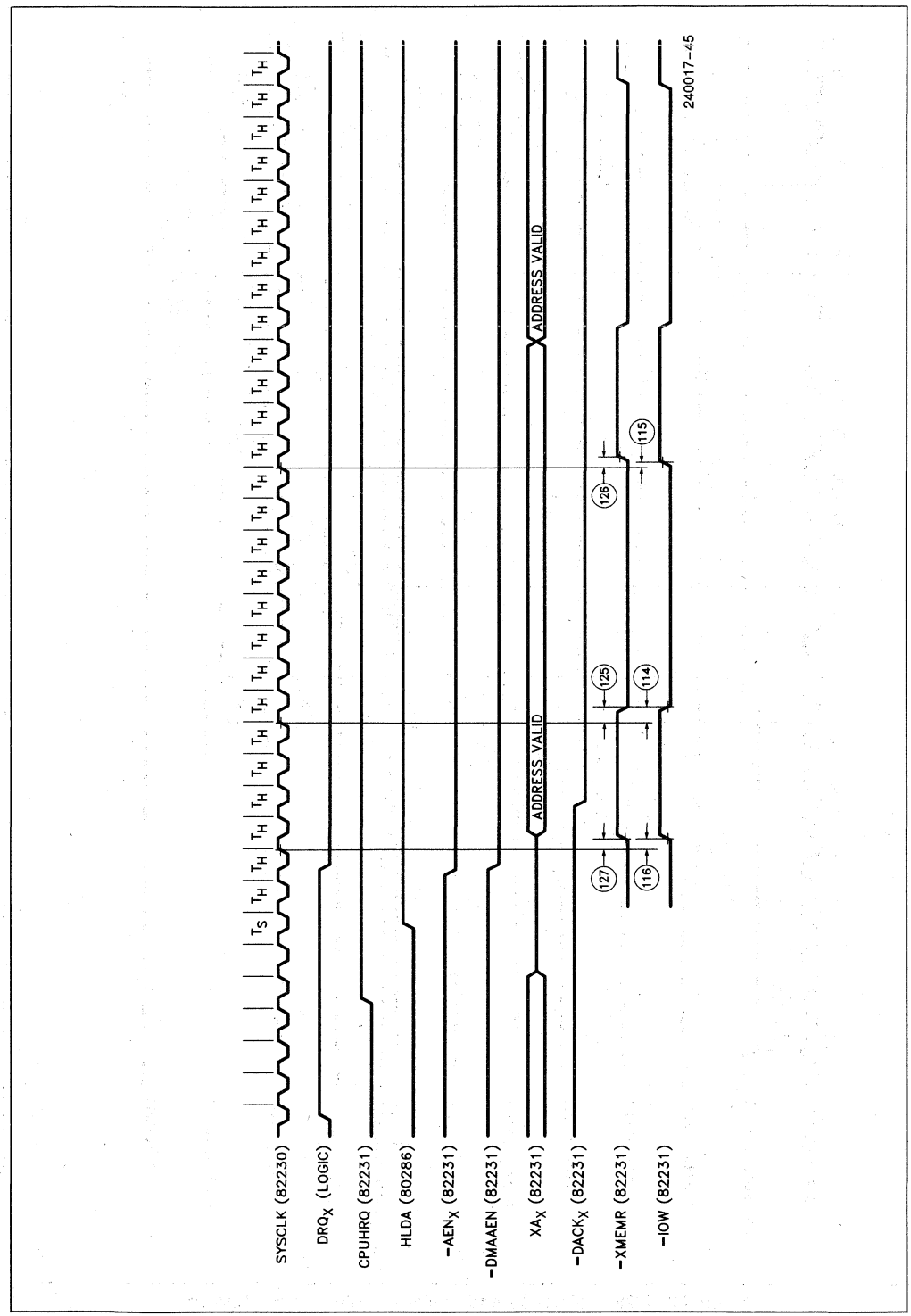


Figure 31. 82231 DMA I/O Write Timing (Block Transfer Shown)

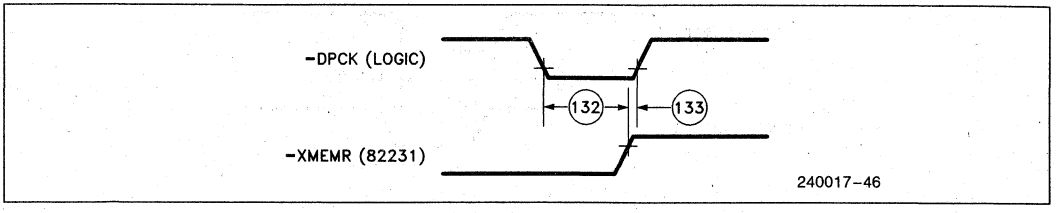


Figure 32. 82231 DPCK Timing

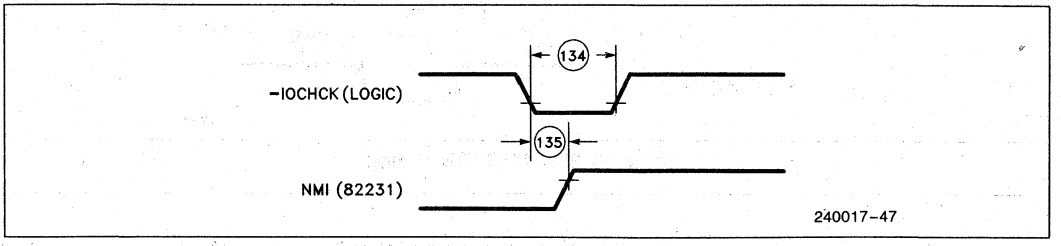


Figure 33. 82231 IOCHCK and NMI Timing

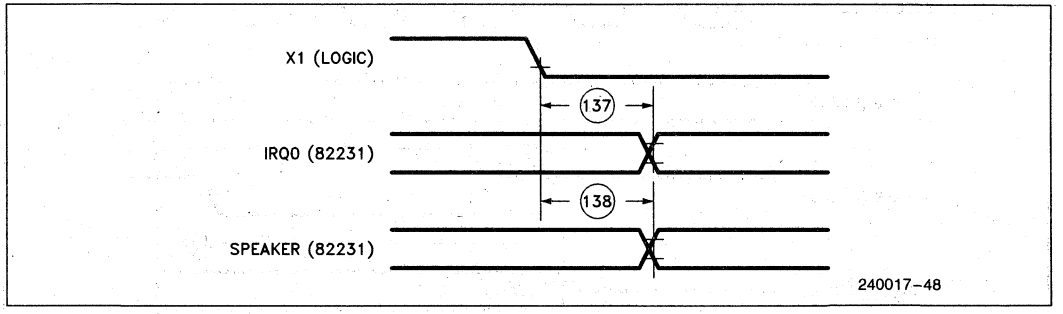


Figure 34. 82231 IRQ0 and SPEAKER Timing

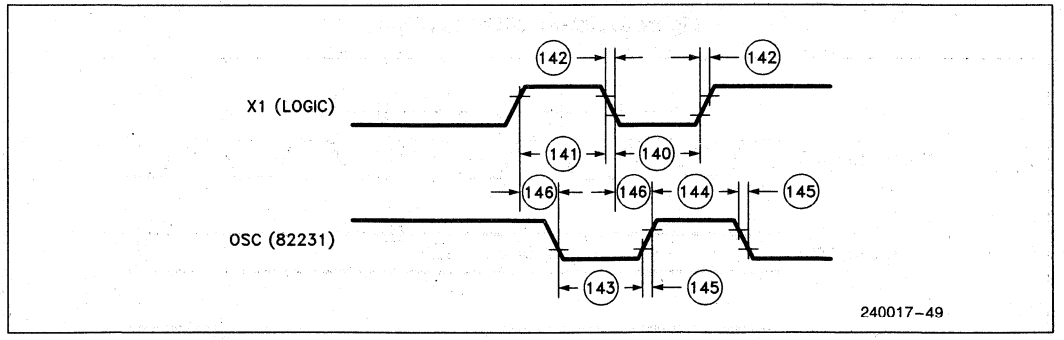


Figure 35. X1 and OSC Timing

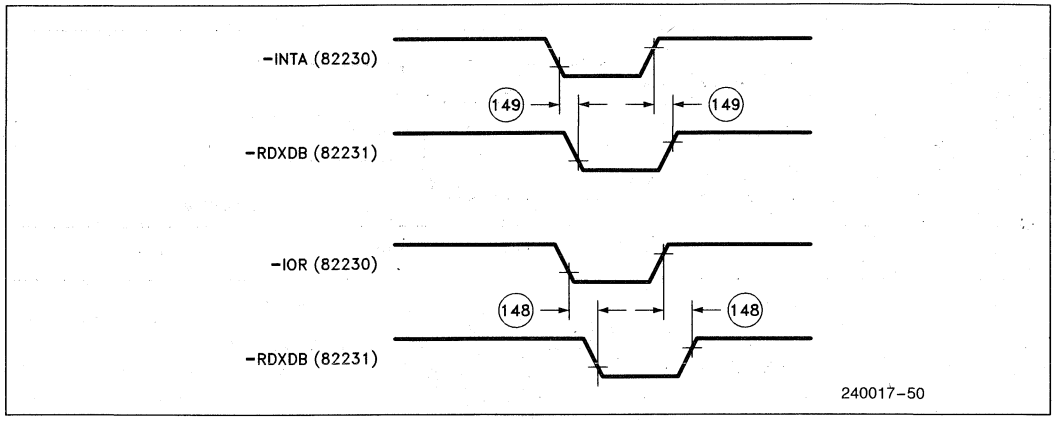


Figure 36. 82231 RDXDB Timing

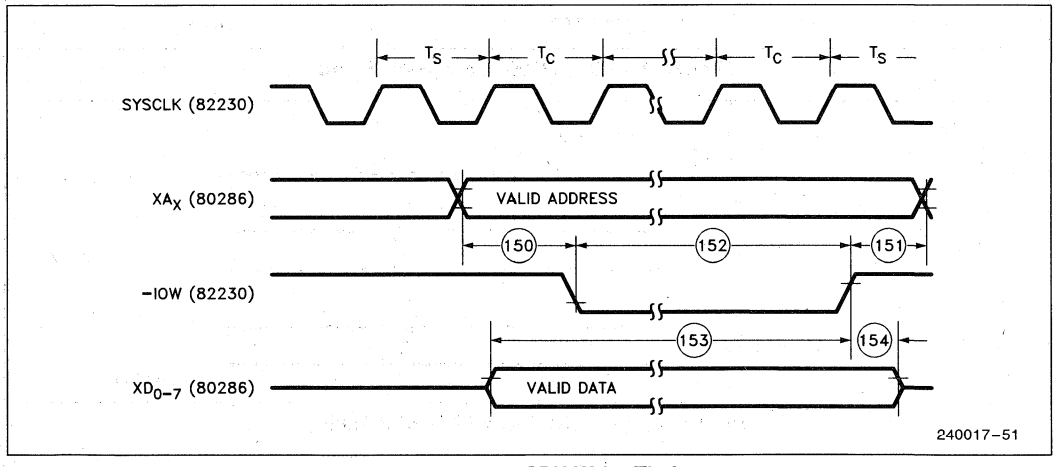


Figure 37. 82231 CPU Write Timing

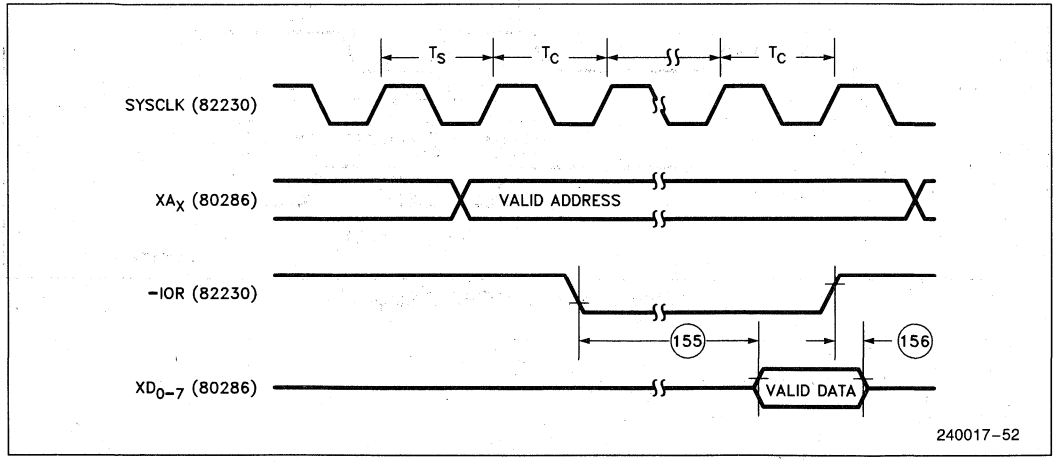


Figure 38. 82231 CPU Read Timing

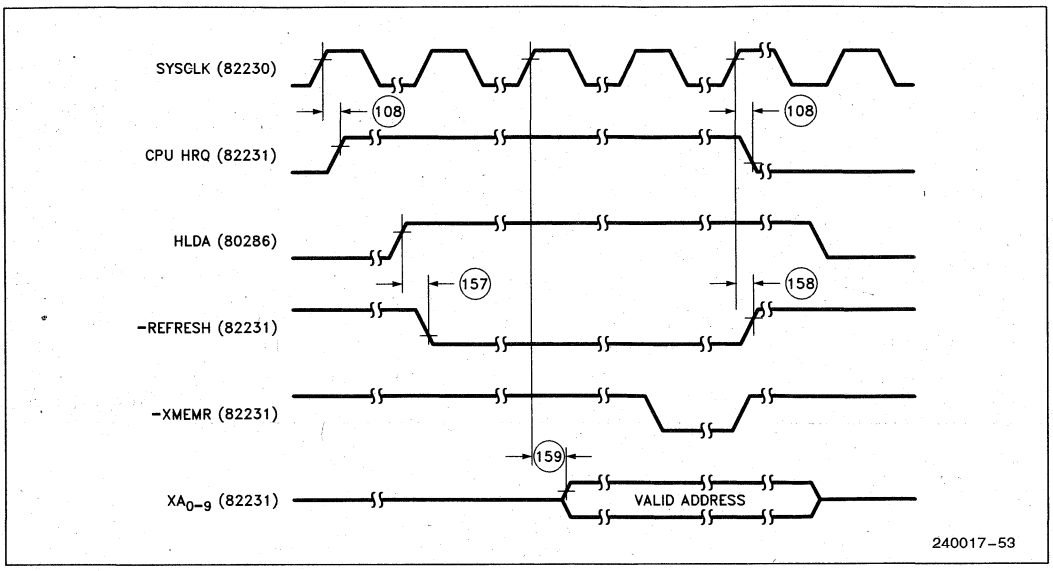
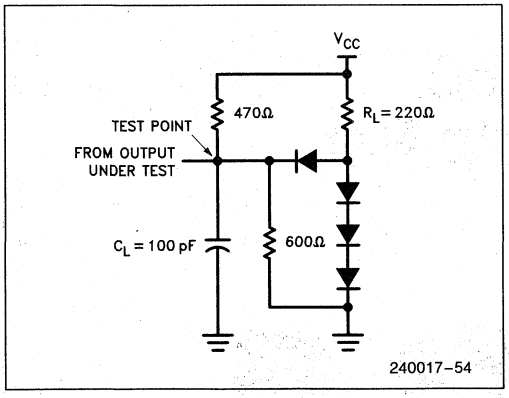


Figure 39. 82231 REFRESH Timing

TEST LOADS



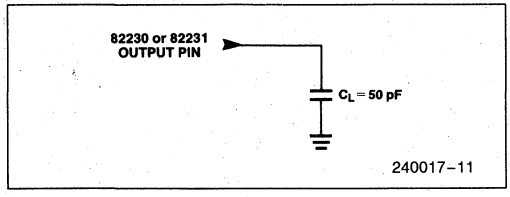
Output Load Circuit

For 82230 pins
- −IOR
- −IOW
- SA0

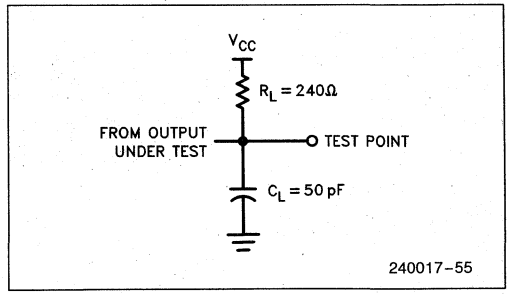
82231 pins
- −IOR
- −IOW

SYSCLK
- −MEMR
- −MEMW

OSC



Output Load Circuit for All Other Pins



Load Circuit for Open-Collector Output

REVISION HISTORY

The following list represents the differences between this and the -003 version of the data sheet.

1. The 82231 DMA I/O Read Timing diagram was corrected.

August 1990

82350
EISA Chip Set

Order Number: 290220-003

82350 EISA CHIP SET

CONTENTS

PAGE

EISA TERMINOLOGY 1-176
EISA System Introduction 1-176
82350 EISA Chip Set Highlights 1-176
EISA System Block Diagrams 1-177

MECHANICAL DATA 1-179
Package Information 1-179

Introduction 1-179
Package Thermal Specification 1-182

82357 INTEGRATED SYSTEM PERIPHERAL (ISP) 1-183

82358 EISA BUS CONTROLLER (EBC) 1-272

EISA TERMINOLOGY

ISA BUS— The bus used in Industry Standard Architecture compatible computers. In the context of an EISA system, it refers to the ISA subset of the EISA bus.

EISA BUS— Extended ISA bus, a superset of the ISA bus. It includes all ISA bus features, along with extensions to enhance performance and capabilities.

HOST CPU— The main system processor, located on a separate Host Bus. This uses the EBC and other system board facilities to interface to the EISA bus.

CPU CYCLE— 386™ CPU and/or the 82385 subsystem, or 80486 CPU is the master running the cycle.

EISA MASTER— A 16-bit or 32-bit bus master that uses the EISA signal set to generate memory or I/O cycles. The bus controller will convert the EISA control signals to ISA signals, when necessary.

ISA MASTER—A 16-bit bus master that uses the ISA subset of the EISA bus for generation of memory or I/O cycles. This device must understand 8-bit or 16-bit ISA slaves, and route data to the appropriate byte lanes. It is not required to handle any of the signals associated with the extended portion of the EISA bus.

EISA SLAVE— An 8-bit, 16-bit or 32-bit memory or I/O slave device that uses the extended signal set of the EISA bus to accept cycles from various masters. It returns information about its type and width using extended and ISA signals.

ISA SLAVE— A 16-bit or 8-bit slave that uses the ISA subset of the EISA bus to accept cycles from various masters. It returns ISA signals to indicate its type and width.

DMA SLAVE— An I/O device that uses the DMA signals (DREQ, DACK#) of the system board ISP to perform a direct memory access.

ISACMD— The ISA command signals (IORC#, IOWC#, MRDC#, MWTC#)

ASSEMBLY/DISASSEMBLY— This occurs when the master/slave data bus size are mismatched. The EBC runs multiple cycles to route bytes to the appropriate byte lanes (byte swapping). For example, if the 32-bit CPU is accessing an 8-bit slave, the EBC. will need to run four cycles to the 8-bit slave and route the bytes to appropriate byte lanes

CYCLE TRANSLATION— This is performed by the EBC when the master and slave are on different busses (Host/EISA/ISA). The EBC will translate the master protocol to the slave protocol (Host master accessing EISA slave).

EISA System Introduction

Extended Industry Standard Architecture (EISA) is a high performance 32-bit architecture based upon the Industry Standard Architecture (ISA) (PC AT*). The wide acceptance of the 32-bit 386 microprocessor family has led to this interest in extending ISA to 32-bits. EISA's advanced capabilities and 32-bit architecture can unleash the full potential of the 386 and i486™ CPUs.

The EISA consortium has defined the EISA bus in response to the demand for a 32-bit high performance ISA compatible system. The open industry standard allows for industry wide participation, compatibility, and differentiation.

EISA brings advances in performance and convenience to the user. It provides 32-bit memory addressing and data transfers for CPU, DMA and bus masters allowing 33 Mbyte/second transfer rate for DMA and bus masters on the EISA bus. EISA provides a specification for auto-configuration of add-in cards that will eliminate the need for jumpers and switches on EISA cards. Interrupts are shareable and programmable. Figure 1 and 2 show the types of busses in an EISA system. A new bus-arbitration makes possible a new generation of intelligent bus master add-in cards that bring advanced applications to PCs.

Since the EISA system is 100% compatible with the ISA 8-bit and 16-bit expansion boards and software, ISA cards can be plugged into the EISA connector slots. The EISA slots can be defined as ISA or EISA for ease of compatibility during configuration. The EISA connector is a superset of the ISA connector maintaining full compatibility with ISA expansion cards and software. Simultaneous use of EISA and ISA add-in boards is available with automatic system and expansion board configuration.

82350 EISA Chip Set Highlights

The Intel 82350 EISA chip set is the industry's first 100% EISA/ISA compatible chip set. The 82350

386™ and i486™ are trademarks of Intel Corporation.

*PC AT is a trademark of International Business Machine Corporation.

EISA chip set supports the 33 MHz and 25 MHz 386 CPU or i486 CPU, 82385 Cache Controller, and optional 80387 numerics coprocessor. The EISA chip set includes three chips:

82352 EISA Bus Buffers (EBB) (Optional)

82357 Integrated System Peripheral (ISP)

82358 EISA Bus Controller (EBC)

Information on the 82352 EBB device is located in a separate data sheet.

The ISP performs the DMA functions of the system and is fully compatible with ISA functions. It integrates seven 32-bit DMA channels, five 16-bit timer/counters, two eight channel interrupt controllers, and provides for multiple NMI control and generation. It provides refresh address generation and keeps track of pending refresh requests when the bus is unavailable. The ISP supports multiple EISA bus masters while offering intelligent system arbiter services which grant the bus on a rotational basis.

The EBC is the EISA "engine". It is an intelligent bus controller that controls 8, 16 and 32-bit bus masters and slaves. It provides the state machine interface to Host, ISA and EISA busses and other IC's in the chip set. It offers a simple interface to the 386/i486 CPU and EISA bus. The EBC services as a bridge between the EISA and ISA devices. Data bus size mismatches are handled automatically by the EBC (including byte assembly and disassembly). It also guarantees cache operation on the Host, EISA, and ISA busses.

More information on EBC and ISP devices can be found in the data sheets in this document.

The 82355 Bus Master Interface Chip (BMIC) is a new device for add-in cards that takes advantage of the EISA bus master capabilities. Information on the 82355 BMIC is located in a separate data sheet.

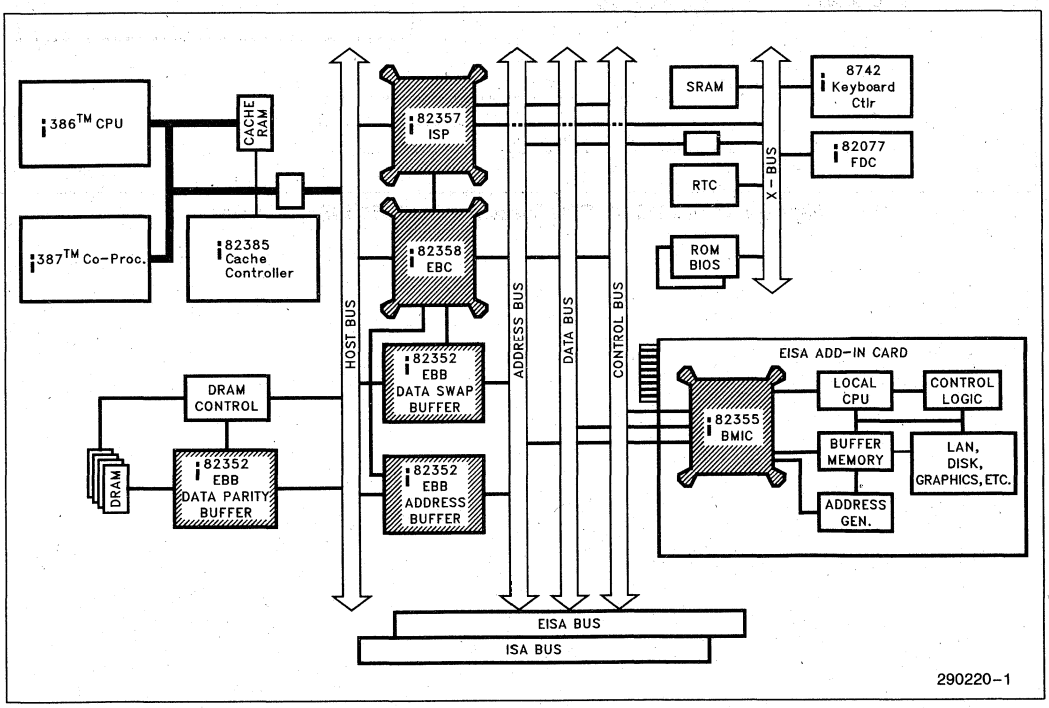


Figure 1. Intel's 386™ CPU System with 82350 EISA Chip Set

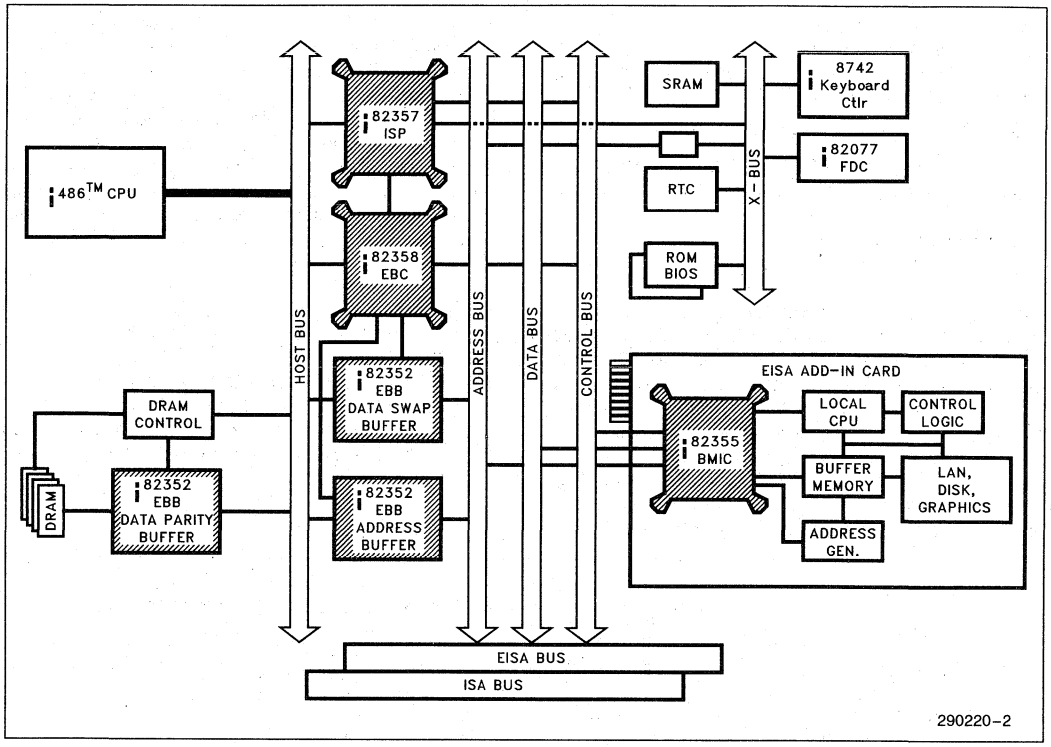


Figure 2. Intel's i486™ CPU System with 82350 EISA Chip Set

MECHANICAL DATA

PACKAGING INFORMATION

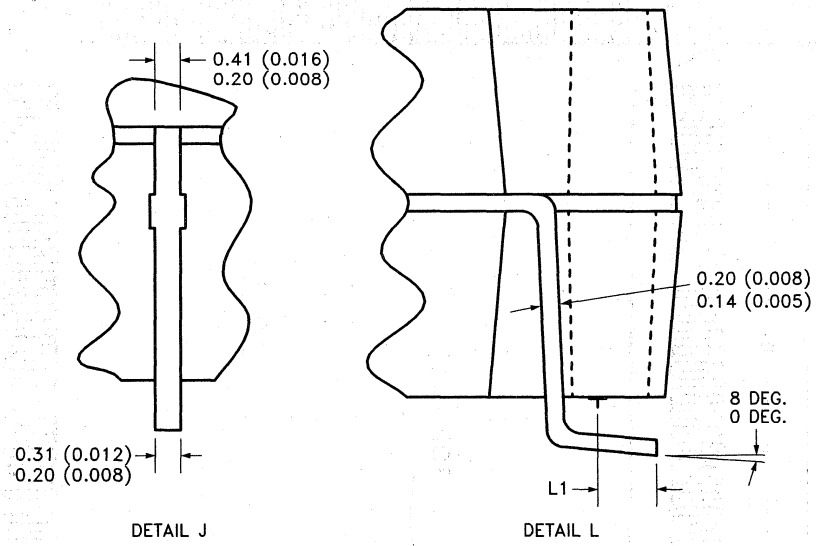
(See Packaging Spec. Order # 231369)

Introduction

The individual components of Intel's EISA Chip Set come in JEDEC standard Gull Wing packages (25 MIL pitch), with "bumpers" on the corners for ease of handling. Please refer to the accompanying table for the package associated with each device, and to the individual component specifications for pinouts. (Note that the individual pinouts are numbered consistently with the numbering scheme depicted in the accompanying figures).

TYPICAL LEAD

Case Outline Drawings
Plastic Fine Pitch Chip Carrier
0.84 mm Pitch

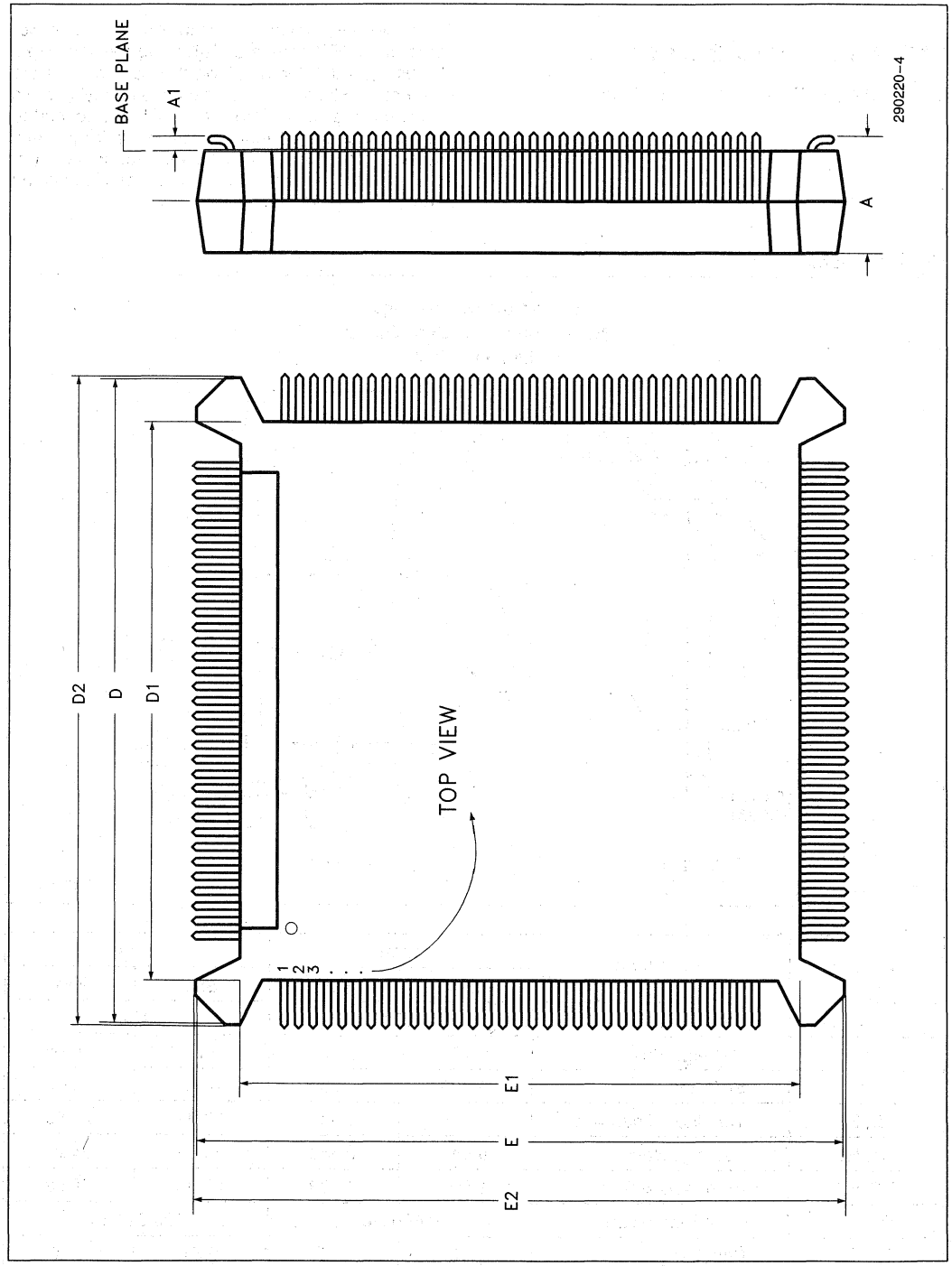


mm (inch)

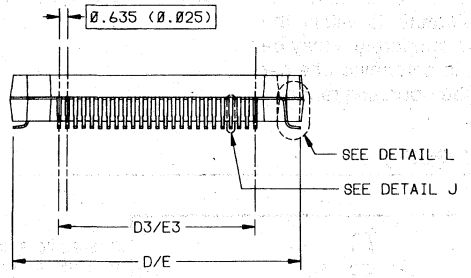
290220-3

Symbol	Description	Inch		mm	
		Min	Max	Min	Max
N	Lead Count	132		132	
A	Package Height	0.160	0.170	4.06	4.32
A1	Standoff	0.020	0.030	0.51	0.76
D, E	Terminal Dimension	1.075	1.085	27.31	27.56
D1, E1	Package Body	0.947	0.953	24.05	24.21
D2, E2	Bumper Distance	1.097	1.103	27.86	28.02
D3, E3	Lead Dimension	0.800 Ref		20.32 Ref	
L1	Foot Length	0.020	0.030	0.51	0.76

PRINCIPAL DIMENSIONS & DATUMS



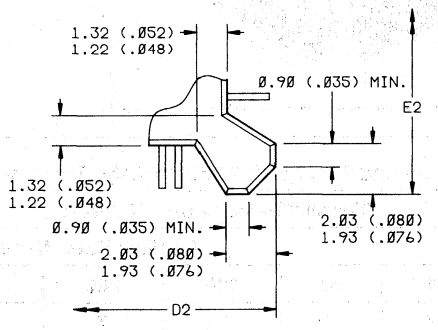
TERMINAL DETAILS



290220-5

mm (inch)

BUMPER DETAIL



290220-6

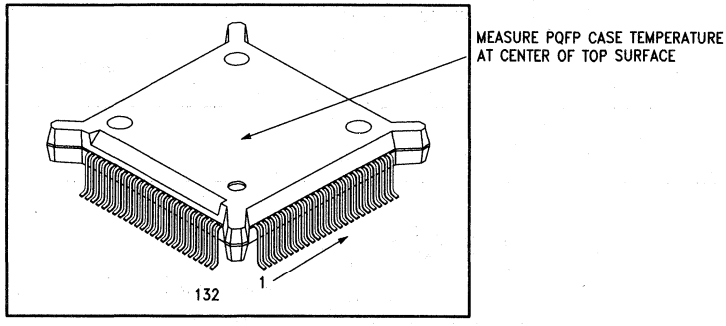
mm (inch)

Package Thermal Specification

The 82357 ISP and 82358 EBC are specified for operation when the case temperature is within the range of 0°C to 85°C. The case temperature may be measured in any environment, to determine whether the device is within the specified operating range.

The PQFP case temperature should be measured at the center of the top surface opposite the pins, as shown in the figure below.

PLASTIC QUAD FLAT PACK (PQFP)



290220-7

Table 2. 82357 ISP and 82358 EBC PQFP Package Thermal Characteristics

Thermal Resistance— °C/Watt							
Parameter	**Air Flow Rate (ft/min)**						
	0	**50**	**100**	**200**	**400**	**600**	**800**
θ Junction—Case	7	7	7	7	7	7	7
θ Case to Ambient	22	21	19.5	17.5	14.5	12	10

NOTES:
1. Table 2 applies to the PQFP device plugged into a socket or soldered directly into the board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$.

PROCESS NAME:
1.2μ CHMOS III P-well

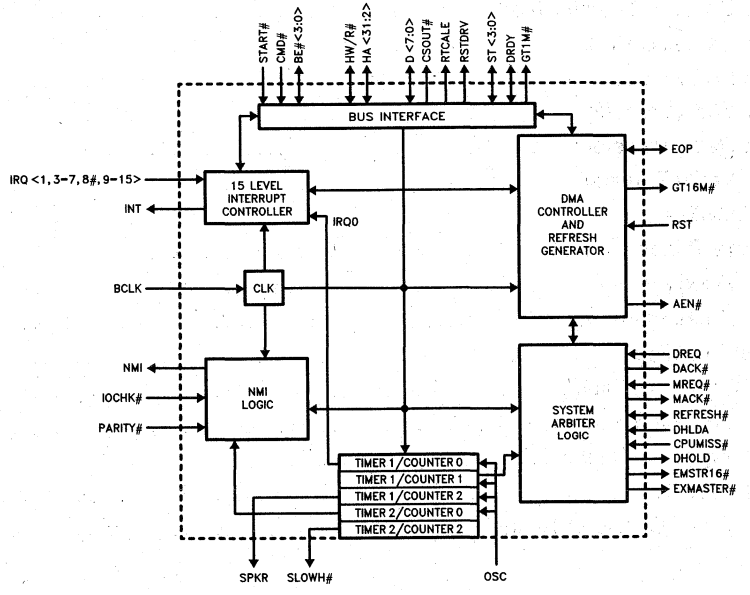
I_{CC} AT HOT WITH NO RESISTIVE LOADS:
150 mA Max at 85°C.

82357
INTEGRATED SYSTEM PERIPHERAL (ISP)

- **Provides Enhanced DMA Functions**
  - **— ISA/EISA DMA Compatible Cycles**
  - **— All Transfers are Fly-By Transfers**
  - **— 32-Bit Addressability**
  - **— Seven Independently Programmable Channels**
  - **— Provides Timing Control for 8-, 16-, and 32-Bit DMA Data Transfers**
  - **— Provides Timing Control for Compatible, Type "A", Type "B", and Type "C" (Burst) Cycle Types**
  - **— 33 Mbytes/sec Maximum Data Transfer Rate**
  - **— Provides Refresh Address Generation**
  - **— Supports Data Communication Devices and Other Devices That Work from a Ring Buffer in Memory**
  - **— Incorporates the Functionality of Two 82C37A DMA Controllers**
- **Provides High Performance Arbitration**
  - **— For CPU, EISA/ISA Bus Masters, DMA Channels, and Refresh**
- **Incorporates the Functionality of Two 82C59A Interrupt Controllers**
  - **— 14 Independently Programmable Channels for Level-or-Edge Triggered Interrupts**
- **Five Programmable 16-Bit Counter/Timers**
  - **— Generates Refresh Request Signal**
  - **— System Timer Interrupt**
  - **— Speaker Tone Output**
  - **— Fail-Safe Timer**
  - **— Periodic CPU Speed Control**
  - **— 82C54 Programmable Interval Timer Compatible**
- **Provides Logic for Generation/Control of Non-Maskable Interrupts**
  - **— Parity Errors for System and Expansion Board Memory**
  - **— 8 μs and 32 μs Bus Timeout**
  - **— Immediate NMI Interrupt via Software Control**
  - **— Fail-Safe Timer**
- **132-Pin PQFP Package**
  (See Packaging Spec: Order #231369)

1

82357 Internal Block Diagram



290253-78

386™ and i486™ are Trademarks of Intel Corporation.

September 1990
Order Number: 290253-003

CONTENTS

PAGE

82357 INTEGRATED SYSTEM PERIPHERAL 1-183
FUNCTIONAL HIGHLIGHTS 1-183

1.0 ISP SYSTEM INTERFACE ILLUSTRATION 1-189

2.0 FUNCTIONAL OVERVIEW 1-189
2.1 Master and Slave Modes 1-189
2.2 DMA Controller 1-189
2.3 System Arbiter 1-190
2.3.1 Priorities of Requesters 1-191
2.3.2 Preemption of EISA Masters and DMA Devices 1-191
2.3.3 Arbitration During Non-Maskable Interrupts 1-192
2.3.4 DREQ and DACK Latency Control 1-192
2.4 Refresh Generation 1-192
2.5 Interrupt Controller 1-193
2.6 Non-Maskable Interrupt (NMI) 1-194
2.7 Interval Timers 1-195
2.8 Register Access 1-196

3.0 DMA CONTROLLER 1-196
3.1 DMA Transfer Modes 1-196
3.1.1 Single Transfer Mode 1-196
3.1.2 Block Transfer Mode 1-197
3.1.3 Demand Transfer Mode 1-197
3.1.4 Cascade Mode 1-197
3.2 Transfer Types 1-197
3.2.1 Write Transfer 1-197
3.2.2 Read Transfer 1-197
3.2.3 Verify Transfer 1-197
3.3 Autoinitialize 1-197
3.4 Channel Priority 1-198
3.4.1 Fixed Priority 1-198
3.4.2 Rotating Priority 1-198
3.5 Buffer Chaining 1-198
3.6 DMA Enhanced Timing 1-200
3.6.1 Compatible Timing 1-200
3.6.2 Type "A" Timing 1-200
3.6.3 Type "B" Timing 1-200
3.6.4 Type "C" (Burst Timing) 1-200

CONTENTS (Continued)

PAGE

3.7 Register Description ... 1-200
3.7.1 Stop Registers (Ring Buffer Data Structure) ... 1-200
3.7.2 DMA Memory Low Page Register ... 1-202
3.7.3 DMA Memory High Page Register ... 1-202
3.7.3.1 Address Compatibility Mode ... 1-203
3.7.4 Current Address Register (READ/WRITE) ... 1-203
3.7.4.1 Address Shifting when Programmed for 16-Bit I/O Count By Words ... 1-203
3.7.5 Current Word Register (READ/WRITE) ... 1-204
3.7.6 Base Page, Base Address and Base Word Count Registers ... 1-204
3.7.7 Command Register ... 1-204
3.7.8 Mode Register ... 1-205
3.7.9 Extended Mode Registers (WRITE ONLY) ... 1-205
3.7.9.1 8-Bit I/O, "Count By Bytes" Mode ... 1-206
3.7.9.2 16-Bit I/O, "Count By Words" (Address Shifted) Mode ... 1-206
3.7.9.3 16-Bit I/O, "Count By Bytes" Mode ... 1-206
3.7.9.4 32-Bit I/O "Count By Bytes" Mode ... 1-206
3.7.9.5 EOP Input/Output Selection ... 1-206
3.7.9.6 Stop Register Selection ... 1-206
3.7.9.7 Summary of the DMA Transfer Sizes ... 1-206
3.7.10 Request Register (Write Only) ... 1-207
3.7.11 Mask Register ... 1-207
3.7.12 Status Register ... 1-208
3.7.13 Set Chaining Mode Register ... 1-209
3.7.14 Set Chaining Mode Status Register ... 1-210
3.7.15 Channel Interrupt Status Register ... 1-210
3.7.16 Chain Buffer Expiration Control Register ... 1-211
3.8 Software Commands ... 1-211
3.8.1 Clear Byte Pointer Flip-Flop ... 1-211
3.8.2 Master Clear ... 1-212
3.8.3 Clear Mask Register ... 1-212
3.9 Terminal Count/EOP Summary ... 1-212

4.0 BUS ARBITRATION ... 1-213
4.1 Bus Timeout ... 1-213
4.1.1 8 μs Bus Timeout ... 1-213
4.1.2 32 μs Bus Timeout ... 1-213
4.2 4 μs Limit on DMA Transfers ... 1-213
4.3 EISA Bus Master Status Latch ... 1-214

1

CONTENTS (Continued)

PAGE

5.0 INTERRUPT CONTROLLER 1-215
5.1 Interrupt Controller I/O Address Map 1-215
5.2 Interrupt Definition Table 1-216
5.3 Interrupt Request Register (IRR) and In-Service Register (ISR) 1-216
5.4 Priority Resolver 1-216
5.5 Interrupt Mask Register (IMR) 1-216
5.6 INT (Interrupt) 1-216
5.7 INTA#/(ST2#) (Interrupt Acknowledge) 1-216
5.8 Interrupt Sequence 1-217
5.9 80 x 86 Mode 1-217
5.10 Programming the Interrupt Controller 1-218
5.11 Initialization Command Words 1-219
5.11.1 Initialization Command Words 1 and 2 (ICW1, ICW2) 1-219
5.11.2 Initialization Command Word 3 (ICW3) 1-220
5.11.3 Initialization Command Word 4 (ICW4) 1-221
5.12 Operation Control Words (OCWS) 1-222
5.12.1 Operation Control Word 1 (OCW1)—Read/Write 1-222
5.12.2 Operation Control Word 2 (OCW2) 1-222
5.12.3 Operation Control Word 3 (OCW3) 1-223
5.13 End-Of-Interrupt Operation 1-224
5.13.1 End Of Interrupt (EOI) 1-224
5.13.2 Automatic End Of Interrupt (AEOI) Mode 1-224
5.14 Modes of Operation 1-224
5.14.1 Fully Nested Mode 1-224
5.14.2 The Special Fully Nested Mode 1-224
5.14.3 Automatic Rotation (Equal Priority Devices) 1-225
5.14.4 Specific Rotation (Specific Priority) 1-225
5.14.5 Poll Command 1-225
5.14.6 Cascade Mode 1-226
5.14.7 Edge and Level Triggered Modes 1-226
5.14.7.1 Edge/Level Triggered Control Register (ELCR) 1-226
5.15 Interrupt Masks 1-227
5.15.1 Masking on an Individual Interrupt Request Basis 1-227
5.15.2 Special Mask Mode 1-227
5.16 Reading the Interrupt Controller Status 1-227

CONTENTS (Continued)

PAGE

6.0 NON-MASKABLE INTERRUPT PORTS 1-228
6.1 NMI Status and Control (Port 061h) 1-228
6.2 NMI Extended Status and Control (Port 0461h) 1-229
6.3 Software NMI Generation (Port 0462h) 1-230
6.4 NMI Enable/Disable and Real-Time Clock Address (Port 070h) 1-230

7.0 INTERVAL TIMER 1-230
7.1 Programming the Interval Timer 1-230
7.1.1 Interval Timer Control Word Format 1-231
7.1.2 Interval Timer Control Latch Command 1-231
7.1.3 Interval Timer Read Back Command 1-232

8.0 DETAILED SIGNAL DESCRIPTION 1-233
8.1 Signals Used during Arbitration 1-233
8.2 Signals Used during DMA and Register Access 1-235
8.3 Signals Used during an Interrupt Sequence 1-239
8.4 Counter/Timer, NMI and Miscellaneous Utility Signals 1-239

9.0 BASIC FUNCTION TIMING DIAGRAMS 1-241
9.1 Slave Mode 1-241
9.1.1 EISA Master or Host CPU Cycle to/from ISP 1-241
9.1.2 ISA Master Slave Cycle to/from the ISP—Short Cycle 1-242
9.1.3 ISA Master Slave Cycle to/from the ISP—Long Cycle 1-243
9.2 DMA Master Mode 1-244
9.2.1 Non-Burst DMA Cycles 1-244
9.2.2 Burst DMA Cycle Diagram 1 1-245
9.2.3 Burst DMA Cycle Diagram 2 1-246
9.2.4 Burst DMA Cycle Diagram 3 1-247
9.2.5 EISA Refresh Cycle 1-247
9.3 Arbiter Timing Diagrams 1-248
9.3.1 Arbiter Timing Diagram 1 1-248
9.3.2 Arbiter Timing Diagram 2 1-249
9.3.3 Arbiter Timing Diagram 3 1-250
9.3.4 Arbiter Timing Diagram 4 1-251
9.3.5 Arbiter Timing Diagram 5 1-252
9.3.6 Arbiter Timing Diagram 6—CIP# (ST3#) Timing 1-253

10.0 D.C. SPECIFICATIONS 1-254
10.1 Maximum Ratings 1-254
10.2 D.C. Specification Tables 1-254

1

CONTENTS (Continued)

PAGE

11.0 A.C. SPECIFICATIONS 1-255
11.1 A.C. Characterization Tables 1-255
11.2 A.C. Characteristic Waveforms 1-259
11.2.1 BCLK, OSC Timing 1-259
11.2.2 Slave Mode Timing 1-260
11.2.3 DMA Master Timing 1-261
11.2.4 Arbiter Timing 1-262
11.2.5 Refresh Timing 1-263
11.2.6 Reset Timing 1-263
11.2.7 NMI Timing 1-263
11.2.8 Interrupt Timing 1-264
11.2.9 Counter/Timer Timing 1-264
11.2.10 RTCALE, GT1M# Timing 1-264

12.0 ISP PIN AND PACKAGE INFORMATION 1-265
12.1 Signal Overview 1-265
12.2 132 Package 1-266
12.3 Device Pinout 1-267

13.0 ISP PORT ADDRESS / I/O DECODING 1-268

The 82357 is a multi-function support peripheral that is designed to work in conjunction with the 82358 EISA Bus Controller to provide most of the system functions necessary in EISA specific applications.

The 82357 is comprised of several computer system functions that are typically found in separate LSI and VLSI components. These include: a high performance seven-channel programmable DMA Controller; an arbitration scheme that allows efficient bus sharing among multiple EISA masters, the host CPU, and DMA devices; a 16 level programmable interrupt controller which provides level-or-edge triggered interrupt capability on a channel-by-channel basis; non-maskable interrupt logic for multiple NMI control and generation; refresh address generation and control; and five counter/timers which provide a system timer interrupt, diskette time-out, DRAM refresh requests, and other system timing operations.

The DMA controller on the 82357 provides the timing control signals necessary to support a DMA data transfer rate of 33 Mbytes/sec. The DMA controller includes full function 32-bit addressability with control signal support for the transfer of data between devices of different data path widths using a single channel. Each channel functions independently in several modes.

1.0 ISP SYSTEM INTERFACE ILLUSTRATION

The ISP connects to the Host bus, EISA bus, X bus and EBC (Bus Controller). These connections are illustrated in Figure 1-1.

2.0 FUNCTIONAL OVERVIEW

The following is a brief discussion of the functionality and features of the 82357. The DMA Controller, Arbiter, Interrupt controller, NMI's, and Timer/Counters each have a corresponding detailed section later in this data sheet.

2.1 Master and Slave Modes

The 82357 is either a slave device or a master device.

In slave mode, the ISP monitors the address lines and decodes all bus cycles attempting to read or write any of its internal registers. In slave mode, either an EISA master or the host CPU can read or write to any of the ISP's internal registers. 16-bit ISA masters can read or write to any of the ISP's 82C37 PCAT compatible registers. The registers that cannot be accessed by an ISA master are located in the I/O space of 00H–0FH and 0C0H–0DFH. The ISP will disable these registers upon granting the ISA master the bus. In slave mode, the ISP also detects and responds to interrupt acknowledge cycles.

In master mode, the 82357 becomes the master of the bus system. It may peform either DMA cycles or refresh cycles at this time.

The arbiter on the ISP determines which mode the device is in.

1

2.2 DMA Controller

The DMA circuitry incorporates the functionality of two 82C37 DMA controllers with seven independently programmable channels, (Channels 0–3 and Channels 5–7). DMA Channel 4 is used to cascade the two controllers together and will default to cascade mode in the Mode register. In addition to accepting requests from DMA slaves, the DMA also responds to requests that are initiated by software. Software may initiate a DMA service request by setting any DMA channel Request register bit to a 1.

Any DMA channel may be programmed for 8-, 16-, or 32-bit DMA device size and ISA compatible, Type "A", Type "B", or burst DMA Type "C" modes. The 82357 provides the timing controls and the 82358 EISA Bus Controller performs the data size translations necessary for the DMA transfer. The DMA memory addressing circuitry supports full 32-bit addresses for DMA devices. Each channel includes a 16-bit ISA compatible Current register which holds the 16 least-significant bits of the 32-bit address, a Low Page register which contains the eight second most significant bits, and a High Page register which contains the eight most significant bits of the 32-bit address.

The channels can also be programmed for any of four transfer modes. The transfer modes include single, block, demand, or cascade. Each of the three active transfer modes, (single, block, and demand), can perform three different types of transfers, (read, write, or verify). The DMA Controller also features refresh address generation, buffer chaining, auto-initialization, and support for a Ring Buffer Data Structure in memory. Stop registers are used to help support Data Communication or devices that work from a Ring Buffer in memory (refer to Section 3.7.1 "USE OF STOP REGISTERS").

The DMA controller is at any time either in master mode or slave mode. In master mode, the DMA controller is either servicing a DMA slave's request for DMA cycles, generating refresh cycles, or allowing a 16-bit ISA master to use the bus via a cascaded DREQ signal. In slave mode, the 82357 monitors the bus and decodes and responds to I/O read and write commands that address its registers.

When the DMA is in master mode and servicing a DMA slave, it works in conjunction with the 82358 EISA bus controller to create bus cycles on the system bus. The DMA places addresses and the memory read/write (HW/R#) signal on the host CPU bus. It instructs the bus controller when to start and what type of bus cycle to run with the ST0 and ST1 lines. The bus controller informs the DMA when to place a new address on the bus with the DRDY signal.

2.3 System Arbiter

The system arbiter evaluates requests for the bus coming from several sources which include the DREQ lines (DMA channels), MREQ# lines, refresh requests (Timer 1 Counter 1), and the host CPU (CPUMISS#). The DREQ lines are used by 8-, 16-, or 32-bit DMA slave devices to request DMA service and by existing 16-bit ISA masters to request the bus; the MREQ# lines are used by new 16- or 32-bit

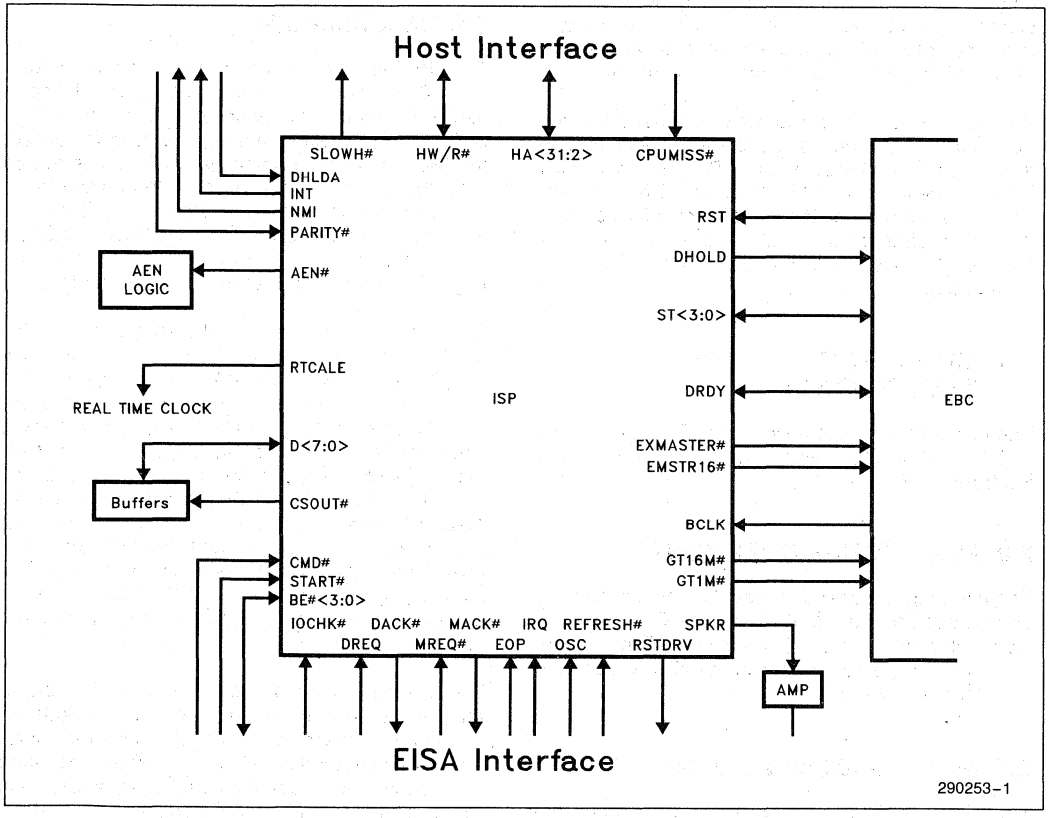


Figure 1-1. ISP System Interface

EISA masters for general bus request; the CPUMISS# line is used by the host CPU when requesting bus access; and Timer 1 Counter 1 specifically is used to generate internal requests for refresh. The default master of the bus is the CPU. It is granted the bus if there are no requesters.

The DMA channels, when used by 16-bit ISA masters for direct access to the bus, must be placed in Cascade mode. In Cascade mode, the DMA controller will respond to DREQ with DACK# and EXMASTER16#, but HW/R#, ST0, ST1, A<31:2>, and BE<3:0># outputs will be disabled.

2.3.1 PRIORITIES OF REQUESTERS

The ISP uses a three way rotating priority arbitration method. At each level, the devices which are considered equal are given a rotating priority. On a fully loaded bus, the order in which the devices are granted bus access is independent of the order in which they assert a bus request, since devices are serviced based on their position in the rotation. The arbitration scheme assures that DMA channels access the bus with minimal latency. As an example, in a system with DMA-2, a CPU, and two masters, all requesting the bus continuously, the grant sequence would be as follows:

DMA-2 CPU DMA-2 Master-1 DMA-2 CPU DMA-2 Master-2 and Repeat.

The priorities and the assignments are as shown in Figure 2-1.

The DREQ lines can be placed in either fixed or rotating priority. The default mode is fixed and by programming the Command registers, the priority can be modified for rotate mode. The MREQ# lines are placed in rotate mode and cannot be changed.

An example of programming the Command register to rotating priority is shown in Figure 2-2.

2.3.2 Preemption of EISA Masters and DMA Devices

An EISA bus master or a DMA slave device (that is not programmed for compatible timing) will be preempted from the bus by another device that requests use of the bus. This will occur regardless of the priority of the pending request. An EISA bus master must release the bus within 64 BCLKs (8 μs) after the arbiter drives its MACK# line inactive. If the bus is not released within the allowable time, a bus timeout (NMI) is generated and the RESDRV signal is driven active to reset the offending bus master. For DMA devices not using compatible timing mode, the DMA controller stops the DMA transfer and releases the bus within 32 BCLK (4 μs) of a preemption.

If the main CPU is currently using the bus and another device requests use of the bus, the ISP will immediately drive HOLD active and wait for the CPU or CACHE to drive DHLDA active. The CPU will be allowed to hold the bus as long as required to finish its current cycle without a 8 μs timeout (NMI) being generated. However, if the slave device that the CPU is addressing does not release EXRDY or CHRDY within 32 μs (256 BCLKs), a timeout (NMI) will be generated (refer to the following paragraph).

A slave which does not release EXRDY or CHRDY can cause the CMD# active time to exceed 32 μs (256 BLCKs). To prevent the system from locking up, a bus timeout (NMI) will be generated. The counting of the 256 BLCKs starts at the request for the bus from another source. This will allow Burst cycles to run unimpeded if there are no pending requests.

16-bit ISA masters using the DREQ lines and cascaded DMA channels cannot be preempted from the bus via the ISP unless CMD# active exceeds 32 μs (256 BLCKs).

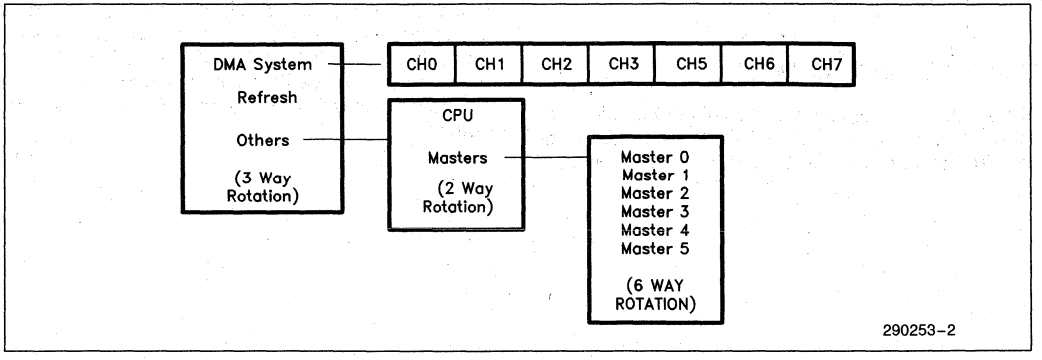


Figure 2-1. Arbitration Priority

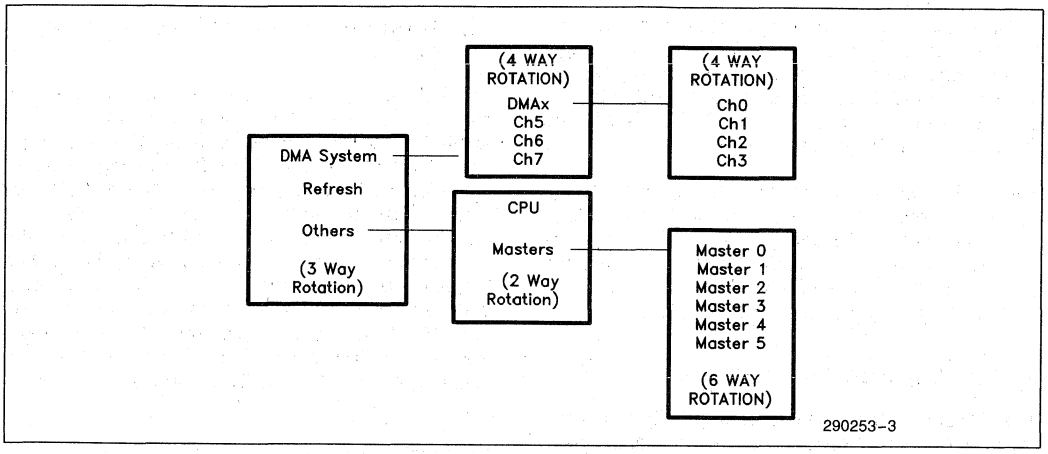


Figure 2-2. DMA Channels in Rotating Priority

2.3.3 ARBITRATION DURING NON-MASKABLE INTERRUPTS

If a non-maskable interrupt (NMI) is pending at the CPU, and the CPU is requesting the bus, then the external bus masters and the DMA controller will be by-passed each time they come up for rotation. This will give the CPU the bus bandwidth it requires to process the interrupt as fast as possible. The bus masters will still execute if the CPU is able to execute out of a local cache and does not require the bus.

2.3.4 DREQ AND DACK LATENCY CONTROL

The ISP arbiter maintains a minimum DREQ to DACK latency on DMA channels programmed to operate in compatible timing mode. This is to support older devices such as the 8272A. This is implemented as an eight BCLK delay for DREQs coming into the Priority resolution logic. The DREQs are effectively delayed by eight BCLKs prior to being seen by the arbiter logic. Software requests will not have this minimum request to DACK latency.

2.4 Refresh Generation

The refresh system uses the combined functions of the Interval Counter/Timers, Arbiter, and DMA. Interval Counter 1/Timer 1 generates an internal refresh request, the Arbiter detects a Refresh signal from either the Counter/Timer or the REFRESH# input and determines when the refresh will be done, and the DMA drives the refresh address out onto the host bus. Counter 1 Timer 1 should be programmed to provide a request for refresh about every 15 μs.

Requests for refresh cycles are generated by two sources: Timer 1 Counter 1 and 16-bit ISA masters that activate REFRESH# when they have bus ownership. EISA bus masters need not supply refresh cycles since the refresh controller can preempt the bus master and perform the necessary refresh cycles. 16-bit ISA bus masters that hold the bus longer than 15 μs must supply memory refresh cycles.

Each time an internal refresh request is not serviced within the normal 15 μs interval, a counter is incremented. The counter counts up to four incomplete refresh requests. When a request cycle occurs, the pending refresh counter is decremented. Only one refresh cycle will be run and the bus will be released to the requester with the highest priority. If more refreshes are queued up, the bus will immediately be arbitrated for again, without waiting for the normal 15 μs interval. If a refresh request is sensed while four refresh requests are pending, the incoming refresh request will be dropped. The bus is requested whenever one or more pending refreshes are recorded.

The DMA controller drives the refresh address out onto the LA<15:2> address bus (14 bits of refresh address) and also enables the BE<3:0># lines so that they can be translated to SA<1:0> lines. The High and Low Page register contents will also be placed on the LA<31:16> bus during refresh. The refresh cycle lasts from the leading edge of START# through the rising edge of CMD# (two BCLKs) unless wait states are added by the memory slave negating EXRDY (EISA slaves) or CHRDY (ISA slaves). The 82358 bus controller, upon seeing REFRESH#, knows to run refresh cycles instead of DMA cycles.

The refresh address bit order on the LA<15:2> and SA<15:0> bus is as follows:

13	12	11	10	9	8	1	0	7	6	5	4	3	2	1	0	Refresh Counter Bits
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LA< >, SA< > Addresses

The 14-bit refresh counter will be reset to 0 upon RST. The Page register is located at I/O address 08FH and can be either read or written. The refresh High Page register is located at I/O address 048FH and can also be either read or written. When writing to the Low Page register, the upper Page register is not cleared to zero.

2.5 Interrupt Controller

The 82357 provides an ISA compatible interrupt controller which incorporates the functionality of two 82C59 interrupt controllers. The two controllers are cascaded so that 14 external and two internal interrupts are possible. The master interrupt controller provides IRQ <7:0> and the slave interrupt controller provides IRQ <15:8#> (see Figure 2-3). The two internal interrupts are used for internal functions only and are not available externally. IRQ2 is used to cascade the two controllers together and IRQ0 is used as a system timer interrupt and is tied to interval Timer 1, Counter 0. The remaining 14 interrupt lines (IRQ1, IRQ3–IRQ15) are available for external system interrupts. IRQ13 is also shared internally with the chaining interrupt as well as being available for external use.

In addition to the ISA features, the ability to do interrupt sharing is included. A register is defined (ELCR) which allows an edge and level sense selection to be made on an individual channel by channel basis instead of on a complete bank of channels. Only the interrupt channels that connect to the EISA/ISA bus may be programmed for level sensitive mode. IRQ (0, 1, 2, 8#, 13) must be programmed for edge sensitive operation. IRQ8# is active low edge sensitive only.

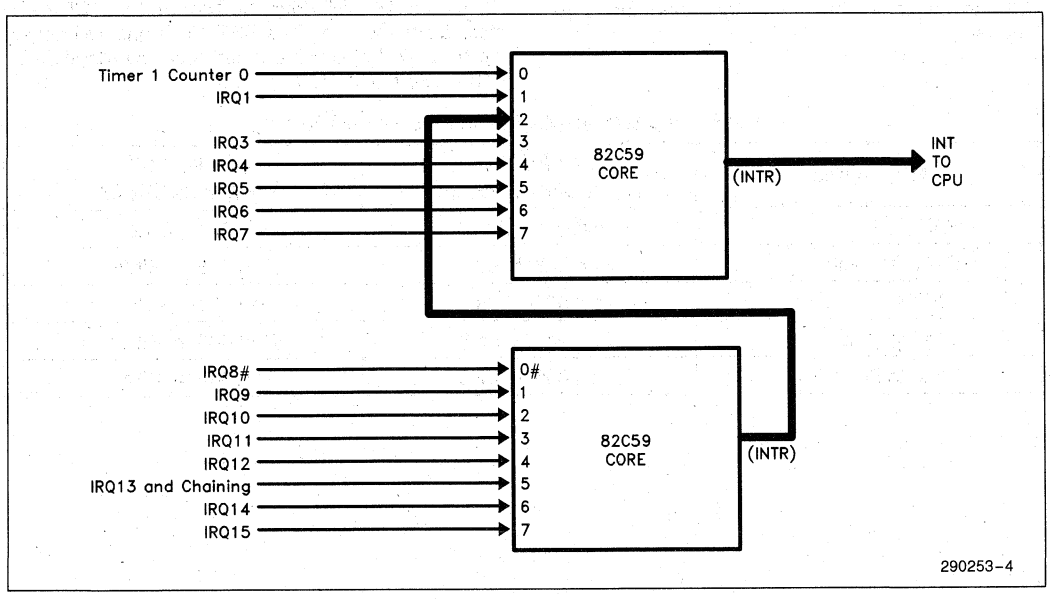


Figure 2-3. Block Diagram of the Interrupt Controller

2.6 Non-Maskable Interrupt (NMI)

An NMI is an interrupt requiring immediate attention and has priority over the normal interrupt lines (IRQX). The 82357 indicates error conditions by generating a non-maskable interrupt. An NMI can be caused by either a hardware or software mechanism.

NMI interrupts are caused by the following conditions:

1. Parity errors on the mother board memory. The system board reports any parity errors from its memory system on the PARITY# line.
2. Parity errors on the add-in memory boards on the ISA expansion bus. IOCHK# is driven low when this error occurs.
3. Timeout of the fail-safe timer Counter 0 on the interval Timer 2 used to prevent the system from locking up. This NMI is sensed with a rising edge detect latch.
4. Timeout of an 8 μs 32-bit "bus" master timeout. If a 32-bit bus master retains the bus more than 8 μs after MACK# goes inactive, the 82357 will drive the NMI and RESDRV signals active together. The RESDRV signal will remain active until the NMI has been reset.

NOTE:
An NMI will not be generated in the case of the CPU holding onto the bus longer than the 8 μs timeout.

5. Timeout of the 32 μs CMD# active timer.
6. Software writing to the NMI I/O interrupt port (0462H). This is a special port which, when written, causes an immediate NMI interrupt, provided port 070h is enabled.

The NMI logic incorporates four different 8-bit registers. These registers are addressed as Port(061h), Port(070h), Port(0461h), and Port(0462h). The status of Ports (0461h) and (061h) are read by the CPU to determine which source caused the NMI. Bits set to 1 in these ports show which device requested an NMI interrupt. After the NMI interrupt routine processes the interrupt, the NMI status bits are cleared by the software. This is done by setting the corresponding enable/disable bit high. Port(070h) is the mask register for the NMI interrupts. This register can mask the NMI signal and also disable or enable all NMI sources. Writing to Port(0462h) with any data will cause an immediate NMI interrupt if enabled.

If it is desired to reset the system bus without resetting other devices in the system (standard system board devices are not reset), the Port(0461h) Bit <0> can be written with a (1). This bit should be held in this state for the desired RSTDRV active time and then returned to its normal state (0).

If a 32-bit bus master tries to hold the bus beyond the 8 μs limit, or if CMD# is active for more than 32 μs, the ISP will drive the NMI and RSTDRV signals active together. The RSTDRV signal will remain active until the NMI has been reset by resetting bit <3> to 0 in I/O Port 0461h.

NMI Source Enable/Disable and Status Port Bits

NMI Source	IO Port Bit for Status Reads	IO Port Bit for Enable/Disable
PARITY#	Port 061h, Bit 7	Port 061h, Bit 2
Fail Safe Timer	Port 0461h, Bit 7	Port 0461h, Bit 2
IOCHK#	Port 061h, Bit 6	Port 061h, Bit 3
Bus Timeout	Port 0461h, bit 6	Port 0461h, Bit 3
Write to Port 0462h	Port 0461h, Bit 5	Port 0461h, Bit 1

To ensure that all NMI requests are serviced, the NMI service routine software needs to incorporate a few very specific requirements. These requirements are due to the edge detect circuitry of the host microprocessor, 80386 or 80486. The software flow would need to be the following:

1. NMI is detected by the processor on the rising edge of the NMI input.
2. The processor will read the status stored in Ports 061h, and 0461h to determine what sources caused the NMI. The processor may then reset the register bits controlling the sources that it has determined to be active. Between the time the processor reads the NMI sources and resets them, an NMI may have been generated by another source. The level of the NMI will then remain active. This new NMI source will not be recognized by the processor because there was no edge on NMI.
3. The processor must then disable all NMI's by writing Bit <7> of Port 070h high and then enable all NMI's by writing Bit <7> of Port 070h low. This will cause the NMI output to transition low then high if there are any pending NMI sources. The CPU's NMI input logic will then register a new NMI.

2.7 Interval Timers

The 82357 contains five counter/timers that are equivalent to those found in the 82C54 programmable interval timer. The counter timers are addressed as though they are contained in two separate 82C54 timers. Timer 1 contains three counters and Timer 2 contains two counters. Counter 1 of Timer 2 is not implemented in EISA systems. Each timer provides three frequencies or counters for the system. The 8 MHz counters use BCLK for a clock source; the others use a division of the 14.31818 MHz OSC input. The 14.31818 MHz OSC input is either divided by 12 or 48 to provide the necessary frequencies.

The outputs of the timers are directed to key system functions. Interval Timer 1, Counter 0 is connected to the interrupt controller IRQ0 and provides a system timer interrupt for a time-of-day, diskette time-out, or other system timing functions. Counter 1 generates a refresh-request signal and Counter 2 generates the tone for the speaker.

Interval Timer 2, Counter 0 is the fail-safe timer that can generate NMI interrupts on the NMI line at regular intervals as a means of preventing the system from locking up. Counter 1 is not implemented. Counter 2 is used to slow down the CPU by means of pulse-width modulation and is tied to the SLOWH# output. To use the slow function, the counter must be programmed. If the counter is not programmed, SLOWH# will not go active. Counter 2 is placed in the one-shot mode and is triggered by the refresh request signal generated by Timer 1/ Counter 1 only. If the counter has been programmed, the Counter 2 output (SLOWH#) will stop the CPU for the programmed period of the one-shot every time a refresh request occurs. To enable the one-shot, select Mode 1 (one-shot) operation by writing to I/O address 4Bh a value of 92h.

NOTE:
Refresh cycles will not necessarily be generated during the time the SLOWH# signal is active, the Arbiter will determine when the refresh cycle will be placed on the bus.

Because the slow function depends upon the refresh-request frequency of another counter, chaining the refresh-request frequency will affect the period of Counter 2 output (SLOWH#) signal. Timer 2 Counter 2 is not configured for the one-shot mode and is not programmed for a counter value until a speed reduction in the system is required. At such time, the value programmed depends on the system speed desired.

Figure 2-4 lists the Interval Timer Functions.

Interval Timer Functions		
	Interval Timer 1	**Interval Timer 2**
Function	**Counter 0 System Timer**	**Counter 0 Fail-Safe Timer**
Gate	Always On	Always On
Clock In	1.193 MHz (OSC/12)	0.298 MHz (OSC/48)
Out	INT-1 IRQ0	NMI Interrupt
	Counter 1 Refresh Request	
Gate	Always On	
Clock In	1.193 MHz (OSC/12)	
Out	Refresh Request	
	Counter 2	**Counter 2**
Gate	Programmable Port 61h	Refresh Request
Clock In	1.193 MHz (OSC/12)	8 MHz (BCLK)
Out	Speaker	CPU Speed Control (SLOWH#)

Figure 2-4. Interval Timer Functions

2.8 Register Access

To the system, the ISP appears as an 8-bit EISA slave. The following signals are used to access the ISP's internal registers:

D<7:0>, HA<15:2> and BE<2:0>#, HW/R#, ST<2:0>, CMD#, START#, DRDY, and CSOUT#

The number of BCLKs required to complete an ISP access depends on the master accessing the ISP and the register being accessed. Non-ISA masters will always see a default 8-bit I/O cycle (6 BCLKs). An ISA master will either see an 8-bit I/O cycle without wait states (3 BCLKs) or an 8-bit I/O cycle with wait states (6 BCLKs) depending on the register being accessed.

Because the ISP does not have a designated output signal to instruct the master of the ISP's bus size, the 82358 bus controller will do all necessary bus conversions and use the 8-bit ISA timing to access the ISP registers. In order to lengthen bus cycles begun by ISA masters for slower ISP registers, the DRDY signal is used to activate the EBC's CHRDY signal. The EBC will combinatorially connect the DRDY signal to CHRDY if an 8-bit I/O is the target of an ISA master's cycle. This will ensure that the CMD# pulse width will be at least five BCLKs long for the slower registers.

3.0 DMA CONTROLLER

3.1 DMA Transfer Modes

The ISP DMA supports four transfer modes: Single, Block, Demand, and Cascade.

NOTE:
Memory to Memory Transfers are not supported by the ISP.

3.1.1 SINGLE TRANSFER MODE

In Single Transfer mode the device is programmed to make one transfer only. The word count will be decremented and the address decremented or incremented following each transfer. When the word count "rolls over" from zero to FFFFFFH, or an external End of Process (EOP) is encountered, a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so. If Chaining is enabled the next chain buffer will be enabled if available.

DREQ must be held active until DACK# becomes active in order to be recognized. If DREQ is held active throughout the single transfer, the bus will be released to the CPU after a single transfer. The bus will be immediately requested again, and, upon winning the bus, another single transfer will be performed. This allows other devices a chance to execute if they require the bus.

3.1.2 BLOCK TRANSFER MODE

In Block Transfer mode the device is activated by DREQ to continue making transfers during the service until a TC, caused by word count going to FFFFFFH, or an external EOP, is encountered. DREQ need only be held active until DACK becomes active. An Autoinitialization will occur at the end of the service if the channel has been programmed for it. In this mode, it is possible to lock out other devices for a period of time (including refresh) if the transfer count is programmed to a large number.

3.1.3 DEMAND TRANSFER MODE

In Demand Transfer mode the device is programmed to continue making transfers until a TC is encountered or an external EOP is encountered, or until DREQ goes inactive. Thus transfers may continue until the I/O device has exhausted its data capacity. After the I/O device has had a chance to catch up, the DMA service is reestablished by means of a DREQ. During the time between services when the system is allowed to operate, the intermediate values of address and word count are stored in the DMA controller Current Address and Current Word Count registers. A TC can cause an Autoinitialize at the end of the service if the channel has been programmed for it.

3.1.4 CASCADE MODE

This mode is used to cascade more than one DMA controller together for simple system expansion. This allows the DMA requests of the additional device to propagate through the priority network circuitry of the preceding device. The priority chain is preserved and the new device must wait for its turn to acknowledge requests. In this architecture, Channel 0 of the second controller (ch4) is used to cascade the first controller to provide a total of seven channels.

Cascade mode is also used to allow direct access of the system by 16-bit ISA bus masters. These devices use the DREQ and DACK signals to arbitrate for the system bus and then they drive the address and command lines to control the bus.

In Cascade Mode, the DMA controller will respond to DREQ with DACK# but the HW/R#, address, and ST0–ST3 outputs will be disabled.

Channel 4 is used to connect the second half of the DMA system, this channel is not available for any other purpose.

3.2 Transfer Types

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write and Verify.

3.2.1 WRITE TRANSFER

Write transfers move data from an I/O device to memory starting with the DMA activating HW/R# and the ST<3:0> lines, the Bus Controller will then activate IORC# and the appropriate EISA or ISA control signals to indicate a memory write depending on which bus the memory is determined to be.

3.2.2 READ TRANSFER

Read transfers move data from memory to an I/O device starting with the DMA deactivating HW/R# and activating the ST<3:0> lines, the Bus Controller will then activate IOWC# and the appropriate EISA or ISA control signals to indicate a memory read, depending on which bus the memory is determined to be.

3.2.3 VERIFY TRANSFER

Verify transfers are pseudo transfers. The DMA controller operates as in Read or Write transfers generating addresses, and producing TC, etc. However, no ST<3:0> signals are activated so the Bus Controller does not activate the memory and I/O control lines. Only the DACK lines will go active. Since no EISA cycles are broadcasted in this mode, the LA bus is not copied to the SA bus. Internally the DMA controller will count BCLKs so that the DACK lines have a defined pulse width. This pulse width is nine BCLKs long. If Verify transfers are repeated during Block or Demand DMA requests, each additional pseudo transfer will add eight BCLKs. The DACK lines will not be toggled for repeated transfers.

3.3 Autoinitialize

By programming a bit in the Mode register, a channel may be set up as an Autoinitialize channel. During Autoinitialize initialization, the original values of the Current page, Current address and Current Word Count registers are automatically restored from the Base Page, Address, and Word count registers of that channel following TC. The Base registers are loaded simultaneously with the Current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not set when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to perform another DMA service, without CPU intervention, as soon as a valid DREQ is detected.

3.4 Channel Priority

For priority resolution the DMA consists of two logical channel groups—channels 0–3 and channels 4–7. Each group may be in either Fixed or Rotate mode, as determined by the Command register.

3.4.1 FIXED PRIORITY

The initial fixed priority structure is as follows:

High priority	Low priority
(0, 1, 2, 3)	5, 6, 7

Channel 0 has the highest priority, then 1, 2, 3, 5, 6, and channel 7 has the lowest priority.

3.4.2 ROTATING PRIORITY

Rotation allows for "fairness" in priority resolution. The priority chain rotates so that the last channel serviced is assigned the lowest priority in the channel group (0–3, 5–7).

Channels 0–3 rotate as a group 4, and are always placed between Channel 5 and Channel 7 in the priority list.

Channels 5–7 rotate as a group of 4—this is, the three channels (5–7) plus the channel 0–3 group.

Figure 3-1 will demonstrate rotation:

	High	Low
Initial setting —		
Group 0–3 is in rotation mode, Group 4–7 is in fixed mode:	(0, 1, 2, 3)	5, 6, 7
After servicing channel 2 —	(3, 0, 1, 2)	5, 6, 7
After servicing channel 3 —	(0, 1, 2, 3)	5, 6, 7
Both groups in rotation mode:		
Initial setting —	(0, 1, 2, 3)	5, 6, 7
After servicing channel 0 —	5, 6, 7, (1, 2, 3, 0)	
After servicing channel 5 —	6, 7, (1, 2, 3, 0),	5
After servicing channel 6 —	7, (1, 2, 3, 0),	5, 6
After servicing channel 7 —	(1, 2, 3, 0),	5, 6, 7
(Notice that the first service caused double rotation)		

Figure 3-1

3.5 Buffer Chaining

The buffer chaining mode of a channel is useful for transferring data from a peripheral to several different areas of memory within one transfer operation (from the DMA device's viewpoint). This is accomplished by causing the DMA to interrupt the host CPU for more programming information (or signal an EISA master via TC, if an EISA master needs to program the DMA) while the previously programmed transfer is still in progress. Upon completion of the previous transfer, the DMA controller will then load the new transfer information automatically. In this way, the entire transfer can be completed without interrupting the operation of the DMA device. This mode is most useful for DMA single-cycle or demand modes where the transfer process allows time for the CPU to execute the interrupt routine.

The buffer chaining mode of a channel may be entered by programming the address and count of a transfer as usual. After the initial address and count is programmed, the Base registers are selected via the "Set Chaining Mode" register "Chaining mode Enabled" bit. The address and count for the second transfer and both the "Chaining mode Enabled" and the "program complete" bits of the Set Chaining Mode register should be programmed at this point, before starting the DMA process. When, during the DMA process, the Current Buffer is expired, the Base address, page, and Count registers will be transferred to the Current registers and a signal that the buffer has been expired is sent to the programming master.

This signal will be an IRQ13 if the master is the Host CPU, or a TC if the programming master is an EISA device. The type of programming master is indicated in the DMA's Set Chaining Mode Register, Bit 4. If the Host CPU is the programming master for the Channel, TC will be generated only if the Current buffer expires and there is no Next Buffer stored in the Base registers.

Upon the expiration of a Current Buffer, the new Base register contents should be programmed and both the "Chaining mode Enabled" and "program complete" bits of the "Set Chaining Mode" register should be set. This resets the interrupt, if the Host CPU was the programming master, and allows for the next Base register to Current register transfer. If the "program complete" bit is not set before the current transfer reaches TC, then the DMA controller will set the Mask Bit and the TC bit in the Status register and stop transferring data. In this case, an over-run is likely to occur. To determine if this is the case, a read from the Status register or the Mask register can be done (the Mask register has been made readable). If the channel is masked or has registered a TC, the DMA channel has been stopped and the full address, count, and chaining mode must be programmed to return to normal operation.

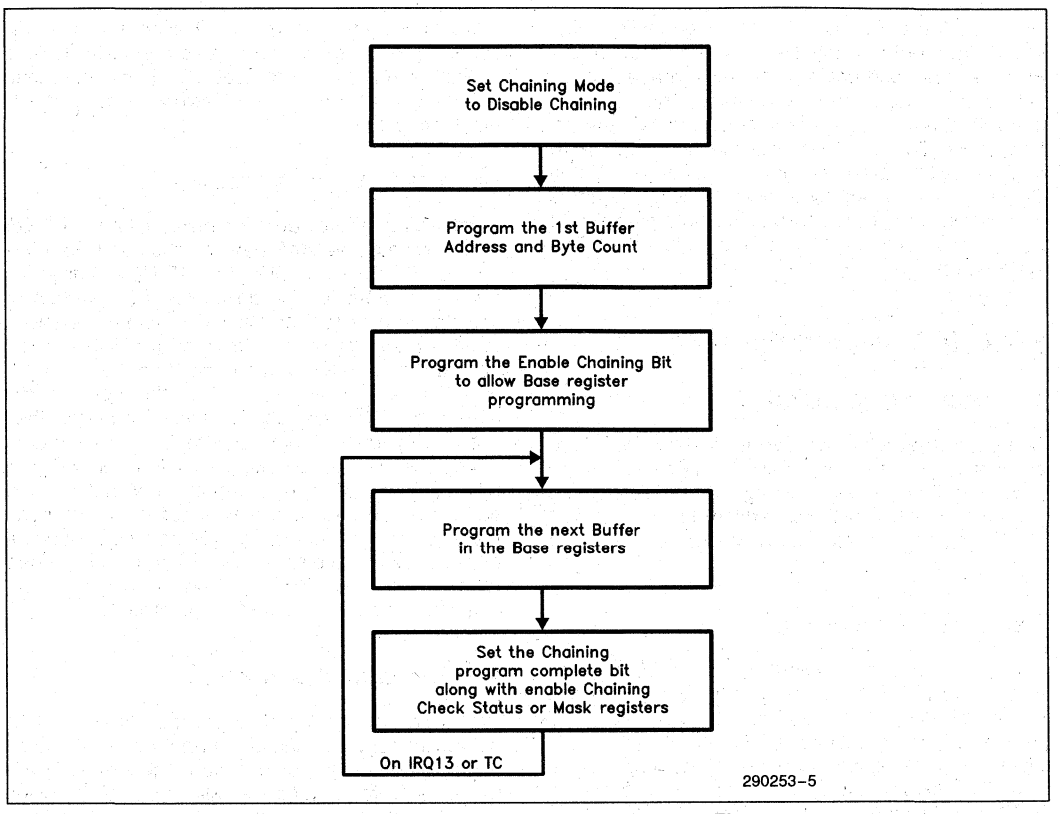


Figure 3-2. Buffer Chaining Programming Flow

Note that if the Host CPU is the programming master, an interrupt will only be generated if a Current Buffer expires and chaining mode is enabled, it will not occur during initial programming. The "Channel Interrupt Status" register will indicate pending interrupts only. That is, it will indicate an empty Base register with "Chaining Mode" enabled. When Chaining mode is enabled, only the Base registers are written by the processor, and only the Current registers can be read. The Current registers are only updated on a TC.

3.6 DMA Enhanced Timing

3.6.1 COMPATIBLE TIMING

Compatible timing is provided for DMA slave devices, which, due to some design limitation, cannot support one of the faster timings. Compatible timing runs at 1125 ns/single cycle and 1000 ns/cycle during the repeated portion of a BLOCK or DEMAND mode transfer.

3.6.2 TYPE "A" TIMING

Type "A" timing is provided to allow shorter cycles when used with EISA memory. If 8/16-bit ISA memory is decoded, the system automatically reverts to ISA DMA type compatible timing on a cycle-by-cycle basis. Type "A" timing runs at 875 ns/single cycle and 750 ns/cycle during the repeated portion of a BLOCK or DEMAND mode transfer. Type "A" timing varies from compatible timing primarily in shortening the memory operation to the minimum allowed by EISA memory. The I/O portion of the cycle (data setup or write, I/O read access time) is the same as with compatible cycles. The actual active command time is shorter, but it is expected that the DMA devices which provide the data access time or write data setup time should not require excess IORC# or IOWC# command active time. Because of this, most ISA DMA devices should be able to use type "A" timing.

3.6.3 TYPE "B" TIMING

Type "B" timing is provided for 8/16-bit ISA or EISA DMA I/O devices which can accept faster I/O timing. Type "B" only works with EISA memory. If 8/16-bit ISA memory is decoded, the system automatically reverts to ISA DMA type compatible timing on a cycle by cycle basis. Type "B" timing runs at 750 ns/single cycle and 500 ns/cycle during the repeated portion of a BLOCK or DEMAND mode transfer. Type "B" timing requires faster DMA slave devices than compatible timing in that the cycles are shortened so that the data setup time on I/O write cycles is shortened and the I/O read access time is required to be faster. Some of the current ISA devices should be able to support type "B" timing, but these will probably be more recent designs using relatively fast technology.

3.6.4 TYPE "C" (BURST TIMING)

Burst timing is provided for newly designed EISA DMA devices. The DMA slave device needs to monitor the EXRDY and IORC# or IOWC# signals to determine when to change the data (on writes) or sample the data (on reads). This timing will allow up to 33 Mbytes per second transfer rate with a 32-bit DMA device and 32-bit memory. Note that 8- or 16-bit DMA devices are supported (through the programmable DMA address increment) and that they use the "byte lanes" natural to their size for the data transfer. As with all bursts, the system will revert to two BCLK cycles if the memory does not support burst. When a DMA burst cycle accesses non-burst memory and the DMA cycle crosses a page boundary into burstable memory, the EBC will continue performing non-burst cycles. This will not cause a problem since the data is transferred correctly.

3.7 Register Description

DMA Channel 4 is used to cascade the two DMA controllers together and should not be programmed for any mode other than cascade. The Mode register will default to cascade mode. Special attention should also be taken when programming the Command and Mask registers as related to channel 4 (refer to the Command and Mask register descriptions, Sections 3.7.7 and 3.7.11).

3.7.1. STOP REGISTERS (RING BUFFER DATA STRUCTURE)

To support a common data communication data structure (the ring buffer), a set of new DMA registers have been provided. These registers are called Stop registers. Each channel has 22-bits of register location associated with it. The 22-bits is distributed between three different registers (one 6-bit and two 8-bit). The Stop registers can be enabled or disabled by writing to the channel's corresponding Extended Mode register.

The ring buffer data structure reserves a fixed portion of memory, on doubleword boundaries, to be used for a DMA channel. Consecutively received frames or other data structures are stored sequentially within the boundaries of the ring buffer memory.

The beginning and end of the ring buffer area is defined in the Base Address register and the Base Address register + the Base Byte/Transfer Count. The incoming frames (data) are deposited in sequential locations of the ring buffer. When the DMA reaches the end of the ring buffer, indicating the byte count has expired, the DMA controller (if so programmed) will Autoinitialize. Upon autoinitialization, the Current Address register will be restored from the Base Address register, taking the process back to the start of the ring buffer. The DMA will then be available to begin depositing the incoming bytes in the ring buffers sequential locations—providing that the host CPU has read the data that was previously placed in those locations. The DMA determines that the CPU has read certain data by the value that the CPU writes into the Stop register.

Once the data of a frame is read by the CPU, the memory location it occupies becomes available for other incoming frames. The Stop register prevents the DMA from over writing data that has not yet been read by the CPU. After the CPU has read a frame from memory it will update the Stop register to point to the location that was last read. The DMA will not deposit data into any location beyond that pointed to by the Stop register. The last address transferred before the channel is masked is the first address that matches the Stop register. (See Table 3-1). The Stop registers store values to compare against A<23:2> only, so the size of the ring buffer is limited to 16 Mbytes.

Since I/O writes should match the I/O slave size, 8-bit writes should be used to program the ISP registers. When writing to the DMA registers, the DMA channels should be masked.

The Bus Controller provides I/O recovery for back-to-back CPU to 8-bit I/O cycles. For EISA master accesses, I/O recovery of at least one BCLK must be provided by software.

For example:

If the stop register = 00001Ch, the last three transfers will be:

Table 3-1

	By Bytes	By Words	By Dwords
Increment	XX00001Ah XX00001Bh XX00001Ch	XX000018h XX00001Ah XX00001Ch	XX000014h XX000018h XX00001Ch
Decrement	XX000021h XX000020h XX00001Fh	XX000023h XX000021h XX00001Fh	XX000027h XX000023h XX00001Fh

Figure 3-3 is a diagram of a ring buffer data structure.

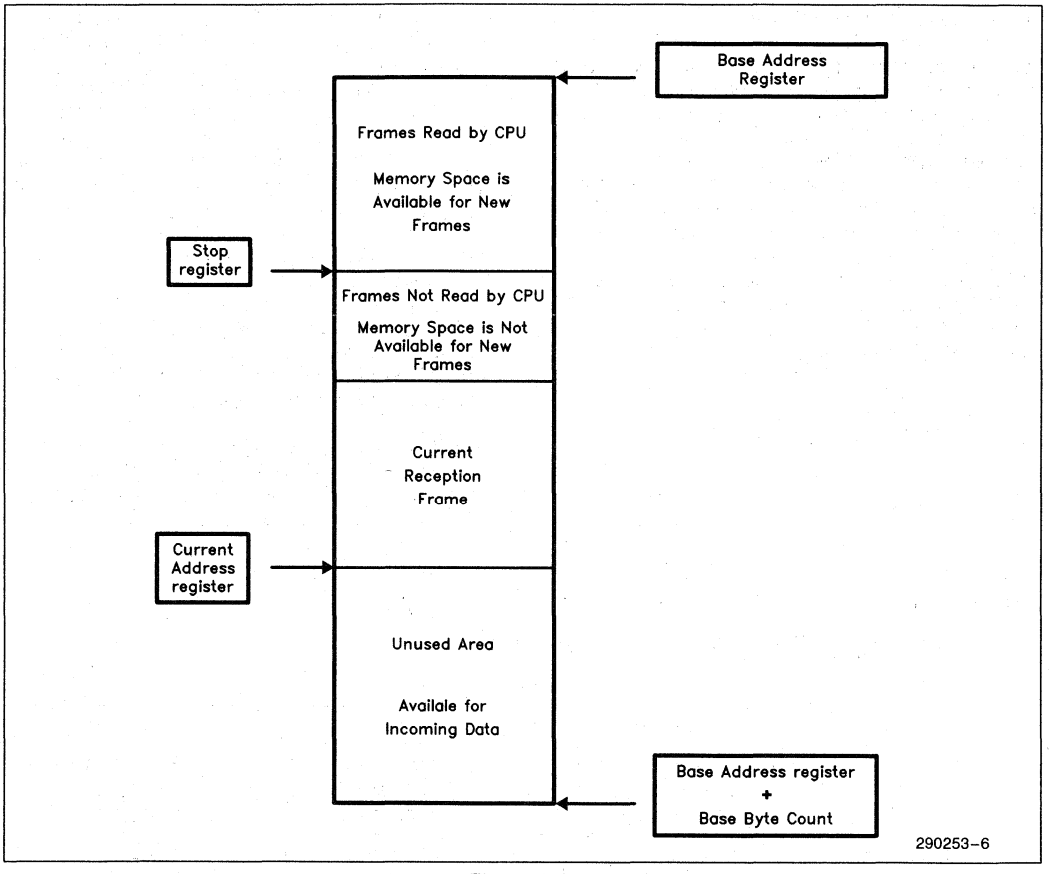


Figure 3-3

3.7.2 DMA MEMORY LOW PAGE REGISTER (READ/WRITE)

Each channel has an 8-bit Low Page register associated with it. The DMA memory Low Page register contains the eight second most-significant bits of the 32-bit address (16–23). It works in conjunction with the DMA controller's High Page register and Current Address register to define the complete (32-bit) address for the DMA channels and corresponds to the "Current Address" register for each channel. This 8-bit register is read or written directly by the processor or bus master. It may also be re-initialized by an Autoinitialize back to its original value. Autoinitialize takes place only after a TC or EOP.

3.7.3 DMA MEMORY HIGH PAGE REGISTER (READ/WRITE)

Each channel has an 8-bit High Page register. The DMA memory High Page register contains the eight most-significant bits of the 32-bit address (24–31). It works in conjunction with the DMA controller's Low Page register and Current Address register to define the complete (32-bit) address for the DMA channels and corresponds to the "Current Address" register for each channel. This 8-bit register is read or written directly by the processor or bus master. It may also be re-initialized by an Autoinitialize back to its original value. Autoinitialize takes place only after a TC or EOP.

This register is reset to 00h during the programming of both the low page register and the Current Address register. Thus, if this register is not programmed after the other address and Low Page registers are programmed, then its value will be zero. In this case, the DMA channel will operate the same as an 82C37 (from an addressing standpoint). This is the address compatibility mode.

If the high 8-bits of the address are programmed after the other addresses, then the channel will modify its operation to increment (or decrement) the entire 32-bit address. This is unlike the 82C37 "Page" register in the original PCs which could only increment to a 64K boundary (for 8-bit channels) or 128K (for 16-bit channels). This is extended address mode. In this mode, the EISA bus controller should generate the signals MRDC# and MWTC# only for addresses below 16 Mbytes.

3.7.3.1 Address Compatibility Mode

Whenever the DMA is operating in Address Compatibility mode, the addresses do not increment or decrement through the HIGH and LOW Page registers, and the high page register is set to 00h. This is compatible with the 82C37 and Page register implementation used in the PC AT*. This mode is set when any of the lower three address bytes of a channel are programmed. If the upper byte of a channel's address is programmed last, the channel will go into Extended Address Mode. In this mode the high byte may be any value and the address will increment or decrement through the entire 32-bits.

After RST all channels will be set to Compatibility Mode. The Master Clear command will also reset the proper channels to Compatibility Mode. The mode bits are stored in individual flip-flops on a per-channel basis.

3.7.4 CURRENT ADDRESS REGISTER (READ/WRITE)

Each channel has a 16-bit Current Address register. This register holds the value of the 16 least significant bits (0–15) of the full 32-bit address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written to or read from by the microprocessor or bus master in successive 8-bit bytes. It may also be re-initialized by an Autoinitialize back to its original value. Autoinitialize takes place only after a TC or EOP.

3.7.4.1 Address Shifting when Programmed for 16-bit I/O Count by Words

To maintain compatibility with the implementation of the DMA in the PC AT which used the 82C37, the DMA will shift the addresses when the Extended Mode register is programmed for, or defaulted to, transfers to/from a 16-bit device count by words. The address shifting is shown in Table 3-2. Note that the least significant bit of the Low Page resister is dropped in 16-bit shifted mode.

Table 3-2

Output Address	8-Bit I/O Programmed Address	16-Bit I/O Programmed Address (Shifted)	32-Bit I/O Programmed Address	16-Bit I/O Programmed Address (No Shift)
A0 A<16:1> A<31:17>	A0 A<16:1> A<31:17>	"0" A<15:0> A<31:17>	A0 A<16:1> A<31:17>	A0 A<16:1> A<31:17>

* PC AT is a trademark of IBM.

3.7.5 CURRENT WORD REGISTER (READ/WRITE)

Each channel has a 24-bit Current Word Count register. This register determines the number of transfers to be performed. The actual number of transfers will be one more than the number programmed in the Current Word Count register (i.e., programming a count of 100 will result in 101 transfers). The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes from zero to 0FFFFFFH, a TC will be generated.

Following the end of a DMA service it may also be re-initialized by an Autoinitialization back to its original value. Autoinitialize can occur only when a TC occurs. If it is not Autoinitialized, this register will have a count of FFFFFFH after TC.

To maintain compatibility with the 82C37, programming either the low byte, bits<7:0>, or the middle byte, bits<15:8>, will clear the high byte bits<23:16>. This provides compatibility with the previous software that does not know of the existence of the upper byte of the word count.

When the Extended Mode register is programmed for, or defaulted to, transfers to/from an 8-bit I/O, the Word count will indicate the number of bytes to be transferred.

When the Extended Mode register is programmed for, or defaulted to, transfers to/from a 16-bit I/O, with shifted address, the Word count will indicate the number of 16-bit words to be transferred.

When the Extended Mode register is programmed for transfers to/from a 16- or 32-bit I/O, the Word Count will indicate the number of bytes to be transferred. The number of bytes does not need to be a multiple of two or four in this case.

3.7.6 BASE PAGE, BASE ADDRESS AND BASE WORD COUNT REGISTERS (WRITE ONLY)

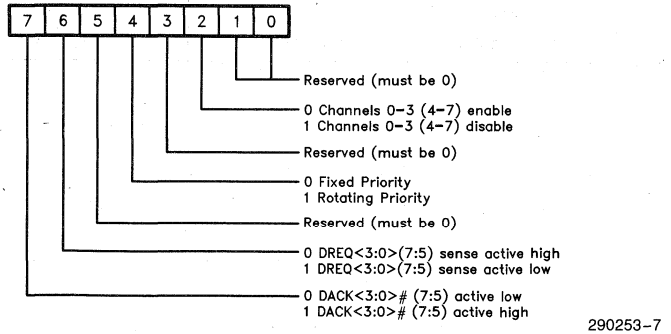
Each channel has a set of Base Page, Base Address and Base Word Count registers. These registers store the original value of their associated Current registers. During Autoinitialize these values are used to restore the Current registers to their original values. The Base registers are written simultaneously with their corresponding Current register in 8-bit bytes in the Program Condition by the microprocessor. These registers cannot be read by the microprocessor.

During Chaining Mode, these registers will store the information about the next buffer in the Chain, if programmed.

3.7.7 COMMAND REGISTER (WRITE ONLY)

This 8-bit register controls the operation of the DMA. It is programmed by the microprocessor in the Program Condition and is cleared by Reset or a Master Clear instruction. Figure 3-4 lists the function of the command bits.

Channels 0–3 port address—08h
Channels 4–7 port address—0D0h

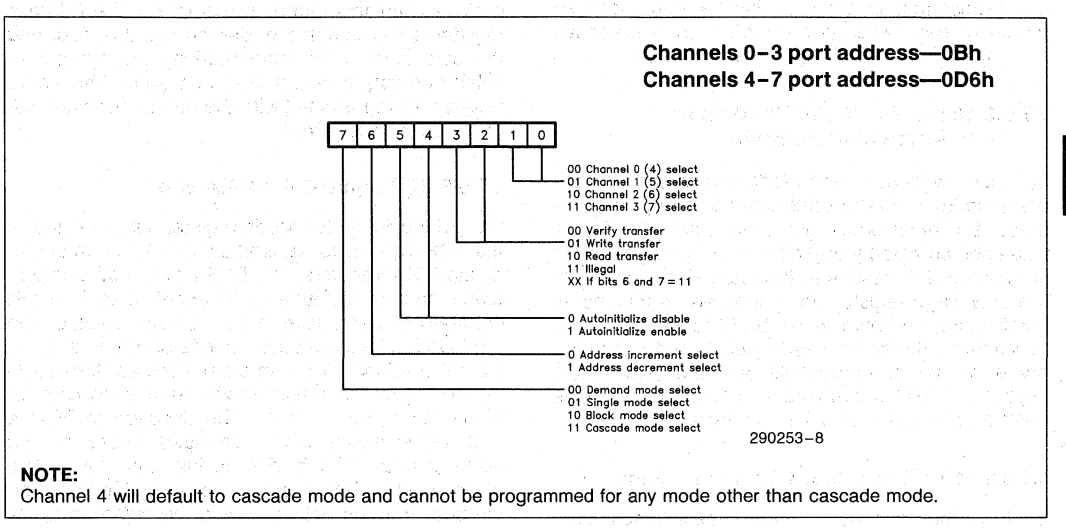


NOTE:
Disabling channels 4–7 will also disable channels 0–3, as channels 0–3 are logically cascaded into channel 4.

Figure 3-4. Command Register (Write Only)

3.7.8 MODE REGISTER (WRITE ONLY)

Each channel has a 6-bit Mode register associated with it. When the register is being written, bits 0 and 1 determine which channel is to be selected. This register is reset upon RST and Master Clear. Its reset value is Verify transfer, Autoinitialize disable, Address increment, Demand mode. Channel 4 defaults to cascade mode. (See Figure 3-5.)



NOTE:
Channel 4 will default to cascade mode and cannot be programmed for any mode other than cascade mode.

Figure 3-5. Mode Register (Write Only)

3.7.9 EXTENDED MODE REGISTERS (WRITE ONLY)

Each channel has a 16-bit Extended Mode register associated with it. The register is used to program the DMA device data size and timing mode. When the register is being written, bits 0 and 1 determine which channel is to be selected.

The default programmed values for channels 0–3 are, 8-bit I/O Count by Bytes, Compatible timing, EOP output and Stop registers disabled. The default values for channels 4–7 are, 16-bit I/O Count by Words with shifted address, Compatible timing, EOP output and Stop register disabled. The default is selected upon reset with RST, it is not selected by Master Clear, or any other programming sequence. (See Figure 3-6.)

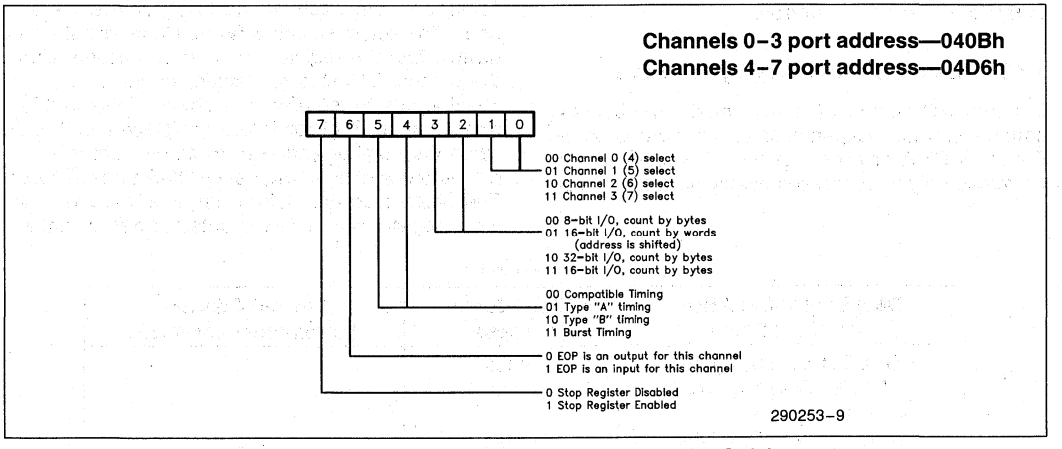


Figure 3-6. Extended Mode Registers (Write Only)

3.7.9.1 8-Bit I/O, "Count By Bytes" Mode

In 8-bit I/O, "count by bytes" mode, the address counter can be programmed to any address. The count register is programmed with the number of bytes minus 1 to transfer. In this mode, byte assembly/disassembly is not available (or necessary) so the timing used when 8- or 16-bit memory is sensed is compatible with the original ISA products.

3.7.9.2 16-Bit I/O, "Count By Words" (Address Shifted) Mode

In "count by words" mode (address shifted), the address counter can be programmed to any even address, but must be programmed with the address value shifted right by one bit. The Page registers are not shifted, this results in the least significant bit of the Low Page register being ignored. In this mode, burst timing and byte assembly/disassembly is not available so the timing used when 8- or 16-bit memory is sensed is compatible with the original ISA products. The Count register is programmed with the number of words minus 1 to be transferred.

3.7.9.3 16-Bit I/O, "Count By Bytes" Mode

In 16-bit "count by bytes" mode, the address counter can be programmed to any byte address. For most DMA devices, however, it should be programmed only to even addresses. If the address is programmed to an odd address, the DMA controller will do a partial word transfer during the first and last transfer if necessary. The bus controller logic will do the byte/word assembly necessary to read or write any size memory device and both the DMA and bus controllers support burst for this mode. In this mode, the Address register is incremented or decremented by two and the byte count is decremented by the number of bytes transferred during each bus cycle. The Count register is programmed with the number of bytes -1 to be transferred.

3.7.9.4 32-Bit I/O, "Count By Bytes" Mode

In 32-bit I/O "count by bytes" mode, the address counter can be programmed to any byte address. For most DMA devices, however, it should be programmed only to addresses evenly divisible by four. If the address is programmed to a value that is not divisible by four, then the DMA controller will do partial transfers for the first and last transfers if necessary. The bus controller logic will do the byte/word assembly necessary to read or write any size memory device and both the DMA and bus controllers support burst for this mode. In this mode, the Address register is incremented or decremented by four and the byte count is decremented by the number of bytes transferred during each bus cycle. The Count register is programmed with the number of bytes minus 1 to be transferred.

3.7.9.5 EOP Input/Output Selection

Bit 6 of the Extended Mode register selects whether the EOP signal is to be used as an input or an output during DMA transfers. The EOP I/O selection is programmable on a channel by channel basis. EOP will generally be used as an output, as was available on the PCAT. The input function was added to support Data Communication and other devices that would like to trigger an autoinitialize when a collision or some other event occurs. The direction of EOP is switched when DACK# is changed. There may be some overlap of the ISP driving the EOP signal along with the DMA slave, however, during this overlap both devices will be driving the signal to a low level (inactive).

3.7.9.6 Stop Register Selection

Bit 7 of this register selects whether the Stop registers associated with this channel are to be used or not. Normally the Stop Registers will not be used. This function was also added to help support Data Communication or other devices that work from a Ring Buffer in memory (refer to Section 3.7.1).

3.7.9.7 Summary of the DMA Transfer Sizes

Table 3-3 lists each of the DMA device transfer sizes. The column labeled "Word Count register" indicates that the register contents represents either the number of bytes to transfer or the number of 16-bit words to transfer. The column labeled "Current Address Register Increment/Decrement" indicates the number added to or taken from the Current Address register after each DMA transfer cycle. The Mode Register determines if the Current Address register will be incremented or decremented.

Table 3-3

DMA Device Data Size and Word Count	Word Count Register	Current Address Increment/Decrement
8-Bit I/O, Count by Bytes	Bytes	1
16-Bit I/O, Count by Words (Address Shifted)	Words	1
16-Bit I/O, Count by Bytes	Bytes	2
32-Bit I/O, Count by Bytes	Bytes	4

3.7.10 REQUEST REGISTER (WRITE ONLY)

Each channel has a Request bit associated with it in one of the two 4-bit Request registers. The Request register is used by software to initiate a DMA request. The DMA responds to the software request as though DREQ<x> is active. These requests are non-maskable and subject to prioritization by the Priority Encoder network (refer to Section 3.4). Each register bit is set or reset separately under software control or is cleared upon generation of a TC. The entire register is cleared upon RST, it is not cleared upon a RSTDRV output. To set or reset a bit, the software loads the proper form of the data word. When the register is being written, bits 0 and 1 determine which channel is to be selected. In order to make a software request, the channel must be in Block Mode. (See Figure 3-7.)

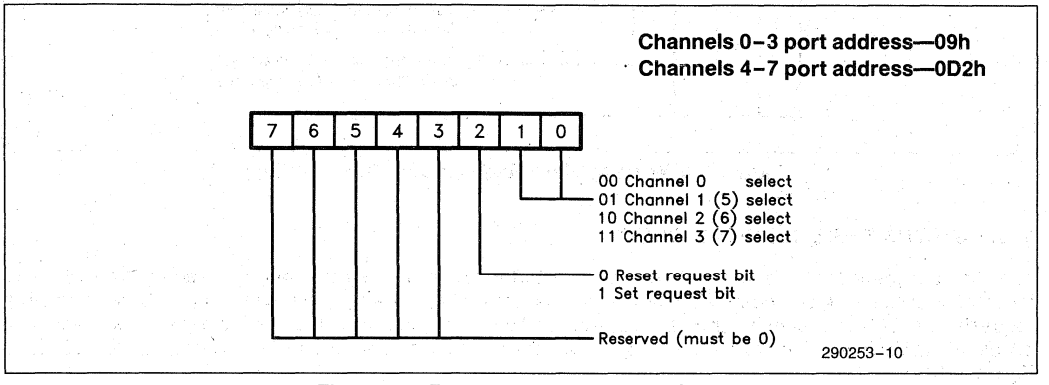


Figure 3-7. Request Register (Write Only)

3.7.11 MASK REGISTER

Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is automatically set when the Current Word Count register reaches terminal count (unless the channel is programmed for autoinitialization or chaining mode). Each bit of the two, 4-bit registers may also be set or cleared under software control. The entire register is also set by a RESET and Master Clear. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. (See Figure 3-8.)

NOTE:
If the channel 4 mask bit is set, the channels logically cascaded into it are also masked.

All four bits of the Mask register may also be read or written with a single command (see Figure 3-9).

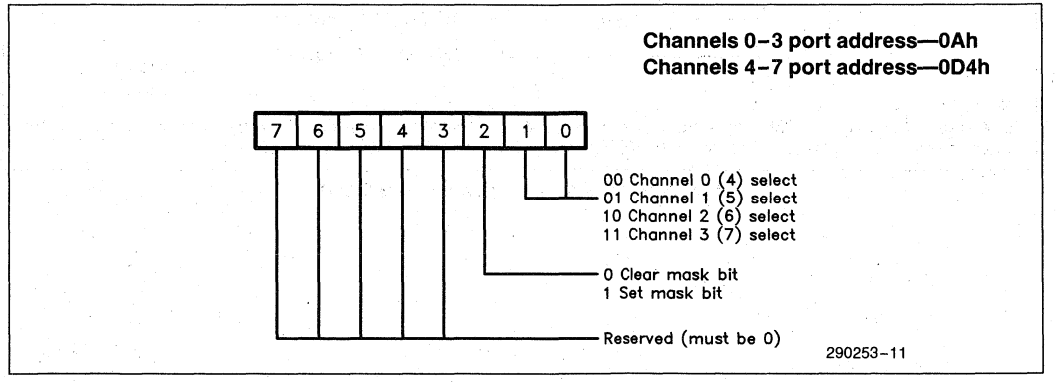


Figure 3-8. Write Single Mask Bit (Write Only)

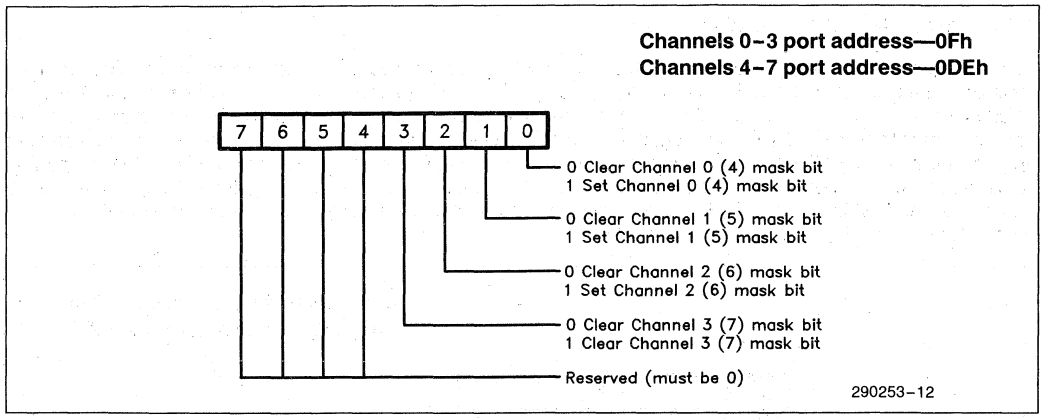
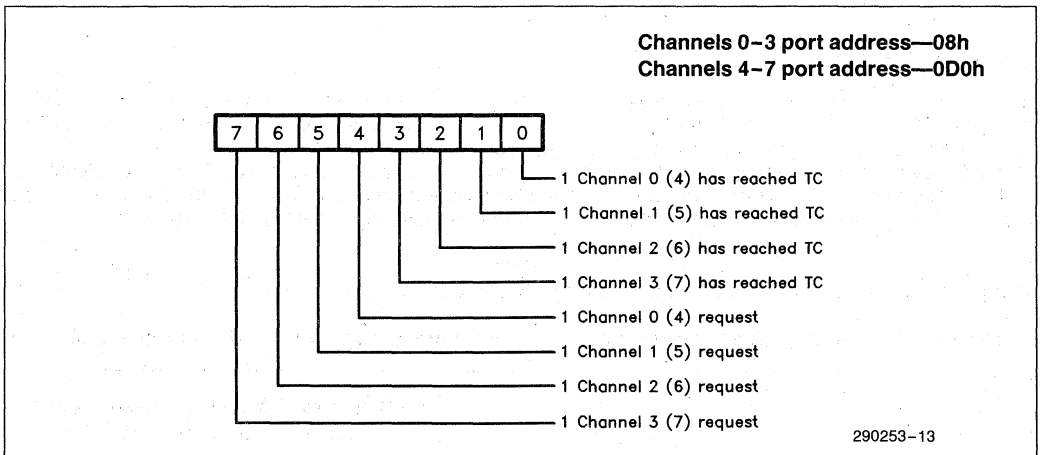


Figure 3-9. Write All Mask Register Bits (Read/Write)

3.7.12 STATUS REGISTER (READ ONLY)

The Status register contains information about the status of the devices that may be read by the CPU. This information includes which channels have reached a terminal count and which channels have pending DMA requests. Bits 0–3 are set every time a TC is reached by that channel. These bits are cleared upon Reset and on each Status Read. Bits 4–7 are set whenever their corresponding channel is requesting service. (See Figure 3-10.)



NOTE:
EISA masters that access DMA registers are not allowed to read the status registers. Only the Host CPU may read the status registers. This is required because the TC bits are cleared upon reading of these registers.

Figure 3-10. Status Register (Read Only)

3.7.13 SET CHAINING MODE REGISTER (WRITE ONLY)

Each channel has a Chaining Mode register associated with it. The Chaining Mode register is used to enable or disable DMA buffer chaining and to indicate when the DMA Base registers are being programmed. When the register is being written, bits 0 and 1 determine which channel is to be selected. The chaining status and interrupt status for all channels can be determined by reading the Set Chaining Mode Status, Channel Interrupt Status, and Chain Buffer Expiration Control registers. The Chaining Mode register is reset to zero upon RST, access (read or write) of a channel's Mode register or Extended Mode register, or a Master Clear. The values upon reset are disable chaining mode and generate IRQ13. (See Figure 3-11.)

Refer to "Buffer Chaining" section for additional information about Buffer Chaining.

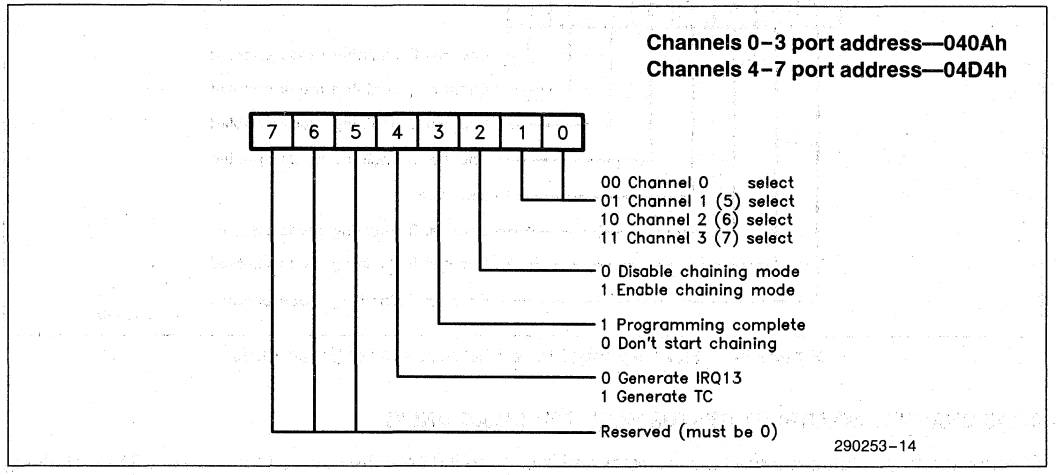


Figure 3-11. Set Chaining Mode Register (Write Only)

The **Enable Chaining mode** bit is used to control the chaining mode logic. If the bit is programmed to a 1 after the initial DMA address and count are programmed, then the Base address and Count registers will become available for programming the next chain buffer.

After the Base registers are programmed (as indicated above), the **Enable Chaining mode** and the **Programming Complete** bits are set to begin a DMA chaining sequence. The DMA channel is then ready to begin transferring data (assuming the mask bit is cleared).

When a chaining mode interrupt or TC (for EISA programming masters) occurs, the next Address and Count registers should be programmed and the **Programming Complete** bit should be set to set up for the next transfer. When the Programming Complete bit is set, the Enable Chaining Mode bit and the Generate IRQ bit both need to be written to the correct state. Upon this programming the interrupt request for that channel is reset, if it was active.

Bit 4 of the Set Chaining Mode register is used to determine the response to the expiration of a DMA buffer. Normally the Host CPU needs to be informed to program the next set of Base registers. In this case bit 4 should be set to a zero to generate an IRQ13. If an EISA bus master is using the DMA to assist in data transfer, then bit 4 can be set to a 1 to generate an EOP(TC) instead of an IRQ13. The EISA master can then use the EOP(TC) in the same way that the CPU uses an interrupt. In this mode the EOP signal will only be driven active while the Channel that caused it is running as determined by the DACK lines.

3.7.14 SET CHAINING MODE STATUS REGISTER (READ ONLY)

This register is read only and is used to determine if Chaining mode for a particular channel is enabled or disabled. A "1" read in this register indicates that the channel's chaining mode is enabled. A "0" indicates that the chaining mode is disabled. All Chaining mode bits are disabled after a reset. After the DMA is used in Chaining mode the CPU will need to clear the Chaining mode enable bit if non-Chaining mode is desired. This bit is programmed in bit 2 of the Set Chaining Mode register. (See Figure 3-12.)

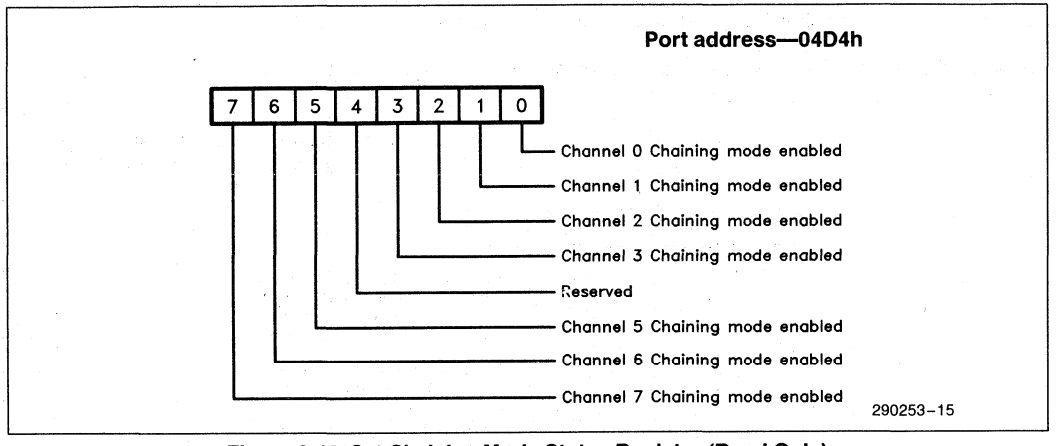


Figure 3-12. Set Chaining Mode Status Register (Read Only)

3.7.15 CHANNEL INTERRUPT STATUS REGISTER (READ ONLY)

Channel Interrupt Status is a read only register and is used to indicate the source (channel) of a DMA chaining interrupt on IRQ13. The DMA controller drives IRQ13 active after reaching terminal count, with chaining mode enabled. It does not drive IRQ13 active during the initial programming sequence that loads the Base registers. (See Figure 3-13.)

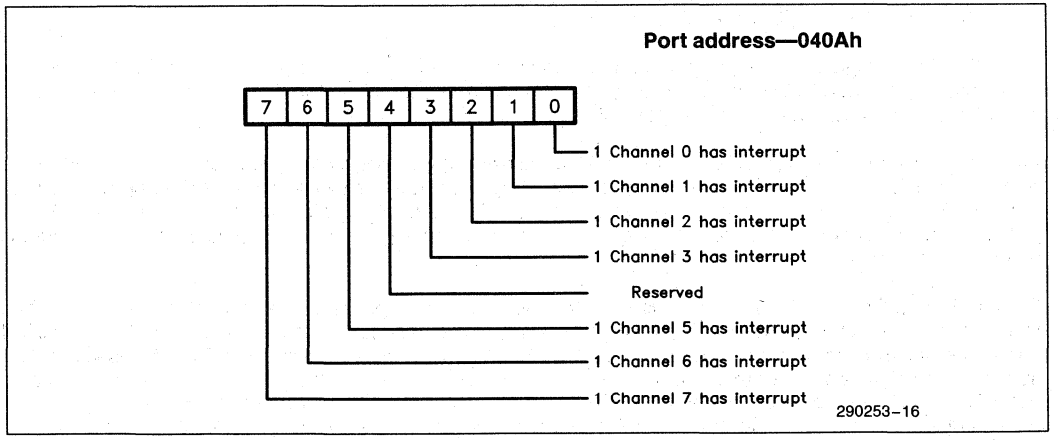


Figure 3-13. Channel Interrupt Status Register (Read Only)

3.7.16 CHAIN BUFFER EXPIRATION CONTROL REGISTER (READ ONLY)

This register is read only and reflects the outcome of the expiration of a chain buffer. If a channel bit is set to a 0, IRQ13 will be activated, otherwise a TC will be issued. This bit is programmed in bit 4 of the Set Chaining Mode register. (See Figure 3-14.)

3.8 Software Commands

These are additional special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The three software commands are: (1) Clear Byte Pointer Flip-Flop (2) Master Clear and (3) Clear Mask Register.

3.8.1 CLEAR BYTE POINTER FLIP-FLOP

This command is executed prior to writing or reading new address or word count information to the DMA. This initializes the flip-flop to a known state so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.

When the Host CPU is reading or writing DMA registers, two Byte Pointer Flip-Flops are used; one for channels 0–3 and one for channels 4–7. Both of these act independently. There are separate software commands for clearing each of them (0Ch for Channels 0–3, 0D8h for Channels 4–7).

An additional Byte Pointer Flip-Flop has been added for use when EISA masters are reading and writing DMA registers. (The arbiter state will be used to determine the current master of the bus.) This Flip-Flop is cleared when an EISA Master performs a write to either 0Ch or 0D8h, there is only one for all eight DMA channels. This new Byte Pointer was added to eliminate the problem of the Host CPU's byte pointer getting out of sync if an EISA Master takes the bus during the Host CPU's DMA programming.

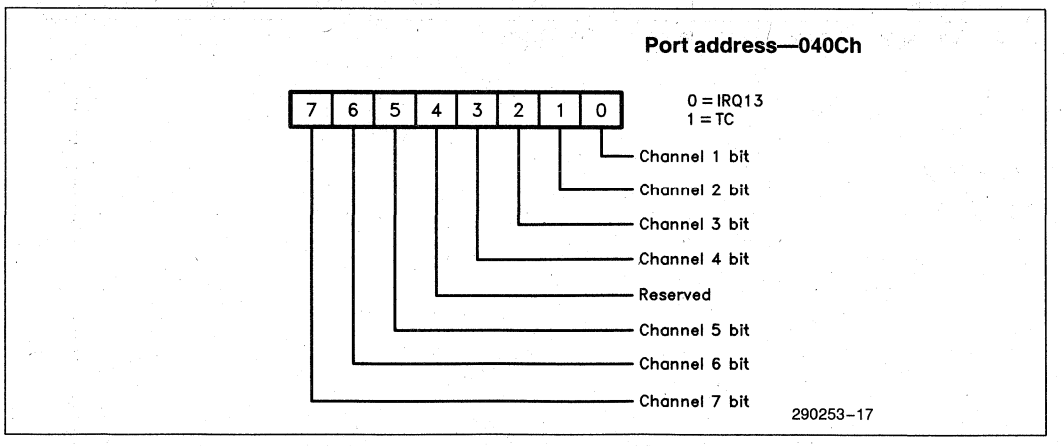


Figure 3-14. Chain Buffer Expiration Control Register (Read Only)

3.8.2 MASTER CLEAR

This software instruction has the same effect as the hardware Reset. The Command, Status, Request, and Internal First/Last Flip-Flop registers are cleared and the Mask register is set. The DMA controller will enter the idle cycle.

There are two independent Master Clear Commands, 0Dh which acts on channels 0–3, and 0DAh which acts on channels 4–7.

3.8.3 CLEAR MASK REGISTER

This command clears the mask bits of all four channels, enabling them to accept DMA requests. I/O port 0Eh is used for Channels 0–3 and I/O port 0DCh is used for Channels 4–7.

3.9 Terminal Count/EOP Summary

Table 3-4 summarizes the events that will happen as a result of a terminal count or external EOP when running DMA cycles in various modes.

Table 3-4. Terminal Count/EOP Summary

Event								
Word Counter Expired	Yes	X	Yes	X	Yes	X	X	X
Stop Reg. Limit Reached	X	X	X	X	X	X	Yes	Yes
EOP Input	X	Asserted	X	Asserted	X	Asserted	X	X
Conditions								
AUTOINIT	No	No	Yes	Yes	No	No	X	X
Chain and Base Loaded	No	No	X	X	Yes	Yes	X	X
Result								
Status TC	set	set	set	set	—	—	—	—
Mask	set	set	—	—	—	—	set	set
SW Request	clr	clr	clr	clr	—	—	—	—
Current Reg.	—	—	load	load	load	load	—	—

(Load = load current from base, — no change, X = Don't Care)

4.0 BUS ARBITRATION

4.1 Bus Timeout

A bus timeout will cause an NMI to the CPU and will activate RSTDRV.

NOTE:
A bus timeout will not occur during the time the system CPU has control of the bus.

4.1.1 8 μs BUS TIMEOUT

An 8 μs bus timeout is determined by counting 64 BCLKs, beginning with the rising edge of BCLK after a MACK# is negated. After the 64 BCLKs (on the rising edge of BCLK), MREQ# and ST3 (CIP#) are sampled. If MREQ# is still active, a bus timeout will occur. Note that MREQ# can be inactive and the bus is still owned by the master until ST3 (bus cycle in progress) goes inactive. The Arbiter will wait until ST3 (bus cycle in progress) goes inactive before granting the bus to another requester.

Figure 4-1 illustrates the longest that MREQ# can be held active without causing a Bus Time out. It also shows when the last bus master cycle must begin with a START# pulse, which in turn will activate ST3 (CIP#).

4.1.2 32 μs BUS TIMEOUT

To prevent the case of a slave from locking up the system when it does not release EXRDY or CHRDY, a bus timeout will occur if CMD# is active for more than 256 BCLKs. The counting of the 256 BCLKs will be conditioned on a request for the bus from another source. This will allow Burst Cycles to run unimpeded if there are no pending requests. (Although there should be a request at least every 15 μs from Refresh). The 32 μs bus timeout is independent of the type of master on the bus.

4.2 4 μs Limit on DMA Transfers

If the ISP DMA is the master of the bus and another request for the bus is received by the Arbiter and the DMA is not doing compatible timing transfers, a 4 μs timer will be started. Upon the expiration of the 4 μs timer, the DACK will be inactivated after the current DMA cycle has completed. The bus will then be arbitrated for and granted to the highest priority requester.

The 4 μs timer is not used in compatible timing mode, it is also not used for 16-bit ISA masters cascaded through the DMA DREQ lines.

If the DMA channel that was preempted by the 4 μs timer was operating in Block mode, an internal bit will be set so that the channel will be arbitrated for again, independent of the state of DREQ. This bit is reset on RSTDRV.

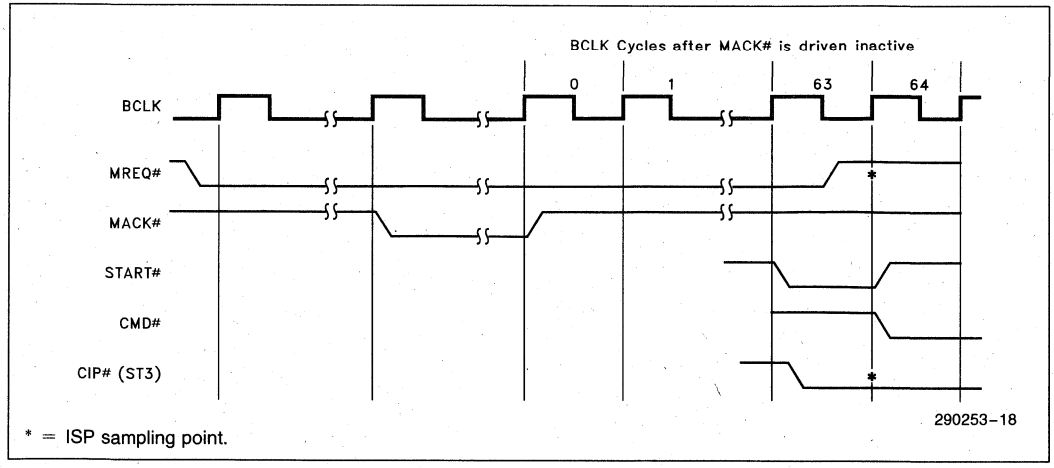


Figure 4-1. 8 μs Bus Timeout

4.3 EISA Bus Master Status Latch

In order to simplify testing of EISA bus master operations, a CPU readable status latch is included which contains information about which EISA bus master most recently had control of the bus. This latch is located at port address 0464h and is read only. (See Figure 4-2.) A read value of (0) indicates that the slot was most recently granted the bus.

An NMI service routine may read this latch to determine which Bus Master controlled the Bus when a Bus preempt Timeout occurred.

Port 0465h is reserved for an additional eight Bus Master status latch slots, it is not implemented on the ISP.

Note that the bits for slot 7 and 8 are driven by the ISP, though they will always be inactive (high).

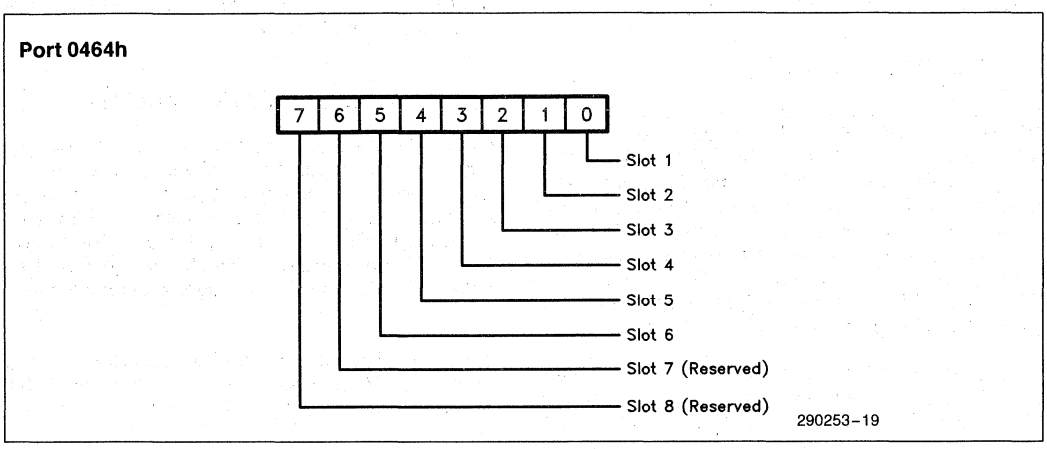


Figure 4-2. EISA Bus Master Status Latch

5.0 INTERRUPT CONTROLLER

The Interrupt Controller consists of two separate 82C59 cores. Interrupt Controller 1 (CNTRL-1) and Interrupt Controller 2 (CNTRL-2) are initialized separately, and can be programmed to operate in different modes. The default settings are: 80x86 Mode, Edge Sensitive (IRQ0–15), Normal EOI, Non-Buffered Mode, Special Fully Nested disabled, and Cascade Mode. CNTLR-1 is connected as the master Interrupt Controller and CNTRL-2 is connected as the slave Interrupt Controller.

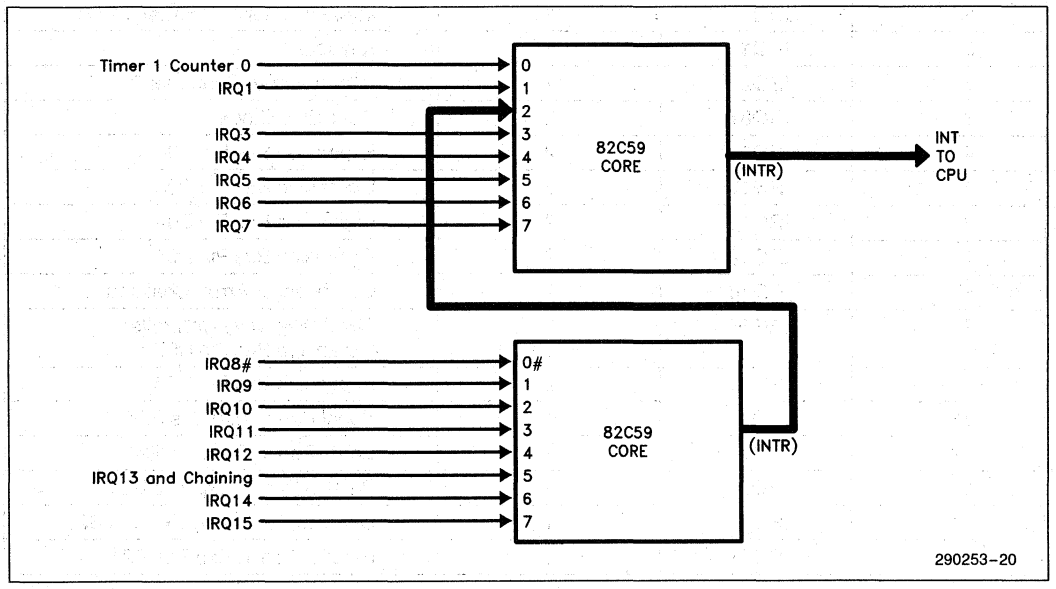


Figure 5-1. Interrupt Controller

5.1 Interrupt Controller I/O Address Map

Table 5-1 lists the I/O port address map for the interrupt registers.

Table 5-1

Interrupts	I/O Address	# of Bits	Register
IRQ<7:0>	0020h	8	CNTLR-1 Control Register
IRQ<7:0>	0021h	8	CNTRL-1 Mask Register
IRQ<7:0>	04D0h	8	CNTRL-1 Edge/Level Control Register
IRQ<15:8>	00A0h	8	CNTLR-2 Control Register
IRQ<15:8>	00A1h	8	CNTRL-2 Mask Register
IRQ<15:8>	04D1h	8	CNTRL-2 Edge/Level Register

5.2 Interrupt Definition Table

IRQ0 and IRQ2, illustrated in Table 5-2, are connected to the interrupt controllers internally.

Table 5-2

Priority	Label	Controller	Typical Interrupt Source
1	IRQ0	1	Interval Timer 1, Counter 0 OUT
2	IRQ1	1	Keyboard
3–10	IRQ2	1	Interrupt from Controller 2
3	IRQ8#	2	Real Time Clock
4	IRQ9	2	Expansion Bus Pin B04
5	IRQ10	2	Expansion Bus Pin D03
6	IRQ11	2	Expansion Bus Pin D04
7	IRQ12	2	Expansion Bus Pin D05
8	IRQ13	2	Coprocessor Error, Chaining
9	IRQ14	2	Fixed Disk Drive Controller Expansion Bus Pin D07
10	IRQ15	2	Expansion Bus Pin D06
11	IRQ3	1	Serial Port 2, Exp Bus B25
12	IRQ4	1	Serial Port 1, Exp Bus B24
13	IRQ5	1	Parallel Port 2, Exp Bus B23
14	IRQ6	1	Diskette Controller, Exp Bus B22
15	IRQ7	1	Parallel Port 1, Exp Bus B21

5.3 Interrupt Request Register (IRR) and In-Service Register (ISR)

The interrupts at the IRQ input lines are handled by two registers in cascade, the Interrupt Request register (IRR) and the In-Service register (ISR). The IRR is used to store all the interrupt levels which are requesting service; and the ISR is used to store all the interrupt levels which are being serviced.

5.4 Priority Resolver

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during Interrupt Acknowledge cycles.

5.5 Interrupt Mask Register (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

5.6 INT (Interrupt)

This output goes directly to the CPU interrupt input.

5.7 INTA#/(ST2#) (Interrupt Acknowledge)

INTA# pulses will cause the Interrupt Controller system to release vectoring information onto the data bus. The format of this data depends on the system mode (μPM) of the Interrupt Controller (programmed for x86 mode in EISA systems). The ISP uses the ST2# input as the Interrupt Acknowledge line.

5.8 Interrupt Sequence

The powerful features of the Interrupt Controller in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The following shows the interrupt sequence for an x86 type system (8080 mode must never be selected by EISA software).

1. One or more of the INTERRUPT REQUEST lines are raised high, setting the corresponding IRR bit(s).
2. The Interrupt Controller evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an INTA# pulse.
4. Upon receiving an INTA# from the CPU, the highest priority ISR bit is set and the corresponding IRR bit is reset. The Interrupt Controller does not drive the Data Bus during this cycle.
5. The CPU will initiate a second INTA# pulse. During this pulse, the Interrupt Controller releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second INTA# pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present at step four (i.e., the request was too short in duration) the Interrupt Controller will issue an interrupt level 7.

5.9 80x86 Mode

In the x86 mode the processor produces only two Interrupt Acknowledge cycles. The Interrupt Controller uses the first interrupt acknowledge cycle to internally freeze the state of the interrupts for priority resolution. The first controller (CNTRL-1), as a master, issues the interrupt code on the cascade lines (internal to the ISP) at the end of the INTA# pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle, the master (CNTRL-1) or slave (CNTRL-2), will send a byte of data to the processor with the acknowledged interrupt code composed as in Table 5-3 (note the state of the ADI mode control is ignored and A5–A11 are unused in 80x86 mode).

Table 5-3. Content of Interrupt Vector Byte for 80x86 System Mode

	D7	D6	D5	D4	D3	D2	D1	D0
IRQ7,15	T7	T6	T5	T4	T3	1	1	1
IRQ6,14	T7	T6	T5	T4	T3	1	1	0
IRQ5,13	T7	T6	T5	T4	T3	1	0	1
IRQ4,12	T7	T6	T5	T4	T3	1	0	0
IRQ3,11	T7	T6	T5	T4	T3	0	1	1
IRQ2,10	T7	T6	T5	T4	T3	0	1	0
IRQ1,9	T7	T6	T5	T4	T3	0	0	1
IRQ0,8	T7	T6	T5	T4	T3	0	0	0

T7–T3 represent the interrupt vector address (refer to Section 5.11.1).

5.10 Programming the Interrupt Controller

The Interrupt Controller accepts two types of command words generated by the CPU or bus master:

1. Initialization Command Words (ICWs): Before normal operation can begin, each Interrupt Controller in the system must be brought to a starting point—by a sequence of two to four bytes timed by I/O write pulses.

An I/O write to CNTRL-1 or CNTRL-2 base address with D4 = 1 and A0 = 0, is interpreted as ICW1. For EISA systems, two I/O writes to "base address + 1" must follow the ICW1. The first write to "base address + 1" performs ICW2, the second write performs ICW3. A third write to "base address + 1" performs ICW4. The base address for CNTRL-1 is 020h, and the base address for CNTRL-2 is 0A0h.

ICW1 starts the initialization sequence during which the following automatically occur:

a) The edge sense circuit is reset, which means that following initialization, an interrupt request (IRQ) input must make a low-to-high transition to generate an interrupt.

b) The Interrupt Mask register is cleared.

c) IR7 input is assigned priority 7.

d) The slave mode address is set to 7.

e) Special Mask Mode is cleared and Status Read is set to IRR.

f) If IC4 = 0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no Auto-EOI, MCS-80, 85 systems).

2. Operation Command Words (OCWs): These are the command words which command the Interrupt Controller to operate in various interrupt modes. These modes are:

a) Fully nested mode

b) Rotating priority mode

c) Special mask mode

d) Polled mode

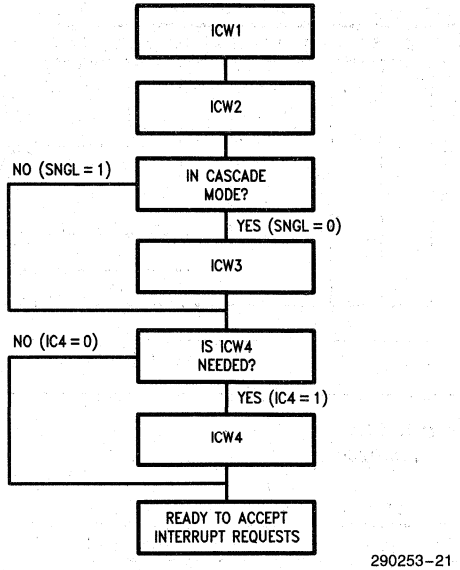
The OCWs can be written into the Interrupt Controller anytime after initialization.

The Base I/O address for CNTRL-1 is 020h; for CNTRL-2, 0A0h. Table 5-4 lists the initial values set-up at power-up by the BIOS.

Table 5-4. Initial Interrupt Controller Values

Port	Value	Description of Contents
020h	11h	CNTLR-1, ICW1
021h	08h	CNTLR-1, ICW2 Vector Address for 000020h
021h	04h	CNTLR-1, ICW3 Indicates Slave Connection
021h	01h	CNTLR-1, ICW4 8086 Mode
021h	B8h	CNTLR-1, Interrupt Mask (may vary)
04D0h	00h	CNTRL-1, Edge/Level Control Register
0A0h	11h	CNTLR-2, ICW1
0A1h	70h	CNTLR-2, ICW2 Vector Address for 0001C0h
0A1h	02h	CNTLR-2, ICW3 Indicates Slave ID
0A1h	01h	CNTLR-2, ICW4 8086 Mode
04D1h	00h	CNTRL-2, Edge/Level Control Register
0A1h	BDh	CNTLR-2, Interrupt Mask (may vary)

Figure 5-2 illustrates the sequence software must follow to load the interrupt controller Initialization Command Words (ICWs). The sequence must be executed for CNTRL-1 and CNTRL-2.



NOTE:
All ICWs must be programmed prior to programming the OCWs.

Figure 5-2. Initialization Sequence

5.11 Initialization Command Words

5.11.1 INITIALIZATION COMMAND WORDS 1 AND 2 (ICW1, ICW2)

In EISA systems the interrupt controllers are programmed for x86 mode. In an x86 system A15–A11 are inserted in the five most significant bits of the vectoring byte and the Interrupt Controller sets the three least significant bits according to the interrupt level. A10–A5 are ignored and ADI (Address interval) has no effect. (See Figures 5-3 and 5-4.)

ICW1 initializes the interrupt controller as follows:

LTIM: This bit is disabled in the EISA system. Its function is replaced by the Edge/Level Triggered Control register (ELCR). It allows each interrupt input to be programmed to either Edge or level mode on a channel-by-channel basis (refer to Section 5.14.7.1).

ADI: Ignored for EISA.

SNGL: This bit is set to 0 for EISA. It indicates that there is more than one interrupt controller in the system.

IC4: If this bit is set—ICW4 has to be read. If ICW4 is not needed, set IC4 = 0.

ICW2 initializes the interrupt controller with the five most-significant bits of the interrupt vector address (refer to Section 5.9).

CNTLR-1 (020h) program to 011h
CNTRL-2 (0A0H) program to 011h

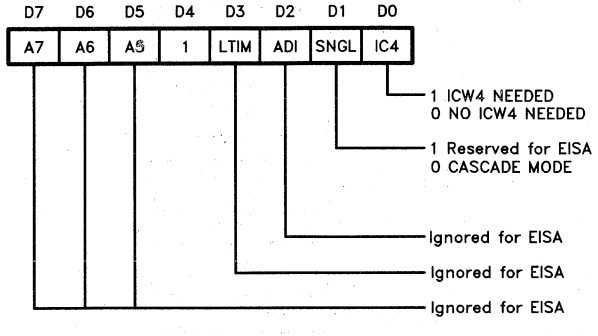


Figure 5-3. ICW1

CNTLR-1 (021h) program to 08h
CNTRL-2 (0A1h) program to 070h

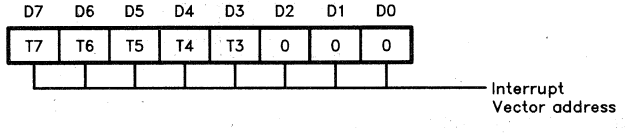


Figure 5-4. ICW2

5.11.2 INITIALIZATION COMMAND WORD 3 (ICW3)

This word is read only in EISA systems. An interrupt request on IRQ2 causes CNTLR-1 to enable CNTLR-2 to present the interrupt vector address during the second interrupt acknowledge cycle. (See Figures 5-5 and 5-6.)

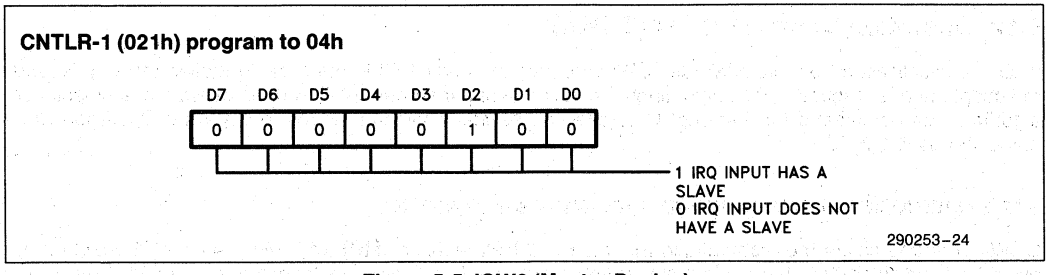


Figure 5-5. ICW3 (Master Device)

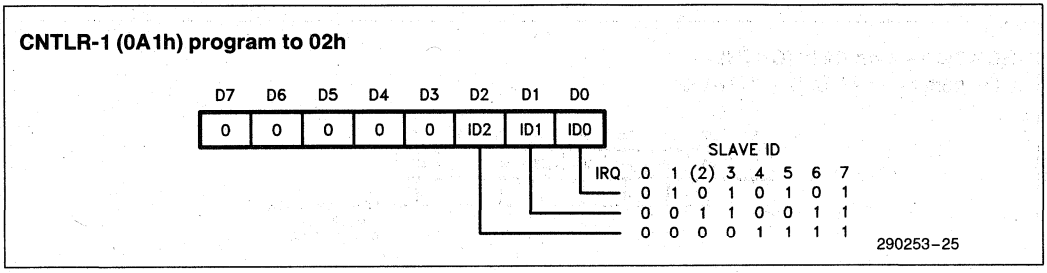


Figure 5-6. ICW3 (Slave Device)

5.11.3 INITIALIZATION COMMAND WORD 4 (ICW4)

(See Figure 5-7.)

SFNM: If SFNM = 1 the special fully nested mode is programmed.

BUF: Programmed to 0 for EISA.

M/S: Ignored for EISA.

AEOI: I AEOI = 1 the automatic end of interrupt mode is programmed.

μPM: Microprocessor mode: μPM = 0 sets the Interrupt Controller for MCS-80, 85 system operation (illegal for EISA systems), μPM = 1 sets the Interrupt Controller for x86 system operation.

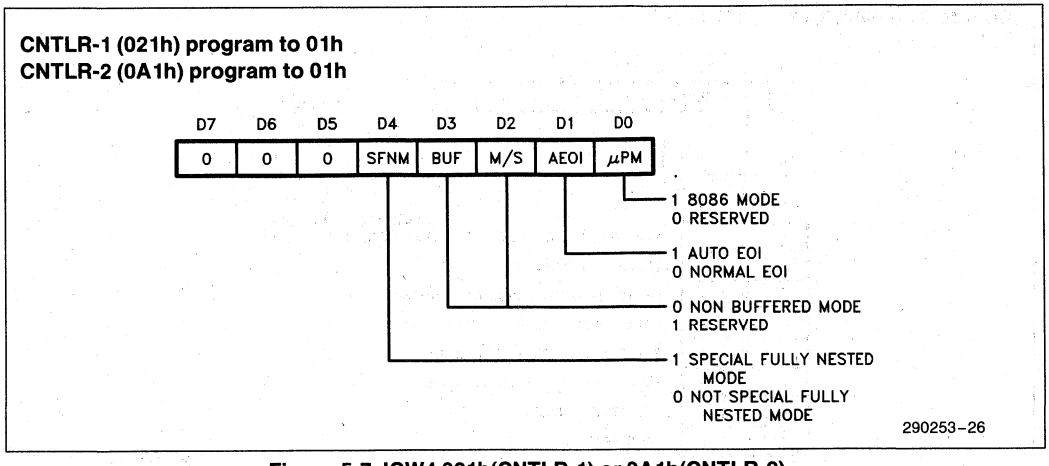


Figure 5-7. ICW4 021h(CNTLR-1) or 0A1h(CNTLR-2)

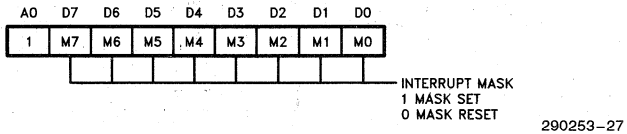
5.12 Operation Control Words (OCWs)

After the Initialization Command Words (ICWs) are programmed into the Interrupt Controller, the chip is ready to accept interrupt requests at its input lines. However, during the Interrupt Controller operation, a selection of algorithms can command the Interrupt Controller to operate in various modes through the Operation Command Words (OCWs).

5.12.1 OPERATION CONTROL WORD 1 (OCW1)—READ/WRITE

OCW1 sets and clears the mask bits in the interrupt Mask register (IMR). M7–M0 represent the eight mask bits. M = 1 indicates the channel is masked (inhibited), M = 0 indicates the channel is enabled. (See Figure 5-8.)

IRQ<7:0>—Port 021h (CNTRL-1)
IRQ<15:0>—Port 0A1h (CNTRL-2)



NOTE:
M0 = Mask for IRQ0(IRQ8#), M1 = Mask for IRQ1(IRQ9), M2 = Mask for IRQ2(IRQ10) and so on . . .

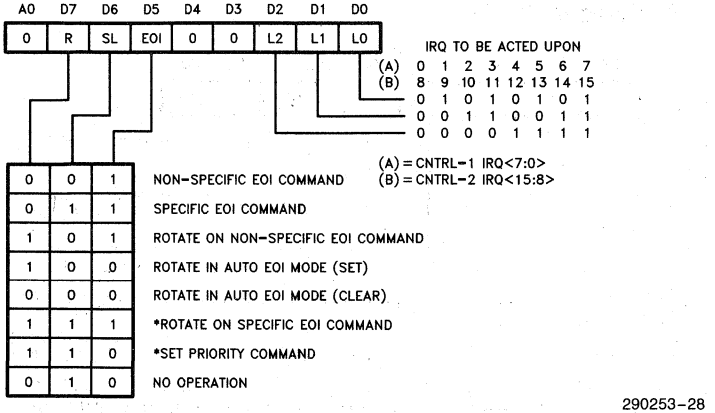
Figure 5-8. OCW1

5.12.2 OPERATION CONTROL WORD 2 (OCW2)

R, SL, EOI—These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

L2, L1, L0—These bits determine the interrupt level acted upon when the SL bit is active. (See Figure 5-9.)

IRQ<7:0>—Port 020h (CNTRL-1)
IRQ<15:0>—Port 0A0h (CNTRL-2)



*L0–L2 are used

Figure 5-9. OCW2

5.12.3 OPERATION CONTROL WORD 3 (OCW3)

ESMM—Enable Special Mask Mode. When this bit is set to 1 it enables the SSM bit to set or reset the Special Mask Mode. When ESMM = 0 the SMM bit becomes a "don't care".

SMM—Special Mask Mode. If ESMM = 1 and SSM = 1 the Interrupt Controller will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the Interrupt Controller will revert to normal mask mode. When ESMM = 0, SMM has no effect. (See Figure 5-10.)

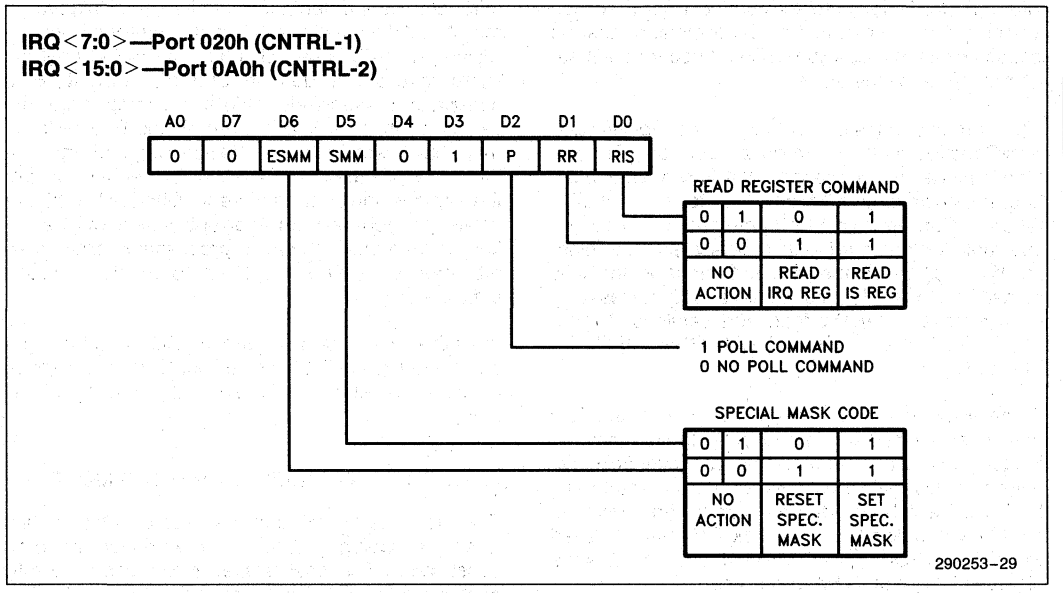


Figure 5-10. OCW3

5.13 End-Of-Interrupt Operation

5.13.1 END OF INTERRUPT (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence INTA# pulse (when AEOI bit in ICW1 is set) or by a command word that must be issued to the Interrupt Controller before returning from a service routine (EOI command). An EOI command must be issued twice if the Cascade mode, once for the master and once for the slave.

There are two forms of EOI commands: Specific and Non-Specific. When the Interrupt Controller is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued the Interrupt Controller will automatically reset the highest IS bit of those that are set, since in the fully nested mode the highest IS level was necessarily the last level acknowledged and serviced. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).

When a mode is used which may disturb the fully nested structure, the Interrupt Controller may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt must be issued which includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and L0–L2 is the binary level of the IS bit to be reset).

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the Interrupt Controller is in the Special Mask Mode.

5.13.2 AUTOMATIC END OF INTERRUPT (AEOI) MODE

If AEOI = 1 in ICW4, then the Interrupt Controller will operate in AEOI mode continuously until reprogrammed by ICW4. In this mode the Interrupt Controller will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse. Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single Interrupt Controller. The AEOI mode can only be used in a master Interrupt Controller and not a slave.

5.14 Modes of Operation

5.14.1 FULLY NESTED MODE

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority from 0 through 7 (0 highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (IS0–7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last INTA#. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal Interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IRQ0 has the highest priority and IRQ7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

5.14.2 THE SPECIAL FULLY NESTED MODE

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

a) When an interrupt request from a certain slave is in service this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IRQ's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

b) When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

5.14.3 AUTOMATIC ROTATION (EQUAL PRIORITY DEVICES)

In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of seven other devices are serviced at most once. (See Figure 5-11.)

There are two ways to accomplish Automatic Rotation using OCW2, the Rotation on Non-Specific EOI Command (R=1, SL=0, EOI=1) and the Rotate in Automatic EOI Mode which is set by (R=1, SL=0, EOI=0) and cleared by (R=0, SL=0, EOI=0).

5.14.4 SPECIFIC ROTATION (SPECIFIC PRIORITY)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IRQ5 is programmed as the bottom priority device, then IRQ6 will have the highest one.

The Set Priority command is issued in OCW2 where: R=1, SL=1; L0–L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command by using the Rotate on Specific EOI command in OCW2 (R=1, SL=1, EOI=1 and L0–L2=IRQ level to receive bottom priority).

5.14.5 POLL COMMAND

In this mode the INT output is not used and the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by software using a Poll command.

The Poll command is issued by setting P="1" in OCW3. The Interrupt Controller treats the next I/O read pulse to the Interrupt Controller as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from the I/O write to the I/O read.

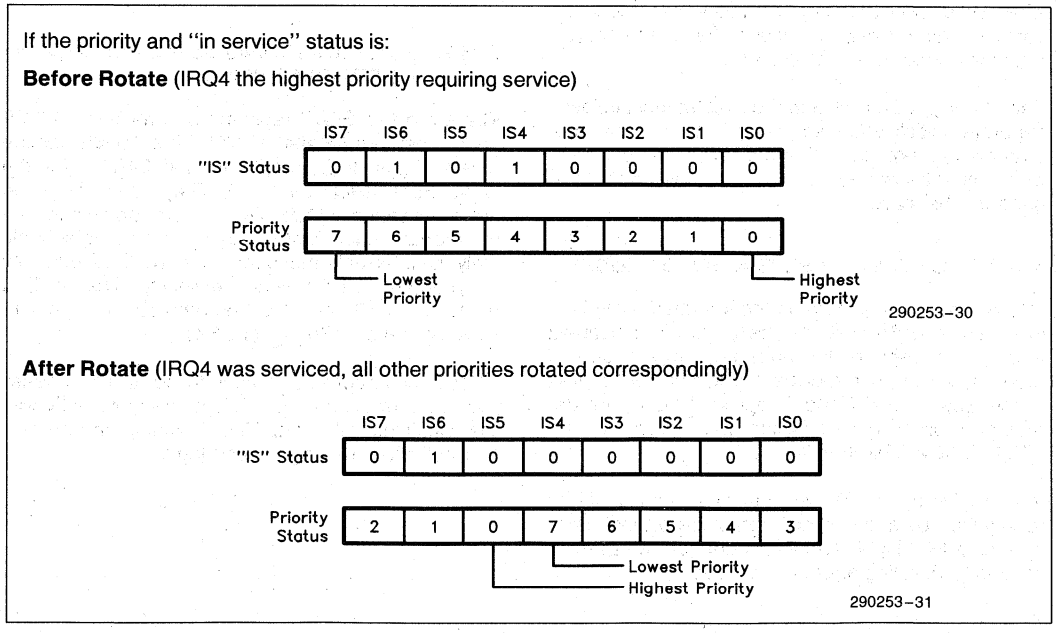


Figure 5-11

The word enabled onto the data bus during I/O read is:

D7	D6	D5	D4	D3	D2	D1	D0
1	--	--	--	--	W2	W1	W0

290253-32

W0–W2: Binary code of the highest priority level requesting service.

1: Equal to "1" if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the INTA# sequence is not needed (saves ROM space).

5.14.6 CASCADE MODE

The Interrupt Controllers in the EISA system are interconnected in a system of one master with one slave to handle up to 15 priority levels.

In a cascade configuration, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will enable the corresponding slave to release the device routine address during byte 2 of INTA#.

Each Interrupt Controller in the system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: one for the master and once for the slave.

5.14.7 EDGE AND LEVEL TRIGGERED MODES

In ISA systems this mode is programmed using bit 3 in ICW1. In EISA systems the LTIM bit is disabled and a new register for level and edge triggered mode selection, per interrupt input, is included. This is the Edge/Level Control register ELCR. The default programming is equivalent to programming the LTIM bit (ICW1 bit 3) to a 0.

If an ELCR bit = "0", an interrupt request will be recognized by a low to high transition on the corresponding IRQ input. The IRQ input can remain high without generating another interrupt.

If an ELCR bit = "1", an interrupt request will be recognized by a "low" level on the corresponding IRQ input, and there is no need for an edge detection. For level triggered interrupt mode, the interrupt request signal must be removed before the EOI command is issued or the CPU interrupt must be disabled. This is necessary to prevent a second interrupt from occurring.

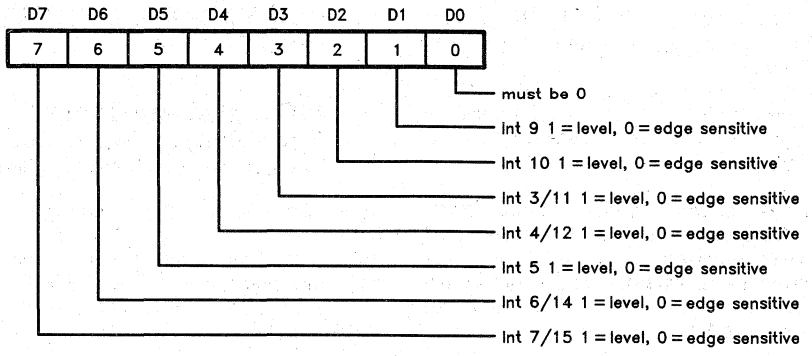
In both the edge and level triggered modes the IRQ inputs must remain active until after the falling edge of the first INTA#. If the IRQ input goes inactive before this time a DEFAULT IRQ7 will occur when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IRQ inputs. To implement this feature the IRQ7 routine is used for "clean up" simply executing a return instruction, thus ignoring the interrupt. If IRQ7 is needed for other purposes a default IRQ7 can still be detected by reading the ISR. A normal IRQ7 interrupt will set the corresponding ISR bit, a default IRQ7 won't. If a default IRQ7 routine occurs during a normal IRQ7 routine, however, the ISR will remain set. In this case it is necessary to keep track of whether or not the IRQ7 routine was previously entered. If another IRQ7 occurs it is a default.

5.14.7.1 Edge/Level Triggered Control Register (ELCR) (Read/Write)

There are two ELCR registers one for each 82C59 bank. They are located at I/O ports 04D0h (for the Master Bank, IRQ<0:1,3:7>) and 04D1h (for the Slave Bank, IRQ<8#:15>). They allow the edge and level sense selection to be made on an interrupt by interrupt basis instead of on a complete bank. Only the interrupts that connect to the EISA bus may be programmed for level sensitivity. That is IRQ (0, 1, 2, 8#, 13) must be programmed for edge sensitive operation. (See Figure 5-12.)

IRQ13 still appears externally to be an edge sensitive interrupt even though it is shared internally with the Chaining interrupt. The Chaining interrupt is "ORed" after the edge sense logic.

IRQ<7:0>—Port 4D0h (CNTRL-1)
IRQ<15:0>—Port 4D1h (CNTRL-2)



290253-33

These registers default to 0 upon RESET

Figure 5-12

5.15 Interrupt Masks

5.15.1 MASKING ON AN INDIVIDUAL INTERRUPT REQUEST BASIS

Each Interrupt Request input can be masked individually by the Interrupt Mask register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set 1. Bit 0 masks IRQ0, Bit 1 masks IRQ1 and so forth. Masking an IRQ channel does not affect the other channels' operation.

5.15.2 SPECIAL MASK MODE

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the Interrupt Controller would have inhibited all lower priority requests with no easy way for the routine to enable them.

That is where the Special Mask Mode comes in. In the Special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level and enables interrupts from all other levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectively enabled by loading the Mask register.

The special Mask Mode is set by OCW3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

5.16 Reading the Interrupt Controller Status

The input status of several internal registers can be read to update the user information on the system. The following registers can be read via OCW3 (IRR and ISR or OCW1 [IMR]).

Interrupt Request register (IRR): 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR).

In-Service register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt Command is issued.

Interrupt Mask register (IMR): 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 0).

The ISR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 1).

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the Interrupt Controller "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.

After initialization the Interrupt Controller is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever I/O read is active, the address is 021h or 061h (OCW1).

Polling overrides status read when P = 1, RR = 1 in OCW3.

6.0 NON-MASKABLE INTERRUPT PORTS

6.1 NMI Status and Control (Port 061h) (Read/Write)

Value on Reset = 00x00000 (Figure 6-1)

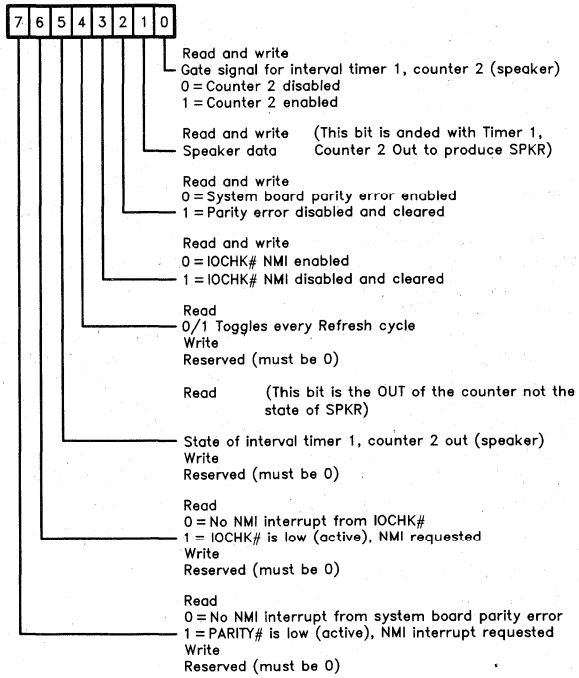
Parity Error from System Memory

Bit 7 is set if the system board drives PARITY# active. This Interrupt is enabled by setting bit 2 to "0". To reset the parity error, set bit 2 to "1" and then clear it to "0".

IOCHK# Error from Expansion Board Memory

Bit 6 is set if an expansion board drives IOCHK# active on the ISA/EISA bus. This interrupt is enabled by setting bit 3 to "0". To reset the interrupt, set bit 3 to 0 and then set it to "1".

Port—061h



290253-34

Figure 6-1

6.2 NMI Extended Status and Control (Port 0461h) (Read/Write)

Value on Reset = 00000000 (Figure 6-2)

This provides additional status and control.

Fail-Safe Timer Timeout

Bit 7 is set if the fail-safe timer count has expired before being reset by a software routine. This interrupt is enabled by setting bit 2 to "1". To reset the interrupt, set bit 2 to "0" then set it to "1".

Bus Timeout

Bit 6 is set if either a 64 BCLK or a 256 BCLK (Bus Timeout) occurs (refer to Section 4.1). The bus timeout interrupt is enabled by setting bit 3 to "1" or disabled by setting bit 3 to "0". To clear the bus timeout interrupt, set bit 3 to "0" and then set it to "1". The ISP drives RSTDRV active when a bus timeout occurs. Clearing the bus timeout status bit causes the ISP to negate RSTDRV.

Bit 4 indicates whether an 8 μs Bus Timeout occurred or not. For example, if Bit 6 = 1 and Bit 4 = 0, a 32 μs Bus Timeout occurred. If Bit 6 = 1 and Bit 4 = 1, an 8 μs Bus Timeout occurred.

Software Generated NMI

Bit 5 is set if an I/O write access occurred to port 0462h. This interrupt is enabled by setting 0461h bit 1 to "1". To reset the interrupt, set port 0461h bit 1 to "0" and then set it to "1".

Bus Reset

Bit 0 can be used to perform a system bus reset without resetting other devices in the system. A system bus reset is done by setting bit 0 to a "1", which drives the RSTDRV signal active on the ISA/EISA bus. Bit 0 should be set long enough for the system bus devices to be properly reset (8 BCLKs), and then bit 0 should be cleared to continue normal operation.

Bit 4 of Port 0461h has a new definition in the ISP B-stepping, which is shown in the Port 0461h Bit Map Table. The ISP stepping can be determined at system RESET:

Bit 4 = 1 (A-Step)

Bit 4 = 0 (B-Step)

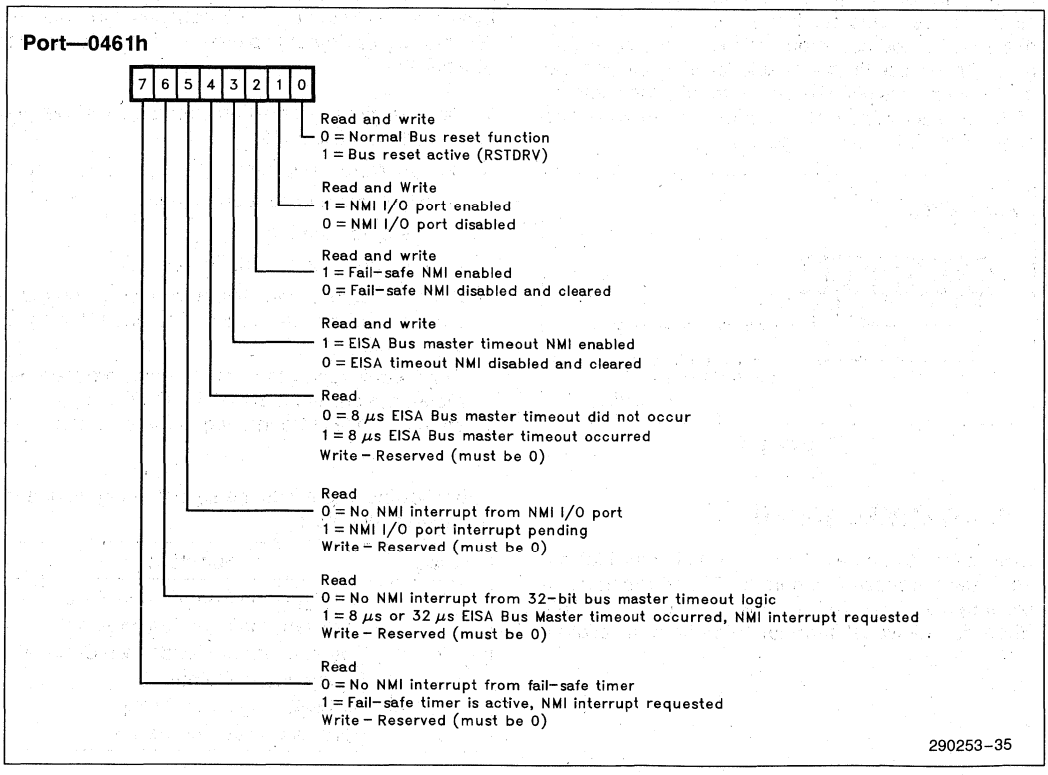


Figure 6-2

6.3 Software NMI Generation (Port 0462h) (Write Only)

A write to this port with any data will cause an NMI. This port provides a software mechanism to cause an NMI if interrupts are enabled. (Figure 6-3.)

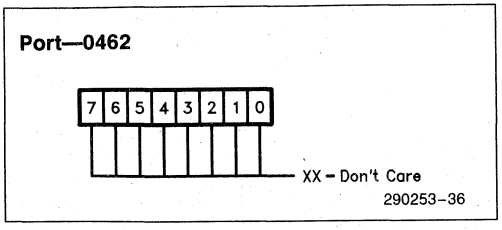


Figure 6-3

6.4 NMI Enable/Disable and Real-Time Clock Address (Port 070h)

The Mask register for the NMI interrupt is at I/O address 070h shown below. The most-significant bit enables or disables all NMI sources including IOCHK#, Fail Safe Timer, PARITY#, Bus Time Out, and the NMI Port. Write an 80h to port 70h to mask the NMI signal. This port is shared with the real-time clock. The real-time clock uses the lower six bits of this port to address memory locations. Writing to port 70h sets both the enable/disable bit and the memory address pointer. Do not modify the contents of this register without considering the effects on the state of the other bits. (Figure 6-4.)

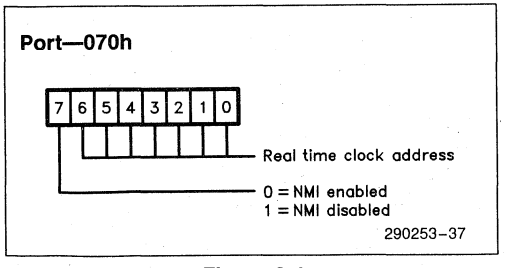


Figure 6-4

7.0 INTERVAL TIMER

EISA systems contain five counter/timers that are equivalent to those found in the 82C54 Programmable Interval Timer (NOTE: refer to the 82C54 Data Sheet for additional information on timer functions).

The following table shows the I/O address map of the interval timer counters:

I/O Port Address	Register Description
040h	Timer 1, System Timer (Counter 0)
041h	Timer 1, Refresh Request (Counter 1)
042h	Timer 1, Speaker Tone (Counter 2)
043h	Timer 1, Control Word Register
048h	Timer 2, Fail-Safe Timer (Counter 0)
049h	Timer 2, Reserved
04Ah	Timer 2, CPU Speed Control (Counter 2)
04Bh	Timer 2, Control Word Register

7.1 Programming the Interval Timer

The counter/timers are programmed by I/O accesses and are addressed as though they are contained in two separate 82C54 interval Timers. Timer 1 contains three counters, timer 2 contains two counters (EISA systems do not implement the middle counter of timer 2).

The interval timer is an I/O-mapped device. Several commands are available:

- Control Word Specifies:
  - — which counter to read or write
  - — the operating mode
  - — the count format (binary or BCD)
- Counter Latch latches the current count so that it can be read by the system. The countdown process continues.
- Read Back reads the count value, programmed mode, the current state of the OUT pins, and the state of the Null Count Flag of the selected counter.

The following Table lists the six operating modes for the interval counters.

Mode	Function
0	Out Signal on End of Count (=0)
1	Hardware retriggerable one-shot
2	Rate Generator (Divide by n Counter)
3	Square Wave Output
4	Software Triggered Strobe
5	Hardware Triggered Strobe

Because the timer counters wake up in an unknown state after power up, multiple refresh requests may be queued up. To avoid possible multiple refresh cycles after power up, program the timer counter immediately after power up.

Programming the interval timer is a simple process:

1. Write a control word.
2. Write an initial count for each counter.
3. Load the least and/or most significant bytes (as required by Control word bits 5, 4) of the 16-bit counter.

7.1.1 INTERVAL TIMER CONTROL WORD FORMAT

The Control Word specifies the counter, the operating mode, the order and size of the COUNT value, and whether it counts down in a 16-bit or binary-coded decimal (BCD) format. After writing the control word, a new count may be written at any time. The new value will take effect according to the programmed mode.

If a counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same counter. Otherwise, the counter will be loaded with an incorrect count. The count must always be completely loaded with both bytes. (See Figure 7-1.)

7.1.2 INTERVAL TIMER COUNTER LATCH COMMAND

The Counter Latch command latches the count at the time the command is received. This command is used to insure that the count read from the counter is accurate (particularly when reading a two-byte count). The count value is then read from each counter's Count register as was programmed by the Control register. (See Figure 7-2.)

Timer 1: Port 043h
Timer 2: Port 04Bh

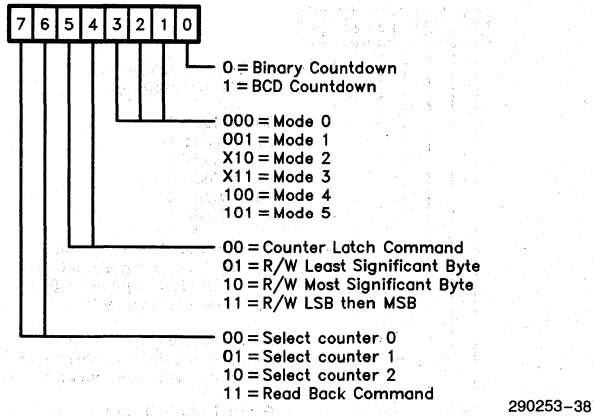


Figure 7-1. Control Word

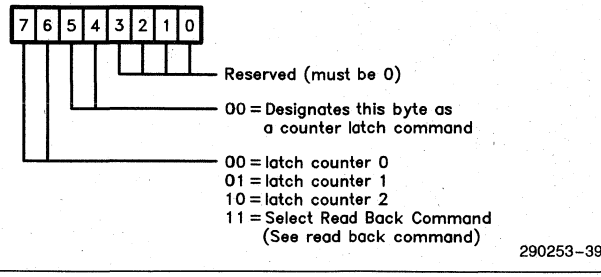


Figure 7-2

7.1.3 INTERVAL TIMER READ BACK COMMAND

The Read-Back command is used to determine the count value, programmed mode, and current states of the OUT pin and Null Count flag of the selected counter or counters. The Read-Back command is written to the Control Word register, which causes the current states of the above mentioned variables to be latched. The value of the counter and its status may then be read by I/O access to the counter address. Figures 7-3 and 7-4 show the format for the Read-Back command and the Status byte.

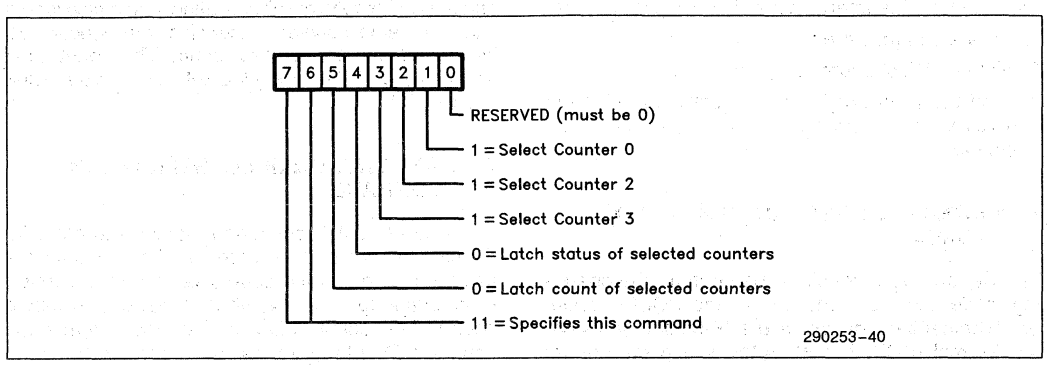


Figure 7-3. Current Read-Back Command Format

Timer 1: Port 040h, 041h, 042h
Timer 2: Port 048h, 04Ah

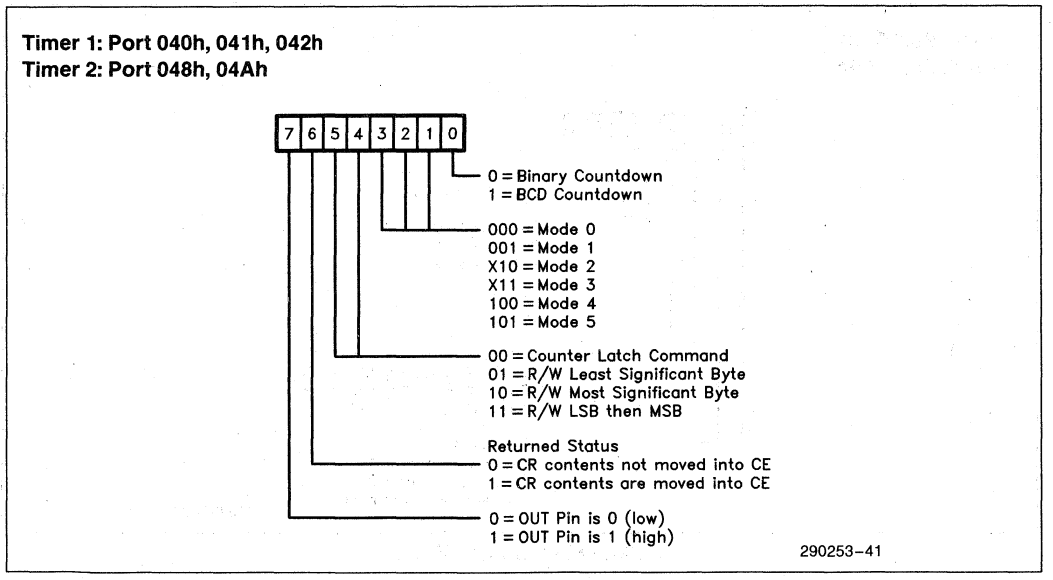


Figure 7-4. Status Byte Format

8.0 DETAILED SIGNAL DESCRIPTION

EISA Bus Cycle/BCLK timing definition points. (Figure 8-1.)

NOTE:
The following definition points are to be used as a reference when reading the detailed signal description.

T0 = rising edge of BCLK at the beginning of START# active.

T1 = falling edge of BCLK at the middle of START# active.

T2 = rising edge of BCLK at the beginning of CMD# active.

T3 = the first falling edge of BCLK after CMD# goes active.

T4A = the first rising edge of BCLK after CMD# goes active.

T4 = rising edge of BCLK at the end of a cycle (CMD# inactive edge).

T5 = the first falling edge of BCLK after CMD# goes inactive.

8.1 Signals Used during Arbitration

DREQ <3:0,7:5> INPUT (DMA REQUEST)

These input lines are used to request DMA service from the DMA subsystem or for a 16-bit ISA master to gain control of the system bus. The active level (high or low) is programmed in the Command registers. When the Command register bit 6 is programmed to 0, they are active high, otherwise they are active low. All inactive to active edges of DREQ are assumed to be asynchronous, they will be sampled on the falling edge of BCLK and used on the next rising edge of BCLK. The request must remain active until the appropriate DACK# goes active. Active to inactive edge sampling in Demand mode or in the case of a cascaded master will have various sampling points as defined below:

ISA Master—The DREQ line is sampled on the rising edge of BCLK, two BCLKs before the DACK# is inactivated. The DREQ is assumed to be asynchronous. The various address and control lines must be floated before DACK# goes inactive and the MASTER16# line must be released as DACK# goes inactive.

Compatible and Type A DMA—The DREQ line is sampled on the rising edge of BCLK, two BCLKs prior to the sampling of DRDY. This is 2.5 BCLKs before the address would be changed. The DREQ is assumed to be asynchronous. (Figure 8-2.)

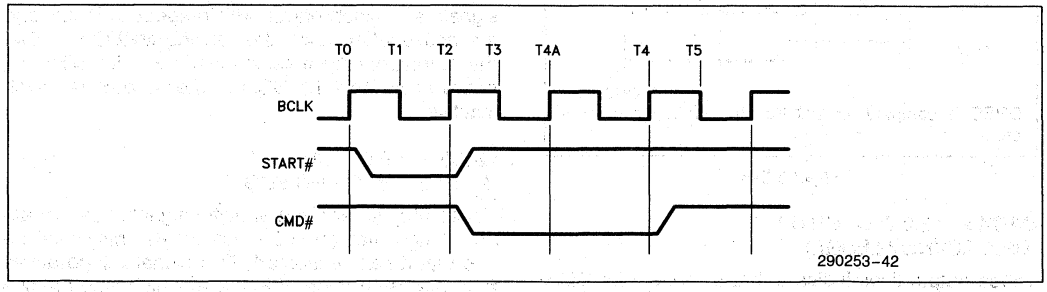


Figure 8-1

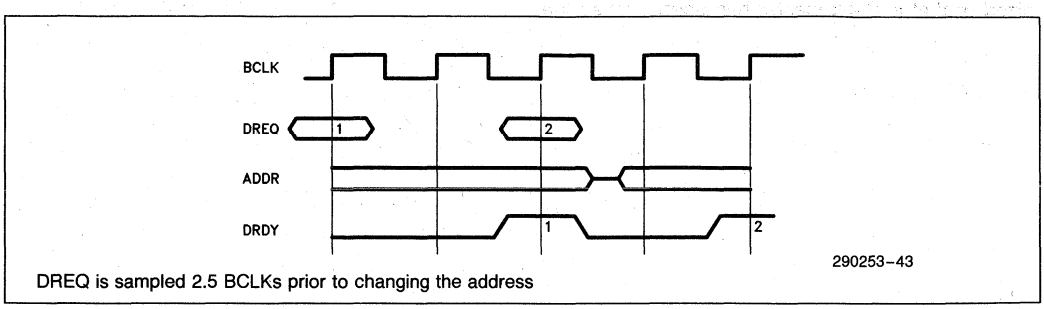
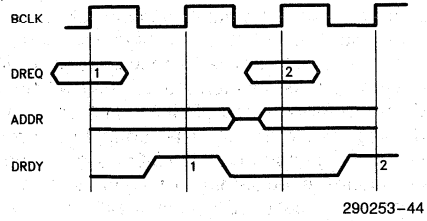


Figure 8-2

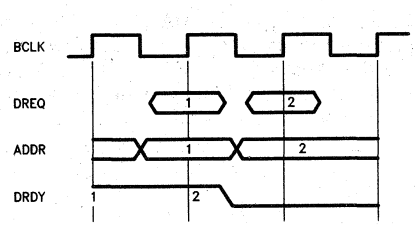
Type B DMA—The DREQ line is sampled on the rising edge of BCLK, 1 BCLK prior to the sampling of DRDY. This is 1.5 BCLKs before the address would be changed. The DREQ is assumed to be asynchronous. (Figure 8-3.)



DREQ is sampled 1.5 BCLKs prior to changing the address

Figure 8-3

Burst DMA—The DREQ line is sampled on the rising edge of BCLK, ½ BCLK after the address is changed on the bus. The DREQ must be synchronous to BCLK. (Figure 8-4.)



DREQ is sampled 0.5 BCLKs after changing the address

Figure 8-4

DACK# <3:0,7:5> OUTPUT
(DMA ACKNOWLEDGE)

These output lines indicate that a request for DMA service from the DMA subsystem has been recognized or that a 16-bit master has been granted the bus. The level of the DACK# lines when active may be programmed to be either high or low. This is accomplished by programming the DMA Command register. These lines should be used to decode the DMA slave device with the IORC# or IOWC# line to indicate selection. If used to signal acceptance of a bus master request, this signal indicates when it is legal to pull MASTER16# low. If the DMA controller has been programmed for a timing mode other than compatible mode, and another device has requested the bus, and a 4 μs time has elapsed, this line will be set inactive and the transfer stopped before the transfer is complete. In this case, the transfer will be restarted at the next arbitration period in which the channel wins the bus. Upon RST these lines are set inactive.

MREQ# <5:0> INPUT (MASTER REQUEST)

These inputs are slot specific signals used by EISA bus masters to request bus access. This signal, once set active, must remain active until the MACK# line indicates that the bus has been granted. The MREQ# line is to be negated on the falling edge of BCLK at or slightly before the end of a master transfer. The LA<>, BE<>#, M-IO#, and W/R lines should be floated on or before the rising edge of BCLK after MREQ# is negated. The end of the last bus cycle is determined by the arbiter via the activation of ST3 (CIP#) which is derived from CMD# in this case. The MREQ# signals are also activated on the falling edge of BCLK, they are always sampled on the rising edge of BCLK. These signals are synchronous with respect to BCLK and are active when low. After driving MREQ# active, the corresponding master must not drive MREQ# active again until 1.5 BCLKs after CMD# is driven inactive.

MACK# <5:0> OUTPUT
(MASTER ACKNOWLEDGE)

These outputs are slot specific signals used to acknowledge that an EISA bus master may use the bus that it has requested. This signal will go active from the rising edge of BCLK at which time the bus master may begin driving the LA<>, BE<>#,

M-IO#, and W/R lines on the next falling edge of BCLK. MACK# will remain active until the rising edge of BCLK when MREQ# is sampled inactive. This line is sampled by EISA masters on the falling edge of BCLK. If another device has requested the bus, this line will be set inactive before MREQ# goes inactive. When the MACK# line goes inactive, the granted device has a maximum of 8 μs to release the MREQ# line and begin a final bus cycle. The MACK# signals will go active from the same rising edge of BCLK that EXMASTER# is activated. These signals are active when low. Upon reset they will be set inactive.

NOTE:
The ISP will deassert the MACK# signal a minimum of one BCLK after asserting it if another device (or refresh) is requesting the bus.

REFRESH# I/O TRI-STATED (REFRESH)

This signal is bidirectional. REFRESH# is used as an output during a refresh to indicate that a refresh cycle is in progress. It should be used to enable the SA<15:0> (or LA<15:2>) address to the row address inputs of all banks of dynamic memory so that when MRDC# (or CMD#) goes active, the entire system memory is refreshed at one time. Memory slaves must not drive any data onto the bus during refresh and should not add wait states since this would affect the entire system throughput. As an output, this signal is driven directly onto the EISA bus. This signal is an output only when the ISP is a master on the bus responding to an internally generated request for Refresh. Upon RST this pin will tri-state.

REFRESH# may be driven by an expansion bus adapter card acting as a 16-bit ISA bus master. As an input, it is assumed to be asynchronous with respect to BCLK. It will be sampled on the falling edge of BCLK and used on the following rising edge of BCLK.

DHLDA INPUT (HOLD ACKNOWLEDGE)

This signal indicates that the system has granted the ISP arbitration logic the use of the host bus. DHLDA is an asynchronous signal with respect to BCLK. It is sampled on the falling edge of BCLK and used on the next rising edge of BCLK. This signal is active high.

CPUMISS# INPUT (CPU CACHE MISS)

This signal from the host CPU or Cache controller subsystem, indicates that a CPU or Cache controller bus cycle is pending and that the System arbitration logic in the ISP should include the CPU as a contender in the next bus arbitration. This line may be tied active in a cacheless system. This signal is asynchronous with respect to BCLK. It is sampled on the falling edge of BCLK and used on the next rising edge of BCLK. It is an active low signal.

DHOLD OUTPUT (CPU HOLD REQUEST)

This output signal is used to request control of the host or differentiation bus. In a simple system this would be making a request from the CPU, in a system with a cache, from the cache. This signal is synchronized to the requirements of the CPU or cache by the EBC. Upon RST this pin is set inactive.

EMSTR16# OUTPUT (ISA 16-BIT MASTER)

This output signal is used to tell the bus controller that an ISA 16-bit master has control of the bus. This will enable the bus controller to do the appropriate translation of signals. This signal is driven active 1 BCLK prior to the BCLK that the ISA Master DACK# signals are driven active, and driven inactive with the same BCLK that the ISA DACK# signals are changed on. This signal is active low. It is set inactive upon RST. EMSTR16# from the ISP must be ANDed with SPWROK from the EBC to generate EMSTR16# to the EBC so that EBC internal signals are initialized before START# is driven active in the first cycle.

EXMASTER# OUTPUT (EISA MASTER)

This output signal is used to tell the bus controller that a new EISA bus master, 16- or 32-bit, has control of the bus. This will enable the bus controller to do the appropriate translations of signals. This signal is both driven active and inactive with the MACK# signals on the rising edge of BCLK. Upon RST this pin is set inactive.

8.2 Signals Used during DMA and Register Access

START# INPUT (START OF CYCLE)

This signal is connected directly to the EISA START#. It is used only in slave mode to indicate the start of a bus cycle. It is sampled on the rising edge of BCLK.

CMD# INPUT (COMMAND)

This signal is connected directly to the EISA CMD#. CMD# is used in slave mode to determine when to tri-state the data buffers after a read cycle. The data buffers will be floated when CMD# goes inactive. The leading edge of this signal is always synchronous, driven from a rising edge of BCLK. The trailing edge is asynchronous only in the case of an ISA bus master.

EOP TRI-STATED (END OF PROCESS)

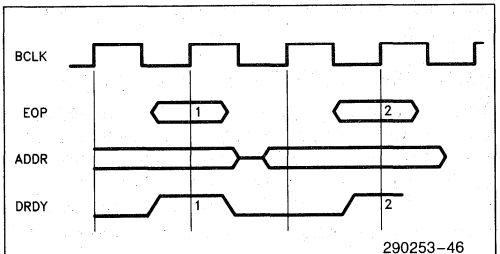
This pin is bidirectional, acting in one of three modes, and is directly connected to the TC line of the ISA/EISA bus. In the first mode, EOPIN, the pin is an input and can be used by a DMA slave to stop a DMA transfer. In the second mode, TCOUT, it is

used as a terminal count output by DMA slaves. An active pulse is generated when the byte counter reaches its last value. In the third mode, INTOUT, it indicates to an EISA programming master that a chaining buffer has been expired and a new chain buffer should be programmed. The timings are as follows:

Eopin Mode

During DMA, using Compatible, Type A or Type B transfers, the EOP pin is sampled by the ISP on the rising edge of BCLK preceding the falling edge that address is changed. If it is sampled active, the address is tri-stated on the falling edge of BCLK, and the transfer is terminated. (Figure 8-5.)

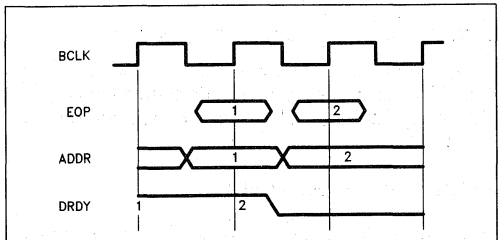
When using Burst cycles, the EOP is sampled at the same time as the DREQ input, ½ BCLK after the address was changed. (Figure 8-6.)



290253-46

EOP is sampled 0.5 BCLKs prior to changing the address

Figure 8-5



290253-47

EOP is sampled 0.5 BCLKs after changing the address

Figure 8-6. Burst Mode EOP Timing

Tcout Mode

In this mode the EOP output behaves differently depending on whether or not the channel is programmed for Burst mode. If the channel is not programmed for Burst mode, the EOP output will go active 1.5 BCLKs after a new address has been output if the byte count expires with that transfer. In burst mode, EOP is driven active along with address since some transfers may only last 1 BCLK cycle. The EOP (TC) will be active until AEN# goes inactive for both Burst and non-Burst DMA (even after the address is tri-stated), except in the following two cases: Case 1—EOP (TC) will go inactive four BCLK cycles before AEN# goes inactive during an Autoinitialization or Buffer Chaining function. Case 2—EOP (TC) will go inactive one BCLK cycle before AEN# goes inactive if the DMA channel that is currently in use is configured for Type "B" timing.

Intout Mode

In this mode the EOP signal has the same behavior as the Chaining Interrupt to the host processor (IRQ13). If a chaining buffer is expired, EOP will go active on the falling edge of BCLK. Only the currently active channel's chaining interrupt will be reflected on this pin. Other channels with active chaining interrupts pending will not affect the EOP pin.

Whenever all the DMA channels are not in use, the EOP pin is kept in output mode and inactive.

BE<3:0># I/O TRI-STATED (BYTE ENABLES)

BE<3:0># are the byte enables of EISA bus. BE<2:0># are bidirectional and BE3# is an output only. In master mode BE<3:0># are outputs and are translated by the EBC to HBE<3:0># or ISA signals, whichever are appropriate. In Master mode the ISP will directly drive the EISA bus. The BE lines are always tri-stated on the T3 BCLK. The BE lines are re-enabled by the DMA on the falling edge of BCLK if DRDY is sampled active on the previous rising edge of BCLK.

In slave mode BE<2:0># are used in conjunction with HA<15:2> to address the ISP's internal registers. BE<2:0># are latched on the T2 edge of BCLK.

HA<31:2> I/O TRI-STATED (HOST ADDRESS BUS)

These signals are Address signals and are connected to the host bus. HA<31:20> and <15:2> are bidirectional, HA<19:16> are output only. In master mode they are outputs and are used in conjunction with BE<3:0># during DMA and Refresh cycles. In slave mode HA<15:2> and <31:20> are inputs. HA<15:2> in conjunction with BE<2:0># are used to address the internal ISP registers. HA<15:2> are latched with transparent latches by the trailing (rising) edge of DADS# (ST0). HA<31:20> are used to decode GT1M#. Upon RST these pins are tri-stated and placed in input Mode.

HW/R# I/O TRI-STATED (HOST WRITE/READ)

This line is bi-directional and connects to the host bus. In slave mode HW/R# is an input and is used to indicate a read cycle when low and a write cycle when high. When in slave mode, HW/R# is latched with transparent latches by the trailing (rising) edge of ST0 (DADS#). In master mode this pin is an output and will have the same timings as HA<31:2>. As an output, it is used during DMA or Refresh cycles by the EBC to propagate the appropriate write/read signals to the EISA Bus. This pin is placed in output mode only during DMA or Refresh cycles. Upon RST this pin is tri-stated and placed in input mode.

GT16M# OUTPUT TRI-STATED (>16 MBYTES)

GT16M# is directly connected to the EBC and indicates that the DMA address on HA<> is greater than 00FFFFFF (16 Mbytes). This signal is an output only when the DMA is the bus master, it is tri-stated otherwise.

The EBC uses this signal during DMA cycles to determine whether or not to generate ISA memory command signals. If it is not driven by another source, it should be pulled up with an external resistor to keep the node, which is an input to the EBC, from floating. This signal has timings similar to the HA<31:2> lines. Upon RST it will be tri-stated.

GT1M# OUTPUT (>1 MBYTE)

GT1M# is directly connected to the EBC and indicates when the address on the HA<> address lines is greater than 000FFFFFh (1 Mbyte). It is decoded by the ISP from the HA<31:20> bus and is used by the bus controller to control the SMRDC# and SMWTC# signals. Its timing is a combinatorial delay from HA<31:20> in all cycles. Upon RST its state depends on the state of the address bus.

RST INPUT (SYSTEM RESET)

This signal is connected directly to the EBC and indicates that the ISP should initialize all of its registers and state machines as well as driving the RSTDRV output active. This signal is asynchronous with respect to BCLK and must have a minimum active pulse width for eight BCLKs. 3.5 BCLKs are required from the inactive edge of RST to the T2 edge of the first slave cycle to the ISP.

BCLK INPUT (BUS CLOCK)

This signal is used as the main clock reference for the ISP and the EISA bus. BCLK is tied directly to the BCLK output of the EBC. The EBC divides the Host CPU Clock (HCLKCPU) by an appropriate number to generate BCLK. The normal frequency range of BCLK is 8.00 MHz to 8.333 MHz with a normal duty cycle of 50%. The high or low time can be stretched by the EBC for synchronization purposes, (REFER TO THE EBC DATA SHEET).

ST<3:0> I/O TRI-STATED WITH WEAK PULLUP (DMA STATUS)

These are synchronous bidirectional control signals between the ISP and the EBC. These pins are output during DMA and refresh cycles. They indicate what type of timing has been programmed for the current cycle and the size of the I/O device involved in the DMA transfer. The EBC takes ST0–ST3 and generates the appropriate command signals for the ISP. The command signals generated from ST0–ST3 are used to control the timing of the cycle. During cycles where the ISP is in slave mode, these pins are inputs. All of these pins have weak pullup resistors to sustain a high level when neither the EBC or ISP are driving the signals. Their definition is as follows:

Master Mode

ST1	ST0	DMA Cycle Timing
0	0	Compatible Cycle Timing
0	1	Type A Timing
1	0	Type B Timing
1	1	Burst Timing

These signals always transition on the falling edge of BCLK.

ST3	ST2	DMA Slave Size or Idle Indication
0	0	8-Bit
0	1	16-Bit
1	0	32-Bit
1	1	Idle

These signals will always go active on the falling edge of BCLK. They will go inactive on the falling edge of BCLK after the last DMA transfer, or on the rising edge of BCLK if a page break is detected (burst mode) or the transfer is about to end (burst mode).

Slave Mode

ST0 has the DMA address strobe function. This signal is used to latch the address and status information on the host bus. Specifically the signals latched with this signal are HA<15:2>, ST1 (M/IO#) and HW/R#. These signals have a setup and hold time around the rising edge of ST0. The leading edge of ST0 also has a setup time to BCLK at T2.

ST1 is the memory or I/O indicator pin (M/IO#), 1=Memory Cycle, 0=I/O Cycle. This signal is latched by the rising edge of ST0.

ST2 is the Interrupt Acknowledge pin (INTA#). When ST2=0, the cycle is an Interrupt Acknowledge. When ST2=1, the cycle is not an Interrupt Acknowledge. This signal is sampled by the ISP in slave mode at the T2 BCLK edge. It has a setup and hold time with respect to that BCLK edge so it must not be pipelined along with the HA lines. Only non-ISA masters can run interrupt acknowledge cycles. Only the host CPU has the control signals to run interrupt acknowledge cycles. ST2 must be tied high with a 1.2K pullup resistor.

ST3 is the cycle in progress pin (CIP#). This is an indication to the ISP System Arbiter that there is a bus cycle in progress. It is used to detect when the bus is free after an EISA master gives it up by driving MREQ# inactive. This signal is sampled on rising edges of BCLK.

These pins are output only when the DMA is the bus master or during Refresh cycles. Upon RST these pins are tri-stated and placed in input Mode.

DRDY I/O TRI-STATED WITH WEAK PULLUP (READY SIGNAL)

This pin is an output in slave mode and an input in master mode. In slave mode it is used by the EBC as an end of cycle indicator and is used to combinatorially drive CHRDY. DRDY is actively driven only when the ISP detects a slave cycle to one of its registers. At all other times it is sustained in a high state internally by a very weak pullup resistor. It is actively driven from the T2 edge of BCLK of an ISP cycle that requires more than a two BCLK wide CMD# pulse width when an ISA device is the bus master. DRDY will not be driven low when the bus master is either the host CPU or an EISA master. If DRDY is driven low, it will be kept low for four BCLKs after the T2 edge of BCLK. It will then be driven high off the rising edge of BCLK. The DRDY output buffer is enabled at T2 and disabled as a result of an inactive CMD# after T4A. DRDY will be asynchronously disabled upon the inactive edge of CMD#.

In master mode DRDY is used to indicate to the DMA controller that the current cycle is completed (DMA-READY) and that the DMA controller should pipeline addresses for Burst DMA transfers (START-PIPELINING). In master mode, DRDY, is sampled 2.5 BCLK periods after ST<3:2> are driven to a non-idle state from an idle one. This indicates START-PIPELINING. If DRDY is sampled high at this point, the DMA controller begins address pipelining. Pipelining is stopped at a page break or at the end of a transfer. DRDY is then sampled on the next BCLK rising edge and on all subsequent BCLK rising edges until ST<3:2> go to idle, to determine DMA-READY. If the DMA is pipelining addresses, DRDY will be sampled until active one last time after ST<3:2> go to the idle state (to complete the last DMA cycle). Upon RST this pin is tri-stated and must be tied high with a 2.4K pullup resistor.

D<7:0> I/O TRI-STATED (DATA LINES)

This bidirectional bus is the ISP's slave mode data bus. In master mode the data lines are not used. These pins are output only when CSOUT# is active during I/O read or Interrupt Acknowledge cycles. That is only when the ISP is being accessed as a result of an I/O read or Interrupt Acknowledge. These pins are inputs during an I/O write cycle to the ISP's internal registers. Upon RST these pins are tri-stated and placed in input mode.

This bus needs to be externally buffered to provide adequate drive to the EISA bus. In ISP master mode, this bus is not used. Upon RST these pins are tri-stated and placed in input Mode.

CSOUT# OUTPUT (SLAVE MODE SELECTED)

This output signal from the ISP indicates when the ISP is being accessed in slave mode. CSOUT# will go active from the rising edge of BCLK at T2 after a slave cycle to or from the ISP has been detected. It will not be active prior to that, i.e., between T0 and T2. CSOUT# is deactivated as a result of an inactive edge of CMD#, this is done asynchronously. CSOUT# will also be active during all interrupt acknowledge cycles. CSOUT# should be used to control the output enable of an EISA local data bus transceiver. This pin is always an output. Upon Reset it will be inactive (high).

AEN# OUTPUT (ADDRESS ENABLE)

This output signal, when inactive (high), indicates that the CPU or another EISA or ISA bus master has control of the bus. When active (low), the DMA controller has control of the bus. AEN# will also go active during Refresh cycles. AEN# is used to disable I/O devices which must not respond during a DMA cycle. This signal is further processed by the system board to form the slot specific AEN# signals. Upon RST this pin is set inactive.

8.3 Signals Used during an Interrupt Sequence

IRQ <15:3,1> INPUT (INTERRUPT REQUEST)

The active polarity of these inputs depends on the programming of the two ELCR registers. Only the EISA bus interrupts may be programmed for active low operation in the ELCR registers, all other interrupts will always be edge-triggered. If an interrupt is programmed for active low operation and shared with open collector drivers, an external pullup will need to be included to restore the inactive high level. Upon RST, the IRQ lines will be placed in edge-triggered mode.

The IRQ inputs must remain active until after the first falling edge of INTA(ST2#). If the IRQ input goes inactive before this time, a DEFAULT IR7 will occur when the CPU acknowledges the interrupt. (Refer to Section 5.14.7.)

INT OUTPUT (CPU INTERRUPT)

This signal is driven by the ISP's interrupt controller subsystem to signal the CPU that an Interrupt request is pending and needs to be serviced. It is asynchronous with respect to BCLK and is always an output. For i486 CPU systems, INT must be externally latched from the ISP until an Interrupt Acknowledge cycle occurs. INT will remain active until after the first INTA(ST2#) pulse. Upon RST the state of this pin is undefined.

8.4 Counter/Timer, NMI and Miscellaneous Utility Signals

OSC INPUT (COUNTER CLOCK)

This is a 14.31818 MHz input clock for use by the timers. This clock is divided by 12 and 48 to form the CLK input on several of the counters. It is asynchronous with respect to BCLK.

IOCHK# INPUT (I/O CHECK BUS ERROR)

This input signal comes from the ISA bus. It is used for parity errors on memory cards plugged into the bus, and for other high priority interrupts. The system board needs a pullup resistor to keep this signal inactive when it is not being driven active by some device. The active **level** of this signal will generate an NMI if this function is enabled in Port 061h. For example, if IOCHK# is active and Port 061h, bit 3 is set to 1 (inactive) no NMI will be caused, however, when Port 061h bit 3 is set to 0 (active) an NMI will be caused, providing that the IOCHK# input is still active and Port 070h bit 7 is enabled (0). This signal is asynchronous with respect to BCLK.

RSTDRV OUTPUT (SYSTEM BUS RESET)

This output signal is used by the EISA bus to disable and reset all installed adapters. More specifically, ISA/EISA bus masters and DMA devices must release and stop requesting the bus until reprogrammed by the system. Any drivers that are connected to the system bus must be tri-stated by this signal. In addition to being activated by RST, RSTDRV can be activated by a write to port 0461h or a bus timeout. A bus timeout will occur if an EISA bus master exceeds its allowable 8 μs time on the bus after being requested to get off by the removal of its MREQ#. Once the RSTDRV is activated by the bus timeout, the timed out master will release the bus allowing the host CPU to gain control of the system. A Bus Timeout will also occur upon the timeout of the 32 μs CMD# active timer. This output drives the EISA bus directly and has a 24 mA drive capability. RSTDRV will remain active as long as the source of the Bus Reset is active.

NOTE:
The 8x42 Keyboard Controller requires clocks (CLKKB) before the end of RSTKBD# (keyboard reset pulse) for keyboard reset to occur. To ensure that RSTKBD# has been driven inactive after the EBC has begun generating CLKKB; RSTDRV and RSTCPU are logically NANDed to generate RSTKBD# for the 8x42.

PARITY# INPUT (PARITY ERROR)

This signal is the main memory parity error input from the system board. The system board reports any parity errors from its memory system on this line. The active **edge** of this signal will generate an NMI if this function is enabled in Port 061h. For example, if PARITY# is active and Port 061h, bit 2 is set to 1 (inactive) no NMI will be caused, and when Port 061h, bit 2 is set to 0 (active) an NMI **will not** be caused even if Port 070h bit 7 is enabled (0). Another active edge on the PARITY# line would need to be detected in order to generate an NMI. This signal is asynchronous with respect to BCLK.

NMI OUTPUT (NON-MASKABLE INTERRUPT)

This signal from the NMI logic is used to force a non-maskable interrupt to the CPU. The CPU registers an NMI when it detects a rising edge on NMI. NMI will remain active until a read from the CPU to one of the NMI registers is detected by the ISP. This signal is set to low upon RST.

SPKR OUTPUT (SPEAKER DRIVE)

This signal is the output of timer 1, counter 2 and is "ANDed" with Port 061h bit 1 to provide Speaker Data Enable. This signal is to be used to drive an external speaker driver device, which, in turn drives the EISA system speaker. SPKR has a 24 mA drive capability. Upon reset, its output state is undefined.

SLOWH# OUTPUT (SLOW DOWN HOST CPU)

This output signal is from the CPU slowdown timer counter OUT, timer 2/counter 2. This counter is used to slow down the main CPU by pulse width modulation of its execution via the CPU's HOLD pin. When the first read is done to an I/O register in the 48h–4Bh range, the SLOWH# pin will be released to follow the output of the SLOWH# timer counter. Counter 2 is triggered by the refresh-request signal generated from timer 1/counter 1. Upon RST, SLOWH# will be set inactive (high).

NOTE:
Refresh cycles will not necessarily be generated during the time the SLOWH# signal is active, the Arbiter will determine when the refresh cycle will be placed on the bus.

RTCALE OUTPUT
(REAL TIME CLOCK LATCH ENABLE)

This active high output signal is provided by the ISP to latch the appropriate memory address into the Real Time Clock. A write to port 070h with the appropriate Real Time Clock memory address that will be written to or read from, will cause RTCALE to go active. RTCALE will go active from the rising edge of BCLK at T2 and will remain active for 1.5 BCLKs. The address is latched on the falling edge of RTCALE. Upon RST this pin is set inactive. (Figure 8-7.)

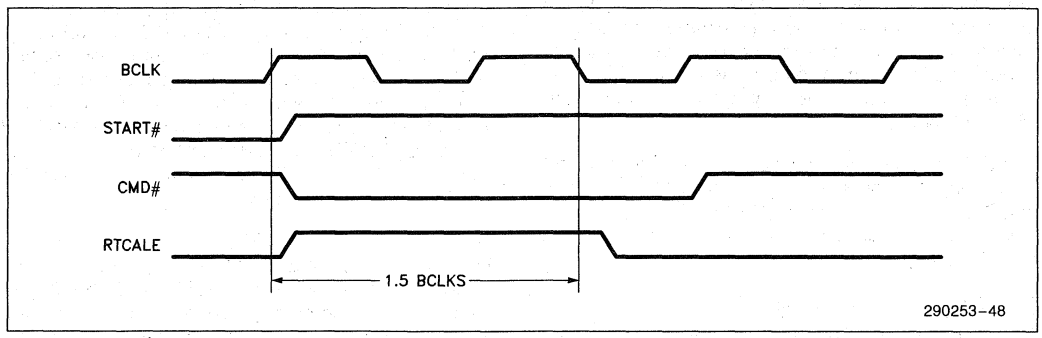


Figure 8-7

9.0 BASIC FUNCTION TIMING DIAGRAMS

The following general notes apply to all of the Basic Function timing diagrams:

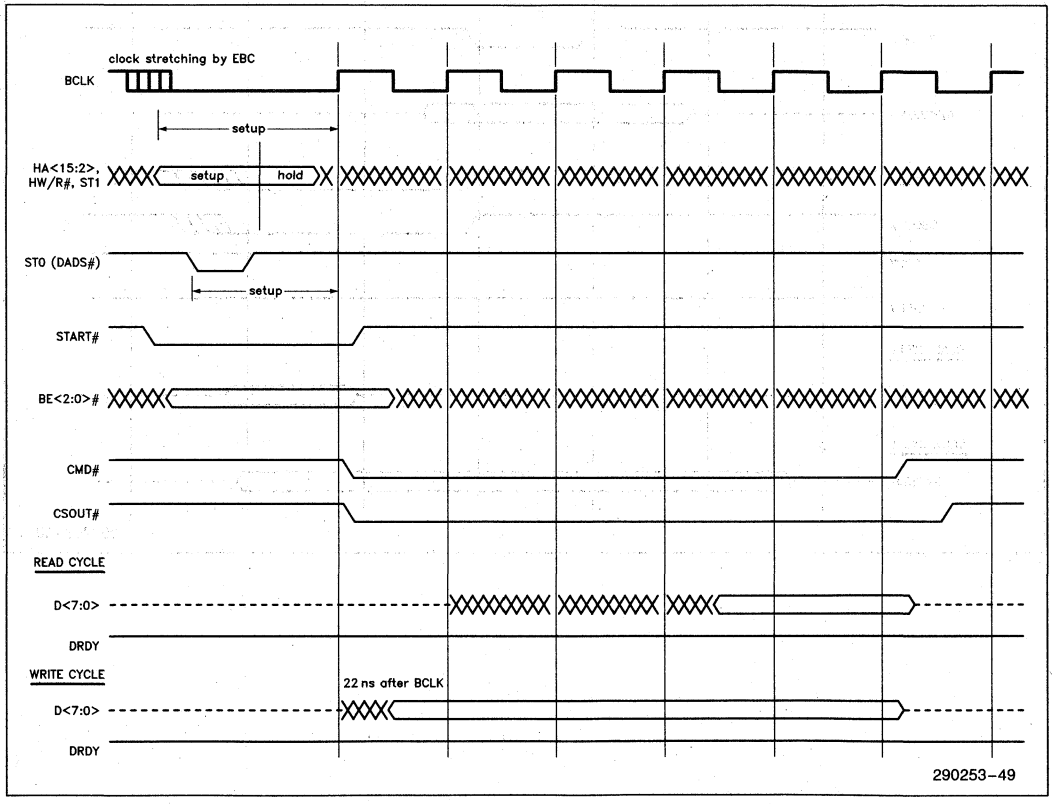
1) * = ISP Sampling point

2) x = Non-sampling points/don't care/Invalid Data

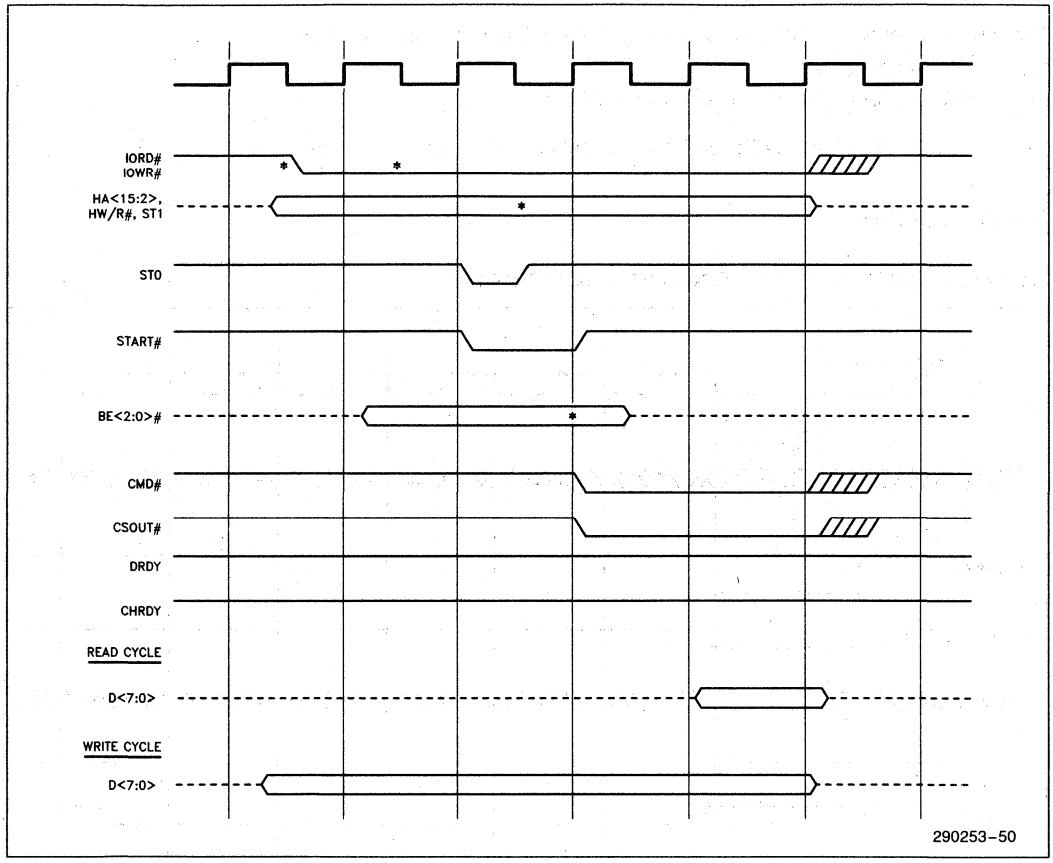
Refer to the 82358 EISA Bus Controller (EBC) data sheet for more timing diagrams.

9.1 Slave Mode

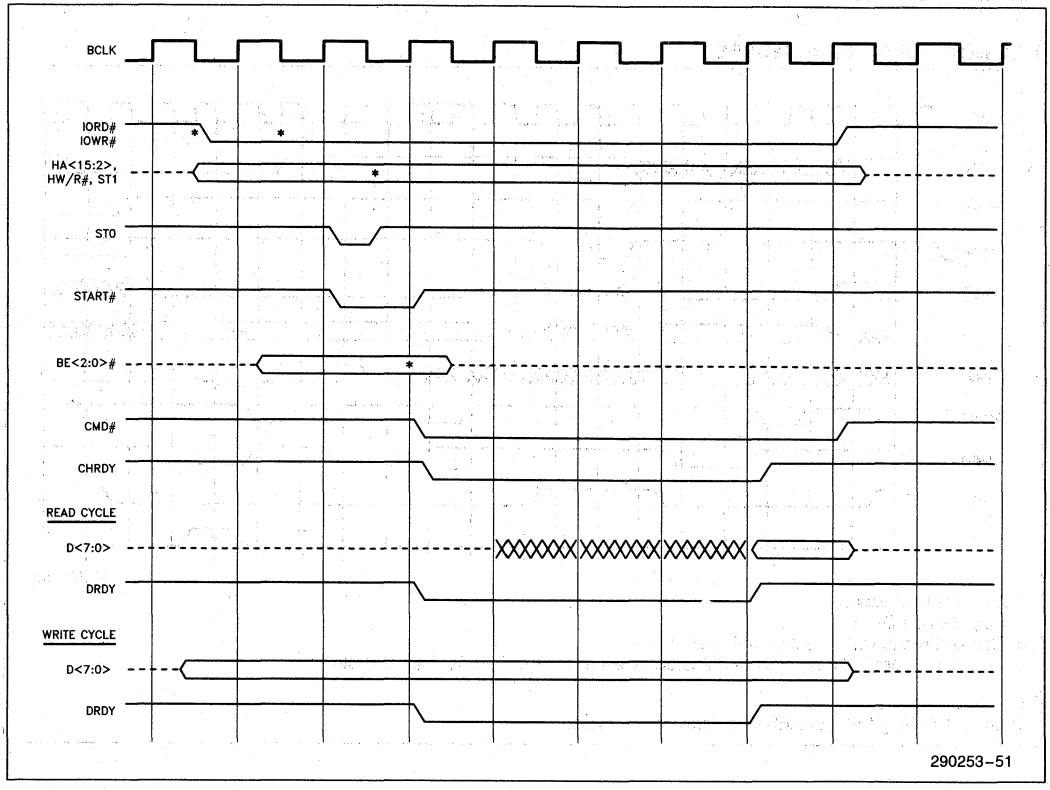
9.1.1 EISA MASTER OR HOST CPU CYCLE TO/FROM ISP



9.1.2 ISA MASTER SLAVE CYCLE TO/FROM THE ISP—SHORT CYCLE

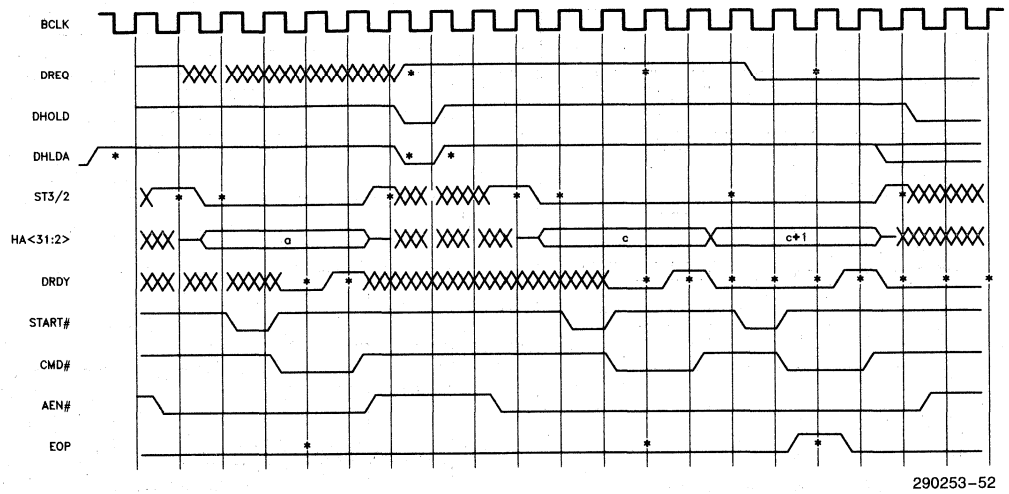


9.1.3 ISA MASTER SLAVE CYCLE TO/FROM THE ISP—LONG CYCLE



9.2 DMA Master Mode

9.2.1 NON BURST DMA CYCLES

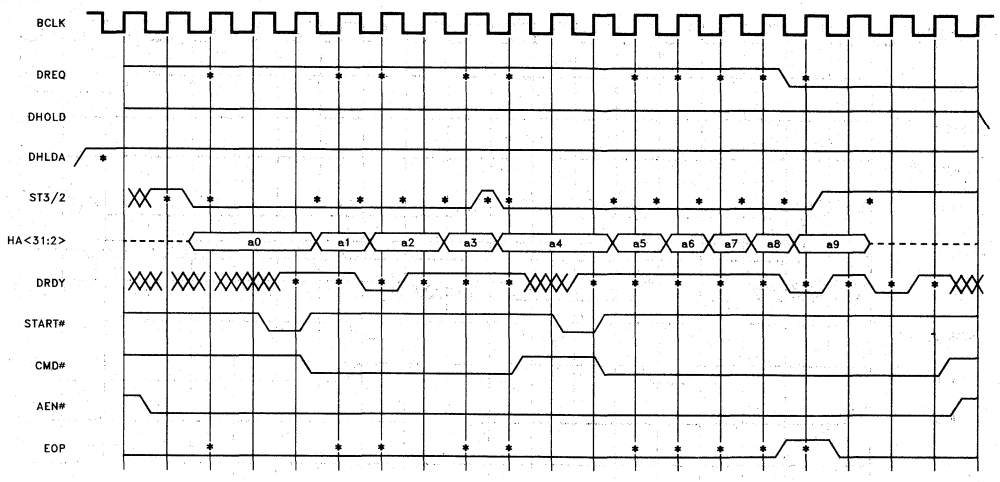


290253-52

Bus cycles illustrated:
a: Single Cycle DMA
c: Demand Mode DMA, cycle terminated with DRDY
c+1: Demand Mode DMA, cycle terminated by either EOP or DREQ, Type B cycle

NOTE:
Type "B" Trailing Edge of AEN# is delayed one cycle.

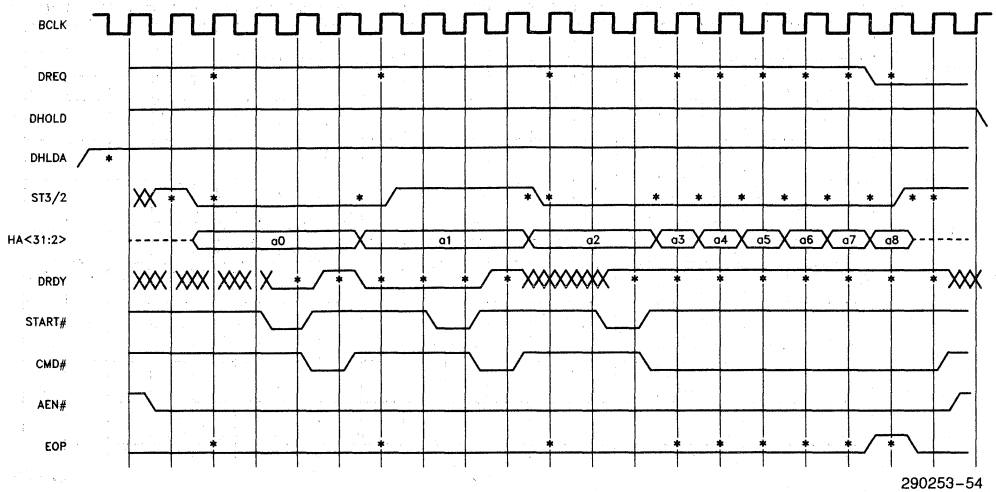
9.2.2 BURST DMA CYCLE DIAGRAM 1



290253-53

Bus Cycles illustrated:
A0: First DMA cycle of Burst, A1 address pipelined, 0 waitstate
A1: 2nd cycle of Burst, A2 address pipelined, 1 waitstate
A2: Continuation of Burst, A3 address pipelined, 0 waitstate
A3: End of Burst (A4 not in same page), A4 address pipelined, 0 waitstate
A4: Start of Burst, A5 pipelined, 0 waitstate
A5: A6 pipelined, 0 waitstate
A6: A7 pipelined, 0 waitstate
A7: A8 pipelined, 0 waitstate
A8: A9 pipelined, 1 waitstate
A9: Last DMA cycle, EOP sampled, DREQ sampled, 1 waitstate

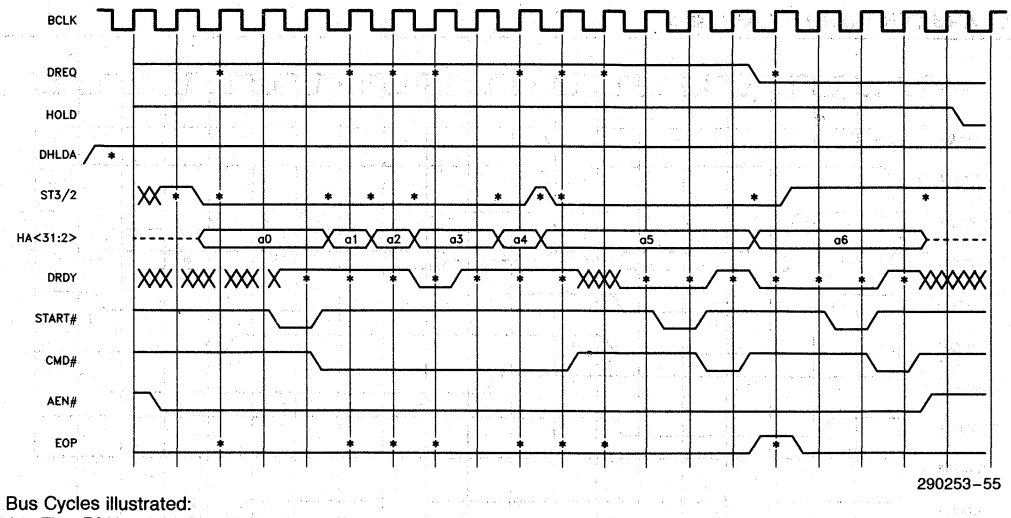
9.2.3 BURST DMA CYCLE DIAGRAM 2



290253-54

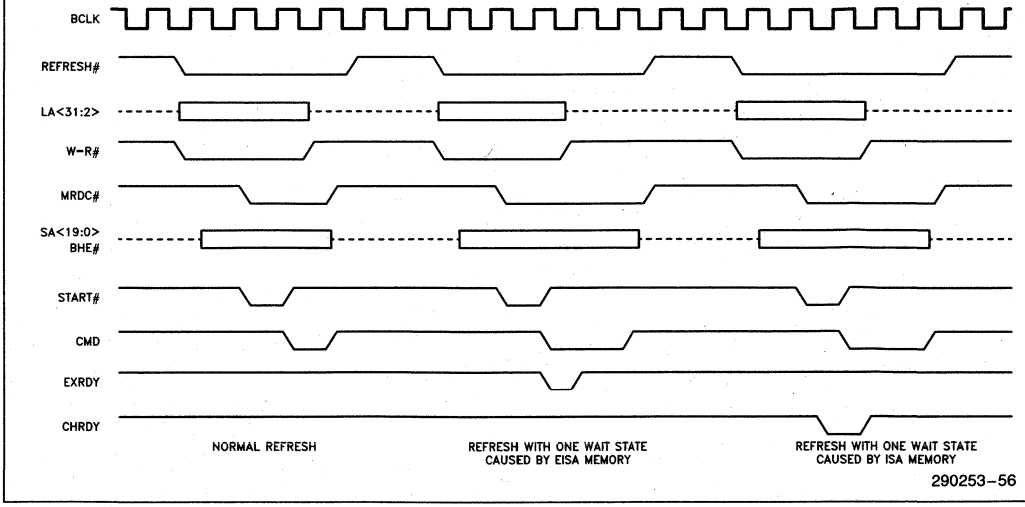
Bus Cycles illustrated:
A0: First DMA cycle, Burst Slave, non-Burst memory, 0 waitstate
A1: 1 waitstate, page break (A2 not in same page)
A2: Start of Burst, A3 address pipelined, 0 waitstate
A3: A4 pipelined, 0 waitstate
A4: A5 pipelined, 0 waitstate
A5: A6 pipelined, 0 waitstate
A6: A7 pipelined, 0 waitstate
A7: A8 pipelined, 0 waitstate
A8: Last DMA cycle, EOP sampled, DREQ sampled, 0 waitstate

9.2.4 BURST DMA CYCLE DIAGRAM 3



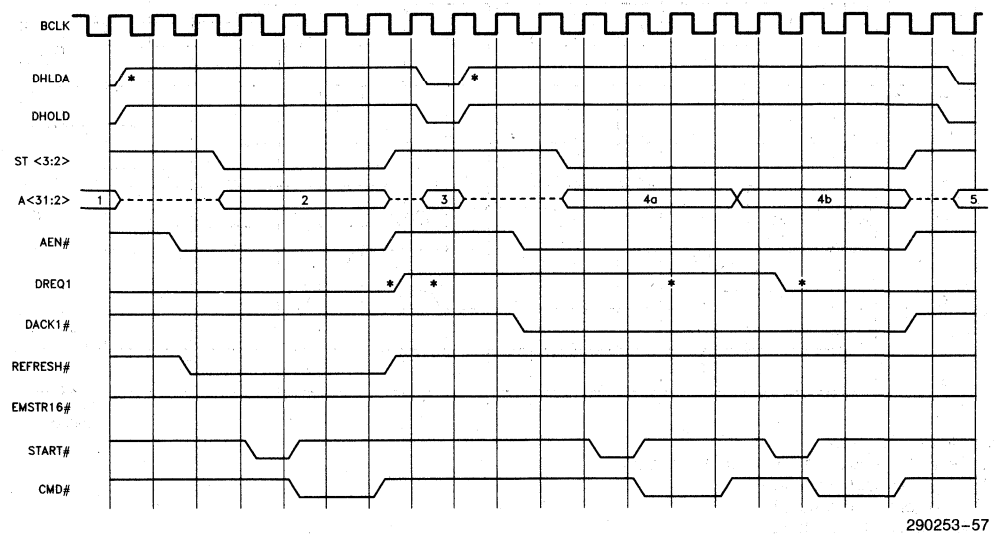
Bus Cycles illustrated:
A0: First DMA cycle, A1 pipelined, 0 waitstate
A1: A2 pipelined, 0 waitstate
A2: A3 pipelined, 0 waitstate
A3: A4 pipelined, 1 waitstate
A4: End of Burst (A5 not in same page), A5 pipelined, 0 waitstate
A5: Burst DMA slave, non-Burst Memory, 0 waitstate
A6: Last DMA cycle, EOP sampled, DREQ sampled, 1 waitstate

9.2.5 EISA REFRESH CYCLE



9.3 Arbiter Timing Diagrams

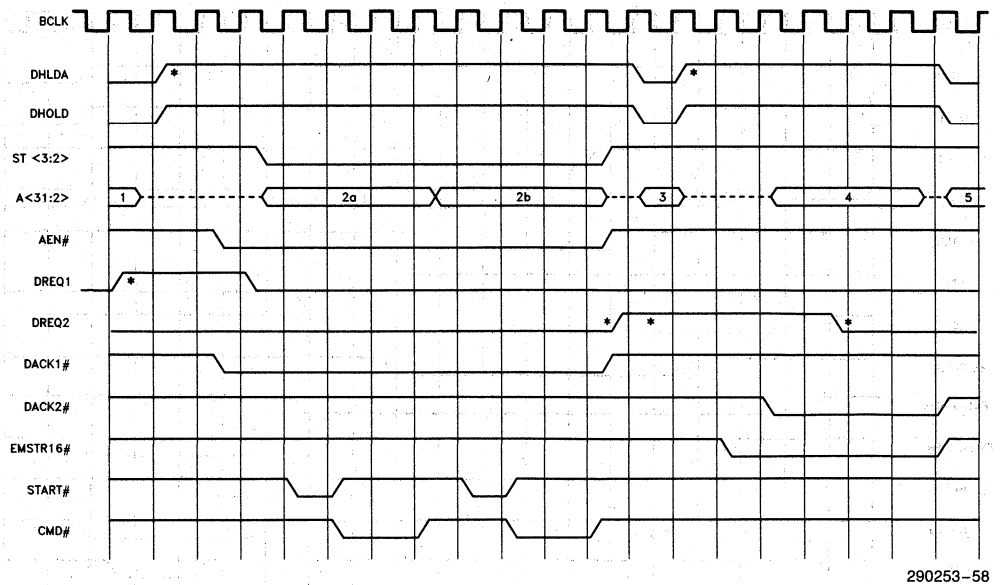
9.3.1 ARBITER TIMING DIAGRAM 1



290253-57

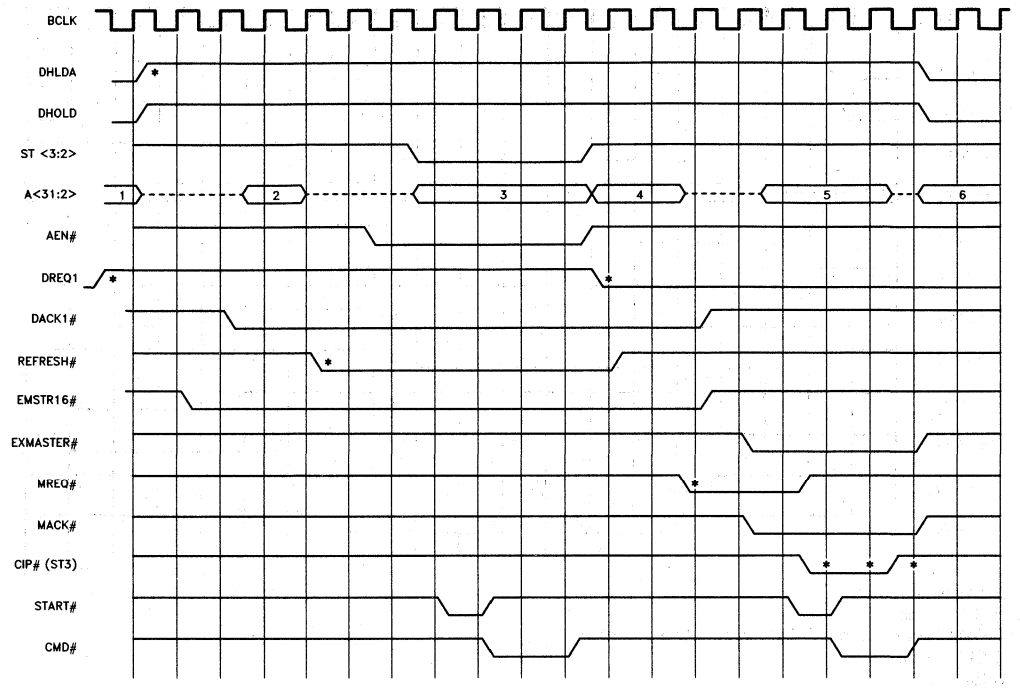
Cycles illustrated
1) CPU Cycle
2) Refresh Cycle
3) CPU Cycle
4) DMA Demand Mode Transfer, two Transfers, DREQ Inactivated
5) CPU Cycle

9.3.2 ARBITER TIMING DIAGRAM 2



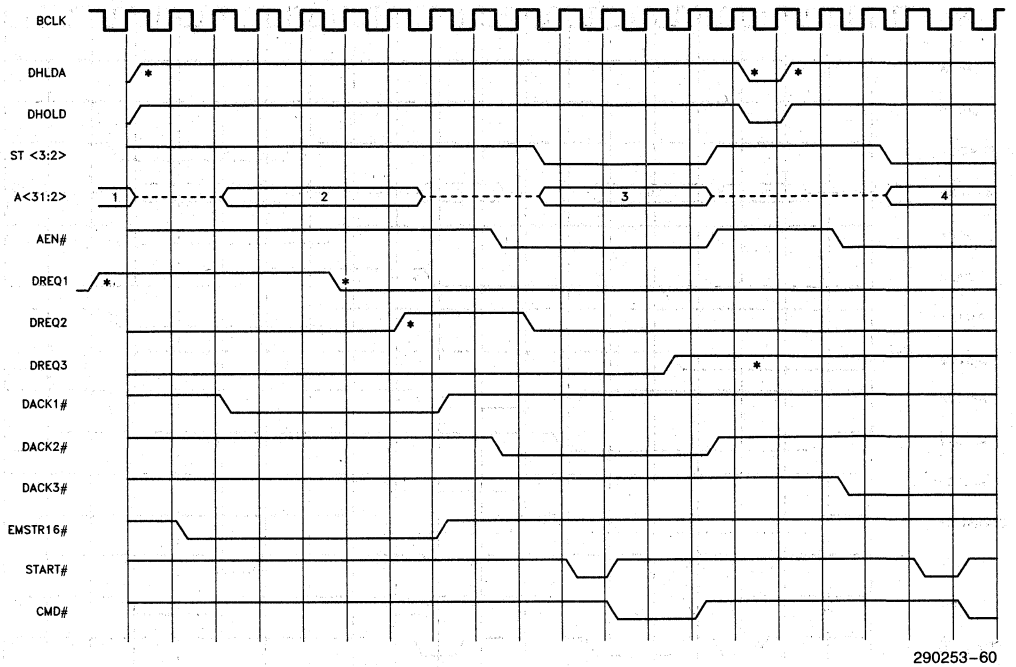
Cycles illustrated
1) CPU Cycle
2) DMA Block Mode, TC Terminated, two Transfers
3) CPU Cycle
4) ISA Master
5) CPU Cycle

9.3.3 ARBITER TIMING DIAGRAM 3



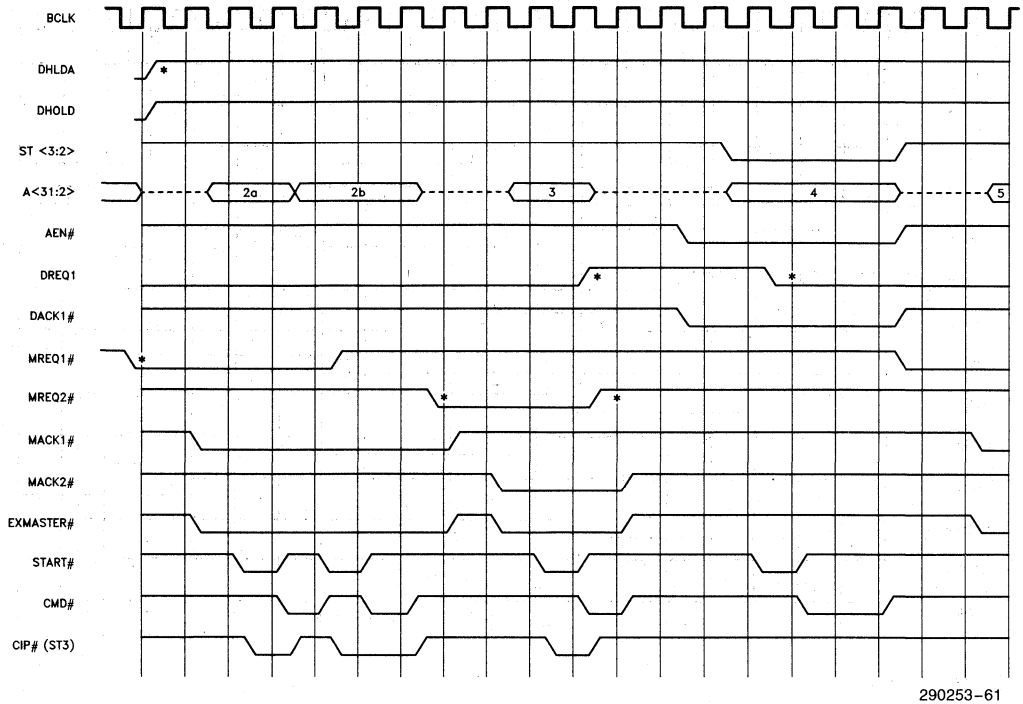
Cycles illustrated
1) CPU Cycle
2-4) ISA Master Cycle, Master Generates REFRESH#, ISA Master Cycle
5) EISA Master Cycle
6) CPU Cycle

9.3.4 ARBITER TIMING DIAGRAM 4



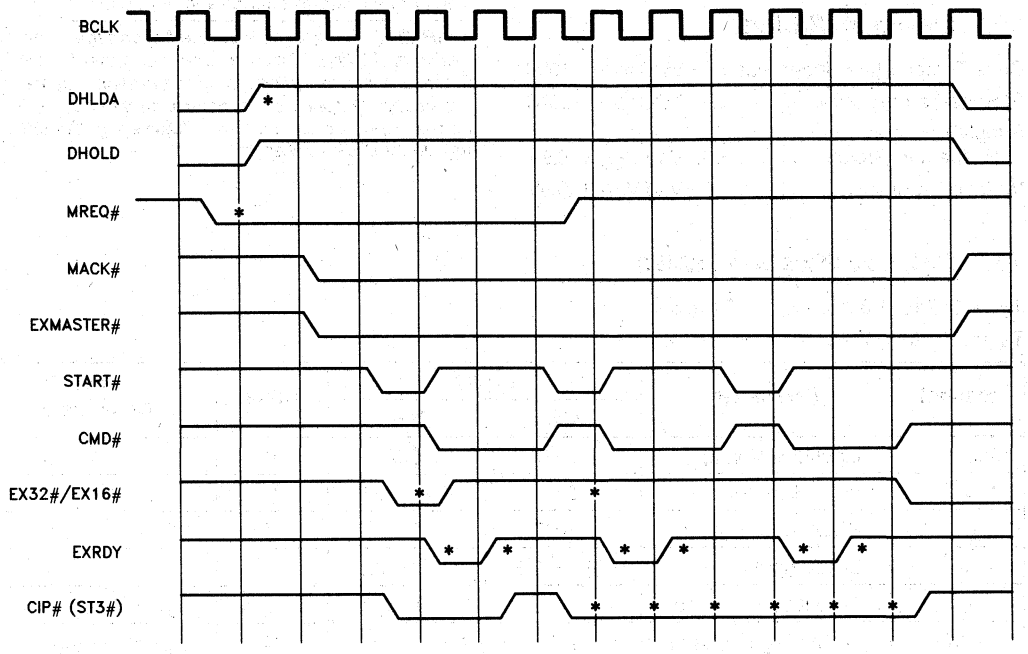
Cycles illustrated
1) CPU Cycle
2) ISA Master
3) Single Cycle DMA, Bus Returned to CPU
4) Demand Mode DMA

9.3.5 ARBITER TIMING DIAGRAM 5



Cycles illustrated
1) CPU Cycle
2) EISA Master1
3) EISA Master2
4) Demand Mode DMA, Type "A"
5) EISA

9.3.6 ARBITER TIMING DIAGRAM 6—CIP# (ST3#) TIMING



290253-62

Cycles illustrated
1) Matched EISA Bus Cycle
2) Mismatched EISA Bus Cycle, EISA Master Drops MREQ, Bus Controller Drops CIP# during Redrive

10.0 D.C. SPECIFICATIONS

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

10.1 Maximum Ratings

Case Temperature Under Bias . . −65°C to + 110°C
Storage Temperature . . . . . −65°C to + 150°C
Supply Voltages with
Respect to Ground . . . . . −0.5V to V_{CC} + 6.5V
Voltage On Any Pin . . . . . −0.5V to V_{CC} + 0.5V

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

10.2 D.C. Specification Tables

T_C = 0°C to +85°C, V_{CC} = 5V ±5%
T_A = 0°C to +70°C

Symbol	Parameter	Limits		Units	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{ILC}	BCLK Input Low	−0.5	0.8	V	
V_{IHC}	BCLK Input High	V_{CC} − 0.8	V_{CC} + 0.5	V	
V_{OL1}	Output Low Voltage		0.45	V	I_{OL} = 5 mA
V_{OH1}	Output High Voltage	2.4		V	I_{OH} = −1 mA
V_{OL2}	Output Low Voltage		0.45	V	I_{OL} = 24 mA(1)
V_{OH2}	Output High Voltage	2.4		V	I_{OH} = −4 mA(1)
V_{OL3}	Output Low Voltage		0.45	V	I_{OL} = 1 mA(1)
V_{OH3}	Output High Voltage	2.4		V	I_{OH} = −250 μA(1)
I_{LI1}	Input Leakage		±15	μA	0V < V_{IN} < V_{CC}
I_{LI2}	Input Leakage		+15	μA	V_{IN} = V_{CC}(2)
I_{LI3}	Input Leakage	−45	−240	μA	V_{IN} = 2.4V(2)
I_{LO}	Output Leakage		±15	μA	0.45 < V_{IN} < V_{CC}
Cap	Cap. In, Out, I/O		12	pF	@ 1 MHz(3)
C_{CLK}	BCLK Cap.		20	pF	@ 1 MHz(3)
I_{CC}	V_{CC} Supply Current		200	mA	BCLK = 8.3 MHz V_{IL} = 0V V_{IH} = 4V

NOTES:
1. V_{OL2} and V_{OH2} apply only to the signals that directly drive the EISA bus and are not slot specific. These are: RSTDRV, REFRESH#, EOP, and BE<3:0>#.
V_{OL3} and V_{OH3} apply only to the signals that do not drive onto buses in the system. These are: INT, EMSTR16#, EXMASTER#, NMI, SLOWH#, GT16M#, GT1M#, DHOLD, AEN#, DRDY, ST<3:0>, and CSOUT#. All other outputs use V_{OL1} and V_{OH1}.
2. I_{L12} and I_{L13} apply only to those pins that include a weak sustaining pullup transistor. These are: ST<3:0>, DRDY and IRQ8#. All other IO and inputs should use I_{LI1}.
3. Sampled only.

11.0 A.C. SPECIFICATIONS

The A.C. specifications given in the following tables consist of output delays and input setup and hold requirements. The A.C. diagram's purpose is to illustrate the clock edges and specific signal edges from which the timing parameters are measured. The reader should not infer any other timing relationships from them. For specific information on timing relationships between the signals, refer to the appropriate functional and pin definition sections.

11.1 A.C. Characterization Tables

The following timing definition points are to be used as a reference when reading the A.C. Characteristic Tables.

EISA Bus Cycle/BCLK timing definition points.

T0 = Rising edge of BCLK at the beginning of START# active.

T1 = Falling edge of BCLK at the middle of START# active.

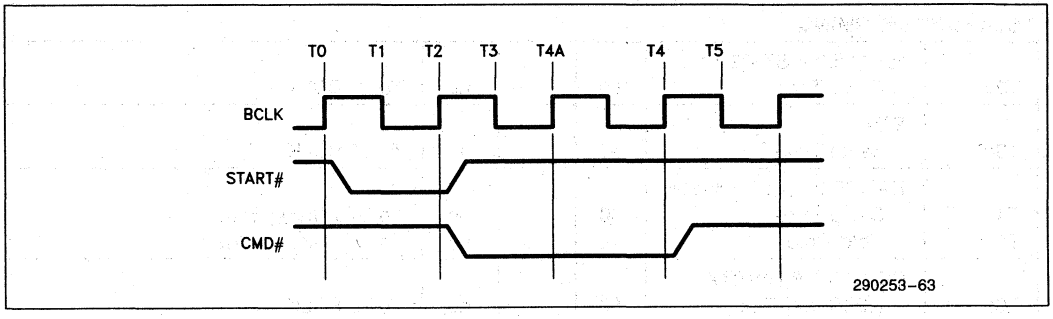
T2 = Rising edge of BCLK at the beginning of CMD# active.

T3 = The first falling edge of BCLK after CMD# goes active.

T4A = The first rising edge of BCLK after CMD# goes active.

T4 = Rising edge of BCLK at the end of a cycle (CMD# inactive edge).

T5 = The first falling edge of BCLK after CMD# goes inactive.



T_C = 0°C to +85°C, V_{CC} = 5V ±5%
T_A = 0°C to +70°C

Symbol	Parameter	Min	Max	Unit	Notes
CLOCK, OSC TIMING					
	BCLK Period				
T1	Period	120		ns	@ 1.5V
			500	ns	Not an ISP Slave Cycle
			170	ns	ISP Slave Cycle
T2a	High Time	55		ns	@ 2.0V
T2b	Low Time	55		ns	@ 0.8V
	OSC				
T77	Period	65	70	ns	For Asynchronous OSC @ 1.5V
T78a	High Time	20		ns	@ 2.0V
T78b	Low Time	20		ns	@ 0.8V
SLAVE MODE TIMING					
	HA<15:2>,ST<2:1>				
T3a	Setup Time	80		ns	To T2 BCLK
	ST0				
T3b	Setup Time	55		ns	To T2 BCLK
	HA<15:2>,ST1,HW/R#				
T4	Setup Time	30		ns	To ST0 Rising Edge
T5	Hold Time	10		ns	To ST0 Rising Edge
	BE<2:0>#,HW/R#				
T6a	(BE)Setup Time	60		ns	To T2 BCLK
T6b	(HW/R#)Setup Time	40		ns	To T2 BCLK
	BE<2:0>#,ST2				
T7	Hold Time	30		ns	
	START#				
T8	Setup Time	30		ns	
T9	Hold Time	0		ns	
	Data (Write)				
T10	Setup Time	10		ns	To T4a BCLK
T11	Hold Time	5		ns	To T4 BCLK
	Data (Read)				
T12	Delay Valid		60	ns	Prior To T4 BCLK
T13	Float		40	ns	From CMD#, Xcvr Must Meet EISA 35 ns
	CSOUT#				
T14	Delay Active		35	ns	From BCLK Rising Edge
T15	Delay Inactive		40	ns	From CMD# Rising Edge
	DRDY				
T16	Delay Active/Inactive		45	ns	From BCLK Rising
T17	Float		45	ns	From CMD# Rising Edge.(1)
	CMD#				
T18	Setup Time	90		ns	Sync (Active/Inactive)
T19	Hold Time	0		ns	Sync (Active/Inactive)

Symbol	Parameter	Min	Max	Unit	Notes
DMA MASTER TIMING					
	HA<31:2>,HW/R#,GT16M#				
T22a	Delay Valid	1	35	ns	Initial Address, HW/R#, GT16#
T22b	Delay Valid	1	30	ns	Subsequent Addresses(1)
T23	Float	1	30	ns	
	ST<3:0>				
T24	Delay Active/Inactive	4	37	ns	
T25	Float	1	40	ns	(1)
	ST<3:2>				
T26	Delay Inactive	4	37	ns	Burst Cycle/Page Break
	DRDY				
T27	Setup Time	20		ns	
T28	Hold Time	15		ns	
	AEN#				
T29	Delay Active/Inactive		40	ns	
	BE<3:0>#				
T30	Delay Valid		45	ns	Direct to EISA to Meet Burst Timings
T31	Float		45	ns	Float from T3 BCLK.(1)
	EOP				
T32	Delay Active		40	ns	Non-Burst
T33	Delay Active/Inactive		40	ns	
T34	Float from DACK		30	ns	From DACK Active.(1)
T35	Setup Time	15		ns	Sync (Burst Mode)
T36	Hold Time	15		ns	Sync (Burst Mode)
T37	Setup Time	15		ns	Async (Non-Burst)
T38	Hold Time	15		ns	Async (Non-Burst)
ARBITER TIMING					
	DREQ				
T39	Setup Time	15		ns	Sync (Trailing Edge)
T40	Hold Time	15		ns	Sync (Trailing Edge)
T41	Setup Time	15		ns	Async (Leading/Trailing Edge)
T42	Hold Time	15		ns	Async (Leading/Trailing Edge)
	MREQ#				
T43a	Setup Time	17		ns	BCLK Rising
T43b	Setup Time	80		ns	BCLK Falling, 0.5 BCLKs after T43a
T44	Hold Time	15		ns	
	CPUMISS#				Asynchronous
T45	Setup Time	15		ns	
T46	Hold Time	15		ns	
	DHLDA				Asynchronous
T47	Setup Time	15		ns	
T48	Hold Time	15		ns	
	DACK#				
T49a	Delay Active/Inactive		50	ns	240 pF (ISA Masters)
T49b	Delay Active/Inactive		50	ns	240 pF (DMA Devices)
	MACK#				
T50	Delay Active/Inactive		40	ns	
	DHOLD				
T51	Delay Active/Inactive		40	ns	

1

Symbol	Parameter	Min	Max	Unit	Notes
ARBITER TIMING (Continued)					
T52	EMSTR16# Delay Active/Inactive		40	ns	
T53	EXMASTER# Delay Active/Inactive		40	ns	
	ST3 (CIP#)				
T54	Setup Time	25		ns	
T55	Hold Time	15		ns	
REFRESH TIMING					
	REFRESH#				
T56	Delay Active		55	ns	
T57	Float Inactive		55	ns	
	REFRESH#				Asynchronous
T58	Setup Time	15		ns	
T59	Hold Time	15		ns	
RESET TIMING					
	RST				Asynchronous
T60	Active Pulse Width	1000		ns	8 BCLKs
	RSTDRV				
T61	Delay Active		60	ns	Bus Timeout
T62	Delay Active		60	ns	Port/RST
T63	Delay Inactive		60	ns	
NMI TIMING					
	NMI				
T64	Delay Active/Inactive		60	ns	From BCLK Falling Edge
T65	Delay Act		200	ns	From OSC Rising Edge
	PARITY#, IOCHK#				Asynchronous
T66	Setup Time	15		ns	
T67	Hold Time	15		ns	
INTERRUPT TIMING					
	INT				
T68	Delay Active/Inactive		500	ns	From IRQ Active/Inactive
T69	Delay Active/Inactive		500	ns	From OSC Rising Edge
T70	Delay Active/Inactive		500	ns	From BCLK Falling Edge
COUNTER/TIMER TIMING					
	SPKR				
T71	Delay Active/Inactive		200	ns	From OSC Rising Edge
T72	Delay Active/Inactive		100	ns	From BCLK Falling Edge
	SLOWH#				
T73	Delay Active/Inactive		200	ns	
RTCALE, GTIM# TIMING					
	RTCALE				
T74	Delay Active		55	ns	
T75	Delay Inactive		40	ns	
	GT1M#				
T76	Delay Active/Inactive		50	ns	

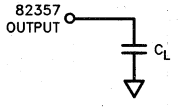
NOTE:
1. Sampled, not 100% tested.

A.C. TEST LOADS

C_L = 240 pF on RSTDRV, BE<3:0>#, REFRESH#, EOP

C_L = 50 pF on DHOLD, GT1M#, GT16M#, ST<3:0>, EMSTR16#, EXMASTER#, CSOUT#, INT, SLOWH#, NMI, DRDY, AEN#, SPKR, and RTCALE

C_L = 120 pF on all other pins (MACK<5:0>#, D<7:0>, HA<31:2>, DACK#)



290253-79

C_L includes all parasitic capacitances.

11.2 A.C. Characteristic Waveforms

In the following timing illustrations (10.2.1–10.2.10) "SET" refers to setup time and "HOLD" refers to hold time. The remaining timings are either output delay (valid, active/inactive edges) or float timings.

A.C. Testing: All non-clock Inputs are driven at 3V for a logic "1" and 0V for a logic "0". A.C. Timings referenced to a BCLK edge are measured from the 2V level on the clock to the 1.5V level on the signal under test. All signals referenced to a non-clock edge are measured from 1.5V to 1.5V; except as noted by the following:

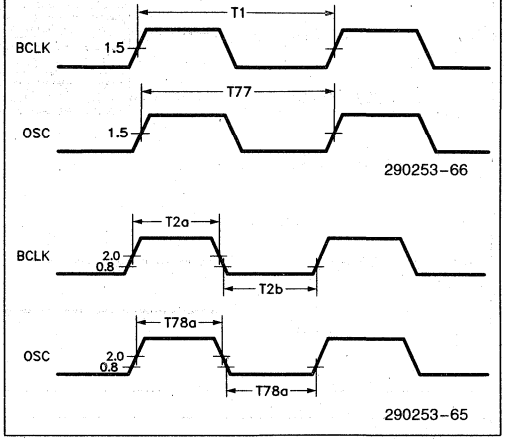
(1) BCLK high time (T2a) measurement is made at 2.0V

(2) BCLK low time (T2b) measurement is made at 0.8V

(3) OSC high time (T78a) measurement is made at 2.0V

(4) OSC low time (T78b) measurement is made at 0.8V

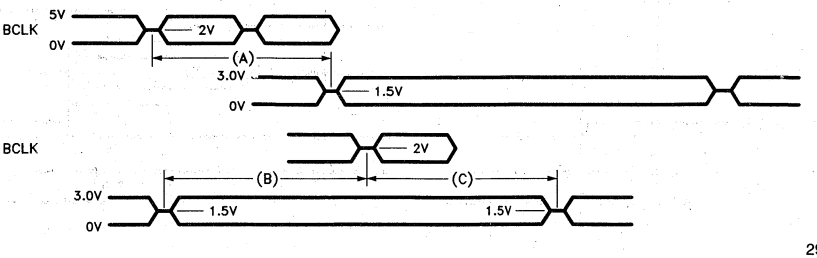
11.2.1 BCLK, OSC TIMING



290253-66

290253-65

A.C. TESTING INPUT, OUTPUT WAVEFORMS



290253-64

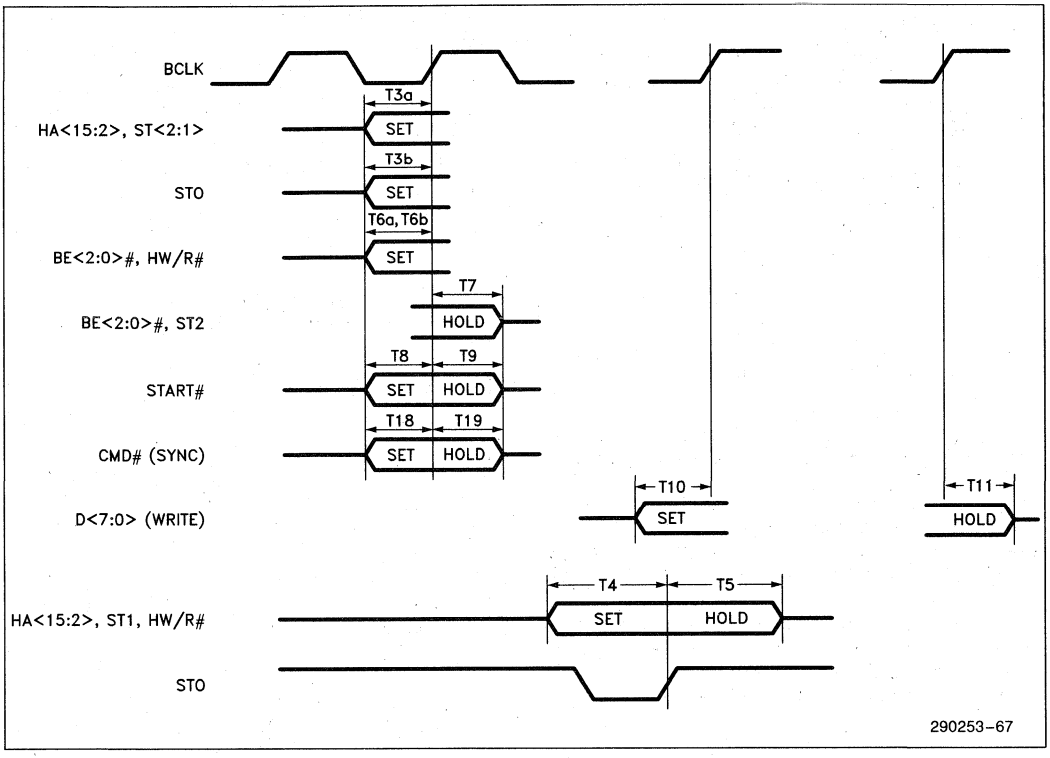
(A) Output delay spec referenced to either a BCLK rising or a BCLK falling edge.

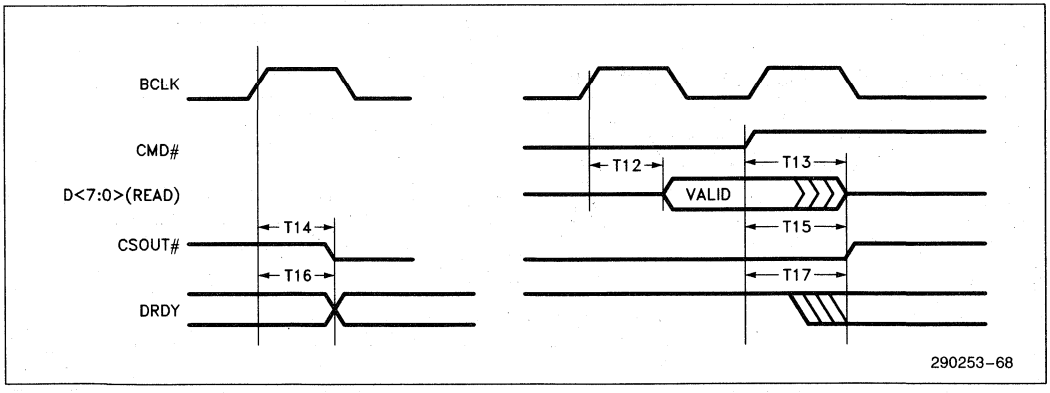
(B) Minimum input setup spec referenced to either a BCLK rising or a BCLK falling edge.

(C) Minimum input hold spec referenced to either a BCLK rising or a BCLK falling edge.

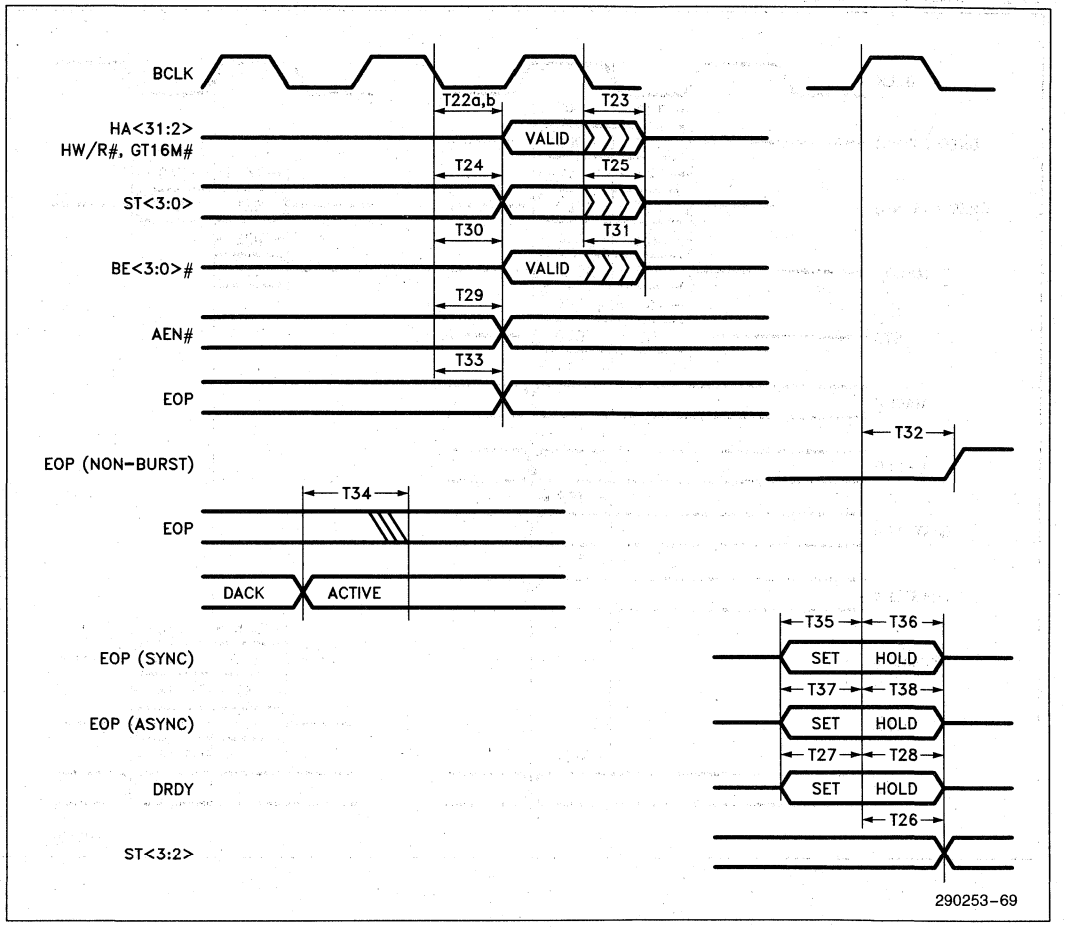
NOTE:
The input waveforms have a $t_r \leq 2.0$ ns from 0.8V to 2.0V.

11.2.2 SLAVE MODE TIMING

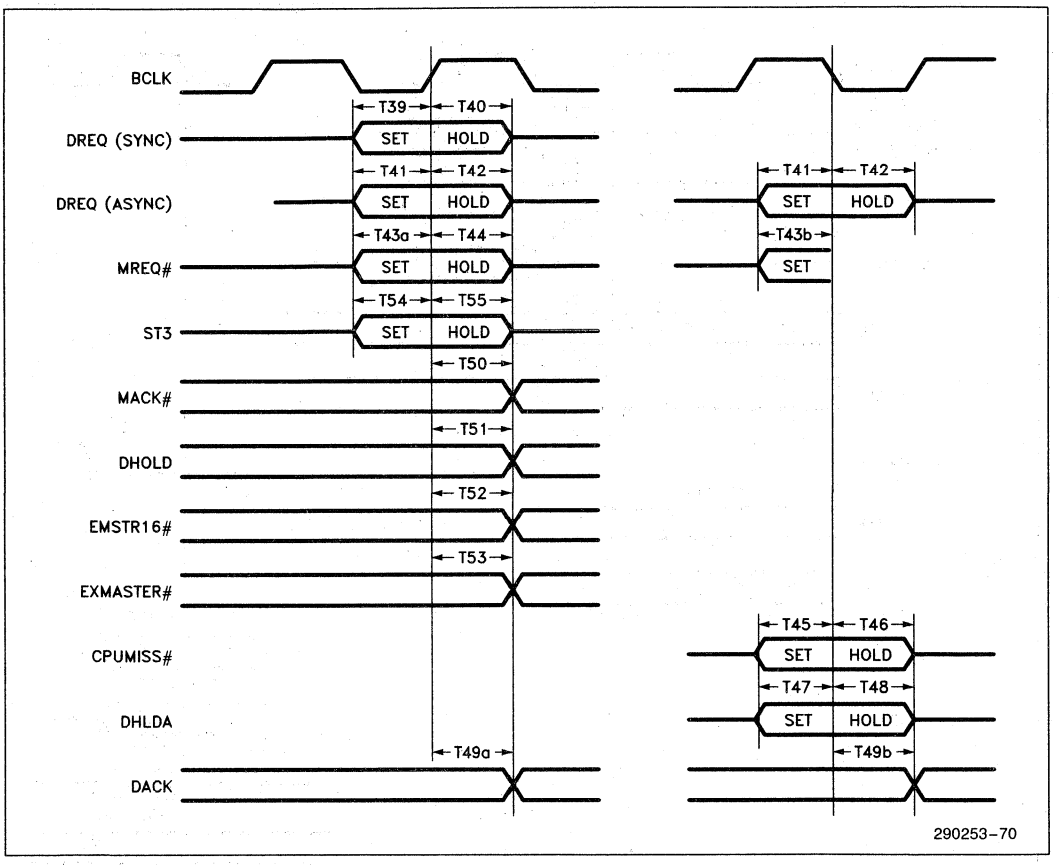




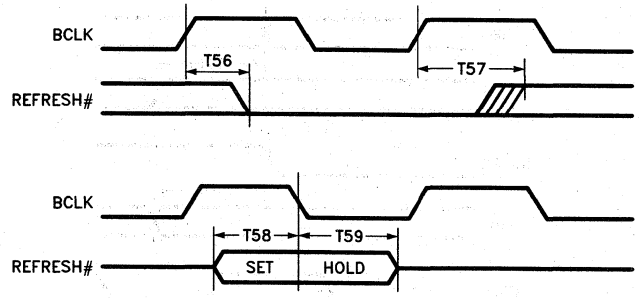
11.2.3 DMA MASTER TIMING



11.2.4 ARBITER TIMING

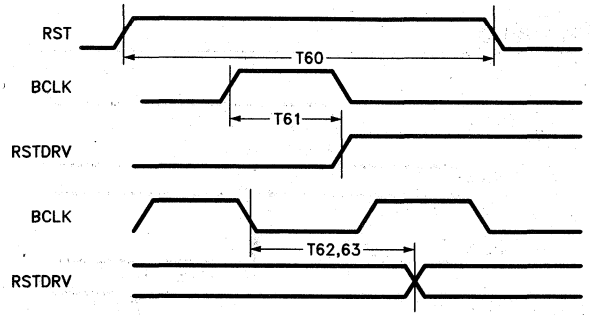


11.2.5 REFRESH TIMING



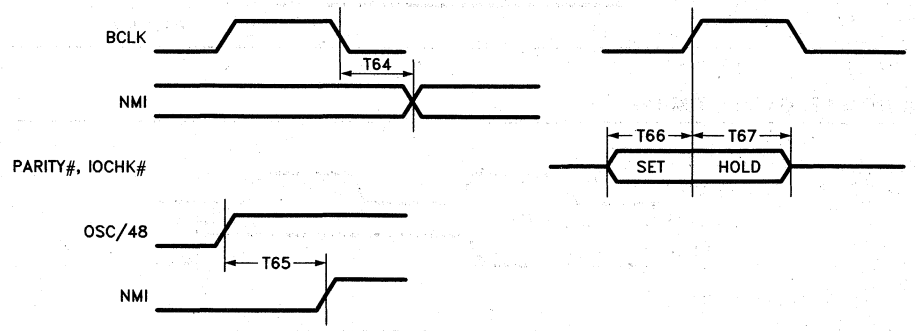
290253-71

11.2.6 RESET TIMING



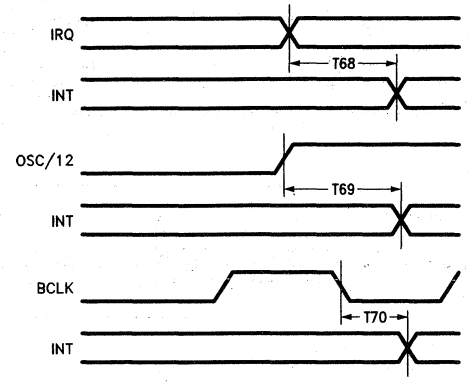
290253-72

11.2.7 NMI TIMING



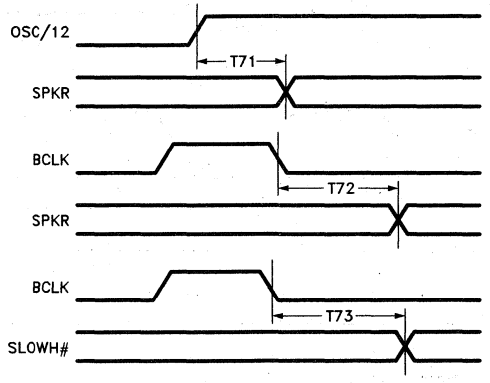
290253-73

11.2.8 INTERRUPT TIMING



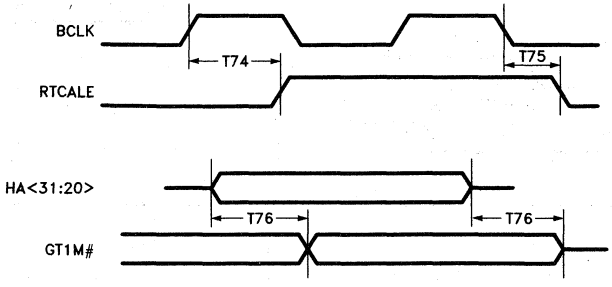
290253-74

11.2.9 COUNTER/TIMER TIMING



290253-75

11.2.10 RTCALE, GT1M# TIMING



290253-76

12.0 ISP PIN AND PACKAGE INFORMATION

12.1 Signal Overview

Name	Type	Pin	Interface	Description
The following pins are used by the ISP for EISA system arbitration:				
DREQ<7:5>	I	100–102	EISA	DMA Request Lines
DREQ<3:0>	I	103–106	EISA	DMA Request Lines
DACK<7:5>#	O	83–85	EISA	DMA Acknowledge Lines
DACK<3:0>#	O	86–89	EISA	DMA Acknowledge Lines
MREQ<5:0>#	I	107–112	EISA	Master Request Lines
MACK<5:0>#	O	92–97	EISA	Master Acknowledge Lines
REFRESH#	I/O	131	EISA	Refresh Control (24 mA)
DHLDA	I	114	Host	CPU Hold Acknowledge
CPUMISS#	I	113	Host	CPU Cache Miss
DHOLD	O	98	EBC	CPU Hold Request
EMSTR16#	O	4	EBC	Early Master 16 Indication to the 82358 Bus Controller
EXMASTER#	O	5	EBC	EISA Bus Master Access Control
The following signals are used by the ISP for either DMA control or in slave mode during a register access:				
START#	I	20	EISA	Start Signal from the EISA Bus
CMD#	I	19	EISA	Command Signal from the EISA Bus
EOP	I/O	80	EISA	End of Process Input and Output
BE<3:0>#	I/O	70–67	EISA	Byte Enables (BE3 is Output Only)
HA<11:2>	I/O	55–64	Host	Address Bus
HA<14:12>	I/O	50–52	Host	Address Bus
HA<21:15>	I/O	35–41	Host	Address Bus (A16–A19 are Output Only)
HA<31:22>	I/O	23–32	Host	Address Bus
HW/R#	I/O	79	Host	Write/Read Status
GT16M#	O	11	EBC	Address Less Than 16 Mbytes
GT1M#	O	12	EBC	Address Less Than 1 Mbyte
RST	I	18	EBC	System Reset
ST<3:0>	I/O	72–75	EBC	Bidirectional Status Signals between the ISP/EBC
DRDY	I/O	78	EBC	Master Mode Ready from EBC and Slave Mode Ready to EBC Terminal Count Output (Directly Drives EISA Bus)
D<7:0>	I/O	42–49	Other	Data Bus
CSOUT#	O	9	Other	ISP Selected in Slave Mode
AEN#	O	82	Other	Address Enable Out from DMA
The following signals are used by the ISP for interrupt control:				
IRQ1	I	128	EISA	Interrupt Request Input
IRQ<15:3>	I	115–127	EISA	Interrupt Request Inputs
INT	O	3	Host	Interrupt Output to CPU

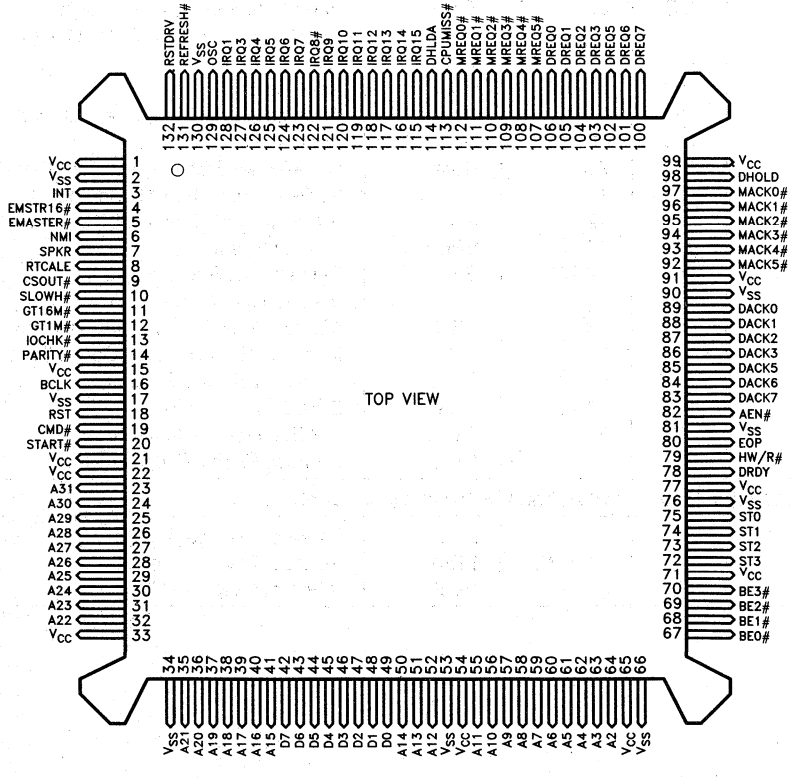
12.1 Signal Overview (Continued)

Name	Type	Pin	Interface	Description
The following signals are used by the ISP for the Timers, NMI Servicing, and Associated Logic:				
OSC	I	129	EISA	14.31818 MHz Clock for Timers
IOCHK#	I	13	EISA	Bus Error Signal
RSTDRV	O	132	EISA	System Bus Reset
PARITY#	I	14	Host	Main Memory Parity Error
SLOWH#	O	10	Host	Slow Down Timer to CPU
NMI	O	6	Host	From NMI Logic
SPKR	O	7	Other	Speaker Driver Output (24 mA Drive)
RTCALE	O	8	Other	Real Time Clock Address Latch Enable
The following pins are used for power:				
V_{CC}	1,15,21,22,33,54,65,71,77,91,99			Power
V_{SS}	2,17,34,53,66,76,81,90,130			Ground
The following pin is the clock input:				
BCLK	I	16	EBC	Operation Clock

Name = Pin Name, **Type** = Input/Output, **Pin** = Pin Location, **Interface** = EBC Interface, EISA Bus, Host Bus, or Other.

12.2 132-Pin Package

82357 Package Pin Out



290253-77

12.3 Device Pinout

Device Pinout—132 Lead PQFP

A Row		
Pin	Label	Type
1	V_{CC}	
2	V_{SS}	
3	INT	O
4	EMSTR16#	O
5	EXMASTER#	O
6	NMI	O
7	SPKR	O
8	RTCALE	O
9	CSOUT#	O
10	SLOWH#	O
11	GT16M#	O
12	GT1M#	O
13	IOCHK#	I
14	PARITY#	I
15	V_{CC}	
16	BCLK	I
17	V_{SS}	
18	RST	I
19	CMD#	I
20	START#	I
21	V_{CC}	
22	V_{CC}	
23	A31	IO
24	A30	IO
25	A29	IO
26	A28	IO
27	A27	IO
28	A26	IO
29	A25	IO
30	A24	IO
31	A23	IO
32	A22	IO
33	V_{CC}	

B Row		
Pin	Label	Type
34	V_{SS}	
35	A21	IO
36	A20	IO
37	A19	O
38	A18	O
39	A17	O
40	A16	O
41	A15	IO
42	D7	IO
43	D6	IO
44	D5	IO
45	D4	IO
46	D3	IO
47	D2	IO
48	D1	IO
49	D0	IO
50	A14	IO
51	A13	IO
52	A12	IO
53	V_{SS}	
54	V_{CC}	
55	A11	IO
56	A10	IO
57	A9	IO
58	A8	IO
59	A7	IO
60	A6	IO
61	A5	IO
62	A4	IO
63	A3	IO
64	A2	IO
65	V_{CC}	
66	V_{SS}	

C Row		
Pin	Label	Type
67	BE0#	IO
68	BE1#	IO
69	BE2#	IO
70	BE3#	O
71	V_{CC}	
72	ST3	IO
73	ST2	IO
74	ST1	IO
75	ST0	IO
76	V_{SS}	
77	V_{CC}	
78	DRDY	IO
79	HW/R#	IO
80	EOP	IO
81	V_{SS}	
82	AEN#	O
83	DACK7#	O
84	DACK6#	O
85	DACK5#	O
86	DACK3#	O
87	DACK2#	O
88	DACK1#	O
89	DACK0#	O
90	V_{SS}	
91	V_{CC}	
92	MACK5#	O
93	MACK4#	O
94	MACK3#	O
95	MACK2#	O
96	MACK1#	O
97	MACK0#	O
98	DHOLD	O
99	V_{CC}	

D Row		
Pin	Label	Type
100	DREQ7	I
101	DREQ6	I
102	DREQ5	I
103	DREQ3	I
104	DREQ2	I
105	DREQ1	I
106	DREQ0	I
107	MREQ5#	I
108	MREQ4#	I
109	MREQ3#	I
110	MREQ2#	I
111	MREQ1#	I
112	MREQ0#	I
113	CPUMISS#	I
114	DHLDA	I
115	IRQ15	I
116	IRQ14	I
117	IRQ13	I
118	IRQ12	I
119	IRQ11	I
120	IRQ10	I
121	IRQ9	I
122	IRQ8#	I
123	IRQ7	I
124	IRQ6	I
125	IRQ5	I
126	IRQ4	I
127	IRQ3	I
128	IRQ1	I
129	OSC	I
130	V_{SS}	
131	REFRESH#	IO
132	RSTDRV	O

13.0 ISP PORT ADDRESS / I/O DECODING

Address Bits					Register Type	Device
Port	FEDC	BA98	7654	3210		
0000h	0000	0000	000X	0000	rw	DMA1 CH-0 Base and Current Address
0001h	0000	0000	000X	0001	rw	DMA1 CH-0 Base and Current Count
0002h	0000	0000	000X	0010	rw	DMA1 CH-1 Base and Current Address
0003h	0000	0000	000X	0011	rw	DMA1 CH-1 Base and Current Count
0004h	0000	0000	000X	0100	rw	DMA1 CH-2 Base and Current Address
0005h	0000	0000	000X	0101	rw	DMA1 CH-2 Base and Current Count
0006h	0000	0000	000X	0110	rw	DMA1 CH-3 Base and Current Address
0007h	0000	0000	000X	0111	rw	DMA1 CH-3 Base and Current Count
0008h	0000	0000	000X	1000	rw	DMA1 status(r) command (w) register
0009h	0000	0000	000X	1001	w	DMA1 Write Request register (w) (reserved) (r)
000Ah	0000	0000	000X	1010	w	DMA1 Write single mask bit (w) (reserved) (r)
000Bh	0000	0000	000X	1011	w	DMA1 Write Mode register (w) (reserved) (r)
000Ch	0000	0000	000X	1100	w	DMA1 Clear byte pointer F/F (w) (reserved) (r)
000Dh	0000	0000	000X	1101	w	DMA1 Master Clear (w) (reserved) (r)
000Eh	0000	0000	000X	1110	w	DMA1 Clear Mask register (w) (reserved) (r)
000Fh	0000	0000	000X	1111	rw	DMA1 Write all mask register bits (w) DMA1 Read all mask register bits (r)
0020h	0000	0000	001X	XX00	rw	INT-1 control register
0021h	0000	0000	001X	XX01	rw	INT-1 mask register
0040h	0000	0000	010X	0000	rw	Programmable Interval Timer 1, 82C54 System Clock (Counter 0)
0041h	0000	0000	010X	0001	rw	Refresh Request (Counter 1)
0042h	0000	0000	010X	0010	rw	Speaker Tone (Counter 2)
0043h	0000	0000	010X	0011	w	Command Mode Register
0048h	0000	0000	010X	1000	rw	Programmable Interval Timer 2, 82C54 Failsafe Clock (Counter 0)
0049h	0000	0000	010X	1001	rw	(reserved)
004Ah	0000	0000	010X	1010	rw	CPU Speed Control (Counter 2)
004Bh	0000	0000	010X	1011	w	Command Mode Register
0061h	0000	0000	0110	0X01	rw	NMI Status (See NMI Interrupts)
0070h	0000	0000	0111	0XX0	w	NMI Enable Register (bit 7 = 0) (w) (reserved) (r)

NOTE:
The above reserved registers (9h–Eh) when read, will return an indeterminate value. Reserved register 49h can not be read or written. When read, 49h will return FFh.

13.0 ISP PORT ADDRESS / I/O DECODING (Continued)

Address Bits					Register Type	Device
Port	FEDC	BA98	7654	3210		
0080h	0000	0000	100X	0000	rw	DMA Page Register Reserved
0081h	0000	0000	100X	0001	rw	DMA Page Register CH-2 Page
0082h	0000	0000	1000	0010	rw	DMA Page Register CH-3 Page
0083h	0000	0000	100X	0011	rw	DMA Page Register CH-1 Page
0084h	0000	0000	100X	0100	rw	DMA Page Register Reserved
0085h	0000	0000	100X	0101	rw	DMA Page Register Reserved
0086h	0000	0000	100X	0110	rw	DMA Page Register Reserved
0087h	0000	0000	100X	0111	rw	DMA Page Register CH-0 Page
0088h	0000	0000	100X	1000	rw	DMA Page Register Reserved
0089h	0000	0000	100X	1001	rw	DMA Page Register CH-6 Page
008Ah	0000	0000	100X	1010	rw	DMA Page Register CH-7 Page
008Bh	0000	0000	100X	1011	rw	DMA Page Register CH-5 Page
008Ch	0000	0000	100X	1100	rw	DMA Page Register Reserved
008Dh	0000	0000	100X	1101	rw	DMA Page Register Reserved
008Eh	0000	0000	100X	1110	rw	DMA Page Register Reserved
008Fh	0000	0000	100X	1111	rw	DMA Page Register Refresh Page
00A0h	0000	0000	101X	XX00	rw	INT-2 control register
00A1h	0000	0000	101X	XX01	rw	INT-2 mask register
00C0h	0000	0000	1100	000X	rw	DMA2 CH-0 Base and Current Address
00C2h	0000	0000	1100	001X	rw	DMA2 CH-0 Base and Current Count
00C4h	0000	0000	1100	010X	rw	DMA2 CH-1 Base and Current Address
00C6h	0000	0000	1100	011X	rw	DMA2 CH-1 Base and Current Count
00C8h	0000	0000	1100	100X	rw	DMA2 CH-2 Base and Current Address
00CAh	0000	0000	1100	101X	rw	DMA2 CH-2 Base and Current Count
00CCh	0000	0000	1100	110X	rw	DMA2 CH-3 Base and Current Address
00CEh	0000	0000	1100	111X	rw	DMA2 CH-3 Base and Current Count
00D0h	0000	0000	1101	000X	rw	DMA2 status(r) command(w) register
00D2h	0000	0000	1101	001X	w	DMA2 Write Request register (w) (reserved) (r)
00D4h	0000	0000	1101	010X	w	DMA2 Write single mask bit (w) (reserved) (r)
00D6h	0000	0000	1101	011X	w	DMA2 Write Mode register (w) (reserved) (r)
00D8h	0000	0000	1101	100X	w	DMA2 Clear byte pointer F/F (w) (reserved) (r)
00DAh	0000	0000	1101	101X	w	DMA2 Master Clear (w) (reserved) (r)
00DCh	0000	0000	1101	110X	w	DMA2 Clear Mask register (w) (reserved) (r)
00DEh	0000	0000	1101	111X	rw	DMA2 Write all mask register bits (w) DMA2 Read all mask register bits (r)

NOTE:
Reserved registers (0084h–0086h and 008Ch–008Eh) can be read or written. Reserved registers (00D2h–00DCh) when read, will return an indeterminate value.

13.0 ISP PORT ADDRESS / I/O DECODING (Continued)

Address Bits					Register Type	Device
Port	FEDC	BA98	7654	3210		
0400h	0000	0100	0000	0000	rw	(reserved)
0401h	0000	0100	0000	0001	rw	DMA1 CH-0 Base/Current count high
0402h	0000	0100	0000	0010	rw	(reserved)
0403h	0000	0100	0000	0011	rw	DMA1 CH-1 Base/Current count high
0404h	0000	0100	0000	0100	rw	(reserved)
0405h	0000	0100	0000	0101	rw	DMA1 CH-2 Base/Current count high
0406h	0000	0100	0000	0110	rw	(reserved)
0407h	0000	0100	0000	0111	rw	DMA1 CH-3 Base/Current count high
0408h	0000	0100	0000	1000	rw	(reserved)
0409h	0000	0100	0000	1001	rw	(reserved)
040Ah	0000	0100	0000	1010	rw	DMA1 Set Chaining mode (w) Interrupt Status (r)
040Bh	0000	0100	0000	1011	rw	DMA1 Ext Write Mode register (w) (reserved) (r)
040Ch	0000	0100	0000	1100	ro	Host CPU/EISA Master Control
040Dh	0000	0100	0000	1101	ro	Stepping Level register
040Eh	0000	0100	0000	1110	rw	ISP Test Register (reserved)
040Fh	0000	0100	0000	1111	rw	ISP Test Register (reserved)
0461h	0000	0100	0110	0001	rw	Extended NMI and reset control
0462h	0000	0100	0110	0010	wo	NMI I/O interrupt port (casual)
0464h	0000	0100	0110	0100	ro	Last 32-bit bus master granted (L)
0480h	0000	0100	1000	0000	rw	(reserved)
0481h	0000	0100	1000	0001	rw	DMA High Page Register CH-2 Page
0482h	0000	0100	1000	0010	rw	DMA High Page Register CH-3 Page
0483h	0000	0100	1000	0011	rw	DMA High Page Register CH-1 Page
0484h	0000	0100	1000	0100	rw	(reserved)
0485h	0000	0100	1000	0101	rw	(reserved)
0486h	0000	0100	1000	0110	rw	(reserved)
0487h	0000	0100	1000	0111	rw	DMA High Page Register CH-0 Page
0488h	0000	0100	1000	1000	rw	(reserved)
0489h	0000	0100	1000	1001	rw	DMA High Page Register CH-6 Page
048Ah	0000	0100	1000	1010	rw	DMA High Page Register CH-7 Page
048Bh	0000	0100	1000	1011	rw	DMA High Page Register CH-5 Page
048Ch	0000	0100	1000	1100	rw	(reserved)
048Dh	0000	0100	1000	1101	rw	(reserved)
048Eh	0000	0100	1000	1110	rw	(reserved)
048Fh	0000	0100	1000	1111	rw	DMA High Page Register Refresh
04C2h	0000	0100	1100	0010	rw	(reserved)
04C6h	0000	0100	1100	0110	rw	DMA2 CH-5 Base/Current count high
04CAh	0000	0100	1100	1010	rw	DMA2 CH-6 Base/Current count high
04CEh	0000	0100	1100	1110	rw	DMA2 CH-7 Base/Current count high

NOTE:
When read from or written to, all of the above reserved registers (except for 40Ah and 40Bh) will not respond and CSOUT# will not be asserted. Reserved registers 40Ah and 40Bh when read, will return an indeterminate value.

13.0 ISP PORT ADDRESS / I/O DECODING (Continued)

Address Bits					Register Type	Device
Port	FEDC	BA98	7654	3210		
04D0h	0000	0100	1101	0000	rw	INT-1 Edge Level Control Register
04D1h	0000	0100	1101	0001	rw	INT-2 Edge Level Control Register
04D2h	0000	0100	1101	0010	rw	(reserved)
04D3h	0000	0100	1101	0011	rw	(reserved)
04D4h	0000	0100	1101	0100	rw	DMA2 Set Chaining mode (w) Chaining mode (r)
04D5h	0000	0100	1101	0101	rw	(reserved)
04D6h	0000	0100	1101	0110	rw	DMA2 Ext Write Mode register (w) (reserved) (r)
04D7h	0000	0100	1101	0111	rw	(reserved)
04D8h	0000	0100	1101	1000	rw	(reserved)
04D9h	0000	0100	1101	1001	rw	(reserved)
04DAh	0000	0100	1101	1010	rw	(reserved)
04DBh	0000	0100	1101	1011	rw	(reserved)
04DCh	0000	0100	1101	1100	rw	(reserved)
04DDh	0000	0100	1101	1101	rw	(reserved)
04DEh	0000	0100	1101	1110	rw	(reserved)
04DFh	0000	0100	1101	1111	rw	(reserved)
04E0h	0000	0100	1110	0000	rw	DMA CH-0 Stop Reg Bits <7:3>
04E1h	0000	0100	1110	0001	rw	DMA CH-0 Stop Reg Bits <15:8>
04E2h	0000	0100	1110	0010	rw	DMA CH-0 Stop Reg Bits <23:16>
04E3h	0000	0100	1110	0011	rw	(reserved)
04E4h	0000	0100	1110	0100	rw	DMA CH-1 Stop Reg Bits <7:2>
04E5h	0000	0100	1110	0101	rw	DMA CH-1 Stop Reg Bits <15:8>
04E6h	0000	0100	1110	0110	rw	DMA CH-1 Stop Reg Bits <23:16>
04E7h	0000	0100	1110	0111	rw	(reserved)
04E8h	0000	0100	1110	1000	rw	DMA CH-2 Stop Reg Bits <7:2>
04E9h	0000	0100	1110	1001	rw	DMA CH-2 Stop Reg Bits <15:8>
04EAh	0000	0100	1110	1010	rw	DMA CH-2 Stop Reg Bits <23:16>
04EBh	0000	0100	1110	1011	rw	(reserved)
04ECh	0000	0100	1110	1100	rw	DMA CH-3 Stop Reg Bits <7:2>
04EDh	0000	0100	1110	1101	rw	DMA CH-3 Stop Reg Bits <15:8>
04EEh	0000	0100	1110	1110	rw	DMA CH-3 Stop Reg Bits <23:16>
04EFh	0000	0100	1110	1111	rw	(reserved)
04F0h	0000	0100	1111	0000	rw	(reserved)
04F1h	0000	0100	1111	0001	rw	(reserved)
04F2h	0000	0100	1111	0010	rw	(reserved)
04F3h	0000	0100	1111	0011	rw	(reserved)
04F4h	0000	0100	1111	0100	rw	DMA CH-5 Stop Reg Bits <7:2>
04F5h	0000	0100	1111	0101	rw	DMA CH-5 Stop Reg Bits <15:8>
04F6h	0000	0100	1111	0110	rw	DMA CH-5 Stop Reg Bits <23:16>
04F7h	0000	0100	1111	0111	rw	(reserved)
04F8h	0000	0100	1111	1000	rw	DMA CH-6 Stop Reg Bits <7:2>
04F9h	0000	0100	1111	1001	rw	DMA CH-6 Stop Reg Bits <15:8>
04FAh	0000	0100	1111	1010	rw	DMA CH-6 Stop Reg Bits <23:16>
04FBh	0000	0100	1111	1011	rw	(reserved)
04FCh	0000	0100	1111	1100	rw	DMA CH-7 Stop Reg Bits <7:2>
04FDh	0000	0100	1111	1101	rw	DMA CH-7 Stop Reg Bits <15:8>
04FEh	0000	0100	1111	1110	rw	DMA CH-7 Stop Reg Bits <23:16>
04FFh	0000	0100	1111	1111	rw	(reserved)

NOTE:
When read from or written to, reserved registers (4D2h, 4D3h, 4D5h, and 4D7h–4DFh) will not respond and CSOUT# will not be asserted. Reserved registers (4E3h, 4EFh, 4EBh, 4EFh–4F3h, 4F7h, 4FBh, and 4FFh when read, will return a value of FFh. Reserved register 4D6h when read, will return an indeterminate value.

82358
EISA BUS CONTROLLER (EBC)

- **Provides EISA/ISA Bus Cycle Compatibility**
  - **— EISA/ISA Standard Memory or I/O Cycle**
  - **— EISA/ISA Wait State Cycles**
  - **— ISA No Wait State Cycle**
  - **— EISA Burst Cycles**
- **Interfaces Host (CPU) Bus to EISA/ISA Bus**
- **Supports 386™ & i486™ Microprocessors**
  - **— 25 MHz & 33 MHz 386 Systems**
  - **— 25 MHz & 33 MHz i486 Systems**
- **Translates Host Bus Cycles to EISA/ISA Bus Cycles**
- **Generates ISA Signals for EISA Masters**
- **Generates EISA Signals for ISA Masters**
- **Supports 8-, 16-, or 32-Bit DMA Cycles**
  - **— Type A, B, or C (Burst) Cycles**
  - **— Compatible Cycles**
- **Supports Host and EISA/ISA Refresh Cycles**
- **Generates Control Signals for Address and Data Buffers**
- **Supports Byte Assembly/Disassembly for 8-, 16-, or 32-Bit Data Transfers**
- **Cache Controller (82385) Interface to Maximize Performance for 386 Based Systems**
- **Supports I/O Recovery Mechanism**
- **Generates 82385, CPU, and System Software Resets**
- **132-Pin PQFP Package**
- **Low Power CHMOS Technology**

The 82358 EISA Bus Controller (EBC) is part of Intel's 82350 EISA chip set. The EBC interfaces either the 386 or i486 microprocessor to the Extended Industry Standard Architecture (EISA) bus. The 82358 (EBC) is designed to facilitate bus cycles between the Host (CPU) and EISA/ISA bus. The EBC generates the appropriate data conversion and alignment control signals to implement an external byte assembly/disassembly mechanism for transferring data of equal or different widths between the Host, Industry Standard Architecture (ISA) and EISA busses. The EBC translates cycles between EISA, ISA and Host busses.

The EBC is tightly coupled with the 82357 DMA controller (ISP) to run 8-, 16-, or 32-bit EISA/ISA DMA transfers between busses. The EBC features special cache hardware interface signals to implement the highest performance 386 based system with the 82385 cache controller.

The EBC features hardware enforced I/O recovery logic to provide I/O recovery time between back to back I/O cycles.

The EBC provides resets to the 82385, 386, i486, and other devices in the system to provide an integrated synchronous system reset.

October 1990
Order Number: 290252-003

82358 EISA BUS CONTROLLER (EBC)

CONTENTS

PAGE

1.0 INTRODUCTION 1-277
1.1 EBC System Architecture Overview 1-277
1.2 Internal Architecture Overview 1-277
1.2.1 Host Bus Interface Unit 1-277
1.2.2 EISA and ISA Interface Bus Units 1-277
1.2.3 ISP Interface Unit 1-277
1.2.4 Address Buffer Control Unit 1-279
1.2.5 Data Buffer Control Unit 1-279
1.2.6 Cache Support 1-279
1.2.7 Reset Control 1-279
1.2.8 I/O Recovery Support 1-279
1.2.9 Slot Support 1-279
1.2.10 Clock Generation Unit 1-279

2.0 82358 PIN DESCRIPTION 1-280
2.1 Host Bus Interface Signals 1-280
2.2 EISA Bus Interface Signals 1-285
2.3 ISA Bus Interface Signals 1-288
2.4 ISP Device Interface Signals 1-294
2.5 Data Buffer Control Signals 1-295
2.6 Address Buffer Control Signals 1-296
2.7 Reset Control Signals 1-298
2.8 Configuration Signals 1-299
2.9 Miscellaneous Signals 1-299

3.0 BASIC FUNCTIONALITY AND TIMINGS 1-302
3.1 EISA/ISA Bus Cycles 1-302
3.2 EBC Cycle Translation Algorithm 1-302
3.3 Clock Generation 1-306
3.4 Functional Timing Diagram Descriptions 1-306
3.5 Timing Diagrams 1-309

4.0 EBC PINOUT INFORMATION 1-366
4.1 EBC Device Pin List 1-369

5.0 D.C. SPECIFICATIONS 1-371
5.1 Maximum Ratings 1-371
5.2 D.C. Specifications for the EISA Bus Controller (EBC) 1-371

6.0 A.C. SPECIFICATIONS 1-372

TABLES

PAGE

Table 2-1 HM/IO# and M-IO Decode 1-282
Table 2-2 ISA Memory Command Signal Generation 1-285
Table 2-3 BE (3:0) # Signal Translation 1-285
Table 2-4 BCLK Generation from HCLKCPU 1-288
Table 2-5 CHRDY and EXRDY Functional Description 1-292
Table 2-6 ST (3:0) Decode 1-295
Table 2-7 CPU Type/Frequency Indication 1-299
Table 2-8 CLKKB Generation 1-299
Table 3-1 EISA/ISA/DMA Bus Cycles 1-302
Table 3-2 EBC Cycle Translation 1-303

FIGURES

Figure 1-1 EBC Internal Block Diagram 1-278
Figure 2-1 HNA# Timings 1-281
Figure 2-2a Host Master to Host Slave Cycle Snoop Strobe Timings (25 MHz or 33 MHz) ... 1-284
Figure 2-2b EISA Master to Host Slave or DMA Burst Cycle Snoop Strobe Timings (25 MHz) 1-284
Figure 2-2c EISA Master to Host Slave or DMA Burst Cycle Snoop Strobe Timings (33 MHz) 1-284
Figure 2-3 (a–b) Host Master Lock Cycle to EISA Slave 1-287
Figure 2-4 (a–b) BCLK Stretching for 386 and i486 CPUs 1-289
Figure 2-5 (a–c) BCLK Stretching Conditions 1-290
Figure 2-6 CHRDY and EXRDY in ISA Master to EISA Memory Slave Cycle 1-293
Figure 2-7 Host/EISA Bus Data Swap Logic 1-297
Figure 2-8 Host/EISA Bus Address Buffers 1-298
Figure 2-9 (a–b) LIOWAIT# Control 1-300
Figure 2-10 AENLE# Timing Diagram 1-301
Figure 3-1 386 CPU Host Master to 16-Bit ISA/Host/EISA/Host Slave 1-305
Figure 3-1A Clock Generation CLK1 Relation to HCLKCPU 1-306

TIMING DIAGRAM LIST

Figure 3-2 Host Master to Host Memory Slave Standard Read/Write Cycles 1-309
Figure 3-3 Host Master to 32-Bit EISA Memory I/O Slave Standard Read/Write Cycles 1-310
Figure 3-4 Host Master to 32-Bit EISA Memory Slave Two Standard Read Cycles 1-311
Figure 3-5 Host Master to 32-Bit EISA Memory Slave Locked Read Cycle 1-312
Figure 3-6 Host Master to 16-Bit EISA Memory Slave Read Assembly Cycle 1-313
Figure 3-7 Host Master to 16-Bit EISA Memory Slave Disassembly Write Cycles 1-314
Figure 3-8 Host Master to 16-Bit EISA I/O Slave Byte Swap Write Cycle with Wait States ... 1-315
Figure 3-9 Host Master to 16-Bit ISA Memory Slave Assembly Read Cycle 1-316
Figure 3-10 Host Master to 8-Bit ISA Memory Slave Assembly Read Cycle 1-317
Figure 3-11 Host Master to 8-Bit ISA Slave Disassembly Write Cycle 1-318

TIMING DIAGRAM LIST (Continued)

PAGE

Figure 3-12 32-Bit EISA Master to Host Memory Slave Standard Write/Read Cycles 1-319
Figure 3-13 32-Bit EISA Master to 32-Bit EISA I/O Slave Standard Write Cycle 1-320
Figure 3-14 32-Bit EISA Master to 32-Bit EISA Memory Slave 1 Wait State Burst Read Cycle 1-321
Figure 3-15 32-Bit EISA Master to 16-Bit EISA I/O Slave Assembly Read Cycle 1-322
Figure 3-16 32-Bit EISA Master to 16-Bit EISA Memory Slave Assembly/Disassembly Read/Write Cycles 1-323
Figure 3-17 32-Bit EISA Master to 16-Bit EISA Memory Slave No Assembly Write Cycle with Wait States 1-324
Figure 3-18 32-Bit EISA Master to 16-Bit EISA Memory Slave Assembly Read during 0 Wait State Burst Cycles 1-325
Figure 3-19 32-Bit EISA Master to 16-Bit EISA Memory Slave 0 Wait State Burst Disassembly Write Cycle 1-326
Figure 3-20 32-Bit EISA Master to 16-Bit ISA Memory Slave NOWS# Disassembly Write Cycle 1-327
Figure 3-21 32-Bit EISA Master to 16-Bit ISA I/O Slave Standard Read/Write Cycles 1-328
Figure 3-22 32-Bit EISA to 16-Bit ISA I/O Slave 1 Wait State Disassembly Write Cycles 1-329
Figure 3-23 32-Bit EISA Master to 8-Bit ISA Memory Slave NOWS# No Assembly Read Cycle 1-330
Figure 3-24 32-Bit EISA Master to 8-Bit ISA Memory Slave NOWS# Assembly Write Cycle 1-331
Figure 3-25 16-Bit EISA Master to 32-Bit EISA Memory I/O Slave Byte Swap Read/Write Cycle 1-332
Figure 3-26 16-Bit EISA Master to 16-Bit EISA Memory Slave Burst Read Cycle 1-333
Figure 3-27 16-Bit EISA Master to 32-Bit EISA Memory Slave 1 Wait State Byte Swap Burst Read Cycle 1-334
Figure 3-28 16-Bit EISA Master to 16-Bit ISA I/O No Assembly Read Cycle 1-335
Figure 3-29 16-Bit EISA Master to 16-Bit ISA Memory Slave Standard Write Cycle 1-336
Figure 3-30 16-Bit EISA Master to 8-Bit ISA Memory Slave Disassembly Write Cycle 1-337
Figure 3-31 16-Bit ISA Master to 32-Bit EISA Memory Slave 1 Wait State Write Cycle 1-338
Figure 3-32 16-Bit ISA Master to 32-Bit EISA I/O Slave Write Cycle with Wait States 1-339
Figure 3-33 16-Bit ISA Master to 16-Bit EISA Memory Slave BCLK Stretched Read Cycle 1-340
Figure 3-34 16-Bit ISA Master to 8-Bit ISA Memory Slave Write Cycle 1-341
Figure 3-35 16-Bit ISA Master to 8-Bit ISA I/O Slave Standard Read Cycle 1-342
Figure 3-36 Halt/Shutdown Cycle 1-343
Figure 3-37 SPWROK 1-344
Figure 3-38 RSTAR# < 32 Clocks 1-345
Figure 3-39 RSTCPU/HHOLD Signal Interlock 1-346
Figure 3-40 Compatible DMA Read Cycle (Memory Size .GE. I/O Size) 1-347
Figure 3-41 Compatible DMA Write Cycle (Memory Size .GE. I/O Size) 1-347
Figure 3-42 Type A DMA Read Cycle (Memory Size .GE. I/O Size) 1-348
Figure 3-43 Type A DMA Write Cycle (Memory Size .GE. I/O Size) 1-348
Figure 3-44 Type B DMA Read Cycle (Memory Size .GE. I/O Size) 1-349
Figure 3-45 Type B DMA Write Cycle (Memory Size .GE. I/O Size) 1-349

TIMING DIAGRAM LIST (Continued)

PAGE

Figure 3-46 Burst DMA Read Cycle (Memory Size .GE. I/O Size) 1-350
Figure 3-47 Burst DMA Write Cycle (Memory Size .GE. I/O Size) 1-350
Figure 3-48 Burst DMA Read Cycle to Non-Burst Memory (Memory Size .GE. I/O Size) 1-351
Figure 3-49 Burst DMA Write Cycle to Non-Burst Memory (Memory Size .GE. I/O Size) 1-351
Figure 3-50 Type A DMA Read Cycle (Memory Size .LT. I/O Size) 1-352
Figure 3-51 Type B Read Cycle (Memory Size .LT. I/O Size) 1-352
Figure 3-52 Burst DMA Read with Assembly Required (Memory Size .LT. I/O Size) 1-353
Figure 3-53 Refresh Cycle 1-353
Figure 3-54 32-Bit I/O to 32-Bit Memory Burst Read DMA Cycle 1-354
Figure 3-55 32-Bit I/O to 16-Bit Memory Compatible Read DMA Cycle 1-355
Figure 3-56 32-Bit I/O to 16-Bit Memory Type A Write DMA Cycle 1-356
Figure 3-57 32-Bit I/O to 16-Bit Memory Type B Write DMA Cycle 1-357
Figure 3-58 32-Bit I/O to 8-Bit Memory Compatible Read DMA Cycle 1-358
Figure 3-59 16-Bit I/O to 32-Bit Memory Type B Write DMA Cycle 1-359
Figure 3-60 16-Bit I/O to 16-Bit Memory Type A Read DMA Cycle 1-360
Figure 3-61 16-Bit I/O to 32-Bit Memory Burst Read DMA Cycle 1-361
Figure 3-62 16-Bit I/O to 16-Bit Memory Burst Read DMA Cycle 1-362
Figure 3-63 8-Bit I/O to 32-Bit Memory Compatible Write DMA Cycle 1-363
Figure 3-64 8-Bit I/O to 16-Bit Memory Read Type B DMA Cycle 1-364
Figure 3-65 8-Bit I/O to 8-Bit Memory Write Compatible DMA Cycle 1-365

1.0 INTRODUCTION

1.1 EBC System Architecture Overview

The EISA Bus Controller (EBC) is located between the host (CPU) bus and the EISA/ISA bus. The EBC watches cycles initiated on either bus. When a host bus master initiates a cycle and no host slave responds, the EBC forwards the cycle to the EISA/ISA bus. The EBC provides the control for the address and data buffers between the two busses. The EBC also inserts delays between back to back I/O cycles coming from the host bus to the ISA bus for the purposes of I/O recovery.

The EBC will support frequencies of 25 MHz and 33 MHz for the 386 or i486 microprocessor.

The EBC is the bridge between ISA and EISA devices on the EISA/ISA bus. The EBC translates cycles from EISA masters into cycles that ISA slaves can handle. Similarly, it translates cycles initiated by ISA masters into cycles that the EISA slaves understand. The EBC also performs byte assembly/disassembly for data transfers from EISA masters to EISA/ISA slaves.

The EBC interfaces with the 82357 Integrated System Peripheral (ISP), which contains the high performance EISA compatible DMA controller, interrupt controller, refresh logic and other integrated peripheral functions. When requested by the ISP, the EBC runs EISA bus cycles for DMA transfers and memory refreshes on behalf of the ISP.

The EBC is packaged in a 132-pin plastic quad flat pack (PQFP).

The following is a discussion of the major internal functional blocks in the EISA Bus Controller which are shown in Figure 1-1.

1.2 82358 Internal Architecture Overview

1.2.1 HOST BUS INTERFACE UNIT

The Host bus interface unit interfaces to the 386 or i486 CPU host bus. It runs at the operating frequency of the host processor.

The Host bus interface unit monitors host bus cycles when the host CPU is the current master on the host bus. If no device on the host bus responds, then it signals the EISA or ISA interface unit to run a cycle. The Host bus interface unit waits for the EISA or ISA interface unit to complete the cycle, and then it terminates the cycle on the host bus. The Host bus interface unit takes advantage of the 386 address pipelining capabilities for 386 systems.

There is a special protocol using the host bus signal HSTRETCH#, which enables BCLK low time to be stretched in CLK1 increments. This allows slow host slave devices to use HSTRETCH# to run zero wait state EISA/ISA cycles.

1.2.2 EISA AND ISA INTERFACE BUS UNITS

These state machine units interface to the EISA and ISA bus. They run at the EISA bus clock (BCLK) frequency.

The EISA unit interfaces with the EISA bus and executes EISA cycles on behalf of the ISA, host, or ISP master on the EISA bus. In this way, any master can communicate with the EISA bus.

Similarly, the ISA unit interfaces with the ISA bus and executes ISA cycles on behalf of the EISA, host, or ISP master on the ISA bus.

The EISA and ISA units monitor cycles initiated by EISA or ISA masters and watch their corresponding busses to detect a slave response. The correct cycle will then be initiated to the responding slave's bus (ISA or EISA). This provides the bridge between both types of busses (ISA or EISA).

The EISA and ISA units accept cycles from the host interface unit, and will run cycles on the EISA or ISA bus. If necessary, the EISA and ISA units will perform multiple cycles (assembly/disassembly) for every host cycle. When this is done, they signal the host interface unit to terminate the cycle on the host bus. The EISA and ISA units also inform the host interface unit when the host can change the address for the next cycle.

It will run multiple EISA or ISA cycles for data byte assembly or disassembly, if necessary, when the master and slave data bus width does not match.

The EISA and ISA interface units also accept requests from the ISP interface unit to run DMA cycles on the EISA or ISA bus.

1.2.3 ISP INTERFACE UNIT

The ISP interface unit accepts requests from the ISP for DMA or refresh cycles, and then requests the ISA and EISA interface units to run the appropriate cycle. There are four types of DMA cycles: compatible, type A, type B, and type C (burst). If the memory slave width does not match the I/O slave width, the EISA/ISA interface unit will perform the byte assembly or disassembly, depending on the case, through multiple EISA/ISA cycles.

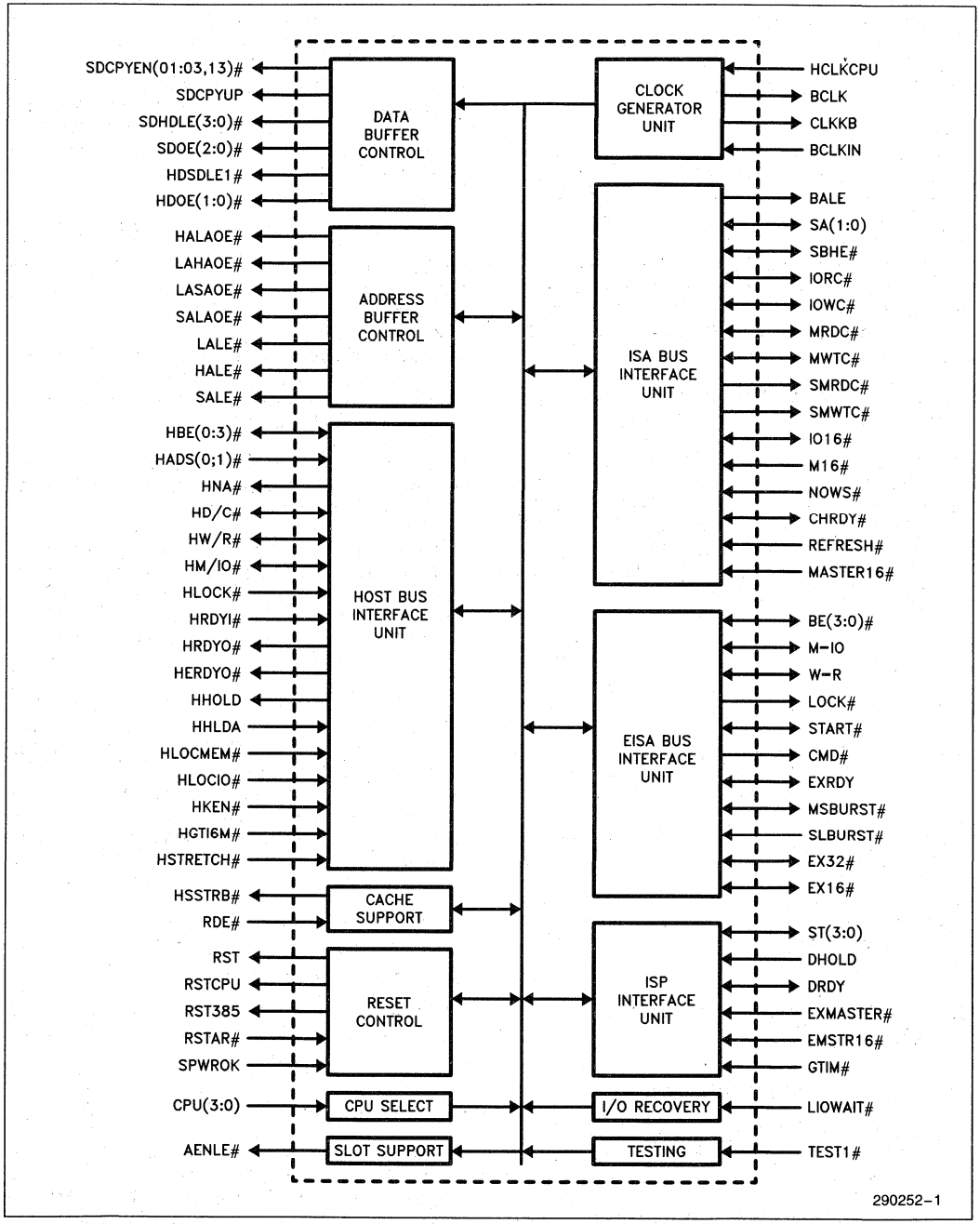


Figure 1-1. EBC Internal Block Diagram

1.2.4 ADDRESS BUFFER CONTROL UNIT

The address buffer control unit controls the external bidirectional address buffers between the host and EISA/ISA bus. It provides the latching clocks and output enables for these buffers. Figure 2-8 shows how the signals from this unit are used.

1.2.5 DATA BUFFER CONTROL UNIT

This unit controls the data buffers between the host and EISA/ISA bus. It provides the latching clocks and the output enables. The data buffer control unit interfaces with the host, EISA, and ISA interface units to provide buffer control during assembly/disassembly of data, which allows copying of data between byte lanes. Figure 2-7 shows how the signals from this unit of the EBC are used when data byte swapping is needed for mismatched master-slave combinations. ISA bus master copy operations are not supported by the EBC and must be externally implemented.

1.2.6 CACHE SUPPORT

The cache support logic instructs the 82385 cache controller when to snoop the system bus (HSSTRB#). HSSTRB# is a synchronized strobe indicating a valid address during non-microprocessor or CPU to memory write cycles.

The cache support has the option (RDE#) to add a wait state by delaying HERDYO# and HRDYO# by 1 CLK1 during a host to EISA bus read to allow for increased cache SRAM write data setup time during cache read miss cycles. See Figure 3-1a for the CLK1 definition.

For i486 special cycles, (flush and writeback only) the EBC treats these as halt cycles, generating a READY signal in zero wait states.

1.2.7 RESET CONTROL

This unit generates 3 types of resets:

1. Reset for the CPU (RSTCPU)

   This results under three conditions: power on, shut down detect from the host bus master, and assertion of the restart signal (RSTAR#). The RSTCPU signal is interlocked with HHOLD, such that only one of them will be active at a time, preventing 386 bus contention when the DMA or EISA/ISA bus master is active.

2. Reset for the 82385 cache controller (RST385)

   This is the same reset as for the CPU, but RST385 is delayed so that the CPU completes its cycles before the 82385 is reset. This prevents the 82385 from erroneously glitching its BADS# output signal. For software resets (RSTAR# active) and shutdown cycles, RST385 active edge is delayed from RSTCPU active edge. For hardware resets (SPWROK active), the EBC generates RST385 and RSTCPU simultaneously.

3. Reset for the ISP (RST)

   This is generated at power on, after BCLK is stable.

1.2.8 I/O RECOVERY SUPPORT

This unit forces a delay between back to back 8-bit and 16-bit ISA I/O cycles originating on the host bus. A controlled delay is inserted depending on the state of the LIOWAIT# pin. LIOWAIT# is sampled on the trailing edge of the I/O command signal (IORC#, IOWC#, CMD#), and the delay is inserted before the leading edge of the next ISA I/O cycle. If the last sub-cycle to an ISA slave is followed by another I/O command signal (IORC#, IOWC#, CMD#) to any slave, a delay will be inserted. No delay is inserted for EISA, ISA or DMA bus master cycles; EISA cycles originating on the host bus; or back to back I/O "sub-cycles" generated as a result of byte assembly or disassembly. EISA master accesses to ISA I/O slaves needs I/O recovery time of 1 BCLK provided by software.

The minimum delay is 1 BCLK following the trailing edge of the ISA I/O command signal prior to issuing START# for the next I/O command signal.

The maximum delay is 3 BCLKs following the trailing edge of a 16-bit ISA I/O command signal or 11 BCLKs following the trailing edge of an 8-bit ISA I/O command signal prior to the leading edge of START# for another separate I/O command cycle. Any other delay lengths between minimum and maximum during I/O recovery is determined by LIOWAIT# becoming inactive.

1.2.9 SLOT SUPPORT

The slot support logic generates an address latch enable (AENLE#) signal when the LA address is valid for the purposes of generating the slot specific AENx# signals.

1.2.10 CLOCK GENERATION UNIT

This unit generates three clocks:

1. Host State Machine Clock (CLK1) runs at the same frequency as the internal clock of the Host CPU. It is synchronized to the entire system board by SPWROK. For 386 systems, CLK1 is generated internally from HCLKCPU divided by two and inverted. For i486 systems, CLK1 is inverted.
2. BCLK—This is the EISA clock that runs at approximately 8 MHz. It is generated by dividing HCLKCPU by 3, 4, 6, or 8, depending on the CPU(3:0) signals, which indicate the CPU type and frequency. This unit also stretches the high or low time of BCLK for synchronization purposes.

3. CLKKB—This is the clock for the keyboard processor. It is generated by dividing HCLKCPU by 3, 5, or 6, also depending on the CPU(3:0) signals.

2.0 82358 PIN DESCRIPTION

This section defines the specific functions of the pins on the EISA bus controller. All signal descriptions apply to both the 386 and i486 CPU unless otherwise stated. The symbol # after the signal name indicates an active low signal.

2.1 Host Bus Interface Signals

HOST CPU CLOCK (HCLKCPU) Input

For 386 and i486 systems, HCLKCPU runs at the same frequency as the external clock of the Host CPU. It is synchronized to the entire system board by SPWROK.

HOST ADDRESS STATUS (HADS(1:0)#) Input

The falling edge of either of these two signals indicate that address, status, and byte enable information is valid on the host bus. In the case of pipelined cycles, HADS(1:0)# are expected to stay active low until HRDYI# has been sampled active by the EBC. These two signals are received by the EBC when there is a master on the Host Bus, and are used to track the start of host bus cycles. When no host slave responds (the CPU is addressing an EISA/ISA slave, including the ISP(1)), then the cycle will be forwarded to the EISA bus.

NOTE:

1. The 82357 ISP (Integrated System Peripheral) contains the DMA controller and the refresh logic, along with other functions. Henceforth, references to ISA slaves in this document include the ISP in the slave mode.

HADS0# and HADS1# are "OR"ed together internally in the EBC so that either one can signal the start of a cycle. They are also used for 64 Kbyte and 128 Kbyte cache designs where an independent state machine generates HADS1#. For example, 128 Kbyte 82385 cache systems use more than one fetch to fill a cache line. The 82385 starts the first cycle by generating HADS0# and the state machine finishes the rest of the fetches by generating its own HADS1# signal. In the circuit, HADS0# connects to the 82385, and HADS1# connects to the state machine.

HOST BYTE ENABLES (HBE3#–HBE0#) Bidirectional

HBE(3:0)# are inputs to the EBC when the Host bus master is addressing an EISA/ISA slave and are translated to BE(3:0)#, SBHE#, SA(1:0).

HBE(3:0)# are outputs during EISA bus master cycles, or when the ISP is performing DMA or refresh cycles, and are derived from BE (3:0)#. They are also outputs during ISA bus master cycles, in which case they are derived from SBHE#, SA(1:0). Therefore HBE(3:0)# are outputs during all cycles except CPU cycles.

HBE3# applies to bits 31–24 of the host data bus, HBE2# to bits 23–16, HBE1# to bits 15–8 and HBE0# to bits 7–0.

HOST NEXT ADDRESS (HNA#) Output

When the Host bus master is addressing an EISA/ISA slave, HNA# is active for one host CLK1, (and then it is inactive and tri-stated) after the address has been latched in the external address buffer, indicating that the CPU can put a new address on the host bus. HNA# is active only during Host bus master cycles, and is tri-stated for other cycles. An external 10K pullup resistor is required if HNA# is used. See Figure 2-1. HNA# TIMINGS.

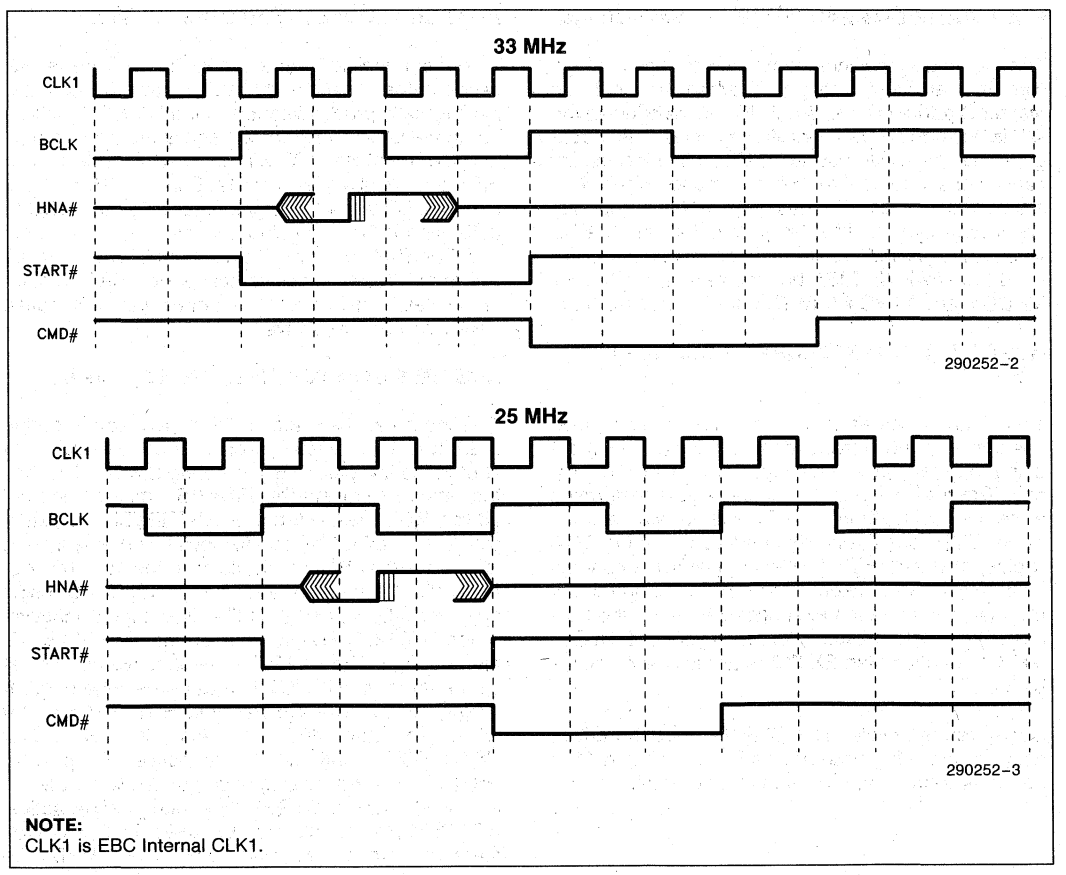
HOST DATA OR CONTROL (HD/C#) Bidirectional

This signal distinguishes between data and control cycles. HD/C# is an input to the EBC when the CPU is master on the Host bus. It is used to decode shutdown and interrupt acknowledge cycles. HD/C# is an output of the EBC when the ISP is performing DMA or refresh cycles, and during EISA/ISA bus master cycles, therefore HD/C# is an output for all cycles except CPU cycles. HD/C# is active (high) whenever it is an output. See note in HM/IO# pin description.

HOST MEMORY OR I/O (HM/IO#) Bidirectional

HM/IO# distinguishes between memory and I/O cycles. The EBC and the address buffers (EBB) share the responsibility of driving HM/IO# and M-IO. Table 2-1 shows how the EBC and the address buffers share this function. For example, when there is an ISA master, M-IO will be driven by the EBC if IORC# or IOWC# is active, and M-IO is an input to the address buffers; HM/IO# will be an input to the EBC, but on the address buffers HM/IO# will follow M-IO. HM/IO# is also used to decode shutdown and interrupt acknowledge cycles.

NOTE:
For back to back cycles where the EBC drives HM/IO# followed by the CPU driving HM/IO#, the EBC may not tri-state HM/IO# before the CPU begins to drive HM/IO#. The resulting potential bus contention does not cause system errors unless the CPU is unable to drive HM/IO# to its correct level before HM/IO# is sampled by another device. This only occurs when the CPU drives HM/IO# low after the EBC drives HM/IO# high (i.e., refresh cycle followed by a CPU write to I/O). The CPU state tracker in the EISA schematics for both 386 microprocessor and i486 microprocessor based systems samples HM/IO# at the end of T1. If the CPU has not driven HM/IO# to the correct logic level by the end of T1, the CPU state tracker will incorrectly interpret the cycle, resulting in system errors. For 386 microprocessor based systems with a 385 cache controller, the HM/IO# signal is usually latched. Since the logic device which latches HM/IO# (74F821 in the EISA schematics for 386 microprocessor based systems) is able to drive HM/IO# quickly even though the EBC has not tri-stated HM/IO#, HM/IO# will contain the correct logic level by the end of T1. However, for i486 microprocessor based systems, the i486 may not be able to drive HM/IO# low before the end of T1. Therefore, a 470Ω resistor is necessary on the HM/IO# line between the i486 microprocessor and the EBC. The 470Ω resistor limits the current on the HM/IO# signal, allowing HM/IO# to be pulled low more quickly.



NOTE:
CLK1 is EBC Internal CLK1.

Figure 2-1. HNA# Timings

Table 2-1. HM/IO# and M-IO Decode

Master	HM/IO#	M-IO
EISA Bus Controller (EBC)		
CPU	Input	Input
DMA (Refresh)	Driven High	Input
EISA	Input	Input
ISA	Input	Driven = IORC# "OR" IOWC#
Address Buffers (EBB)		
CPU	Input	Driven = HM/IO#
DMA (Refresh)	Input	Driven = HM/IO#
EISA	Driven = M-IO	Input
ISA	Driven = M-IO	Input

HOST WRITE OR READ (HW/R#) Bidirectional

This signal distinguishes between write and read cycles. It is an input to the EBC when the Host bus master is addressing an EISA/ISA slave or when the ISP is performing DMA or refresh cycles. It is propagated to the EISA bus as W-R, and if necessary causes the appropriate ISA command. HW/R# is also used to decode shutdown and interrupt acknowledge cycles. HW/R# is an output of the EBC when an EISA/ISA master is in control and it is derived from W-R for EISA bus masters, and IORC#/MRDC# and IOWC#/MWTC# for ISA bus masters.

HOST BUS LOCAL MEMORY (HLOCMEM#) Input

This signal indicates that a host bus memory slave has decoded the current address as its own (without being internally preconditioned to the HM/IO# signal). When the CPU is the master and is performing a memory cycle, then the EBC will not start a cycle on the EISA bus if this signal is active. During DMA cycles, or during EISA/ISA bus master memory cycles, this signal is used by the EBC to determine if the memory address being accessed is on the host bus. For EISA master memory cycles, if HLOCMEM# is active then EX32# is always driven active by the EBC.

During refresh cycles HLOCMEM# must be driven active at the rising edge of START#. T10b and T10c set up and hold requirements must be met.

HOST BUS LOCAL I/O (HLOCIO#) Input

This signal indicates that a host bus I/O slave has decoded the current address as its own (without being internally preconditioned to the HM/IO# signal). When the CPU is the master, and is performing an I/O cycle, then the EBC will not start a cycle on the EISA bus if this signal is active. During EISA/ISA bus master I/O cycles this signal is used by the EBC to determine if the I/O being accessed is on the host bus. For EISA master I/O cycles, if HLOCIO# is active then EX32# is always active. For ISA bus master cycles, if HLOCIO# is active this will cause IO16# to become active.

HOST BUS STRETCH (HSTRETCH#) Input

This input can be used by host bus slaves during EISA/ISA bus master cycles or DMA cycles to stretch the low part of BCLK during CMD#. This has the effect of stalling the EISA/ISA master, without adding BCLK wait states. If HSTRETCH# is sampled active, BCLK will remain low. It is then sampled at every other HCLKCPU rising edge for 386 systems (CLK1 interval) and every HCLKCPU rising edge for i486 systems, until it is sampled inactive. When HSTRETCH# is sampled inactive BCLK will then go high. The HSTRETCH# inactive edge must be set-up to the HCLKCPU rising edge where BCLK is to go high. Note: Do not hold HSTRETCH# low for more than 400 ns, and do not assert HSTRETCH# until CMD# is active. To prevent HSTRETCH# from being driven active while CMD# is inactive, gate HSTRETCH# with CMD# active. HSTRETCH# should not be used in DMA burst memory write cycles. Use EXRDY or CHRDY instead.

HOST BUS READY INPUT (HRDYI#) Input

This signal indicates the cycle termination on the host bus. HRDYI# is monitored when the CPU is addressing host slaves, in order to track CPU cycles. When the CPU is accessing an EISA slave, the EBC does back-to-back EISA bus cycles whenever possible; it does not wait for HRDYI# after the EISA cycle ends, so HRDYI# cannot be delayed by more than 1 CLK1 from HERDYO#.

HOST BUS READY OUTPUT (HRDYO#) Output

HRDYO# is driven for CPU master to EISA/ISA slave, halt, shutdown, and ISP cycles. When the CPU is addressing an EISA/ISA slave, this signal is driven active to indicate that the EBC has completed the cycle, as generated from EXRDY, CHRDY, NOWS#, DRDY, whichever is appropriate. In the next cycle, HRDYO# is driven inactive and then tri-stated. For halt and shutdown cycles, the EBC responds with a HRDYO# in zero CPU wait states (non-pipelined).

HOST BUS EARLY READY (HERDYO#) Output

This is an early version of the ready output from the EBC to be synchronized externally for use with the higher frequency CPUs where HRDYO# does not provide adequate set-up time. During CPU halt and shutdown cycles, HERDYO# is not any earlier than HRDYO#. The need for external synchronization for higher frequency CPUs could cause an extra wait state to be added for halt/shutdown cycles. Unlike HRDYO#, HERDYO# is not tri-stated after being driven inactive.

HOST BUS HOLD REQUEST (HHOLD) Output

This is the hold request to the host, and is driven active by the EBC when the ISP drives DHOLD active to indicate an EISA/ISA bus master requests control or that a DMA device needs service. This is synchronous to the host CPU's internal clock (HCLKCPU/2 for 386 and HCLKCPU for i486), and meets the set-up and hold time requirement for a 25 MHz CPU. At 33 MHz, external synchronization is required. This signal is interlocked with RSTCPU such that only one of them will become active at a time.

HOST HOLD ACKNOWLEDGE (HHLDA) Input

This input to the EBC is synchronous to the host CPU's internal clock and is driven active by the host bus master to indicate that it has relinquished control of the bus.

HOST BUS LOCK (HLOCK#) Input

This signal is driven active by the host bus master when a locked bus cycle is occurring on the Host bus. If the addressed device is on the EISA bus, HLOCK# is propagated to LOCK# on the EISA bus. HLOCK# must be sampled inactive before the EBC will drive LOCK# inactive on the EISA bus. Figure 2-3(a-b) show examples of locked cycles.

HOST SNOOP STROBE (HSSTRB#) Output

This signal is driven by the EBC during any memory write cycle on the host bus; for example, I/O to memory DMA cycles, or EISA/ISA bus master memory write cycles, or CPU memory write cycles. In the 386 system, the 82385 cache controller uses HSSTRB# to maintain cache coherency with data stored in EISA or host memory. To maintain cache coherency in the i486 system, HSSTRB# is tied to EADS# on the i486 CPU.

However, in order to prevent the i486 microprocessor from snooping the bus during host write cycles, the HSSTRB# signal must be qualified with HHLDA active. If the i486 microprocessor is performing a write cycle and snoops the bus, it will invalidate the tag of the cache line which it has just written to its internal cache. For i486 microprocessor systems operating at 33 MHz, the active edge of HSSTRB# needs to be externally synchronized to HCLKCPU with a registered, E-speed PLD in order to meet timing requirements. The HSSTRB# signal must be qualified with HHLDA and synchronized to HCLKCPU in the same registered, E-speed PLD. HHSTRB# can not be delayed by one or more HCLKCPU periods in 25 MHz systems.

See Figures 2-2(a-c) for Snoop Strobe timings during HOST, EISA, and DMA master cycles. During DMA and EISA master non-burst cycles HSSTRB# is generated for one CLK1 duration after START# is driven inactive.

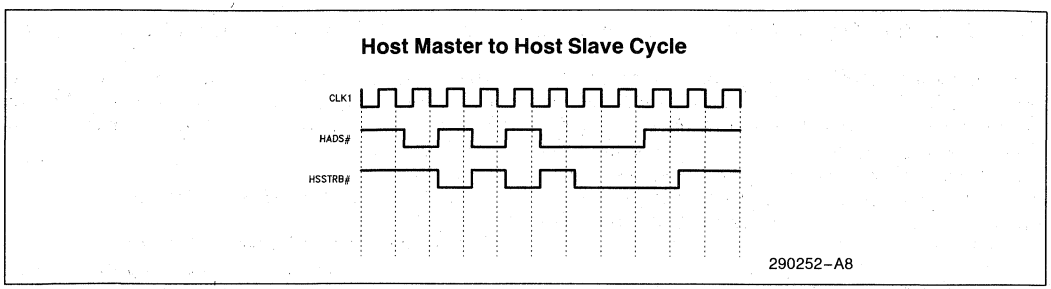


Figure 2-2a. Snoop Strobe Timings (25 MHz or 33 MHz)

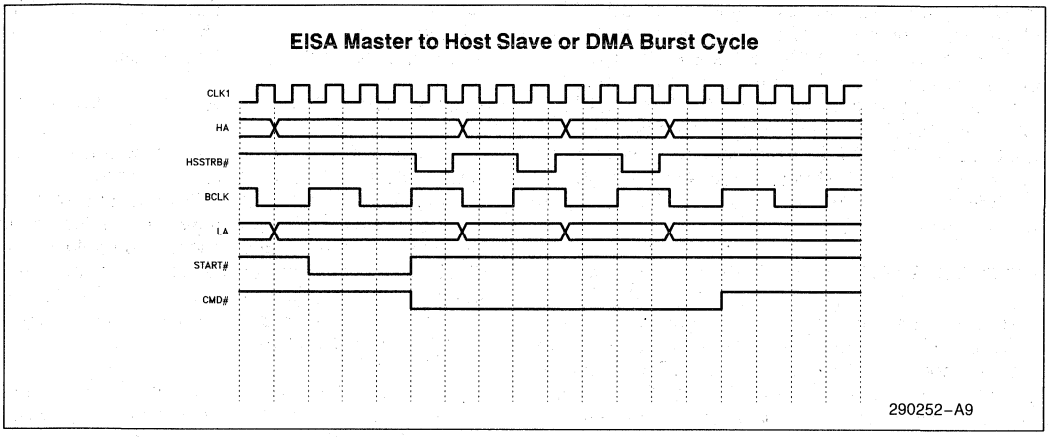


Figure 2-2b. Snoop Strobe Timing (25 MHz)

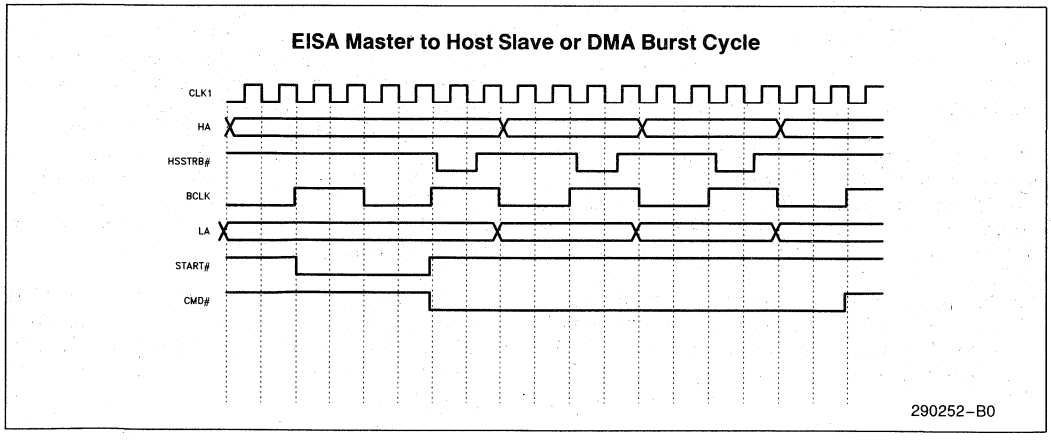


Figure 2-2c. Snoop Strobe Timing (33 MHz)

HOST GREATER THAN 16 MEGABYTES (HGT16M#) Input

This signal indicates the address of the current cycle is greater than the range of 00000000h to 00FFFFFFh (16 Mbytes). HGT16M# is driven by the ISP for all addresses above 16 Mbytes for DMA cycles only. The EBC uses this signal during DMA cycles to determine whether or not to generate ISA memory command signals, MRDC# and MWTC#. These ISA memory command signals (MRDC#, MWTC#) will always be generated during DMA cycles, except when a 32-bit or 16-bit EISA memory slave, or host memory slave responds and HGT16M# is active, as shown in Table 2-2.

Table 2-2. ISA Memory Command Signal Generation

Memory Slave Responding	CPU Master Address Range		DMA Compatible Cycle Address Range		DMA Type A/B Address Range		DMA Type C Address Range	
	<16M	>16M	<16M	>16M	<16M	>16M	<16M	>16M
ISA Slave	ISAMCMD	ISAMCMD	ISAMCMD	ISAMCMD	ISAMCMD	ISAMCMD	NA	NA
EISA SLAVE			ISAMCMD (Note 1)	(Note 2)	(Note 3)	(Note 2)		
Host Slave								

For example, the diagram shows that the EBC will generate MRDC# or MWTC# if only an ISA slave responds to an address below 16 Meg or above 16 Meg during a DMA compatible cycle. Secondly, the EBC will generate MRDC# or MWTC# if only an EISA slave responds to an address below 16 Meg, but if the address is above 16 Meg, MRDC#/MWTC# will not be generated by the EBC during a DMA compatible cycle. Lastly, the EBC will not generate MRDC# or MWTC#, if only a host slave responds to any address during a DMA compatible cycle. If both an ISA and EISA slave respond to an address, the EBC will give the EISA slave by generating only EISA cycles.

NOTES:
ISAMCMD = MRDC# or MWTC#
1. This is necessary due to DMA devices that use ISA Memory command signals to begin a cycle early. No possible conflict occurs because the address is less than 16 Mbytes and all devices decode all 23 bits.
2. The EBC does not generate ISA Memory command signals for this condition because an ISA memory slave may incorrectly respond since the ISA memory slave decodes only 23 bits.
3. For Type A/B cycles the EBC does not generate ISA memory command signals. This implies that the DMA I/O devices that use memory command signals to begin a cycle early may not use Type A/B cycles.

The DMA always generates ISAIOCMDs for I/O cycles, but in read/write memory cycles ISAMCMDs may not get generated. The decision to generate or not generate ISAMCMDs is necessary in case the memory address is above 16 Mbytes, so that ISA slaves and EISA slaves don't respond at the same time.

HOST CACHE ENABLE (HKEN#) Input

HKEN# is used in i486 systems during host master memory read cycles to the EISA/ISA bus. It indicates that the current address is cacheable and therefore all four bytes are required for the line fill in the i486 internal cache, regardless of the HBE(3:0)# value. This signal is derived from the cacheability map logic and must be activated only for memory read cycles.

2.2 EISA Bus Interface Signals

BYTE ENABLES (BE3#-BE0#) Bidirectional

These signals indicate which bytes on the 32-bit EISA data bus are involved in the current cycle. BE(3:0)# are pipelined from one cycle to the next and must be latched by the EISA slave. During standard cycles, they are valid before BALE goes active and remain valid as long as the LA(31:2) address lines remain valid. BE(3:0)# are outputs when the CPU is addressing an EISA/ISA slave, during EISA master cycles where a mismatched slave responds, or DMA cycles where a mismatched slave responds. Under those conditions, BE(3:0)# are derived combinatorially from HBE(3:0)# after the first falling edge of BCLK during CMD# of the previous cycle, except in i486 systems when HKEN# is active (BE(3:0)# is forced to 0000). They are also driven during ISA master cycles, excluding ISA master refresh cycles, and are derived combinatorially from SA(1:0) and SBHE#. They are inputs during EISA bus master cycles or when the ISP is performing DMA or refresh cycles. BE(3:0)# are sampled by the EBC at the end of START#, and are translated to HBE(3:0)#, SA0, SA1, and SBHE#. The switch from inputs to outputs occurs one BCLK after the end of the first START#, and switches back to being inputs with the negation of the last CMD#. BE3# applies to bits 31-24 of the EISA data bus, BE2# to bits 23-16, BE1# to bits 15-8, and BE0# to bits 7-0.

Table 2-3. BE(3:0)# Signal Translation

Input Signals			Output Signals
SBHE#	SA1	SA0	BE(3:0)#
0	0	0	1100
0	0	1	1101
0	1	0	0011
0	1	1	0111
1	0	0	1110
1	0	1	ILLEGAL
1	1	0	1011
1	1	1	ILLEGAL

MEMORY OR I/O CYCLE (M-IO) Bidirectional

This signal is used to distinguish between a memory cycle (high) and I/O cycle (low) during EISA Bus cycles. It is driven during ISA master cycles and is driven low if IORC# or IOWC# go active. M-IO stays low as long as IORC# or IOWC# are active. M-IO is floated during ISA master refresh cycles. At other times M-IO is high and it is initially set high to indicate memory. M-IO is an input during CPU, DMA, or EISA bus master cycles. This signal is pipelined from one cycle to the next, and must be latched by the slave. See Table 2-1 for M-IO decode information.

WRITE OR READ CYCLE (W-R) Bidirectional

This signal is used to distinguish between EISA read (low) and write (high) cycles. W-R is an input to the EBC during EISA master cycles and is propagated to the host bus as HW/R#. It is an output of the EBC for all cycles except EISA master cycles, and is derived from HW/R#. W-R is generated with START#, but is valid only until a half a BCLK after CMD# becomes active.

START CYCLE (START#) Bidirectional

START# provides timing control at the start of the cycle. START# is an input to the EBC during EISA bus master cycles and is used to start cycles on the host bus. START# is an output of the EBC when the CPU is the master, and is generated in response to the beginning of a cycle on the host bus to which no host slave responded. START# is driven active after the address LA(31:2) and M-IO have become valid. It is also driven by the EBC when the ISP requests DMA or refresh cycles, during ISA bus master cycles to EISA slaves, during ISA master I/O cycles to ISA I/O slaves, and when the EBC translates a 16-bit or 32-bit EISA bus master cycle into cycles for an EISA/ISA slave with a smaller data bus size. Essentially, START# is an output of the EBC for all cycles except EISA master cycles, but even during this time START# becomes an output if a mismatched slave, or an ISA slave responds. This switch from input to output happens at the end of the first START#, and switches back to being an input with the negation of the last CMD#. The trailing edge of START# is always triggered from the rising edge of BCLK after one BCLK cycle time.

COMMAND (CMD#) Output

This signal provides timing control within the cycle. It is driven active from the rising edge of BCLK simultaneously with the negation of START# and remains asserted until the end of the cycle. It is generated by the EBC during any EISA cycle, for example; EISA bus master cycles, translated host cycles, translated ISA cycles, or multiple EISA cycles for data assembly/disassembly. It is also active for ISA I/O cycles (for a possible ISP access). CMD# goes inactive from BCLK's rising edge, except during some types of DMA cycles and ISA bus master cycles. CMD# is driven inactive asynchronously from the trailing edge of the ISA command signal during ISA master read cycles.

MASTER BURST (MSBURST#) Bidirectional

This signal is an input to the EBC for an EISA bus master to indicate that the EISA bus master is capable of supporting the next cycle as a burst cycle. MSBURST# is an output during DMA cycles, if DMA burst mode has been decoded. MSBURST# should be sampled by the memory slaves on the rising edge of BCLK.

SLAVE BURST (SLBURST#) Input

This signal is driven active by EISA memory slaves to indicate that they are capable of accepting burst cycles. SLBURST# is sampled on the rising edge of BCLK when the DMA master requests burst cycles.

EISA 32-BIT DEVICE (EX32#) Bidirectional-Open Collector

This signal is driven active by 32-bit EISA slaves to indicate their 32-bit data bus size. When a 32-bit 386 CPU, i486 CPU, or a 32-bit EISA bus master is addressing a slave on the EISA bus, then a slave driving EX32# active indicates a matched data size, and no multiple EISA or ISA cycles are necessary. If no slave on the EISA bus activates EX32#, and if no host bus slave responds (HLOCMEM# or HLOCIO# inactive), then the EBC will assemble or disassemble the data, running multiple EISA or ISA cycles if necessary, and then activate EX32# to indicate completion of the cycle. If a slave on the host bus responds to a 32-bit EISA master cycle, then the EBC activates EX32#. If host memory does not respond during DMA cycles, the EBC samples EX32# to determine the data bus width of the memory being addressed.

When a 16-bit EISA bus master is addressing a 32-bit slave on the EISA bus, EX32# driven active by the slave indicates a mismatched data size. Then the EBC reroutes the data and drives EX32# active to indicate the end of cycle to the master. EX32# is driven for the duration of half a BCLK from the falling edge of BCLK before the rising edge of the last CMD# signal of the cycle.

During refresh cycles, EX32# must be driven active at the rising edge of START#. T26d and T26e setup and hold times must be met.

EISA 16-BIT DEVICE (EX16#) Bidirectional-Open Collector

This signal is driven active by 16-bit EISA slaves to indicate their 16-bit data bus size. When a 16-bit EISA bus master is addressing a slave not on the host bus, EX16# indicates a matched data size, and no multiple ISA cycles are necessary. If no slave on the EISA bus drives EX16# active, and if no host bus slave responds, the EBC will assemble or disassemble the data, running multiple ISA cycles if necessary, and then drives EX16# active to indicate completion of the cycle. If host memory did not respond during DMA cycles, or during ISA master cycles, the EBC samples EX16# to determine the data bus width of the memory being addressed.

When a CPU, or a 32-bit EISA bus master is addressing a 16-bit slave not on the host bus, EX32# sampled inactive indicates a mismatched data size, and multiple EISA cycles are run before the EBC drives EX32# active to indicate the end of cycle to the master. EX32# is driven for half a BCLK from the falling edge of BCLK before the rising edge of the last CMD# signal of the cycle.

EISA READY (EXRDY) Bidirectional, OC

EXRDY indicates the slave is ready to terminate the cycle. It is sampled on the falling edge of BCLK after CMD# has been driven active, and if inactive, each falling edge thereafter. The CMD# signal remains inactive half a BCLK after EXRDY is sampled active. This results in an added wait state in the given cycle. EXRDY# is sampled when the CPU is addressing EISA or ISA slaves, and is propagated to the host bus as HERDYO#, and HRDYO# when the entire assembly/disassembly sequence is done (if required). It is also sampled during all other EISA/ISA bus master cycles, to determine the end of cycle. When the ISP is performing DMA or refresh cycles, EXRDY is sampled if an EISA slave responds with EX16# or EX32#, and is propagated to the ISP as DRDY. This signal is forced low for a BCLK at the start of a potential DMA burst write. Table 2-6 shows EXRDY and CHRDY translation.

LOCKED CYCLE (LOCK#) Output

This signal is driven active by the EBC when the host master is running locked cycles to EISA slaves (HLOCK# has been driven active). This guarantees exclusive memory access during the time LOCK# is asserted. If HLOCK# is sampled active in the first host clock (CLK1) of START#; then LOCK# is driven active. LOCK# remains active until HLOCK# is sampled inactive in the first host clock (CLK1) of START# of another host master cycle, or during a host master idle cycle following three host clocks after the end of CMD#. If there is an idle cycle in between a locked cycle and an unlocked cycle, the LOCK# signal will remain active as long as HLOCK# is active. Figure 2-3a and 2-3b depict the EBC sampling points of HLOCK# and HLOCK's effect on LOCK#. LOCK# may be driven inactive

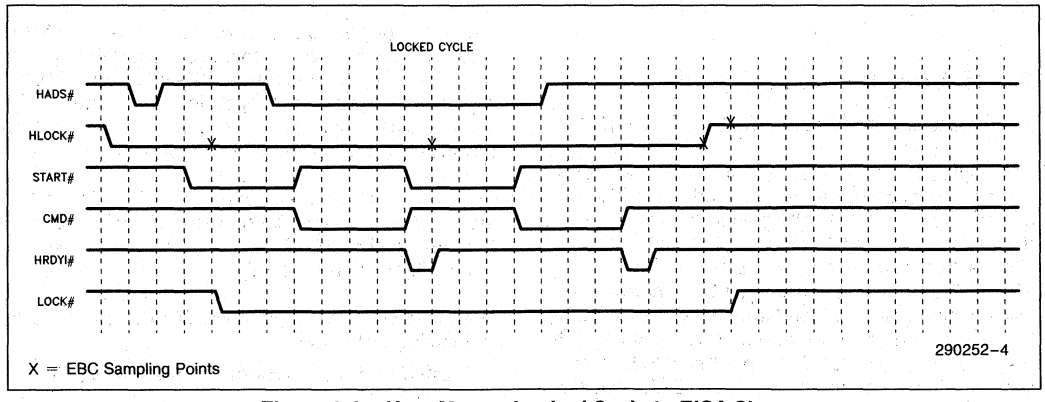


Figure 2-3a. Host Master Locked Cycle to EISA Slave

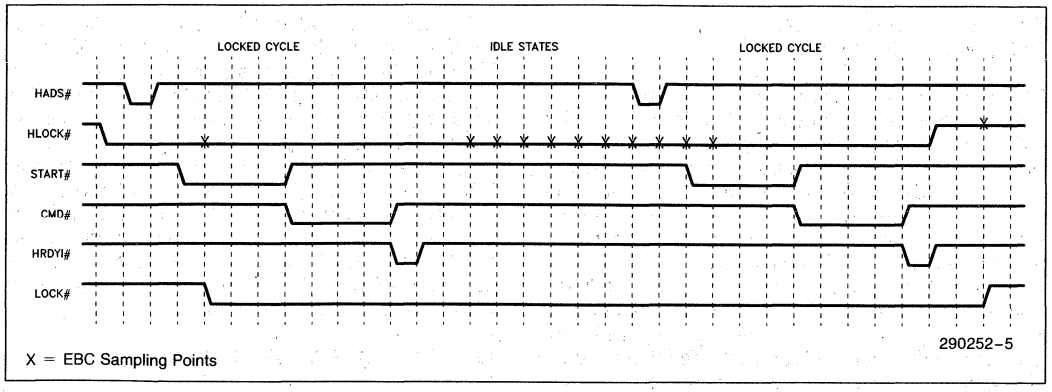


Figure 2-3b. Host Master Locked Cycle to EISA Slave

asynchronous to BCLK. If this occurs in a system with dual ported memory, the EISA bus master may not be able to access the dual ported memory for 1 BCLK after the CPU has driven HLOCK# inactive.

For 386 CPU systems with an 82385 Cache Controller, 2 CLK1 delays must be inserted between LOCK# from the 386 CPU and HLOCK# into the EBC. This is to allow the EBC to continue driving LOCK# on the EISA bus during a write cycle of a LOCK# sequence, since write cycles are posted by the 82385 cache controller.

For 386 CPU systems without the 82385 Cache Controller or for i486 CPU based systems, LOCK# from the CPU is a direct connect to the EBC HLOCK# pin.

2.3 ISA Bus Interface Signals

Bus Address Latch Enable (BALE) Output

This signal is driven active to indicate that a valid address is present on the LA address bus. ISA slaves should use the falling edge of BALE to latch the LA bus. The EBC holds BALE active whenever HHLDA is active, except for EISA bus masters. If EXMASTER# is active while HHLDA is active, BALE will be driven active combinatorially with HHLDA. BALE is driven active for one half BCLK during CPU cycles to the EISA/ISA bus, and during EISA master cycles. In cases of stretching BCLK, BALE could be driven active for as much as a BCLK. Figure 2-5b shows a one BCLK stretch for BALE. For ISA or DMA cycles, BALE is continuously active.

BUS CLOCK (BCLK) Output

The EBC divides the HCLKCPU by a divisor based on how CPU(3:0) are strapped, to generate BCLK. (See Table 2-4.) The maximum frequency is 8.333 MHz with a normal duty cycle of 50%. Figure 2-4a and 2-4b show how the EBC can stretch BCLK multiple ways in 33 MHz and 25 MHz CPU systems.

The high or low time can be stretched to synchronize BCLK to the following three conditions:

a. End of cycle on the EISA bus when HSTRETCH# is sampled active. This results in holding BCLK low until HSTRETCH# has been sampled inactive.

b. When a Host bus master is accessing an EISA or ISA slave, the rising edge of BCLK is synchronized to the falling edge of CMD#.

c. When an ISA master is accessing an EISA slave, the rising edge of BCLK is synchronized to the falling edge of CMD#.

Figure 2-5 (a through c) show the above conditions. Note that ISA master or Host bus master to any non-host slave cycle generates START# and CMD# signals (EISA cycle) except during ISA master to ISA slave cycles. This will result in stretching BCLK as described in the above three conditions.

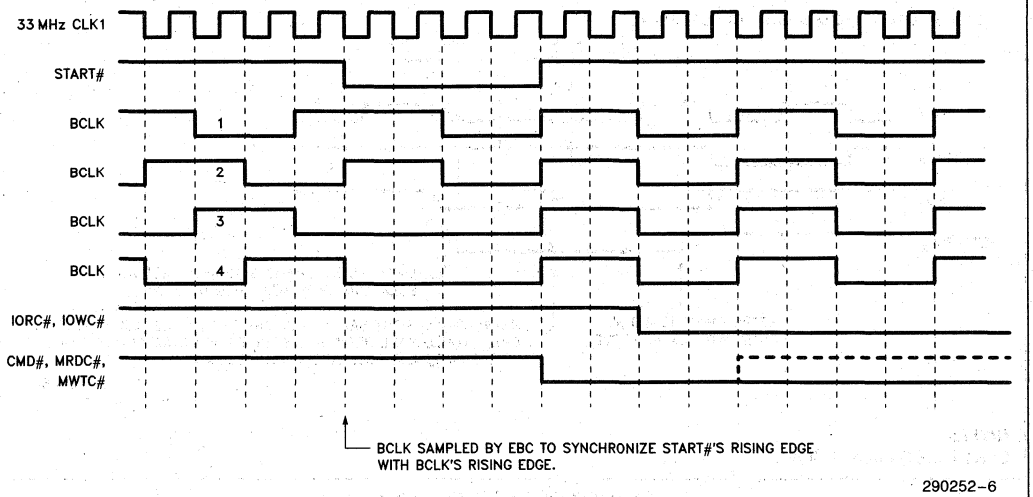
As a result of stretching, BCLK may not have a falling edge during the duration of START#. Events that are synchronized to BCLK edges should be done without regard to frequency or duty cycle.

Note that BCLK needs to drive BCLKIN for the EBC to function correctly.

Table 2-4. BCLK Generation from HCLKCPU

CPU(3:0)	BCLK Freq.	HCLKCPU Divided By
1010 — 25 MHz 386	8.33 MHz	/6
1011 — 33 MHz 386	8.25 MHz	/8
1100 — 25 MHz i486	8.33 MHz	/3
1101 — 33 MHz i486	8.25 MHz	/4

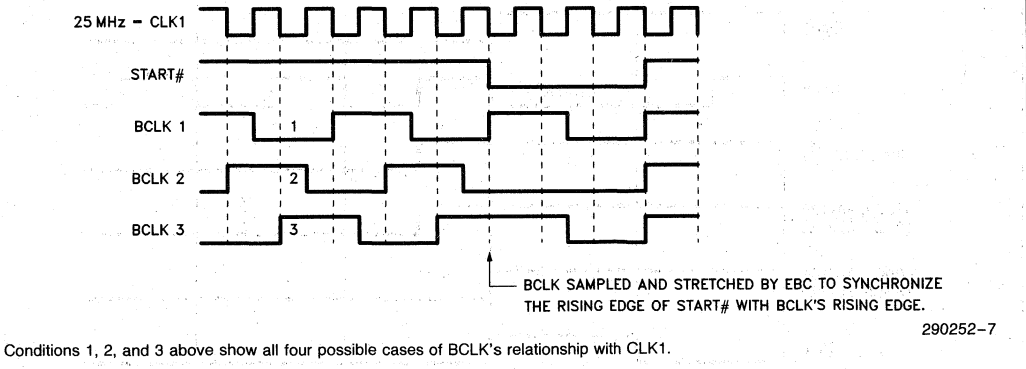
NOTE:
All other bit combinations are reserved.



Conditions 1, 2, 3, and 4 show all four possible cases of BCLK's relationship with CLK1.

NOTE:
CLK1 is EBC Internal CLK1.

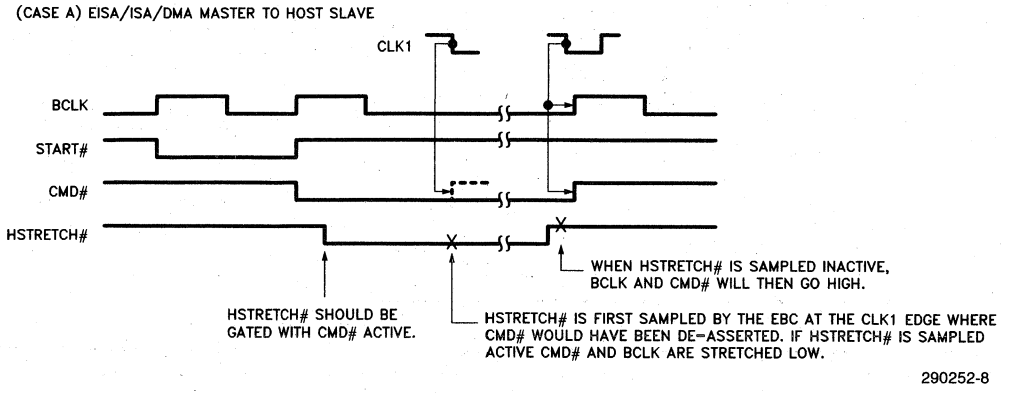
Figure 2-4a. BCLK Stretching for 33 MHz CPUs



Conditions 1, 2, and 3 above show all four possible cases of BCLK's relationship with CLK1.

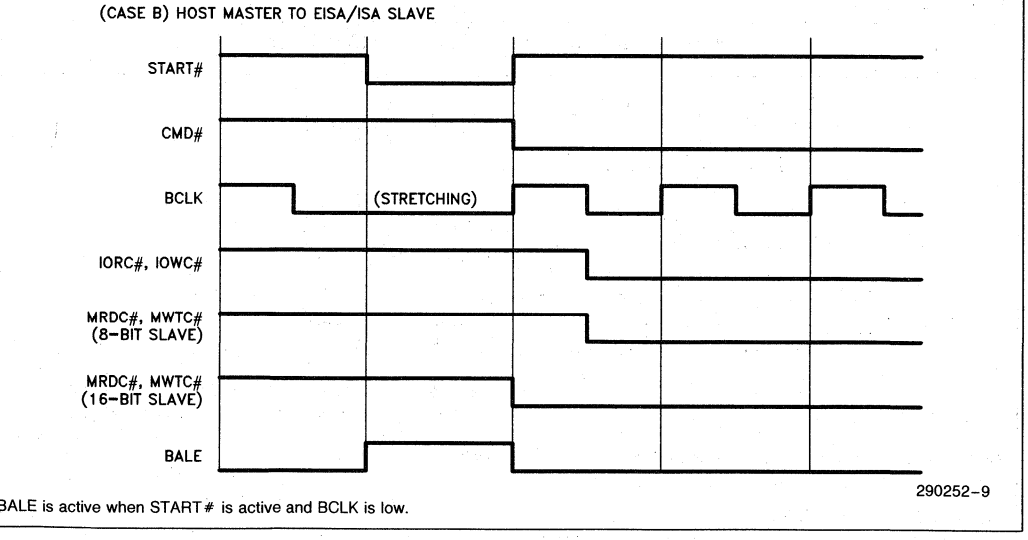
NOTE:
CLK1 is EBC Internal CLK1.

Figure 2-4b. BCLK Stretching for 25 MHz CPUs



NOTE:
CLK1 is EBC Internal CLK1.

Figure 2-5a. BCLK Stretching



BALE is active when START# is active and BCLK is low.

Figure 2-5b. BCLK Stretching

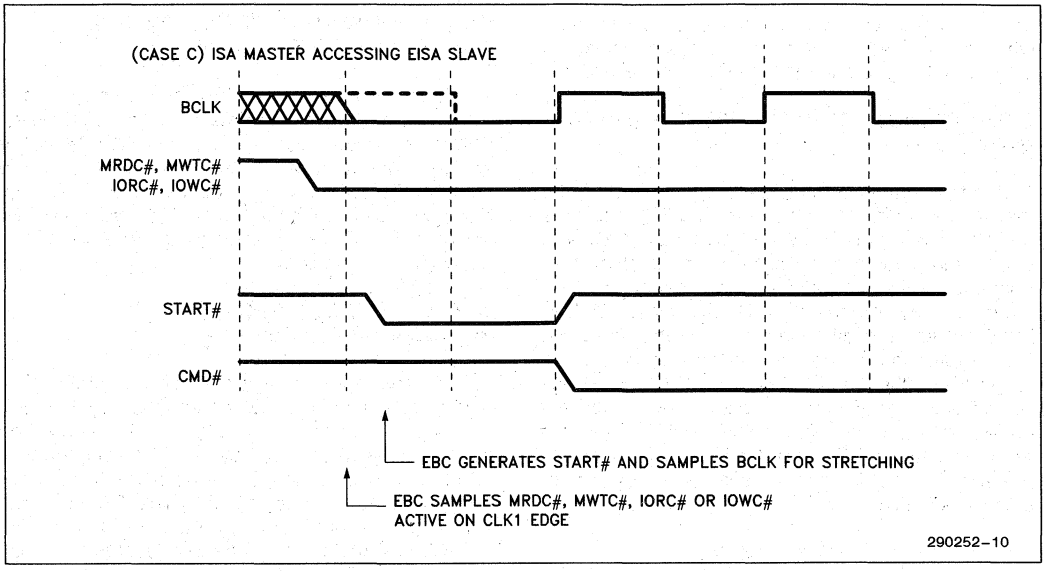


Figure 2-5c. BCLK Stretching

16-BIT MASTER (MASTER16#) Input

This signal indicates a 16-bit EISA or ISA master has control of the EISA bus. MASTER16# is sampled twice: once at the beginning and once at the end of START#. If it was inactive at the first sampling, and active at the second sampling, the EBC treats this as a cycle from a 32-bit EISA master that is downshifting to 16-bits in order to be able to burst.

I/O READ CONTROL STROBE (IORC#)
Bidirectional

This is the command to an ISA I/O slave that it may drive data to the data bus. IORC# is an output signal for all cycles, except when the ISA master owns the bus during a non-refresh cycle. The EBC drives IORC# active when the CPU or an EISA master is addressing an ISA I/O slave, and during DMA memory write cycles. It is an input when an ISA master is in control and is translated to an EISA cycle, whch is always broadcasted (START# and CMD# activated). If no slave on any bus responds, translation defaults to an 8-bit ISA slave.

I/O WRITE CONTROL STROBE (IOWC#)
Bidirectional

This is the command to an ISA I/O slave that it may latch data from the data bus. The EBC drives IOWC# active when the CPU or an EISA master is addressing an ISA I/O slave, and during memory read DMA cycles. It is an input when an ISA master is in control and is translated to EISA control signals.

16-BIT I/O (IO16#)
Bidirectional-Open Collector

16-bit ISA I/O slaves decode SA(15:1) on the ISA bus without regard to IORC# or IOWC#, and drives IO16# active indicating that the 16-bit ISA I/O slaves are capable of performing 16-bit transfers. The EBC drives IO16# during ISA master cycles when a host I/O slave has responded with HLOCIO#. It is the responsibility of 16- and 32-bit EISA I/O slaves to drive IO16# active, when they decode their address (ignoring the M-IO signal). IO16# is an input to the EBC during CPU cycles to the EISA bus and EISA master cycles.

MEMORY READ CONTROL STROBE (MRDC#)
Bidirectional

This is the command to an ISA memory slave that it may drive data to the data bus. MRDC# is an output when the CPU, an EISA master, or DMA is performing a memory read cycle to which neither host bus memory nor EISA memory responded. During memory read compatible DMA cycles, MRDC# is driven active in all cases, except when a 16- or 32-bit EISA memory, or host memory responds and the address is above the range of 00000000h to 00FFFFFFh (HGT16M# has been driven active). MRDC# is also

an output for ISA master refresh cycles. It is an input when an ISA master is in control and is translated to EISA control signals if EX32# or EX16# is active.

MEMORY WRITE CONTROL STROBE (MWTC#) Bidirectional

This is the command to an ISA memory slave that it may latch data from the data bus. The EBC activates MWTC# when the CPU, an EISA master, or DMA is performing a memory write cycle to which neither host bus memory nor EISA memory responded. During memory write compatible DMA cycles, MWTC# is driven active in all cases, except when an EISA memory responds and the address is above the range of 00000000h to 00FFFFFFh (HGT16M# has been driven active). MWTC# is an output for all cases, except when the ISA master owns the bus. It is an input when an ISA master is in control and is translated to EISA control signals if EX32# or EX16# is active.

16-BIT MEMORY (M16#) Input

This signal indicates that the addressed ISA memory is capable of performing 16-bit transfers. M16# is decoded from LA (23:17). M16# is sampled by the EBC during CPU cycles to EISA/ISA bus, EISA master cycles and ISA master cycles to host memory slaves. Table 3-2 EBC Cycle Translation shows M16# decode for slave size determination.

STANDARD MEMORY READ CONTROL STROBE (SMRDC#) Output

This signal is the command to an ISA memory slave that it may drive data to the data bus. The EBC activates SMRDC# during CPU, DMA, or EISA/ISA master read cycles to a 16- or 8-bit ISA memory slave, and the address is within the range 00000000h to 000FFFFFh (GT1M# is inactive), or during refresh cycles. SMRDC# behaves with similar timing to MRDC#. When GTIM# is active, SMRDC# is floated, therefore SMRDC# needs a pullup resistor.

STANDARD MEMORY WRITE CONTROL STROBE (SMWTC#) Output

This is the command to an ISA memory slave that it may latch data from the data bus. The EBC drives SMWTC# active whenever it drives MWTC# active and the address is within the range 00000000h to 000FFFFFh (GT1M# is inactive). SMWTC# behaves with similar timing to MWTC#. When GTIM# is active, SMWTC# is floated, therefore SMWTC# needs a pullup resistor.

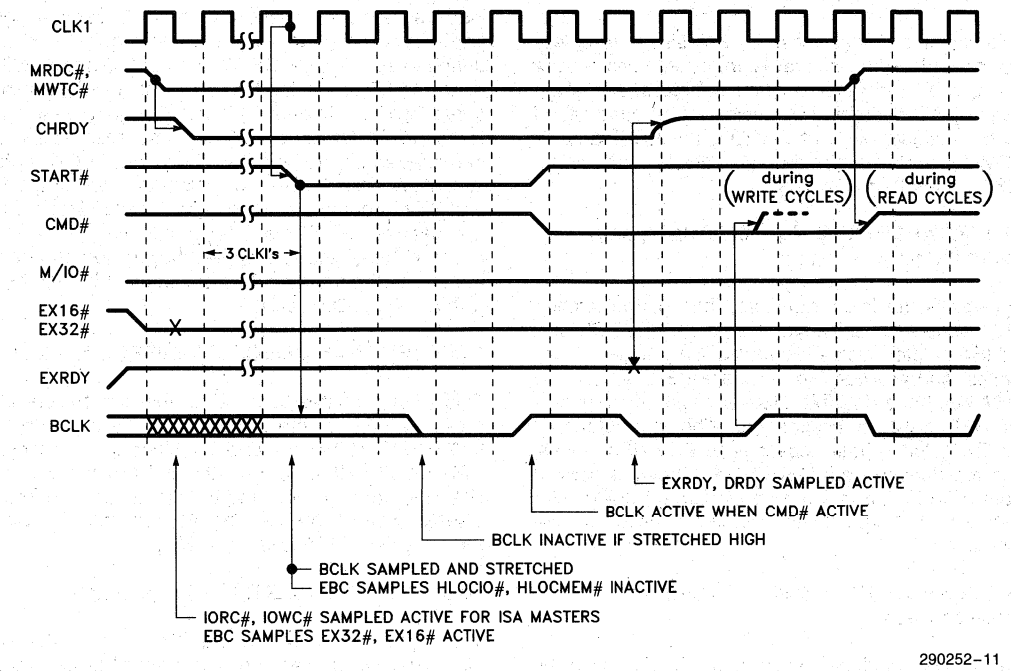
CHANNEL READY (CHRDY) Bidirectional-Open Collector

This is used by ISA slaves to insert wait states. The falling edge is captured asynchronously by the EBC, and is gated on the falling edge of BCLK to determine whether to add wait states. The rising edge is synchronized, and the EBC stops adding wait states on the next rising edge of BCLK. The ISA command signals will remain active at least one BCLK after CHRDY is sampled high. The EBC samples CHRDY during CPU or EISA bus master cycles to ISA slaves, and during DMA accesses to ISA memory. CHRDY takes precedence over NOWS#, i.e. if CHRDY is inactive, NOWS# driven active is ignored. CHRDY is an output during ISA master I/O cycles until EXRDY and DRDY are sampled high when CMD# is active on BCLK's falling edge. CHRDY is also an output during ISA master memory cycles not to ISA memory slaves, until EXRDY and DRDY are sampled high when CMD# is active on BCLK's falling edge. CHRDY is driven inactive asynchronously from ISA command signals for any ISA master to EISA cycle translation including ISA master accessing ISA I/O slave, and is driven active synchronously during CMD# after all other ready signals (EXRDY, DRDY) are sampled high. Table 2-5 below shows CHRDY and EXRDY functional description, and Figure 2-6 shows CHRDY functionality for an ISA master to EISA memory slave cycle.

Table 2-5. CHRDY and EXRDY Functional Description

Master	EXRDY	CHRDY	Conditions
CPU/EISA	Input	Input	(Note 1)
DMA	I/O	Input	(Note 1) EBC drives EXRDY inactive for DMA memory write or I/O read burst cycles.
ISA	Input	Output	EBC drives CHRDY inactive for any I/O slave or any EISA memory slave. CHRDY is a function of EXRDY, and DRDY. (1)

NOTE:
1. EBC samples CHRDY for ISA slaves and samples EXRDY for EISA slaves.



x = EBC Sampling Point

NOTE:
CLK1 is EBC Internal CLK1.

Figure 2-6. CHRDY and EXRDY in ISA Master to EISA Memory Slave Cycle

NO WAIT STATES (NOWS#) Input

This signal is driven active by ISA slaves to delete standard (default) wait states. In order to shorten the cycle, NOWS# must be active before the falling edge of BCLK during ISA cycles. If not superseded by the inactive edge of CHRDY, the EBC samples NOWS# to delete standard wait states from host or EISA cycles to ISA slaves.

SYSTEM ADDRESS BITS 1 and 0 (SA1, SA0) Bidirectional

These are the least significant bits of the latched EISA address bus. They are inputs to the EBC during ISA master cycles, excluding ISA master refresh cycles, and are used to generate BE(3:0)# on the EISA bus (for a possible EISA access) and to generate HBE(3:0)# on the host bus. They are outputs of the EBC during host accesses to EISA/ISA slaves, and are derived from HBE(3:0)#. They are derived from BE(3:0)#, and are outputs during EISA master cycles to ISA slaves, and DMA accesses to ISA memory.

SYSTEM BYTE HIGH ENABLE (SBHE#) Bidirectional

This active low signal indicates the high byte on the ISA data bus (SD(15:8)) is valid. It is an input when an ISA master is in control, excluding ISA master refresh cycles, and is used to generate BE(3:0) on the EISA bus (for a possible EISA access) and to generate HBE(3:0)# on the host bus. SBHE# is an output during host accesses to EISA/ISA slaves and is derived from HBE(3:0)#. SBHE# is also an output during EISA master cycles to ISA slaves, and DMA accesses to ISA memory in which case SBHE# is derived from BE(3:0)#.

REFRESH (REFRESH#) Input

This input indicates the ISP is performing a refresh cycle. During refresh, the EBC generates MRDC# and CMD# signals to refresh the entire system memory.

2.4 ISP Device Interface Signals

ISP HOLD REQUEST (DHOLD) Input

The ISP drives this signal active to request the host bus on behalf of ISA/EISA masters, or when a DMA device requests service. DHOLD is used to generate HHOLD.

ISP READY (DRDY) Bidirectional

DRDY is an indication of the end of a cycle. It is an input to the EBC when the ISP is in the slave mode. During ISA bus master I/O cycles, CHRDY is driven inactive asynchronously, and is kept inactive until DRDY is sampled active at the falling edge of BCLK during CMD# active. DRDY is an output of the EBC during DMA and refresh cycles and has two functions: Two BCLKs after ST3 and ST2 go to their non-idle state, DRDY is active (high) if address pipelining can happen on the host bus. After this, DRDY is the indication of the end of the current transfer, until ST3 and ST2 become inactive again. After ST3 and ST2 are inactive, DRDY may still be inactive for the last transfer. Pipelining is stopped at page breaks or at the end of the transfer. This signal must be tied high with a 2.4K pullup resistor.

GREATER THAN 1 MEGABYTE (GT1M#) Input

This input indicates to the EBC that the current address is above the range from 00000000h to 000FFFFFh (1 Mbyte), and if it is not active during a CPU or EISA/ISA bus master or DMA cycle to an ISA memory slave, then the EBC generates SMRDC# or SMWTC#. The ISP generates GT1M# for all cycles; DMA and non-DMA.

DMA STATUS (ST3–ST0) Bidirectional

These pins are inputs during DMA and refresh cycles. They indicate what type of timing has been programmed for the current cycle, and the size of the I/O device involved in the DMA transfer. the EBC will generate the appropriate command signals for the ISP and will control the timing of the cycle. If the memory which responded does not match the size of the I/O device, and compatible timing has not been selected, the EBC will have to perform assembly or disassembly of data and use EISA/ISA timing. (See Table 2-6.)

During refresh cycles, ST3 and ST2 follow the 8-bit cycle definition (low), and ST1 and ST0 are undefined.

During cycles where the ISP is not the master, these status pins are outputs.

ST0 is the DMA address strobe function for the ISP. The ISP uses ST0 to latch the address, M-IO, and HW/R# information. (1 = ISP internal latch closed; 0 = open.) For CPU cycles, ST0 will be active low for one host CLK1 in the beginning of START#; for EISA cycles, ST0 will be active during START#; and for ISA cycles, it will be kept low continuously.

ST1 is the memory or I/O function (1 = Memory cycle; 0 = I/O cycle). It is derived from HM/IO# during CPU cycles to the EISA bus, and from M-IO during EISA bus master cycles.

Table 2-6. ST(3:0) Decode

ST(3:0)	I/O Device Type & Programmed Timing	Note
0000	8-Bit Device and Compatible Timing	A
0001	8-Bit Device and Type A Timing	B
0010	8-Bit Device and Type B Timing	B
0011	8-Bit Device and Burst Timing	C
0100	16-Bit Device and Compatible Timing	A
0101	16-Bit Device and Type A Timing	B
0110	16-Bit Device and Type B Timing	B
0111	16-Bit Device and Burst Timing	C
1000	32-Bit Device and Compatible Timing	A
1001	32-Bit Device and Type A Timing	B
1010	32-Bit Device and Type B Timing	B
1011	32-Bit Device and Burst Timing	C
11XX	No Operation	

NOTES:
A. The EBC does not assemble or disassemble.
B. Type A/B.
If the DMA is performing Type A/B transfers and the EBC needs to translate to ISA memory cycles (ISA memory slave), or assembly/disassembly is required, then the cycle reverts to memory timings similar to that used with EISA bus masters. The MRDC# and MWTC# signals are not active unless the system must do assembly/disassembly for ISA memory.
C. Burst cycles.
If the DMA is performing a burst transfer and the memory does not support bursting (SLBURST# inactive), the cycle reverts to memory timings similar to the standard memory cycle generated by EISA bus masters.

ST2 is the INTA# function, which is active (low) when the EBC decodes an interrupt acknowledge cycle. An interrupt acknowledge cycle will result in a START-CMD sequence. This signal must be tied high with a 1.2K pullup resistor.

ST3 is the cycle in progress (CIP#) function, indicating an EISA bus master cycle is in progress, and is used by the arbitration logic in the ISP. This active low signal is driven inactive at the end of the last assembly/disassembly cycle.

EISA MASTER (EXMASTER#) Input

This input signal indicates a 16- or 32-bit EISA master has control of the EISA bus. It is used by the EBC in conjunction with MASTER16#, to distinguish between 32-bit EISA masters, 16-bit EISA masters, and 16-bit ISA masters.

EARLY INDICATION OF 16-BIT ISA MASTER (EMSTR16#) Input

This signal from the ISP indicates a 16-bit ISA master is in control, or about to assume control. It gives the EBC enough time to be able to switch the direction of the control signals and the address buffers between the host bus and the EISA bus, when changing from host initiated cycles to ISA master initiated cycles. Since EMSTR16# resets the direction of the control signals, it must be driven active before START# is driven active in the first cycle. This can be accomplished by logically ANDing EMSTR16# from the ISP with SPWROK from the EBC to generate the EMSTR16# signal to the EBC. During refresh cycles, EMSTR16# must be pulsed active.

2.5 Data Buffer Control Signals

COPY ENABLE BETWEEN BYTES (SDCPYEN01#–03#, 13#) Output

This signal enables the byte copy transceiver between the EISA bus data bytes 0, 1, 2 and 3. SDCPYEN01# enables copying between data bits 0–7 and data bits 8–15. SDCPYEN02# enables copying between data bits 0–7 and data bits 16–23. SDCPYEN03# enables copying between data bits 0–7 and data bits 24–31. SDCPYEN13# enables copying between data bits 8–15 and data bits 24–31. The EBC does not support ISA bus master copy operations. These operations must be externally implemented.

COPY UP (SDCPYUP) Output

LOW BYTE TO HIGH BYTE — This signal controls the direction of the byte copy transceivers. When active, the lower bytes are copied on to the higher bytes. The direction is reversed when this signal is inactive. Copying of data bytes is done when the data bus size of the master and slave do not match. The EBC does not support ISA bus master copy operations. These operations must be externally implemented.

SYSTEM (EISA) DATA TO HOST DATA LATCH ENABLES (SDHDLE3-0#) Output

These signals latch the data from the EISA bus into byte latches which can enable data onto the host data bus. SDHDLEO# controls the latching of EISA bus data bits 7-0 to host bus data bits 7-0, SDHDLE1# controls the latching of EISA bus data bits 15-8 to host bus data bits 15-8, SDHDLE2# controls the latching of EISA bus data bits 23-16 to host bus data bits 23-16, and SDHDLE3# controls the latching of EISA bus data bits 31-24 to host bus data bits 31-24.

SYSTEM (EISA) DATA OUTPUT ENABLE (SD0E2-0#) Output

These signals enable the output of the EISA bus data buffers. SDOE0# applies to EISA bus data bits 7-0, SDOE1# to bits 15-8, SDOE2# to bits 31-16. These are enabled during CPU write cycles, EISA/ISA master read cycles from host memory, during DMA read cycles from the Host bus slave, and during byte assembly or disassembly for EISA masters and DMA cycles addressing EISA/ISA slaves with mismatched data bus size.

HOST DATA TO SYSTEM (EISA) DATA LATCH ENABLES (HDSDLE1#) Output

This signal latches the data from the host data bus into byte latches which can enable data onto the EISA data bus. HDSDLE1# controls the latching of host data bus bits 31-0.

HOST DATA OUTPUT ENABLE (HD0E1-0#) Output

These signals enable the output of the host data bus buffers. HDOE0# applies to host bus data bits 15-0 and HDOE1# applies to bits 31-16. These buffers are enabled during CPU read cycles from EISA/ISA slaves, EISA/ISA master, DMA write cycles, during assembly or disassembly for DMA cycles, or for EISA masters addressing EISA/ISA slaves with a mismatched data bus size. These signals are driven active with the falling edge of CMD#.

Figure 2-7. Host/EISA Bus Data Swap Logic shows how the signals in this section are connected.

2.6 Address Buffer Control Signals

HOST ADDRESS BUS TO EISA LA BUS OUTPUT ENABLE (HALAOE#) Output

HALAOE# is active during CPU, DMA and refresh cycles. This signal enables the output of the address buffers from the host address bus bits (31:2) to the EISA LA bus bits (31:2).

HOST ADDRESS LATCH ENABLE (HALE#) Output

This signal enables the latching of the LA address bus on to the host address bus. For EISA and ISA bus masters, HALE# is always held active low to allow the LA bus to propagate to the host address bus.

EISA LA TO EISA SA OUTPUT ENABLE (LASAOE#) Output

This signal enables the output of the address buffers from the EISA LA bus bits (19-2) to the EISA SA bus bits (19-2). It is active during CPU, EISA bus master, DMA and refresh cycles.

EISA LA TO HOST ADDRESS OUTPUT ENABLE (LAHAOE#) Output

This signal enables the output of the address buffers from the EISA LA bus to the host address bus. It is active during EISA/ISA bus master cycles.

LA LATCH ENABLE (LALE#) Output

This signal controls the latching of the host address bus on to the LA address bus. At its trailing edge, the latch closes and the LA addresses are held. This signal is used when CPU address pipelining is required on the host bus. To reset LALE# to a known state in the first cycle, EMSTR16# from the ISP should be ANDed with SPWROK to generate EMSTR16# to the EBC.

EISA SA TO EISA LA OUTPUT ENABLE (SALAOE#) Output

This signal enables the output of the address buffers from the EISA SA bus bits (16-2) to the EISA LA bus bits (16-2). It is active during ISA bus master cycles.

SA LATCH ENABLE (SALE#) Output

This signal controls the latching of the LA address bus (bits 19:2) to the SA address bus. At its trailing edge the latch closes, and the SA addresses are held. For EISA master cycles and CPU cycles, SALE# is active while BALE is active. For DMA cycles that did not succeed as burst cycles, SALE# is driven active one half BCLK before CMD# is driven active and inactive when CMD# is inactive. During regular DMA cycles and DMA burst cycles, SALE# is active on the next rising edge of BCLK on which the ST(3:0) are sampled, and inactive at the trailing edge of START#. Figure 2-8 shows how the signals in this section are connected.

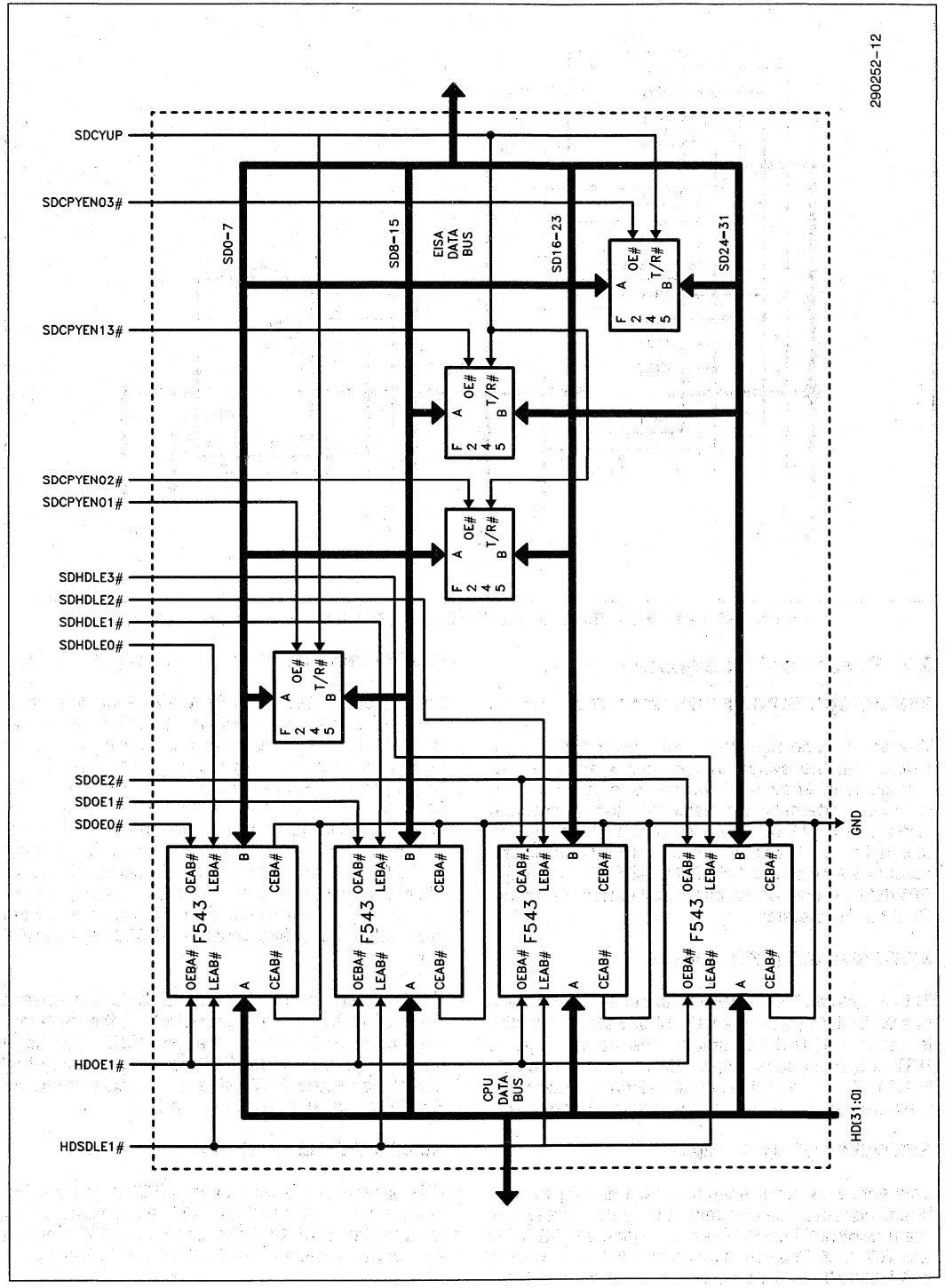


Figure 2-7. Host/EISA Bus Data Swap Logic (EBB 32-Bit Data Mode w/o Parity Equivalent)

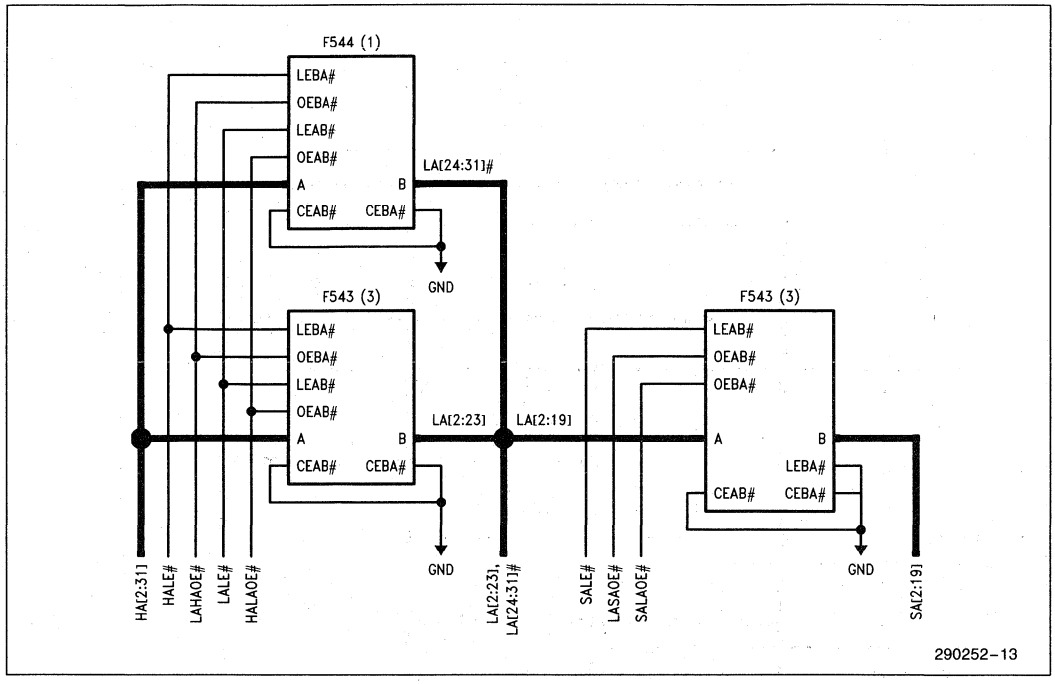


Figure 2-8. Host/EISA Bus Address Buffers (EBB Address Mode Equivalent)

2.7 Reset Control Signals

SYNCHRONOUS POWER OK (SPWROK) Input

The power good signal from the power supply (indicating that the system board has been at proper voltage long enough for proper operation) is synchronized externally to a master 1x clock phase, and is input into the EBC. The EBC uses its leading edge to synchronize the phase of its internal 1x clock, and outputs synchronous RSTCPU, RST385, and RST. SPWROK should be synchronized externally for 386 CPU based systems.

SYSTEM RESET (RST) Output

This is active when SPWROK is active, and is inactive 90 BCLKs (approximately 10 μs) after SPWROK is active and BCLK is stable. The trailing edge of RST is synchronous to the EBC's internal 1x clock. RST, RSTCPU and RST385 are all driven inactive at the same time, after being driven active at power-up.

RESTART (RSTAR#) Input

This active low input is the reset signal from the keyboard controller used to reset the system under program control. The EBC uses it to generate RSTCPU and RST385. The minimum pulse width of RSTAR# is 4 HCLKCPUs. This signal is latched by the EBC.

RESET FOR CPU (RSTCPU) Output

This signal is active when SPWROK is inactive, shutdown is decoded, or when RSTAR# is active. RSTCPU is synchronous to the EBC's internal 1x clock, and remains active as long as RST is active or RSTAR# is active with a minimum pulse width of 34 CLK1 or 68 HCLKCPU cycles. RSTCPU is interlocked with HHOLD so that only one of them is active at a time. If a reset condition other than SPWROK occurs while HHOLD is active, the reset will not be activated until HHOLD is inactive. External retiming of this signal is necessary to meet the setup and hold requirements of the 386 CPU RESET input.

The 8x42 requires clocks (CLKKB) before the end of RSTKBD# (keyboard reset pulse) for the keyboard reset to occur. To ensure that RSTKBD# has been driven inactive after the EBC has begun generating CLKKB, generate RSTKBD# by logically NANDing RSTCPU and RSTDRV from the ISP.

RESET FOR THE 82385 (RST385) Output

This signal is active whenever RSTCPU is active, but delayed by 16 HCLKCPUs. RST385 remains active for a minimum of 18 CLK1 or 36 HCLKCPU cycles and goes inactive with RSTCPU. RST385 active

edge is delayed from the RSTCPU active edge by 16 CLK1 or 32 HCLKCPU cycles only if RSTAR# is activated, otherwise RST385 and RSTCPU are activated concurrently. This is to ensure the CPU is not in the middle of a cycle to the 82385 when the 82385 is reset. External retiming of this signal is necessary to meet the setup and hold requirements of the 82385 RESET input. An E-PAL or PLD device must be used.

2.8 Configuration Signals

CPU TYPE AND FREQUENCY INDICATION CPU(3-0) Input

These pins indicate the type of CPU and its frequency on the host bus. CPU3 indicates the CPU data bus width, CPU2 indicates whether HCLKCPU is a 1x or 2x clock and CPU(1:0), along with CPU2 indicate the frequency of HCLKCPU.

Table 2-7. CPU Type/Frequency Indication

CPU(3:0)
1010 — 32-bits, 2x, 25 MHz (25 MHz 386)
1011 — 32-bits, 2x, 33 MHz (33 MHz 386)
1100 — 32-bits, 1x, 25 MHz (25 MHz i486)
1101 — 32-bits, 1x, 33 MHz (33 MHz i486)

NOTE:
All other bit combinations are reserved.

Bit Definition for CPU(3:0):

Bit 1	Bit 0	
0	0	25 MHz i486
0	1	33 MHz i486
1	0	25 MHz 386
1	1	33 MHz 386

Bit 2 — 1x or 2x mode

Bit 3 — Reserved (Must be 1)

READY DELAY ENABLE (RDE#) Input

This signal is strapped low in systems which require an extra CPU CLK1 delay before HERDYO# and HRDYO# are generated in response to a CPU to EISA bus read cycle. This allows more time for the data to propagate to the host bus. RDE# is pulled up high if the delay is not required.

2.9 Miscellaneous Signals

KEYBOARD CONTROLLER CLOCK (CLKKB) Output

This signal is for use by the keyboard processor and has a nominal duty cycle of 50%. It is generated by dividing HCLKCPU appropriately. Its frequency depends on the CPU(3:0) pins as shown in Table 2-8.

The 8x42 keyboard controller requires clocks (CLKKB) before the end of RSTKBD# (keyboard reset pulse) for the keyboard reset to occur. To ensure that RSTKBD# has been driven inactive after the EBC has begun generating CLKKB, generate RSTKBD# for the 8x42 by logically NANDing RSTCPU and RSTDRV from the ISP.

Table 2-8. CLKKB Generation

CPU(3:0)	CLKKB Freq	HCLKCPU Divided By
1010 — 25 MHz 386	10.00 MHz	/5
1011 — 33 MHz 386	11.00 MHz	/6
1100 — 25 MHz i486	8.33 MHz	/3
1101 — 33 MHz i486	11.00 MHz	/3

NOTE:
All other bit combinations are reserved.

LONG WAIT BETWEEN I/O CYCLES (LIOWAIT#) Input

This signal controls the delay between back to back 8- and 16-bit ISA I/O slave cycles from the CPU. LIOWAIT# is sampled on the rising edge of the I/O command strobe. If LIOWAIT# is sampled active, the START# signal of the current cycle is delayed until LIOWAIT# is sampled inactive. Figure 2-9 (a-b) LIOWAIT# CONTROL shows several cases where LIOWAIT# controls the delay between cycles. Once LIOWAIT# is sampled inactive the next cycle can begin. This guarantees a minimum of 1 added BCLK for I/O recovery time when LIOWAIT# is sampled inactive. The maximum delay is defined by LIOWAIT# being sampled active; three BCLKs for 16-bit I/O slaves, or eleven BCLKs for 8-bit I/O slaves. Delays in between the minimum and maximum are controlled by LIOWAIT# going inactive. No delays are inserted for EISA bus master cycles, CPU cycles to EISA slaves, or I/O assembly/disassembly cycles. EISA master cycles to ISA I/O slaves need I/O recovery time of 1 BCLK provided by software.

AEN LATCH ENABLE (AENLE#) Output

This signal is used on the system board for slot specific AEN control. For a CPU master to an EISA memory/I/O slave or ISA memory slave cycle, AENLE# is driven inactive from the same falling edge of BCLK that drives BALE active while START# is active, and AENLE# is activated one half BCLK after CMD# goes active. AENLE# behaves the same for a CPU master to an ISA I/O slave cycle, except AENLE# is driven active one half BCLK after CMD# goes inactive. For EISA, ISA, or DMA masters, AENLE# is driven active on the rising edge of HHLDA and is inactive on the falling edge of BCLK after START# is active. For ISA slave assembly or disassembly cycles where CMD# is im-

mediately followed by another START# (back to back cycles), AENLE# will remain inactive. Figure 2-10 shows the AENLE# signal functionality.

BUS CLOCK INPUT (BCLKIN) Input

This pin is the source of BCLK from which all BCLK related timings are triggered. BCLKIN is always connected to BCLK with minimum trace length.

TEST1# Input

This active low signal tri-states all outputs of the EBC, except BCLK, SMRDC#, and SMWTC#. Under normal system operation, this input should be pulled high.

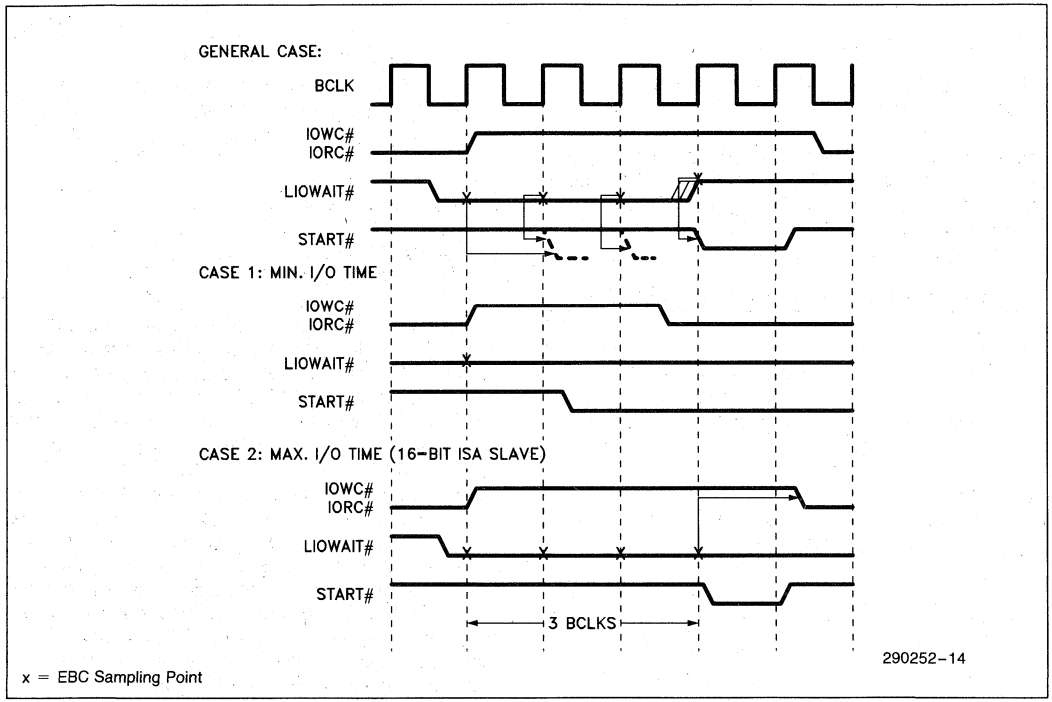


Figure 2-9a. LIOWAIT# Control

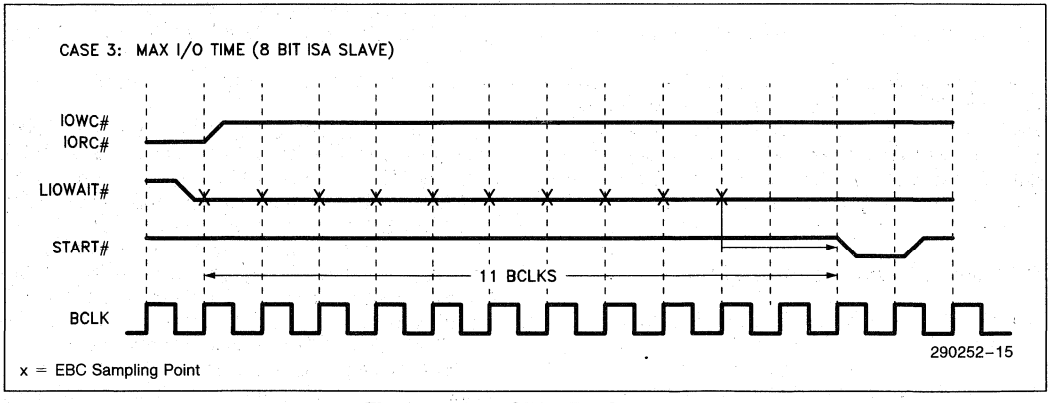


Figure 2-9b. LIOWAIT# Control

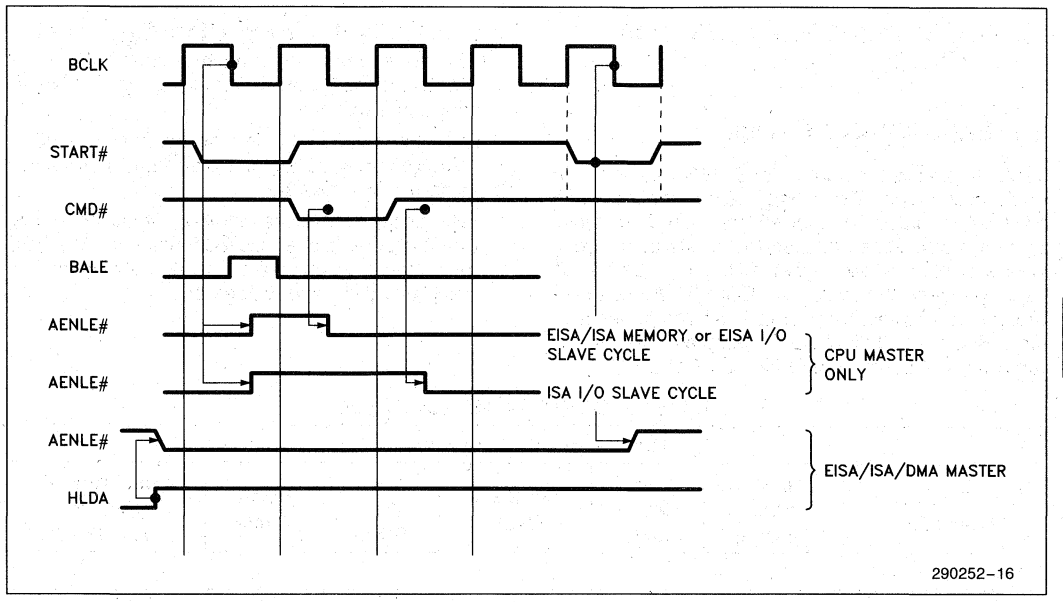


Figure 2-10. AENLE# Timing

3.0 BASIC FUNCTIONALITY AND TIMINGS

3.1 EISA/ISA Bus Cycles

Table 3-1 shows the number of BCLKs that occur for each cycle type. Zero wait states applies when NOWS# is used to shorten the standard cycle. Standard wait states applies to the standard cycle. One wait state applies when CHRDY or EXRDY is used to extend the cycle by 1 BCLK(2). An ISA 8-bit memory read cycle, for example, will run cycles lasting 6 BCLKs; or if NOWS# is sampled active, it will run a no wait state cycle lasting 3 BCLKs.

3.2 EBC Cycle Translation Algorithm

1. EBC determines who is master (Host/EISA/ISA/DMA), depending on which signals are active as shown in Table 3-2.
2. EBC determines who is slave/size (Host/EISA/ISA/DMA), depending on which signals are active as shown in Table 3-2.
3. EBC assembles/disassembles cycles according to the master/slave size combination. The EBC determines which cycle to run depending on which bus the slave resides on.

Table 3-1. EISA/ISA/DMA Bus Cycles

Slave Operation	Bus Size (in Bits)	Number of BCLKs per Cycle: Zero Wait State	Standard Cycle	One Wait State	Burst Cycle	Max. Transfer Rate at 8.33 MHz
ISA Cycles						
Memory Read	16	2	3	4	NA	8.33 Mb/sec
Memory Read	8	3	6	7	NA	2.78 Mb/sec
Memory Write	16	2	3	4	NA	8.33 Mb/sec
Memory Write	8	3	6	7	NA	2.78 Mb/sec
I/O Read	16	3	3	4	NA	5.56 Mb/sec
I/O Read	8	3	6	7	NA	2.78 Mb/sec
I/O Write	16	3	3	4	NA	5.56 Mb/sec
I/O Write	8	3	6	7	NA	2.78 Mb/sec
INTA Read	NA	NA	6	NA	NA	NA
Halt/Shutdown	NA	NA	3	NA	NA	NA
EISA Cycles						
Memory Read	32/16	NA	2	3	1	33.33 Mb/sec
Memory Write	32/16	NA	2	3	1	33.33 Mb/sec
I/O Read	32/16	NA	2	3	NA	16.66 Mb/sec
I/O Write	32/16	NA	2	3	NA	16.66 Mb/sec
DMA Cycles						
Compatible	All	8	8	10	NA	4.17 Mb/sec
Type A	All	6	6	7	NA	5.56 Mb/sec
Type B	All	4	4	5	NA	8.33 Mb/sec
Type C (Burst)	32/16	3	3	4	1	33.33 Mb/sec

The cycles above are measured from the BCLK edge when LA address is valid to the BCLK edge when CMD# goes high (inactive) for ISA/EISA cycles.

NOTES:
1. DMA I/O devices can not add wait states.
2. CHRDY or EXRDY can be used to add a wait state during DMA compatible cycles. When CHRDY is sampled active a wait state of two BCLK duration is added during these cycles. When EXRDY is sampled active, a wait state of one BCLK duration is added.

Table 3-2. EBC Cycle Translation

EBC DETERMINES MASTER

HHLDA	REFRESH#	EXMASTER#	MASTER16#	EMSTR16#	SIZE	MASTER
0	1	1	1	1	32	CPU
1	0	1	1	1	All	REFRESH
1	1	0	1	1	32	EISA*
1	1	0	0	1	16	EISA
1	1	1	0	0	16	ISA
1	1	1	1	1	All	DMA

NOTE:
*For downshifting from 32- to 16-bit EISA burst bus master, the bus master must drive MASTER16# active during START# for each cycle that it generates.

EBC DETERMINES SLAVE

HLOCMEM#	HLOCIO#	EX32#	EX16#	M16#	IO16#	SIZE	SLAVE
0	0	X*	X	X	X*	32	HOST
1	1	0	X	X	X	32	EISA
1	1	1	0	X	X	16	EISA
1	1	1	1	0	1	16	ISA MEMORY
1	1	1	1	1	0	16	ISA I/O
1	1	1	1	1	1	8	ISA

NOTE:
*During EISA master accesses to Host slaves, the EBC drives EX32#. For ISA master accesses to Host slaves, the EBC drives IO16# from HLOCIO#.

TYPES OF CYCLES EBC CAN RUN ON ISA/EISA BUS

DMA	ISA & EISA	EISA Only
Type A	Default (Standard) Memory	Burst
Type B	Default (Standard) I/O	
Type C (Burst)	NOWS# Memory	
Compatible	NOWS# I/O	

EBC DETERMINES WHICH CYCLE TO TRANSLATE

Master Bus	Slave Bus	EBC Bus Translation (Broadcast)*
Host	EISA	No ISA Cycle Broadcasted
Host	ISA	ISA/EISA Cycle Broadcasted (START#, CMD# and ISACMD Signals Generated)
Host	Host	No EISA/ISA Cycle Broadcasted
EISA	Host	No ISA Cycle Broadcasted
EISA	ISA	ISA Cycle Broadcasted
EISA	EISA	No ISA Cycle Broadcasted
ISA	Host	EISA Cycle Broadcasted if Host Slave Drives EX32#
ISA	EISA	EISA Cycle Broadcasted
ISA	ISA	EISA Cycle Broadcasted, except ISA Master to ISA Memory Slave
DMA	All	Both EISA, ISA, and DMA Cycles Broadcasted (See Table 2-2)

NOTES:
*The EBC broadcasts (translates) one bus cycle to another bus cycle if necessary. The three types of busses that can be translated are host, ISA, and EISA. Notice that the EBC can not translate from the ISA/EISA bus to the Host bus, but it can translate from the Host bus to the EISA/ISA bus. The EISA bus's primary broadcast signals are START# and CMD#. The ISA bus's primary broadcast signals are MRDC#, MWTC#, IOWC# and IORC# (ISACMD).

An EISA cycle (signified by START# and CMD#) will occur for all Host bus/slave bus combinations, except:

1. Host master to Host slave

2. ISA master to ISA memory slave

3. Halt/Shutdown cycles

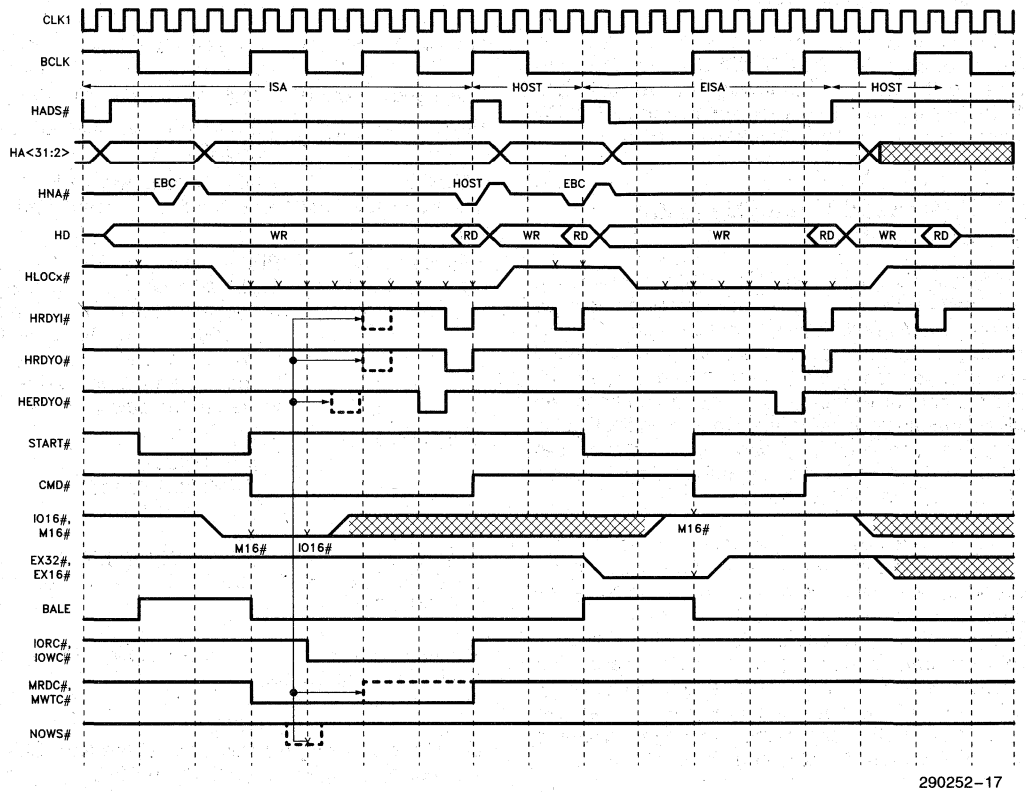
For example, if the CPU on the Host bus is accessing an ISA slave, the EBC will generate an ISA cycle (broadcast) as well as an EISA cycle (broadcast). On the other hand, if the CPU is accessing main memory on the Host bus, the EBC will not generate (broadcast) an EISA or ISA cycle.

EXAMPLE TRANSLATION:

Table 3-2 shows how the EBC determines the master and slave for cycle translations.

The EBC would sample HHLDA inactive, which indicates the CPU is master. HADS# would be active indicating the start of the cycle. The EBC would sample HLOCMEM# inactive indicating the host bus memory slave did not respond. Then it would sample EX16#, EX32#, and IO16# inactive; and M16# active, which indicates a 16-Bit ISA memory slave. Since this is a 32-bit versus 16-bit master to slave mismatch, the EBC runs two 16-bit ISA read cycles to the ISA memory slave to fetch the data. The EBC also controls the data swap buffers so that these two ISA bus cycles are translated to the appropriate bytes.

The EBC starts the first memory read cycle (activates MRDC#). The EBC steers the first transfer of data to the lower word. Then the EBC starts the second read cycle (activates MRDC#), and changes the address by manipulating A0, A1, and BHE# in order to read the second word. The EBC then steers the data to the upper word, and returns a READY# signal to complete the transfer. Figure 3-9 depicts this example.



NOTES:
When Host is accessing EISA or ISA Slave, one wait state is added if RDE# is active.
x = EBC Sampling Point
CLK1 is EBC Internal CLK1.

Figure 3-1. 386 CPU Host Master to 16-Bit ISA/Host/EISA/Host Slave

Figure 3-1 shows a host master to a 16-bit ISA slave cycle followed by a host slave, an EISA slave, and another host slave cycle.

3.3 Clock Generation

HCLKCPU in the 386 CPU has twice the clock frequency of the i486 CPU HCLKCPU signal. CLK1, internal to the EBC, is the inverse of the i486 CPU HCLKCPU signal. CLK1's relationship to HCLKCPU is shown in Figure 3-1A. Clock Generation.

3.4 Functional Timing Diagram Descriptions

In general, the beginning of any cycle starts when the master presents a valid address on the address bus and drives M-IO active to indicate a memory or I/O cycle. The address can be presented before the end of the current cycle to allow address pipelining. The memory or I/O slave decodes the address and drives the appropriate signals active indicating its type, size, and whether it can perform any special timings. EX32# or EX16# are activated by the slave if it can support EISA cycles.

The master drives START# active to indicate the beginning of the current cycle, W-R to indicate if it is a read or write cycle, and BE(3:0)# to indicate the bytes being transferred and their location on the EISA bus. The activation of CMD# by the EBC controls the data transfer to or from the slave. The length of CMD# can vary depending on the type and speed of the devices performing the transfer. EXRDY or CHRDY can be held low by the slave to insert wait states, thereby lengthening the cycle.

In the timing diagrams, some of the signals have been combined or abbreviated. A list of the combined signals follows:

HADS# = HADSO# is anded with HADS1#

RST[CPU:385] = RSTCPU is bussed with RST385

HLOC[M:IO] = HLOCM# is bussed with HLOCIO#

HWR#HMIO#, = HW/R# is bussed with HM/IO#.

A signal level of 10 would indicate a write to I/O.

BURST[MS:SL]# = MSBURST# is bussed with SLBURST#

WR,MIO = W-R is bussed with M-IO

A signal level of 01 would indicate a read to memory.

BE[3:0]# = BE3#, BE2#, BE1#, and BE0# bussed together. A signal level of 0011 would indicate BE3# = 0, BE2# = 0, BE1# = 1 and BE0# = 1, therefore the upper word is involved in the current cycle.

CPEN[13, 3:1] = SDCPYEN13#, SDCPYEN03#, SDCPYEN02#, and SDCPYEN01#.

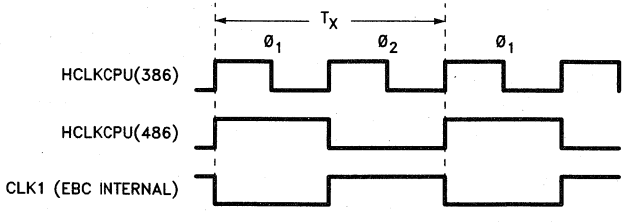
A signal level of 0101 indicates SDCPYEN13# = 0, SDCPYEN03# = 1, SDCPYEN02# = 0, and SDCPYEN01# = 1

All other signals that are bussed follow the same numbering order format.

NOTE:
CLK1 in the Timing diagrams is not the i486 CLK1. It is the CLK1 signal internal to the EBC.

HOST MASTER DIAGRAM HIGHLIGHTS

Figures 3-2 through 3-11 show a Host master to Host slave, 32/16-bit EISA/ISA slave, or 8-bit ISA slave cycles. Locked cycles, burst cycles, assembly or disassembly cycles are shown with and without wait states.



NOTE:
CLK1 in the Timing Diagrams is not the i486 CLK1. It is the CLK1 internal to the EBC.

Figure 3-1A. Clock Generation CLK1 Relation to HCLKCPU

Figure 3-3 shows a Host master reading and writing to a 32-bit EISA memory and I/O slave. In this diagram all of the cycles are standard cycles (2 BCLK).

Figure 3-4 shows a Host master reading from a 32-bit EISA memory slave. It shows two standard back-to-back read cycles with an extra CPU CLK1 delay inserted because RDE# is active. This occurs when RDE# is active during any Host read from the EISA bus. This causes HERDYO# and HRDYO# to be delayed so that more time is allowed for the data to propagate to the Host bus.

Figure 3-5 shows a Host master reading a 32-bit EISA memory slave. The Host master has driven the HLOCK# signal active to guarantee exclusive I/O or memory access.

Figure 3-9 shows a Host master reading from a 16-bit ISA memory slave (HHLDA is low, M16# is active, and MRDC# is active). A standard cycle for an ISA slave is shown (3 BCLK). Since the master and slave sizes are mismatched, byte assembly is required. To assemble the data from a 16-bit slave to a 32-bit master requires two cycles. In the first cycle, the first two bytes (lower word) are read from the slave. In the second cycle, the next two bytes (upper word) are read from the slave, thus completing the assembly.

Figure 3-11 shows a Host master writing to an 8-bit ISA I/O slave. In normal cycles including this one, the falling edge of HADS# indicates the address is valid, and HRDYO# driven active indicates the end of the cycle. Since only the upper word is being written, one byte is written to the ISA I/O slave during each cycle, making this a two cycle transfer. CHRDY is activated to add wait states.

EISA MASTER DIAGRAM HIGHLIGHTS

Figures 3-12 through 3-30 show an EISA master to Host, 32/16-bit EISA/ISA, or 8-bit ISA slave cycles. Standard cycles, Burst cycles, and Assembly or Disassembly cycles are shown with and without wait states.

Figure 3-13 shows a 32-bit EISA master writing to a 32-bit EISA I/O slave. This is a standard cycle (2 BCLK). No assembly or disassembly is required since the master and slave size match.

Figure 3-14 shows a 32-bit EISA master reading from a 32-bit EISA memory slave during a burst cycle. The EBC samples SLBURST# at the rising edge of START# and one BCLK later it samples MSBURST#, except if EXRDY is inactive then the EBC samples MSBURST# one additional BCLK later.

Figure 3-20 shows a 32-bit EISA master writing to a 16-bit ISA slave (mismatched sizes), two bytes are written to the ISA I/O slave during each cycle. The lower word is written first, then the upper word is written to the ISA slave. Two cycles complete this disassembly transfer.

Figure 3-30 shows a 16-bit EISA master writing to an 8-bit ISA memory slave. Since the master/slave size is mismatched, more than one cycle is needed to complete this write cycle. In the first cycle, a byte is written to the slave which becomes the lower byte. In the second cycle, byte swapping occurs by the EBC since CPEN[13,3:1]# = 1110 (SDCPYEN[13,3:1]#). This routes the byte through the data swap buffers into the upper byte location of the 16-bit master and completes the cycle.

ISA MASTER DIAGRAM HIGHLIGHTS

Figures 3-31 through 3-35 show a 16-bit ISA master to 32/16-bit EISA, or 8-bit ISA slave cycles. Standard or assembly cycles with or without byte swapping are shown.

Figure 3-33 shows BCLK stretching during a 16-bit ISA master read from a 16-bit EISA memory slave. During this type of cycle the rising edge of BCLK is synchronized to the falling edge of CMD#. The master and slave size are the same, so only one cycle is needed to transfer the data. CHRDY is pulled low by the slave to add one wait state.

MISCELLANEOUS CYCLE DIAGRAM HIGHLIGHTS

Figures 3-36 through 3-39 shows other cycles such as Halt/shutdown and Reset.

Figure 3-36 shows a Halt/shutdown cycle. There is no START# or CMD# signal active for this cycle. Also, the CPU and 82385 (in a 386 system) are reset for the Shutdown case only.

Figure 3-39 shows the HHOLD/RSTCPU interlock characteristics. HHOLD will never be active when RSTCPU is active. The active SPRWOK signal indicates that the power to the system board is at the correct level. There is a system reset occurring in this diagram. Note that no cycles are run during this time (HADS#, START#, and CMD# are inactive), and other control signals are inactive.

DMA CYCLE TIMING DIAGRAMS

In Figures 3-40 through 3-65, each timing shows a 32-bit, 16-bit, or 8-bit I/O device accessing 32-bit, 16-bit, or 8-bit memory. All four types of cycles are shown:

Compatible, Type A, Type B, and Burst.

During DMA cycles all of the master decode signals equal one. The EBC samples ST(3:0) at the rising BCLK edge one BCLK before START# to determine the I/O (DMA requestor) size. The memory size is determined by the slave decode signal levels.

When the memory size is less than the I/O size, more than one cycle can occur when assembly/disassembly is needed, except during Compatible cycles. Only one cycle will occur for Compatible cycles.

When the memory size is greater than or equal to the I/O size, only one cycle is needed. See EISA specifications for more information.

The EBC sampling points for signals is shown in Figures 3-40 to 3-53 DMA cycle timing diagrams.

DMA Compatible cycles (Figures 3-40 and 3-41) show a standard cycle and what occurs if wait states are added. Two different signal transitions are shown for CMD#, MRDC#/MWTC#, IOWC#, and DRDY. The first transition occurs in standard cycles. The second transition would occur when wait states are added by the slave driving either CHRDY or EXRDY active.

Figure 3-42 through Figure 3-45 DMA Type A and Type B cycles show a standard cycle, with and without wait states added. Two different command signal transitions are shown to reflect these cases. The first transition occurs in standard cycles. The second transition would occur when wait states are added by the slave driving either CHRDY or EXRDY active.

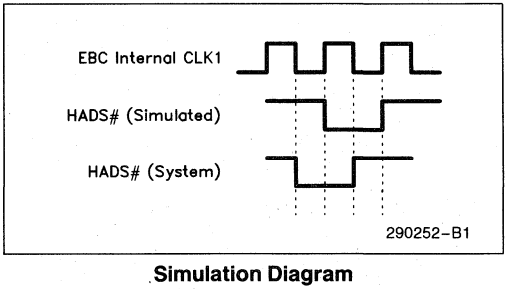
Figure 3-48 and Figure 3-49 Burst DMA cycle to non-burst memory shows standard burst cycles, and where signals would transition if wait states are added. Two different command signal transitions are shown to reflect both of these cases.

Figure 3-53 Refresh cycle shows a standard cycle, and a cycle with wait states added. Two different command signal transitions are shown to reflect both of these cases.

NOTE:

The timing diagrams that follow are very similar to EISA timing specifications, but the timing diagrams show simulations of the EBC device functionality and sampling points and are not taken from the EISA system.

For example, HADS# shown in the timing diagrams does not show system functionality, but serves as a stimulus for the EBC based on the sampling edges of the internal CLK1 signal. See the Simulation Diagram for an example of this.



Simulation Diagram

3.5 Timing Diagrams

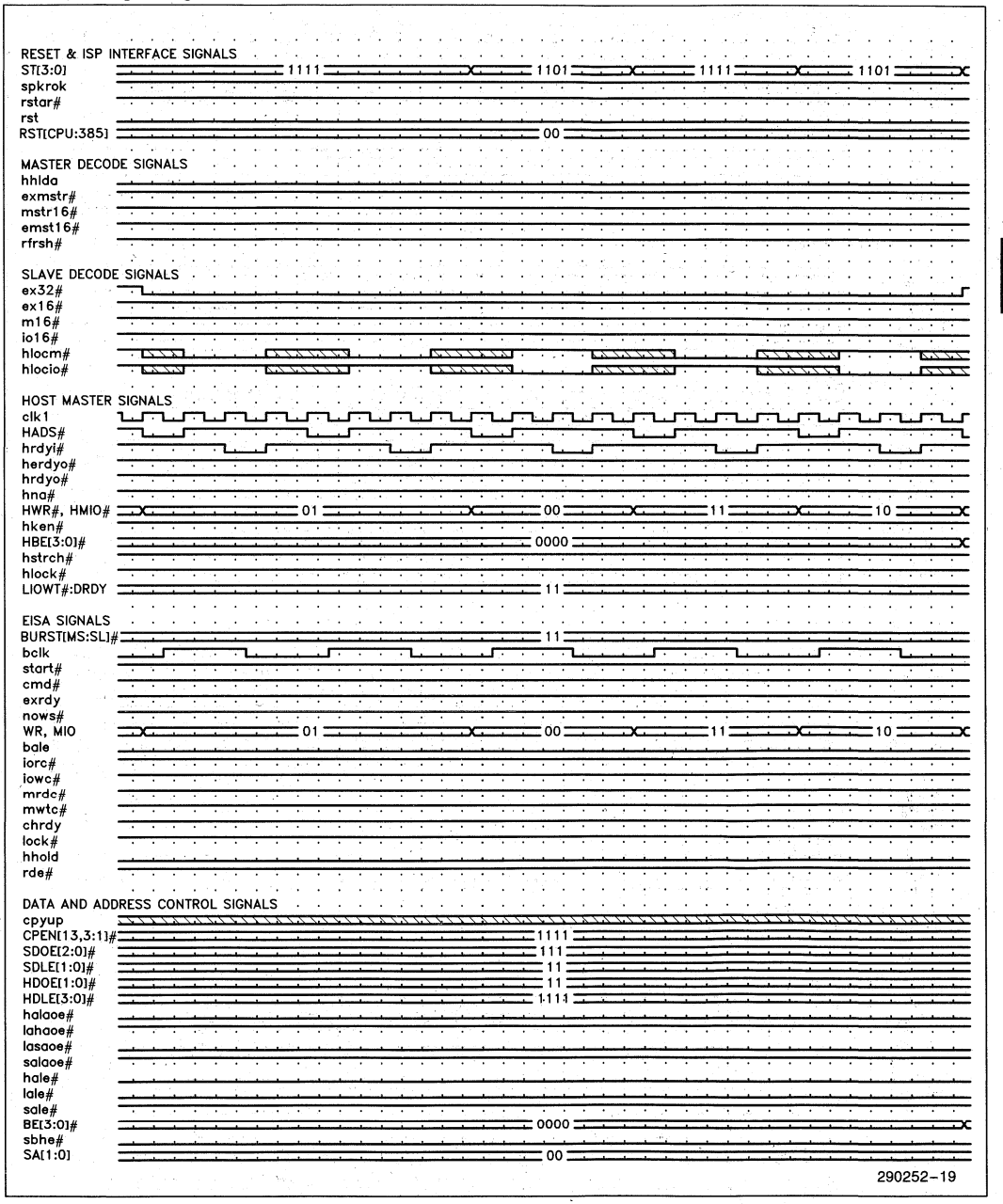


Figure 3-2. Host Master to Host Memory I/O Slave Standard Read/Write Cycles

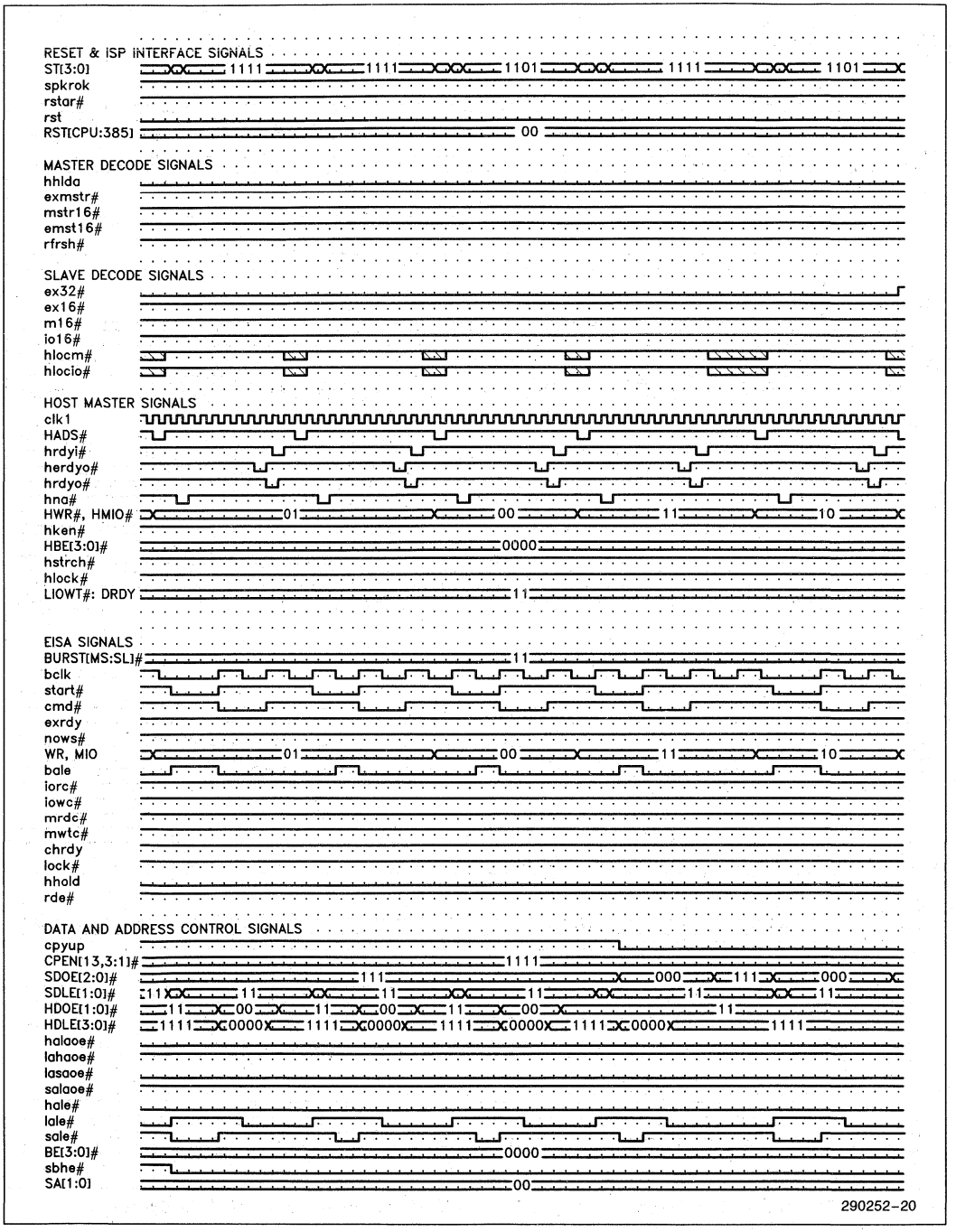


Figure 3-3. Host Master to 32-Bit EISA Memory I/O Slave Standard Read/Write Cycles

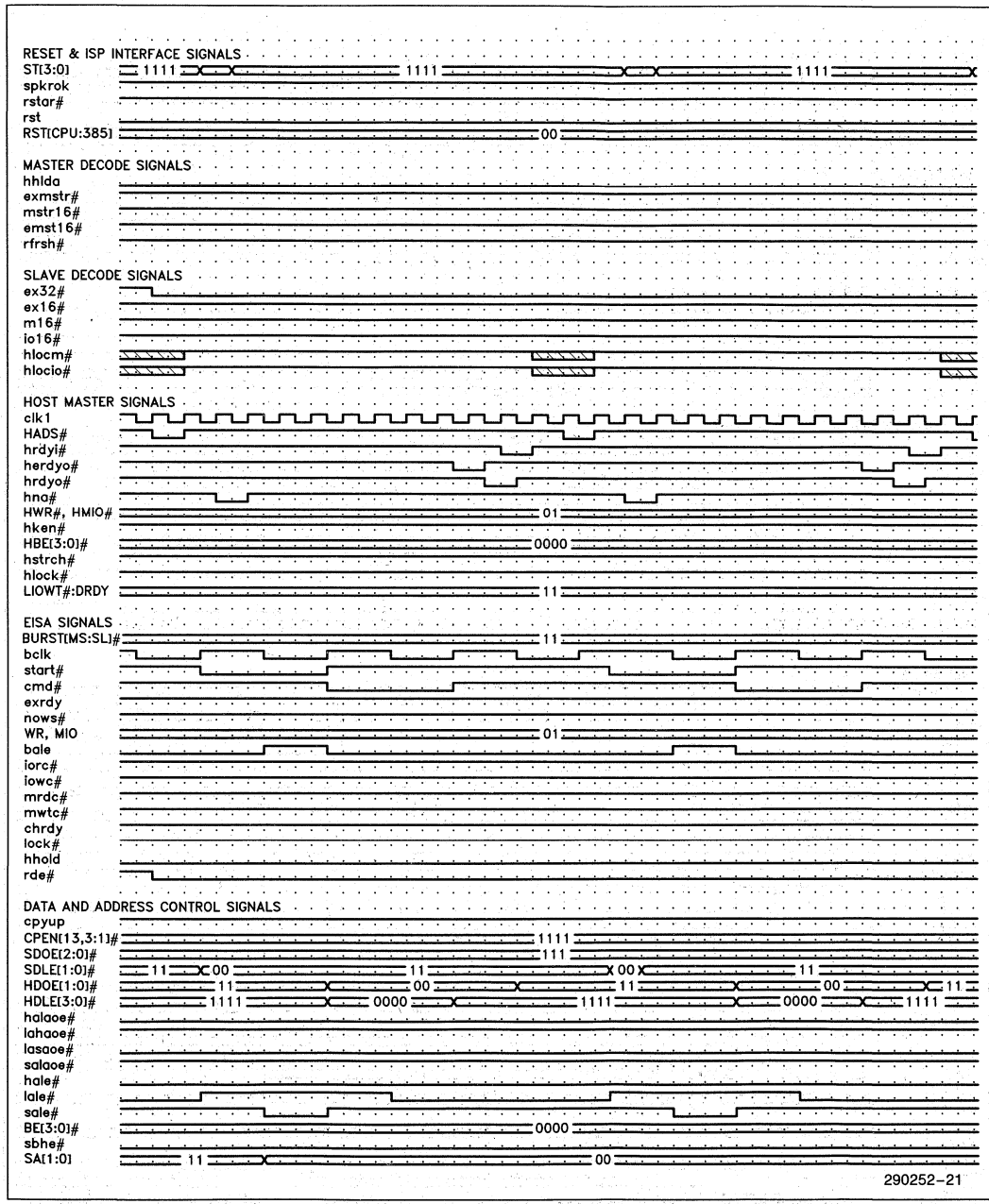


FIGURE 3-4. Host Master to 32-Bit EISA Memory Slave Two Standard Read Cycles

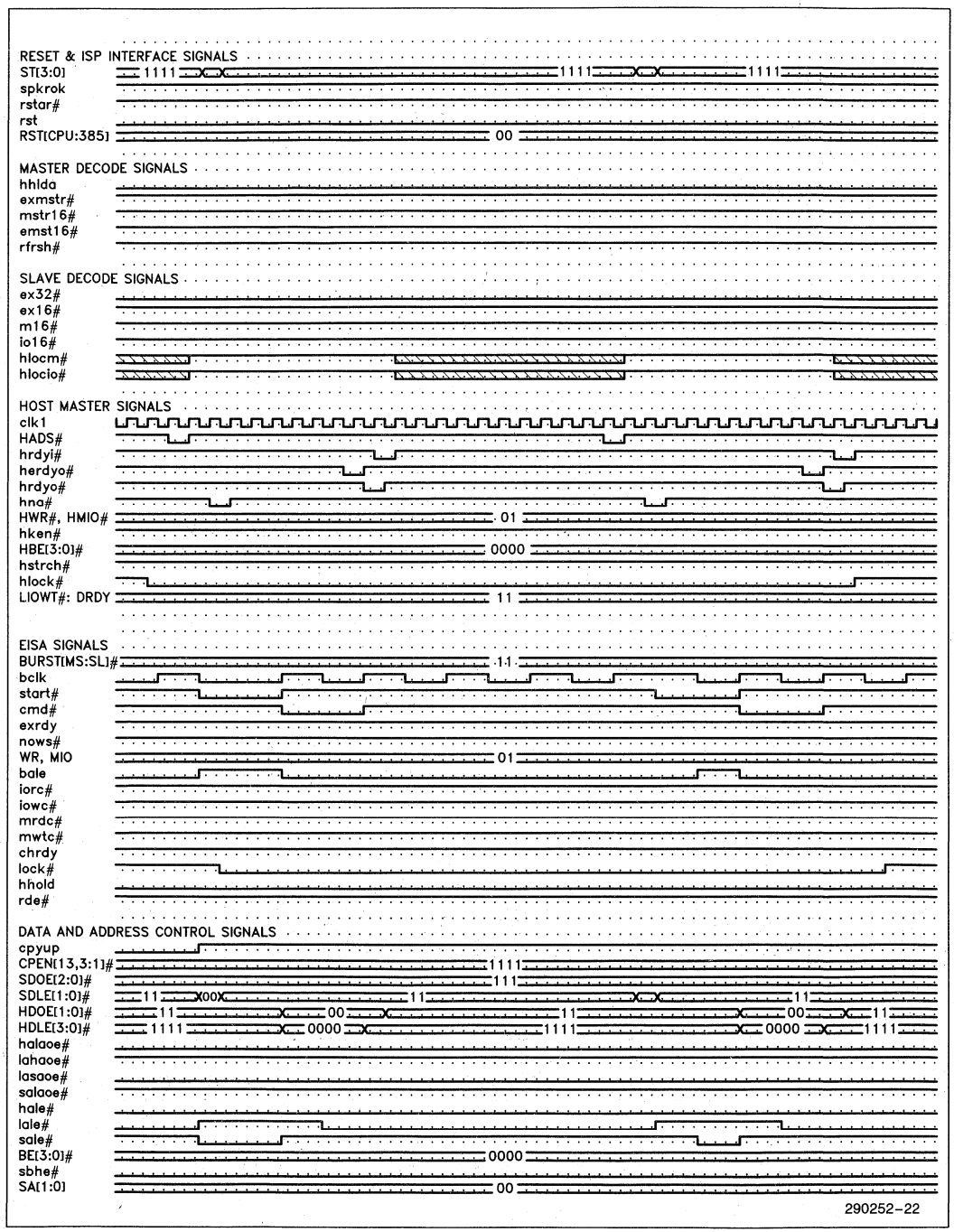


Figure 3-5. Host Master to 32-Bit EISA Memory Slave Locked Read Cycle

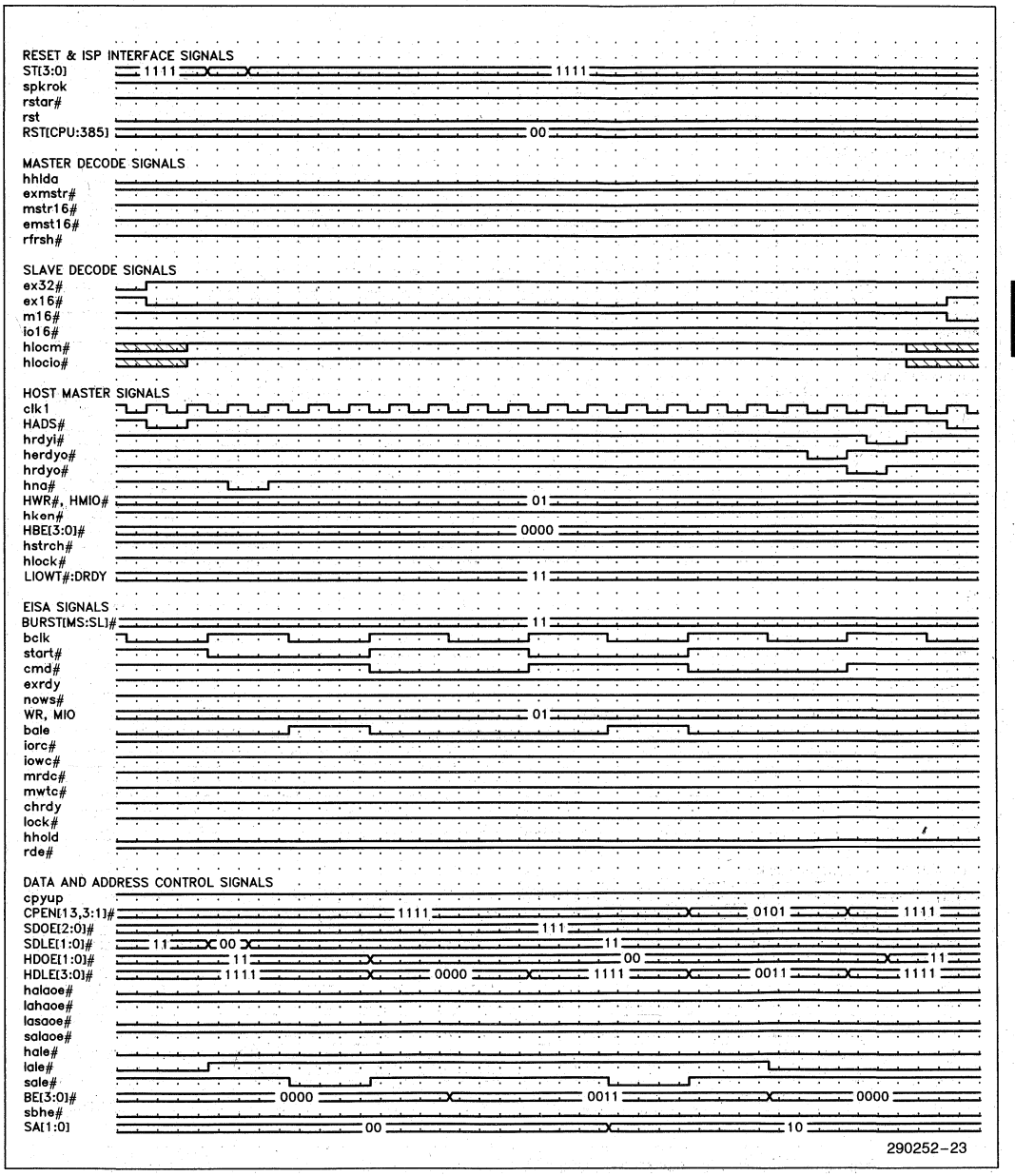


Figure 3-6. Host Master to 16-Bit EISA Memory Slave Read Assembly Cycle

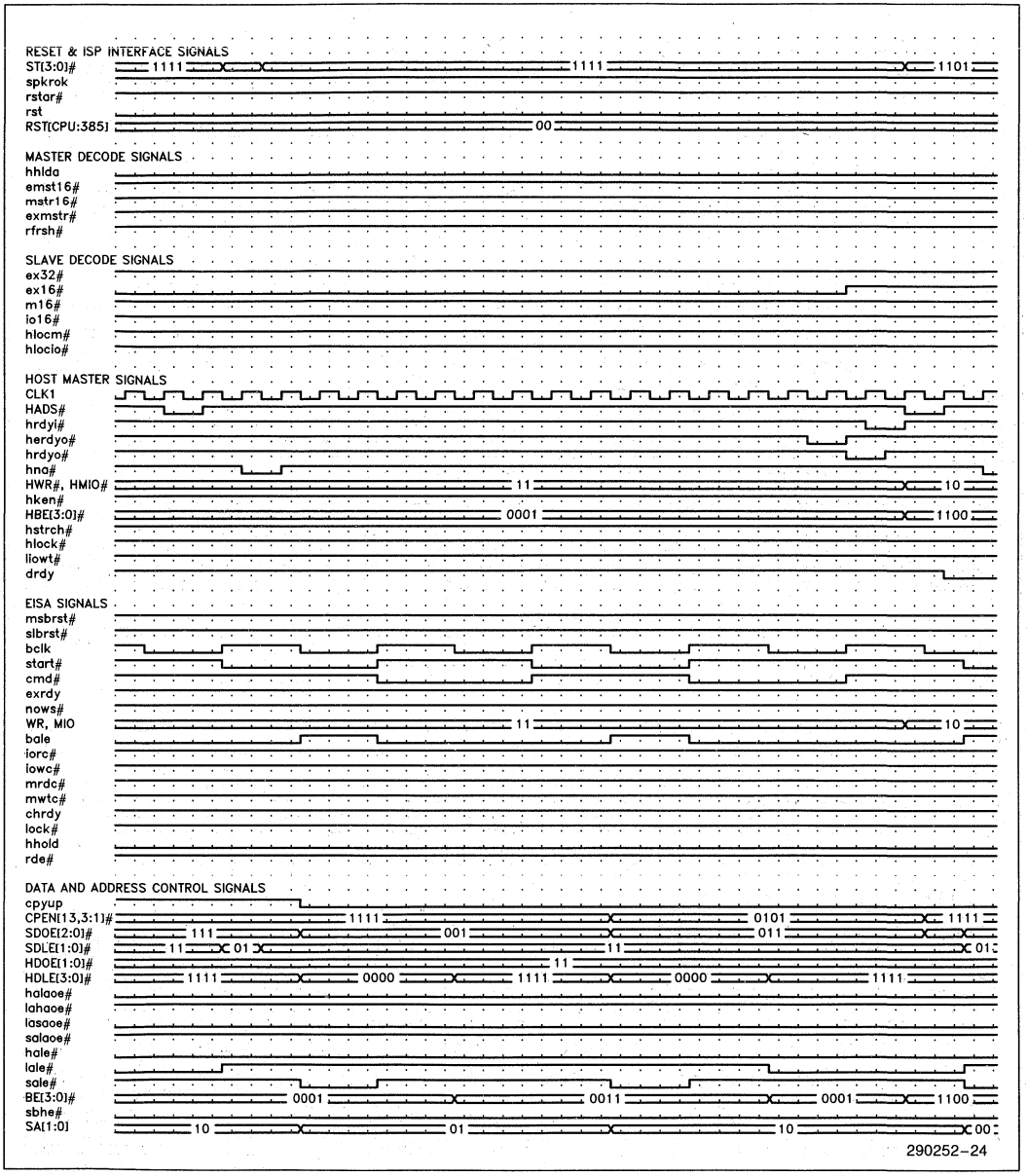


Figure 3-7. Host Master to 16-Bit EISA Memory Slave Disassembly Write Cycles

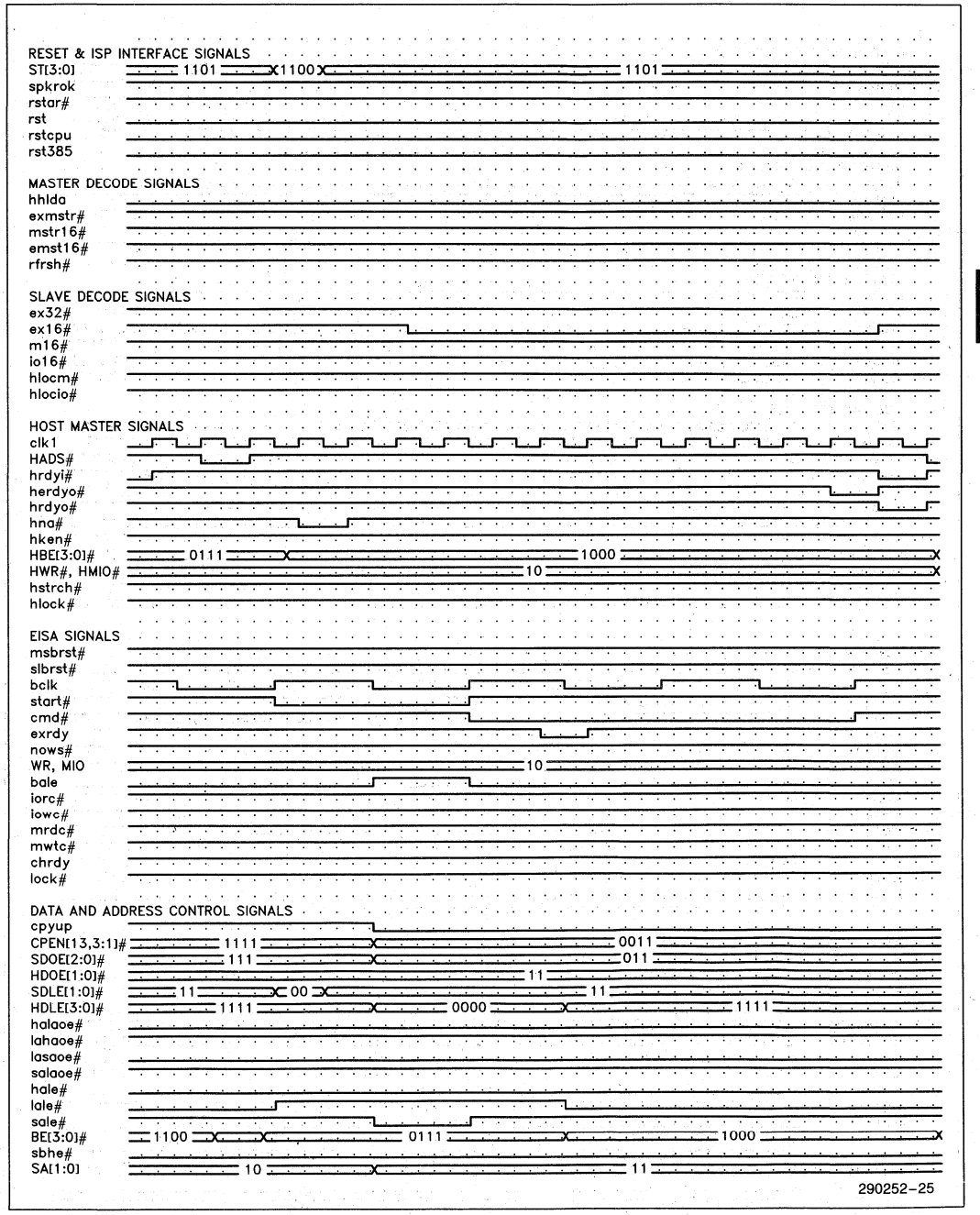


Figure 3-8. Host Master to 16-Bit EISA I/O Slave Byte Swap Write Cycle with Wait States

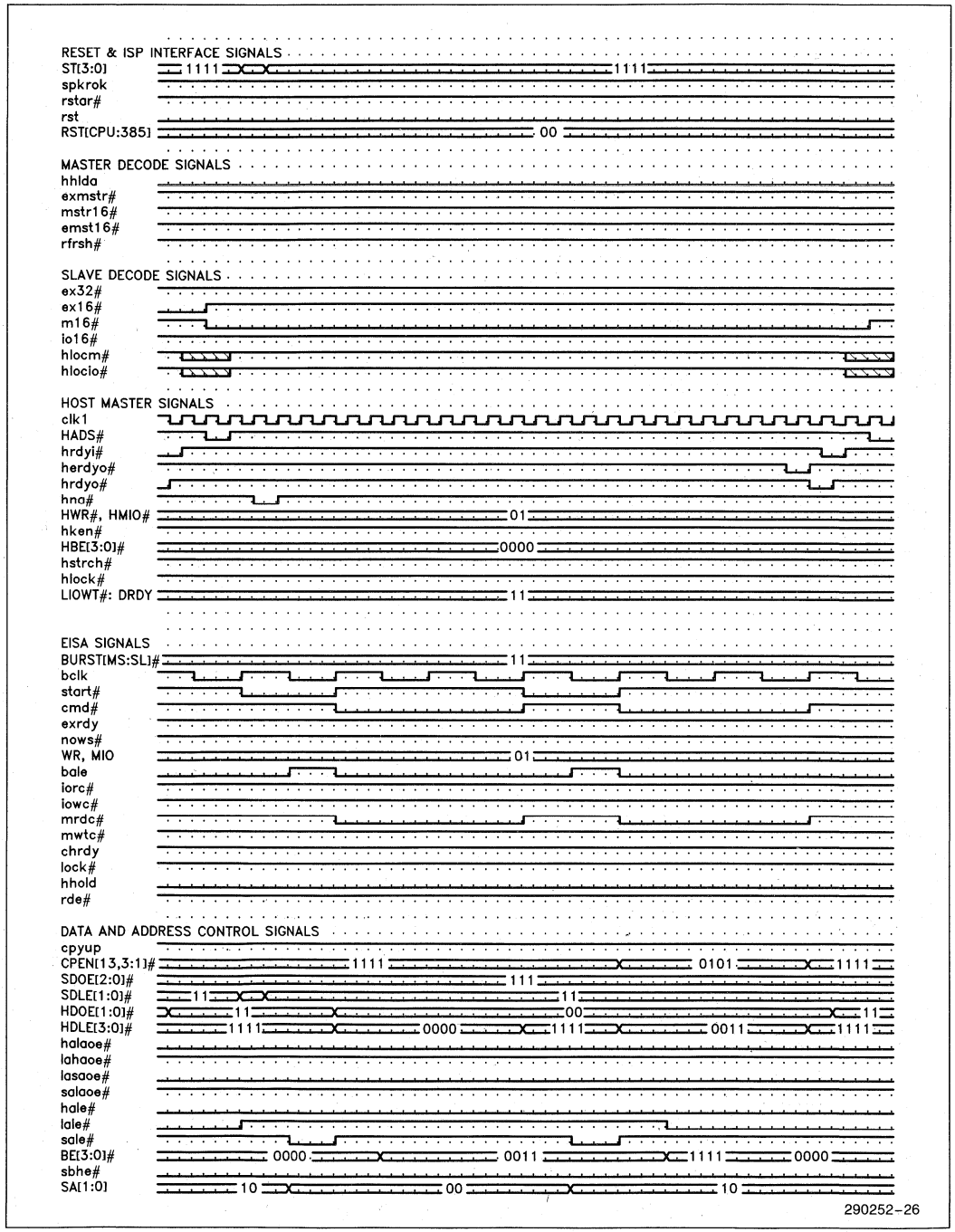


Figure 3-9. Host Master to 16-Bit ISA Memory Slave Assembly Read Cycle

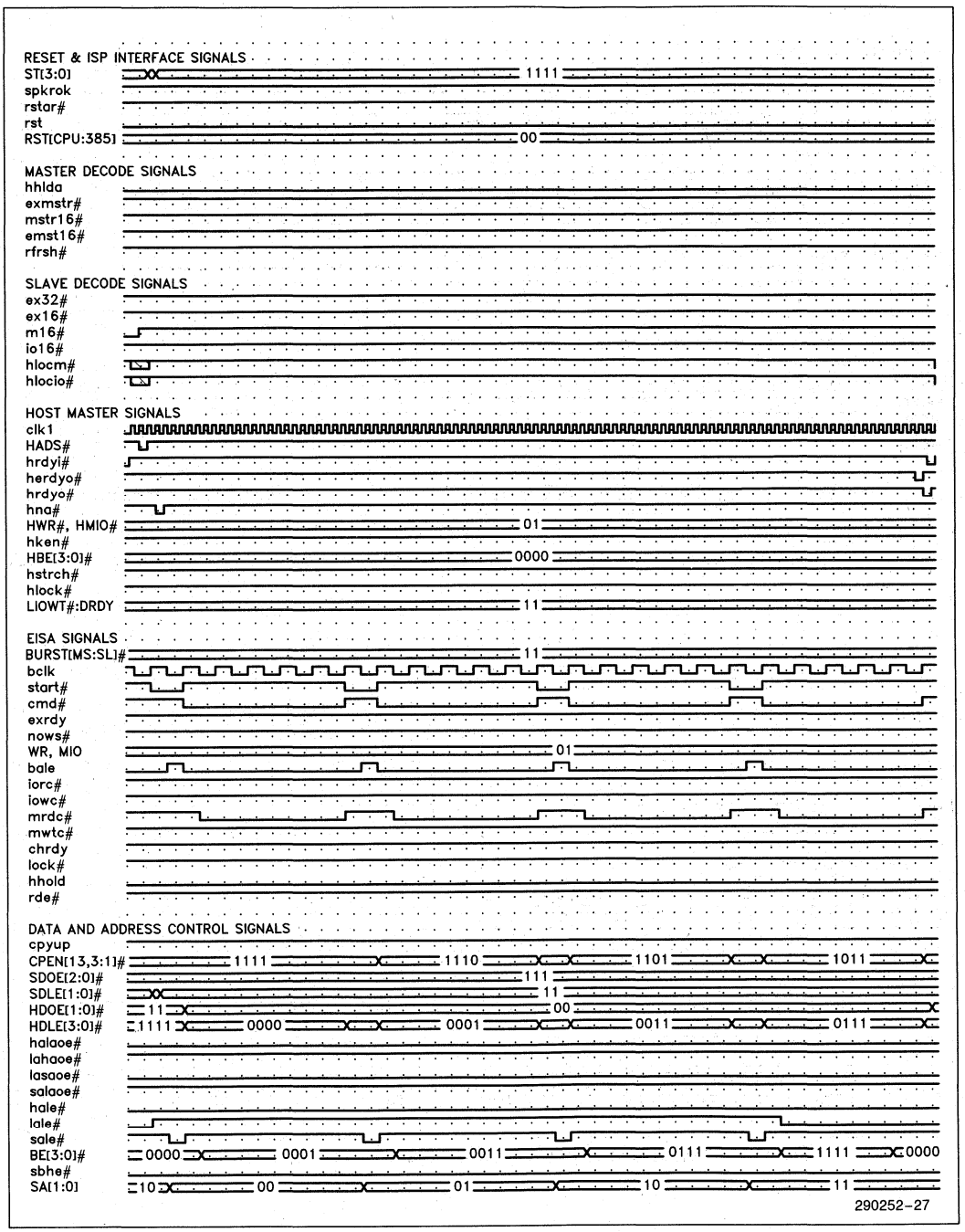


Figure 3-10. Host Master to 8-Bit ISA Memory Slave Assembly Read Cycle

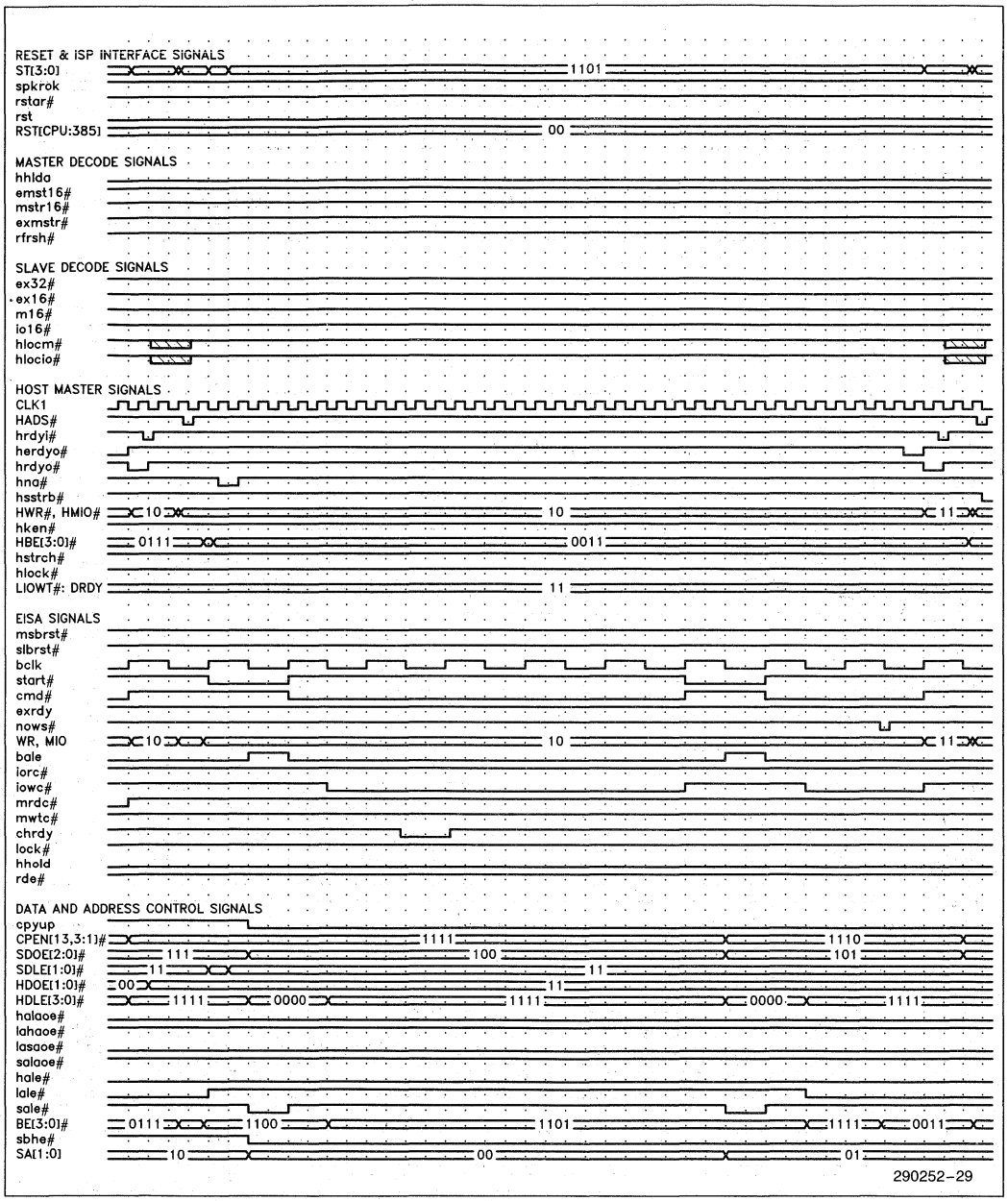


Figure 3-11. Host Master to 8-Bit ISA I/O Slave Disassembly Write Cycle

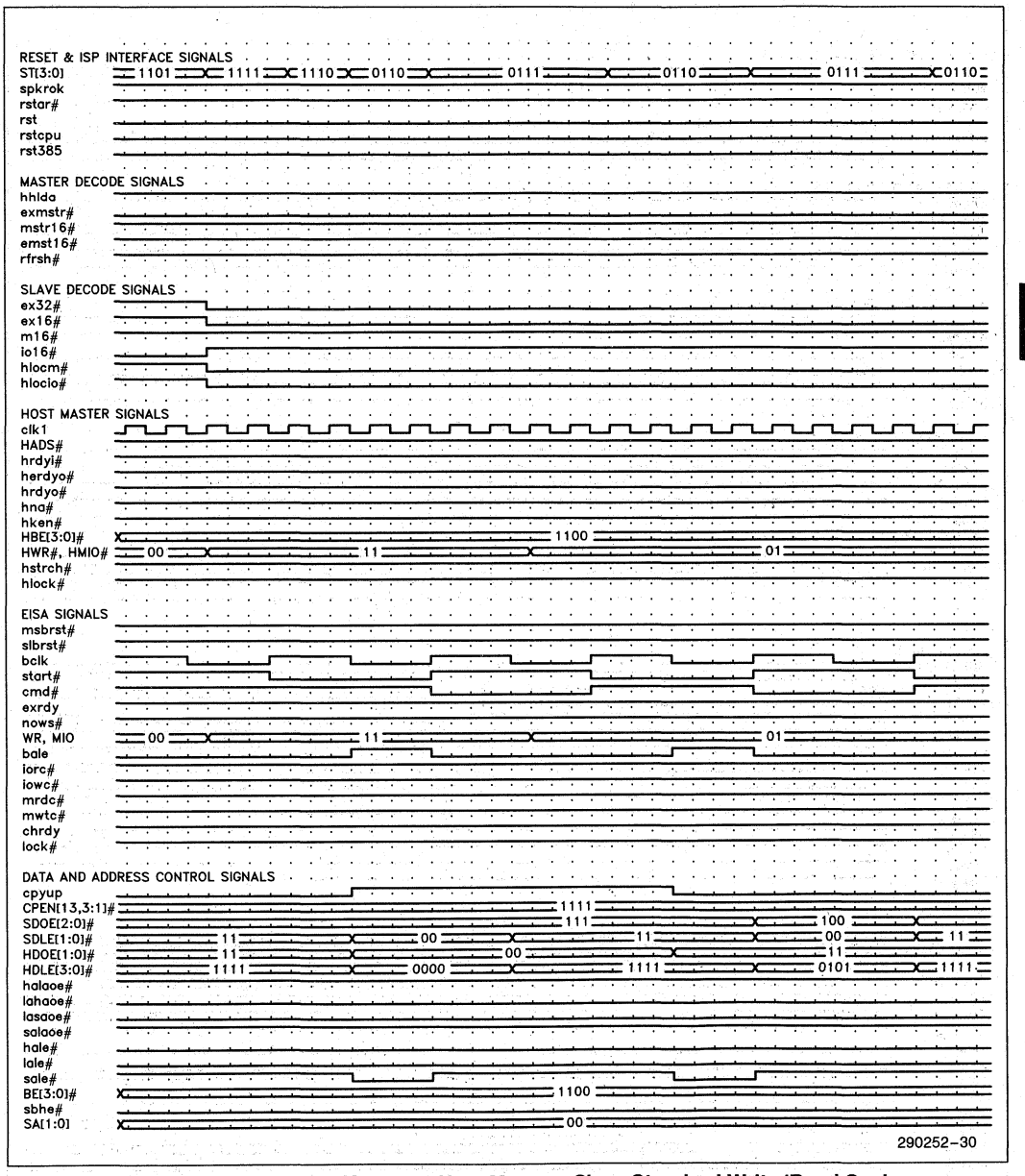


Figure 3-12. 32-Bit EISA Master to Host Memory Slave Standard Write/Read Cycles

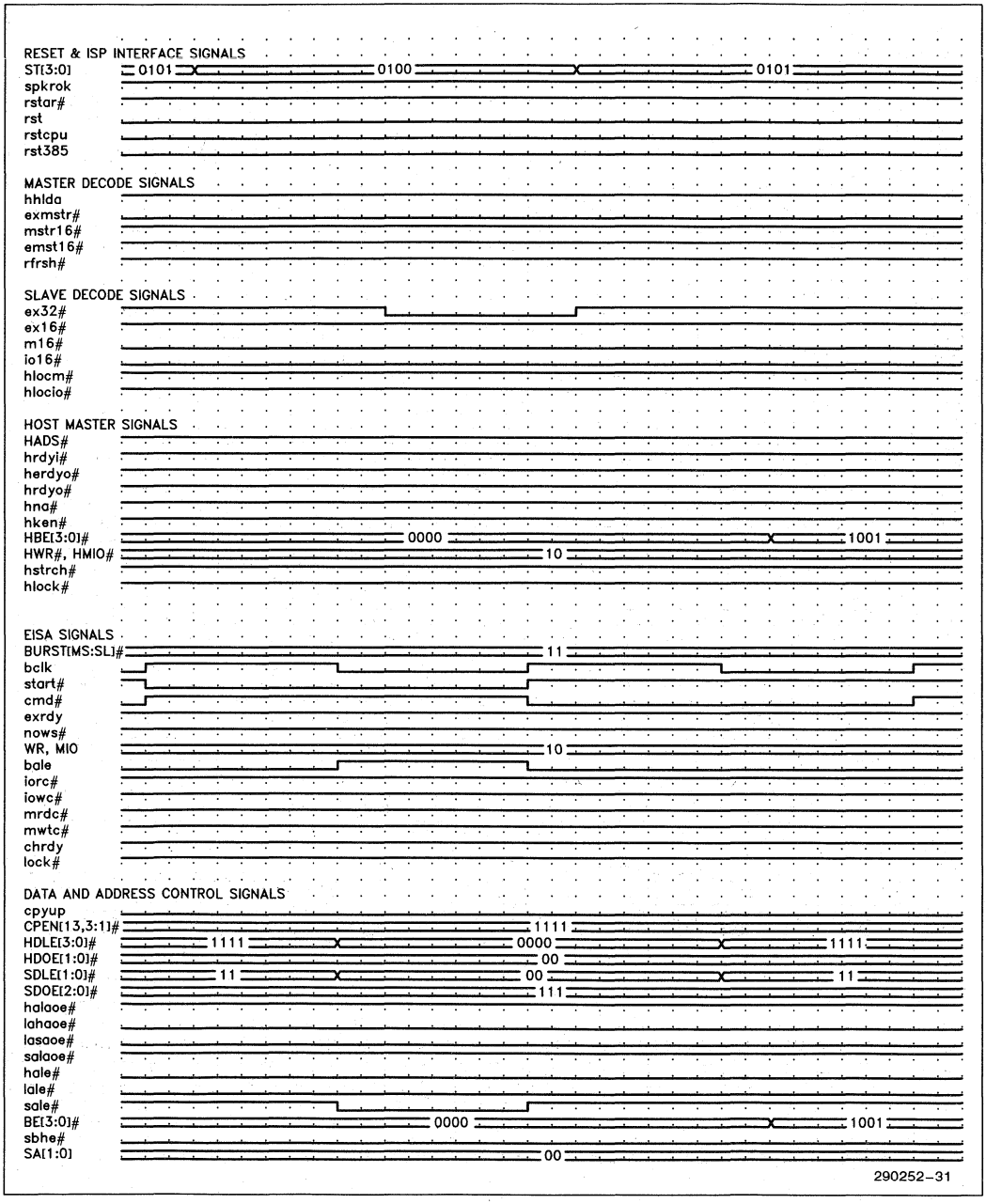


Figure 3-13. 32-Bit EISA Master to 32-Bit EISA I/O Slave Standard Write Cycle

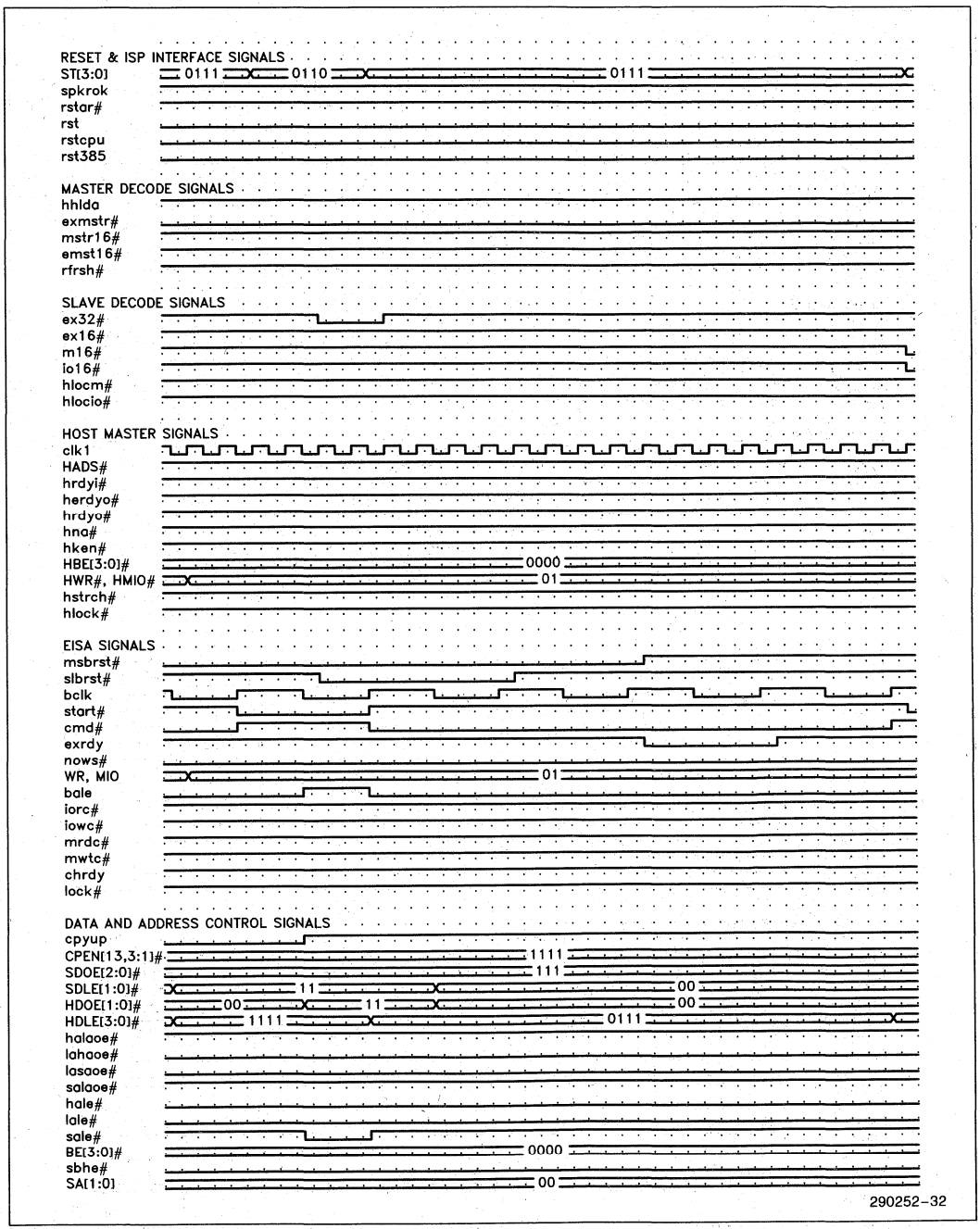
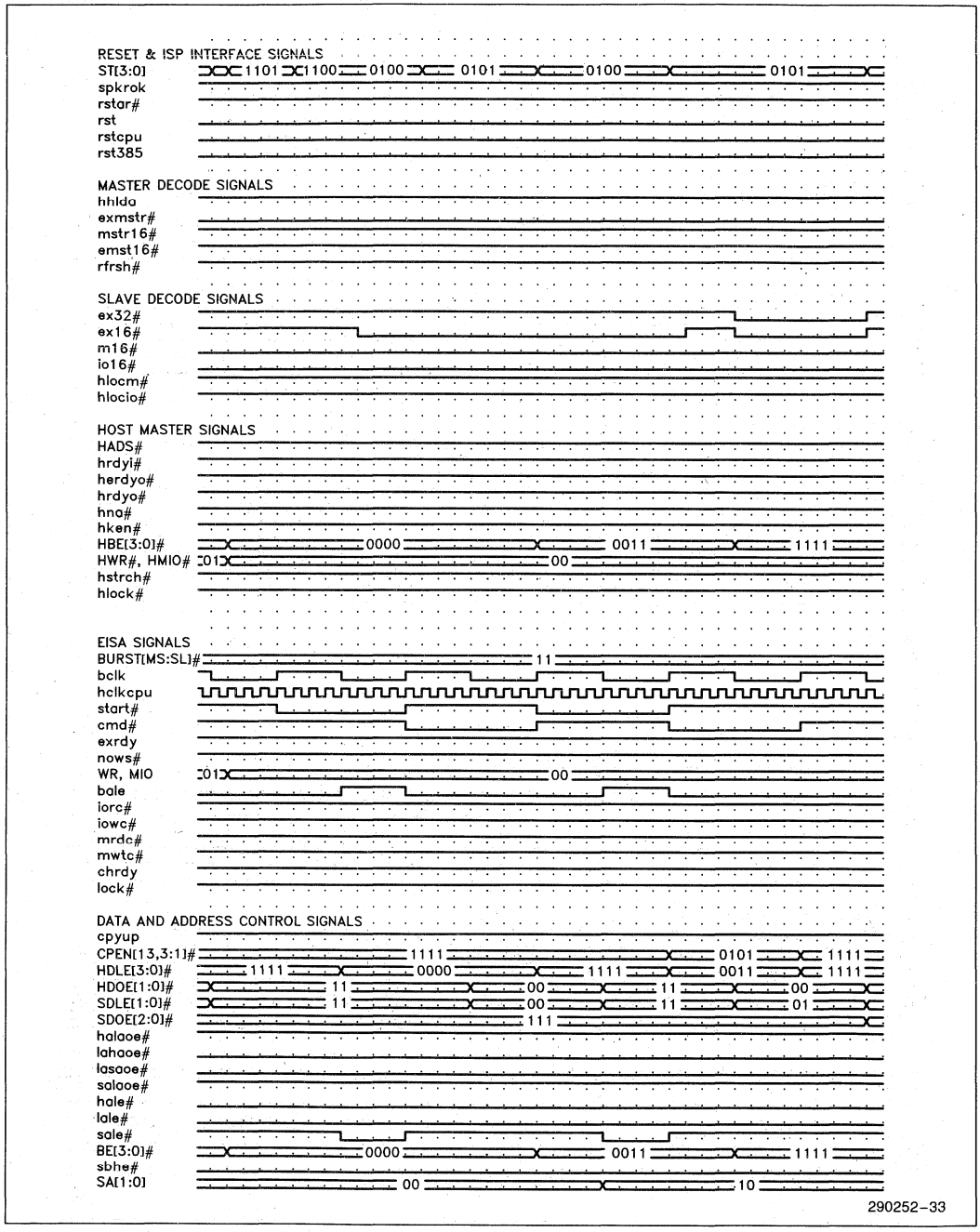


Figure 3-14. 32-Bit EISA Master to 32-Bit EISA Memory Slave 1 Wait State Burst Read Cycle



290252-33

Figure 3-15. 32-Bit EISA Master to 16-Bit EISA I/O Slave Assembly Read Cycle

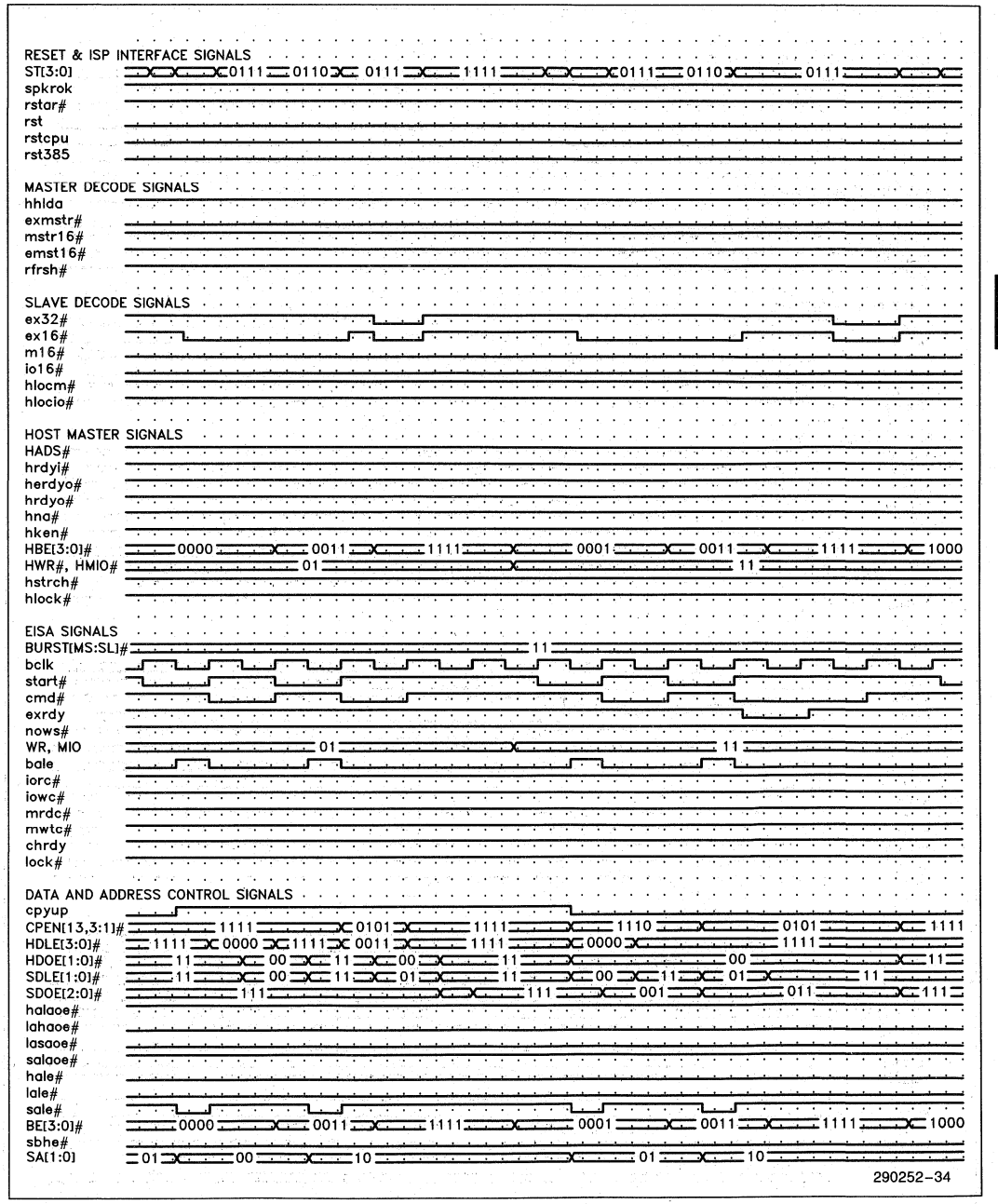


Figure 3-16. 32-Bit EISA to 16-Bit EISA Memory Slave Assembly/Disassembly Read/Write Cycles

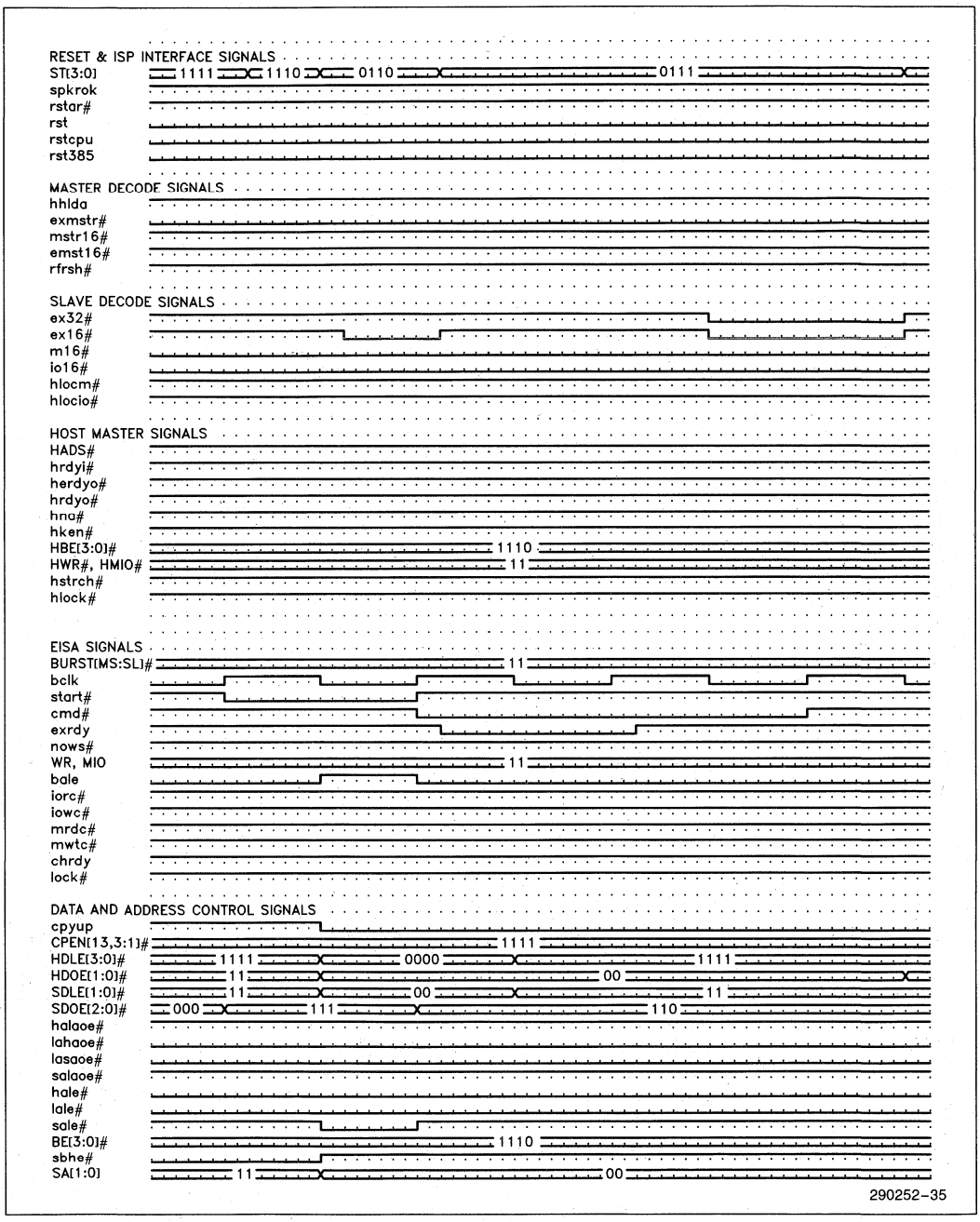


Figure 3-17. 32-Bit EISA Master to 16-Bit EISA Memory Slave No Assembly Write Cycle with Wait States

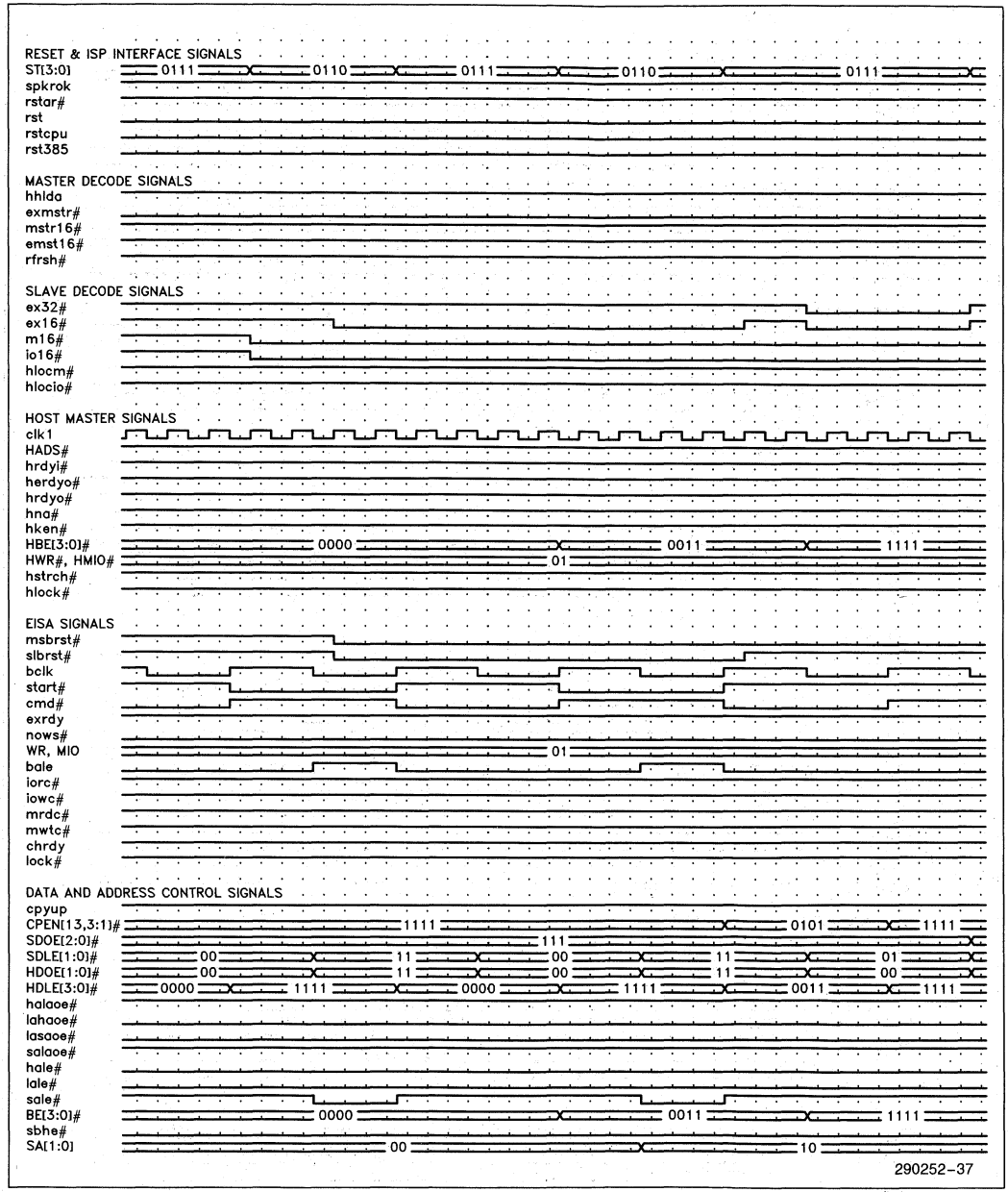


Figure 3-18. 32-Bit EISA Master to 16-Bit EISA Memory Slave Read During 0 Wait State Burst Cycles

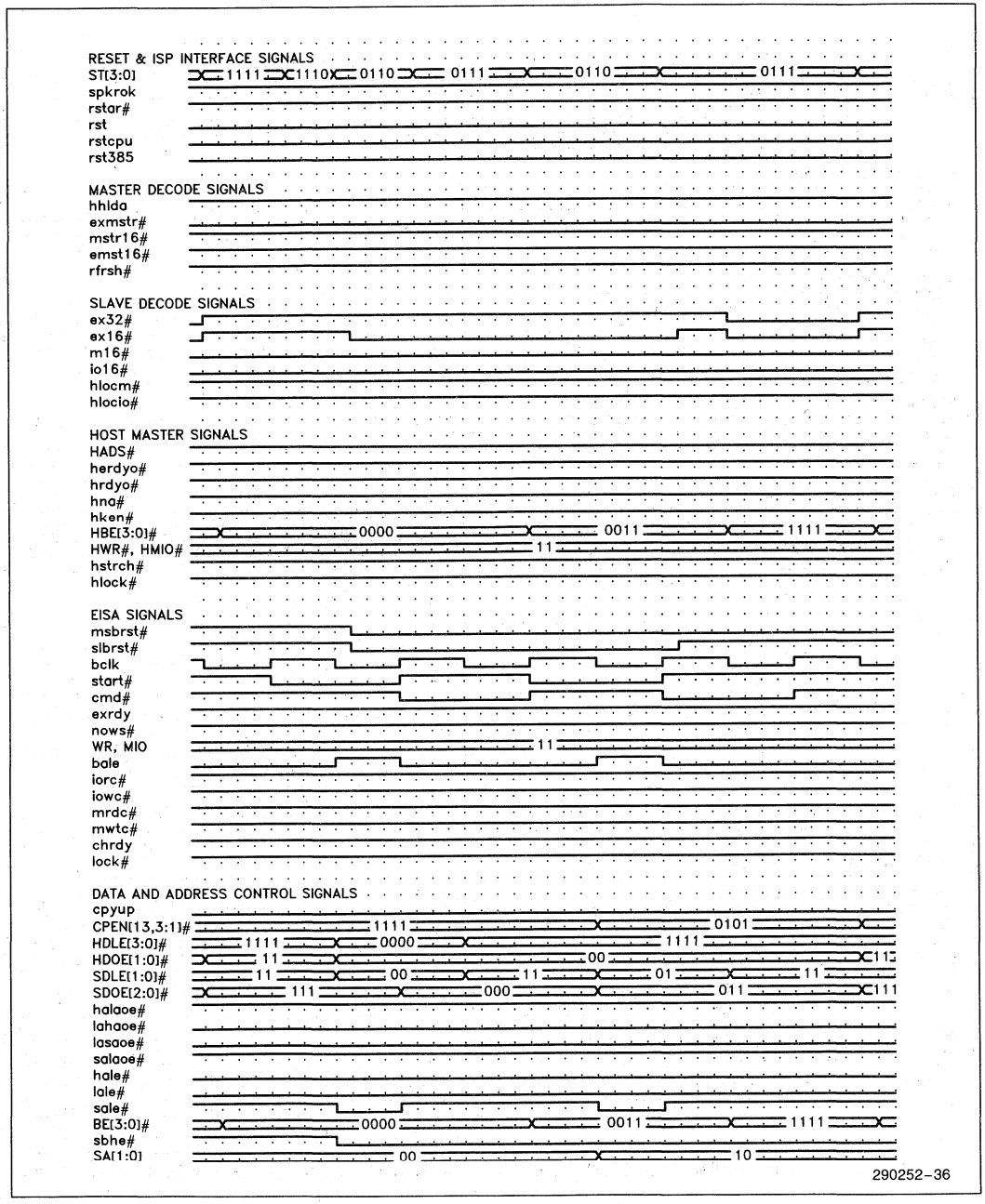
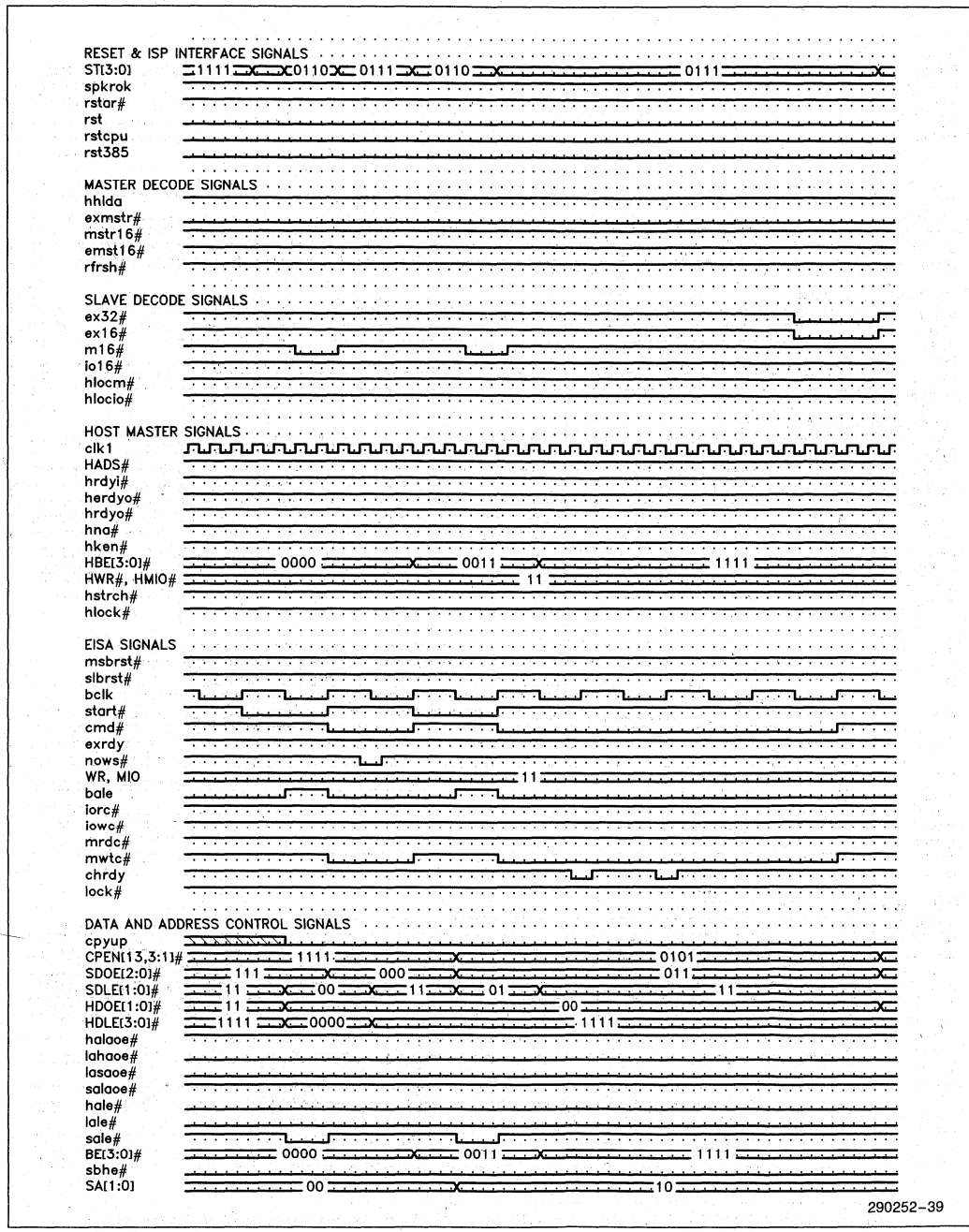


Figure 3-19. 32-Bit EISA Master to 16-Bit EISA Memory Slave 0 Wait State Burst Disassembly Write Cycle



290252-39

Figure 3-20. 32-Bit EISA Master to 16-Bit ISA Memory Slave Disassembly Write Cycle

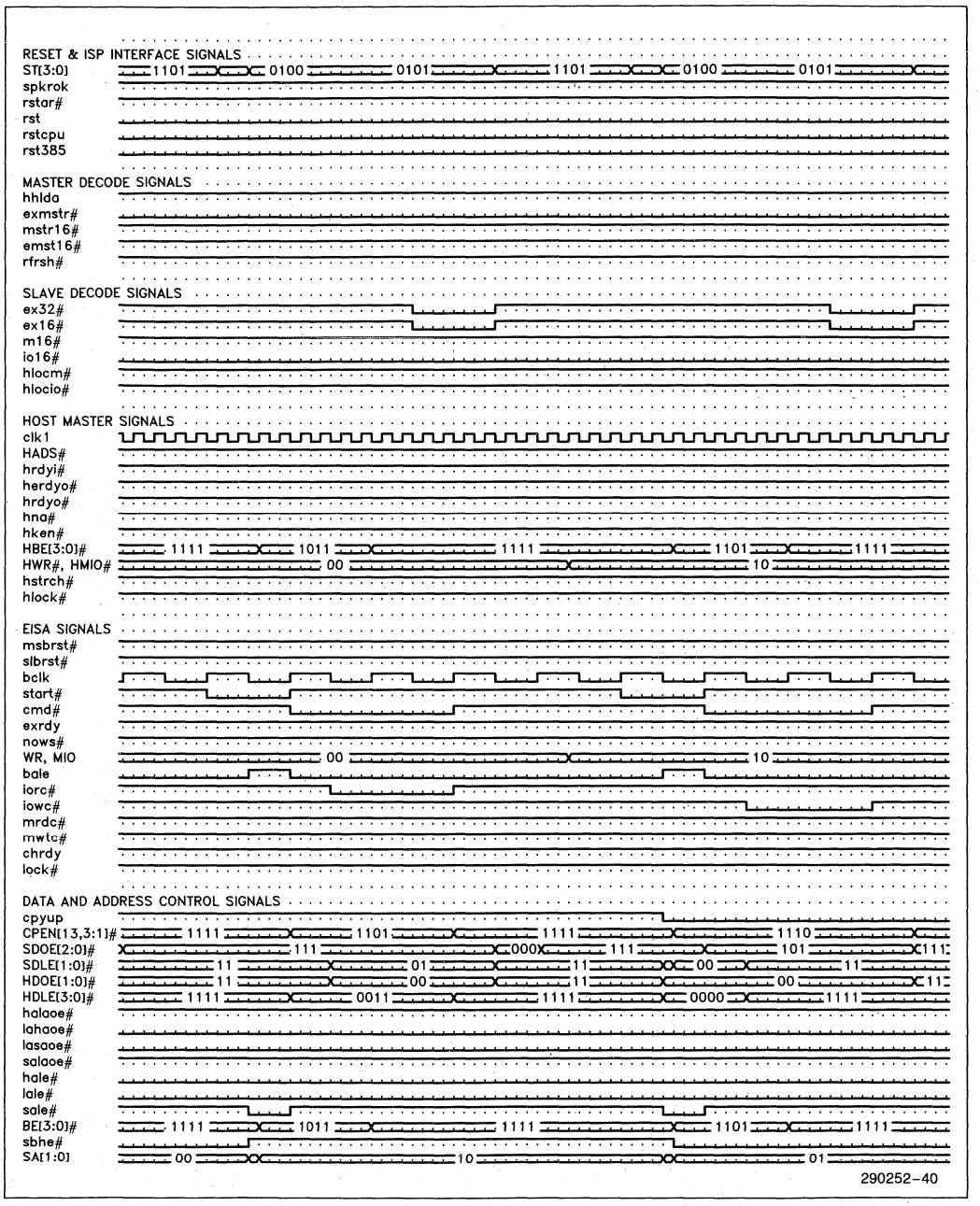


Figure 3-21. 32-Bit EISA Master to 16-Bit ISA I/O Slave Standard Read/Write Cycles

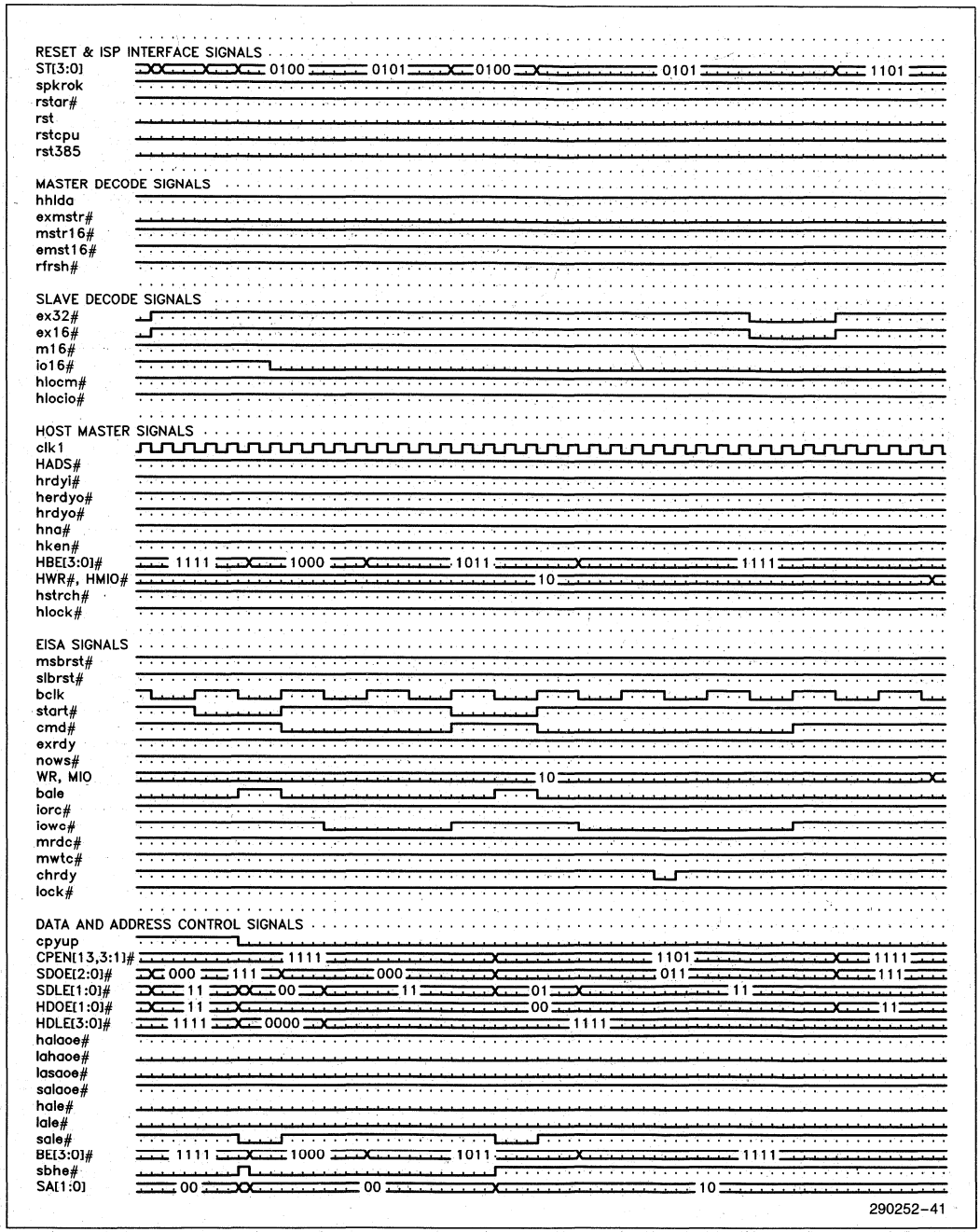


Figure 3-22. 32-Bit EISA to 16-Bit ISA I/O Slave 1 Wait State Disassembly Write Cycles

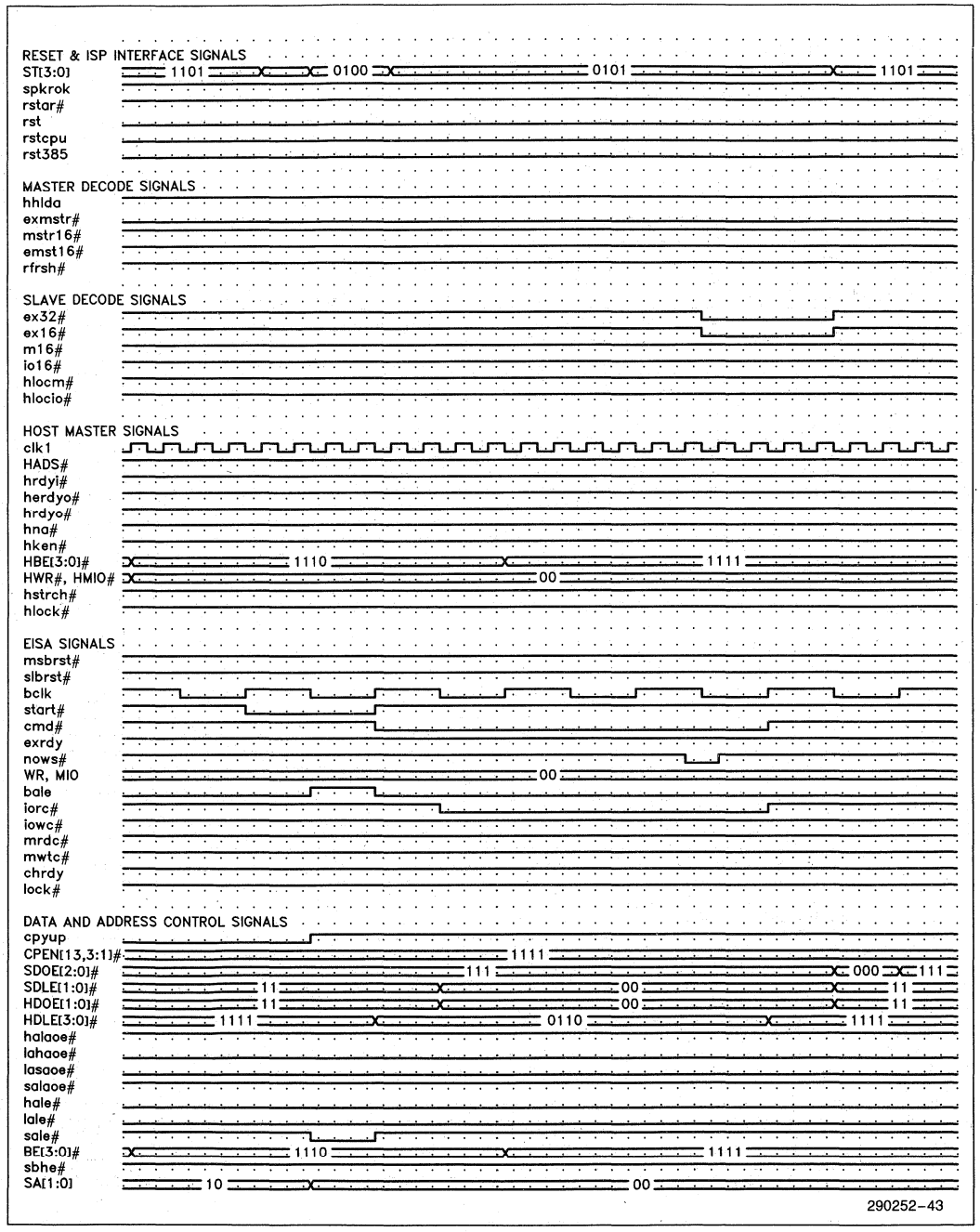


Figure 3-23. 32-Bit EISA Master to 8-Bit ISA I/O Slave NOWS# No Assembly Read Cycle

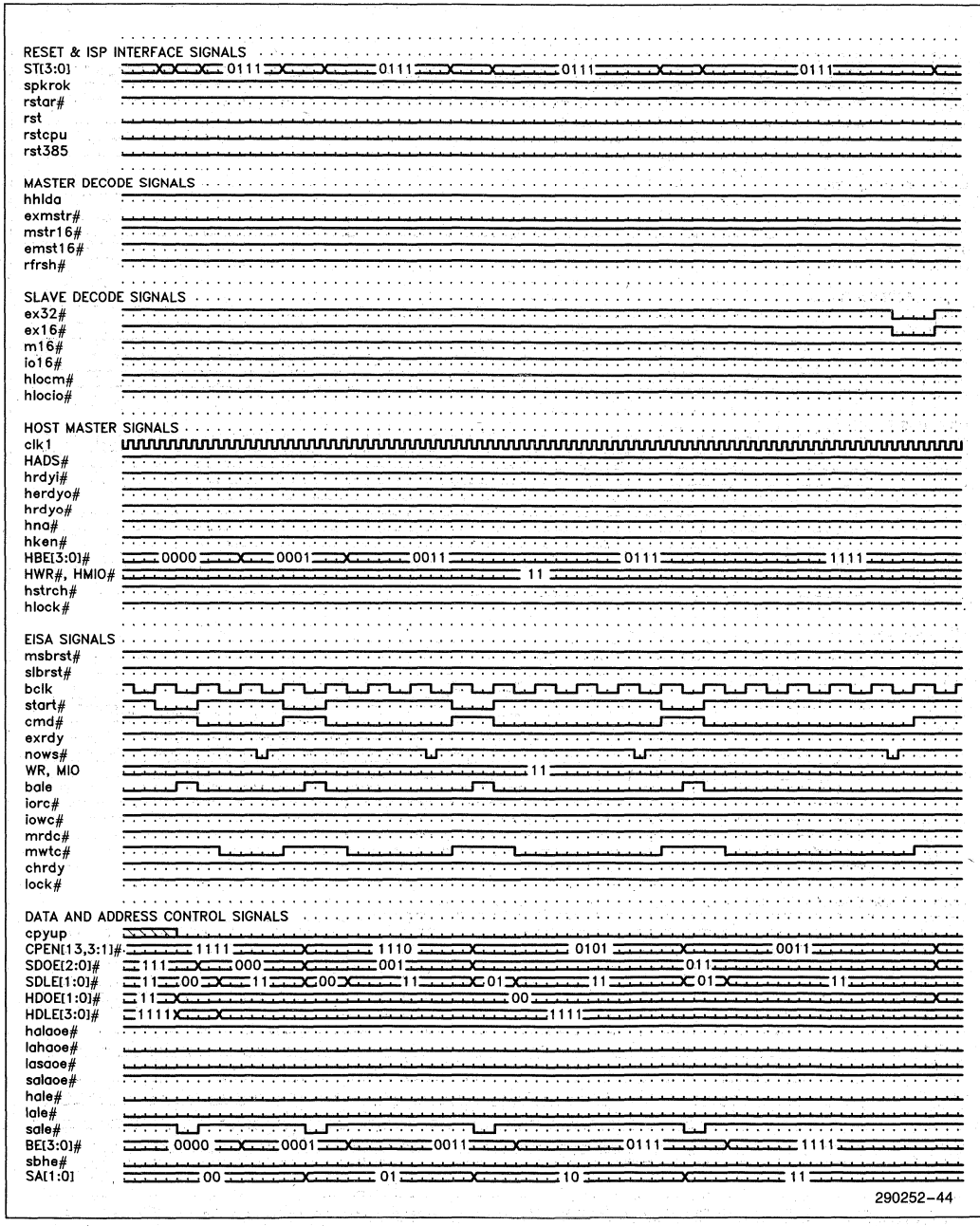


Figure 3-24. 32-Bit EISA Master to 8-Bit ISA Memory Slave NOWS# Assembly Write Cycle

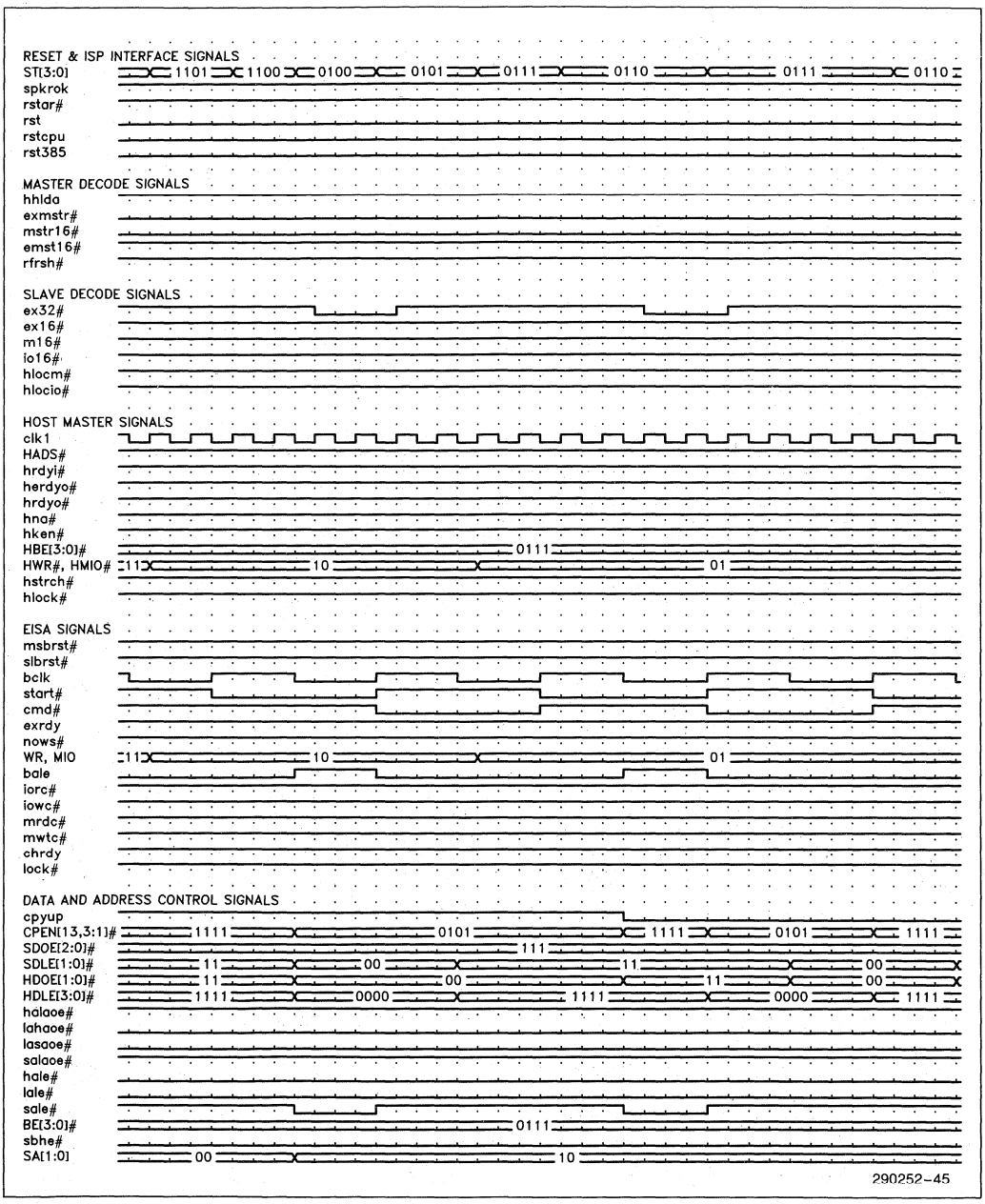


Figure 3-25. 16-Bit EISA Master to 32-Bit EISA Memory/IO Slave Byte Swap Read/Write Cycle

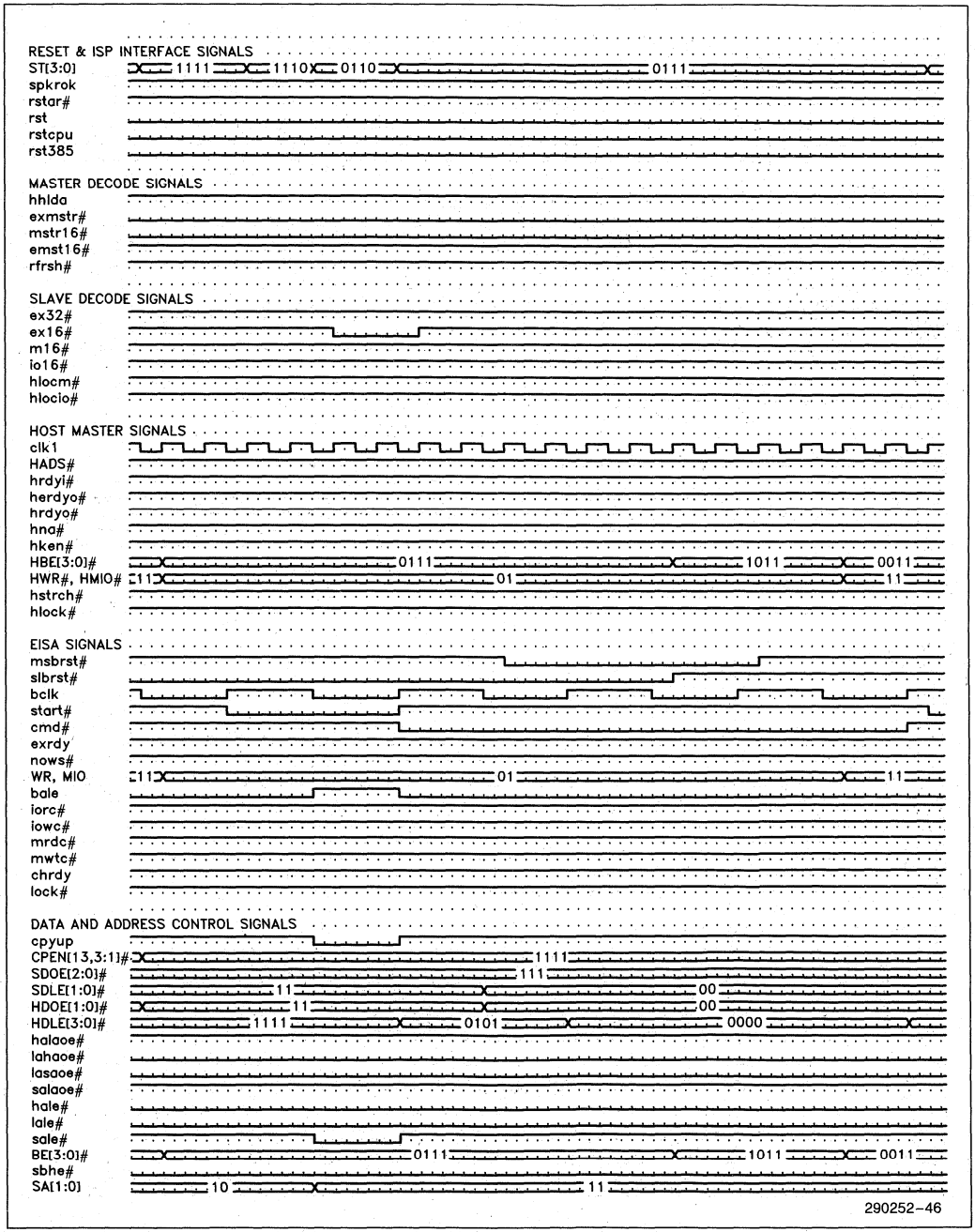


Figure 3-26. 16-Bit EISA Master to 16-Bit EISA Memory Slave Burst Read Cycle

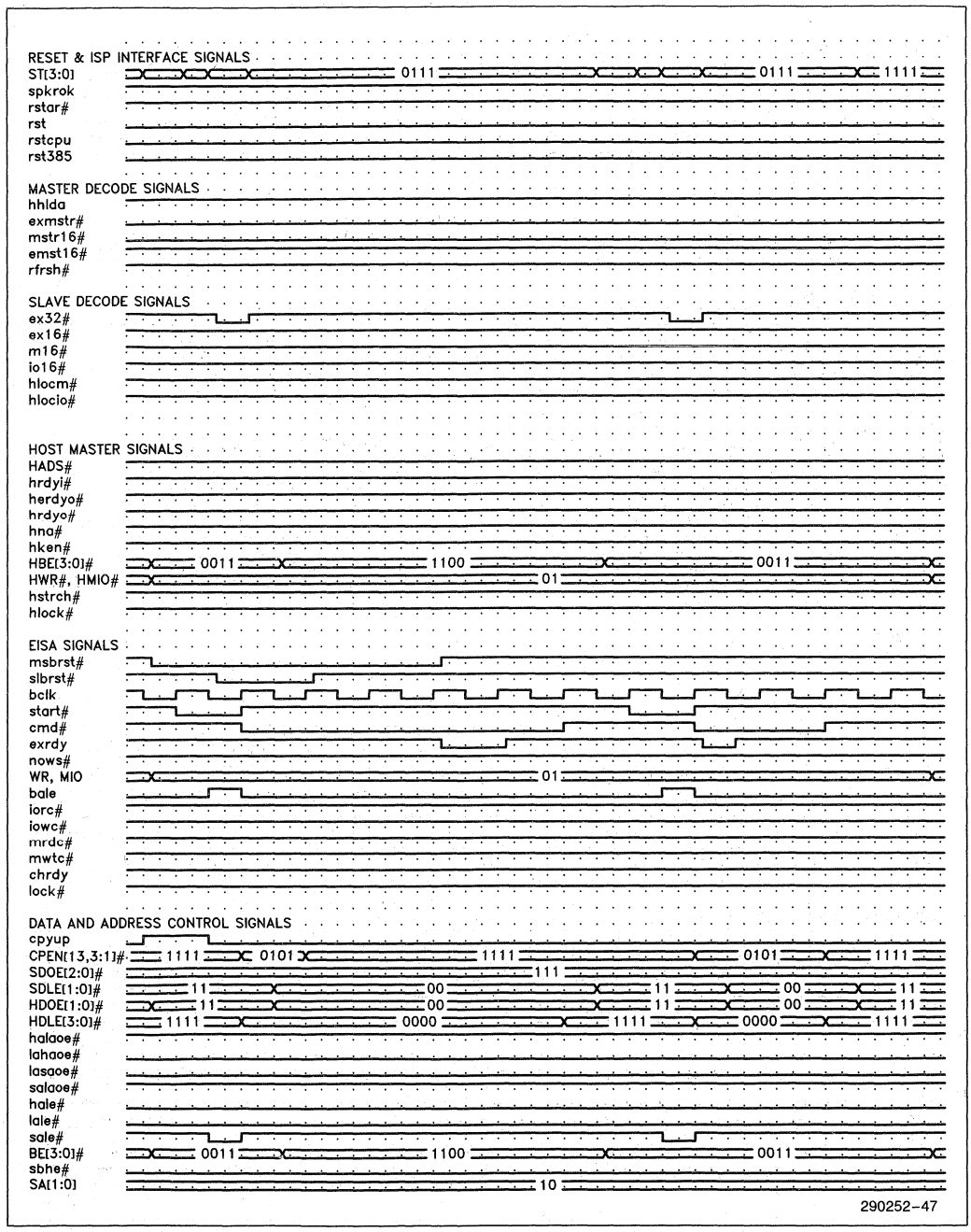
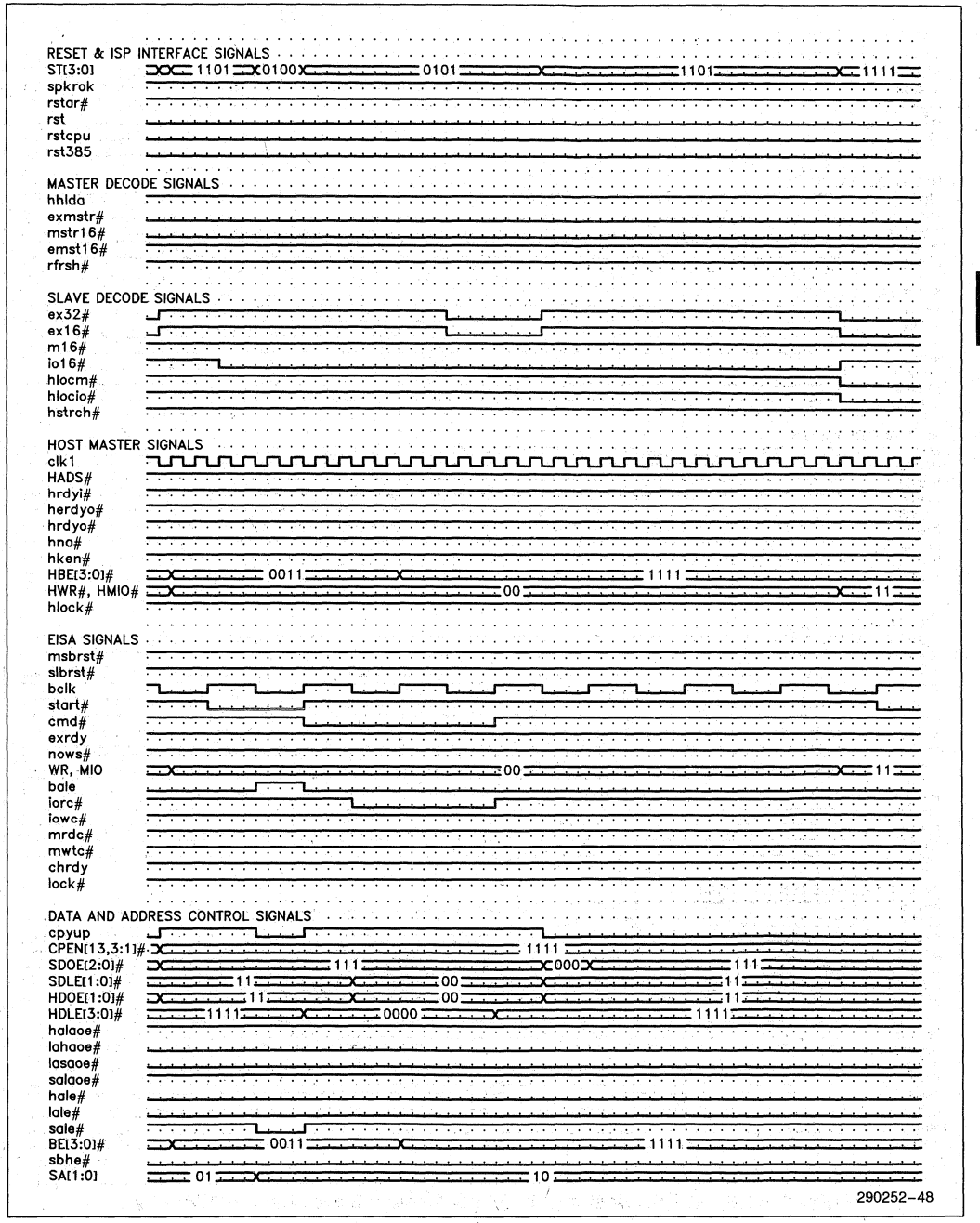


Figure 3-27. 16-Bit EISA Master to 32-Bit EISA Memory Slave 1 Wait State Byte Swap Burst Read Cycle



290252-48

Figure 3-28. 16-Bit EISA Master to 16-Bit ISA I/O No Assembly Read Cycle

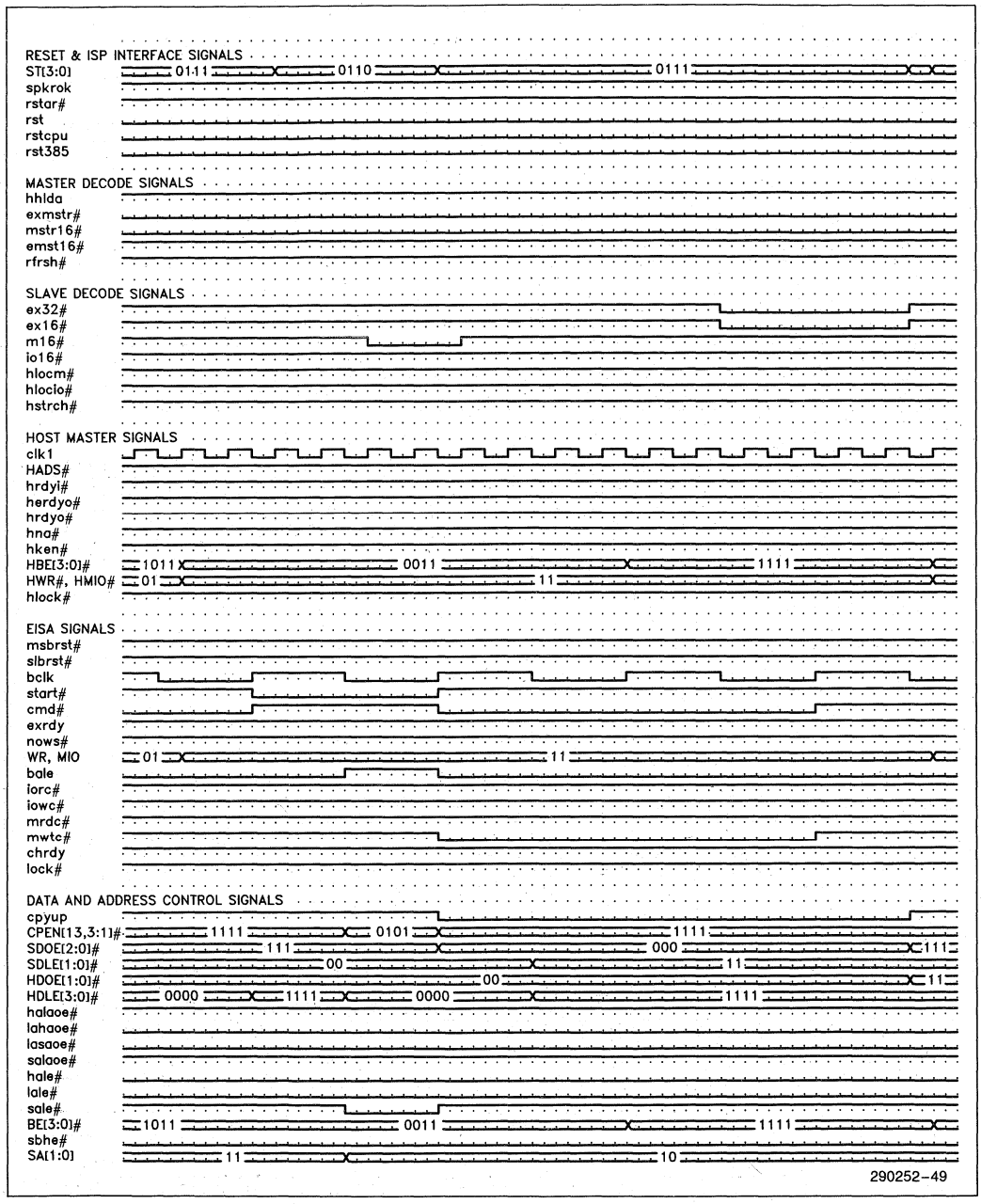


Figure 3-29. 16-Bit EISA Master to 16-Bit ISA Memory Slave Standard Write Cycle

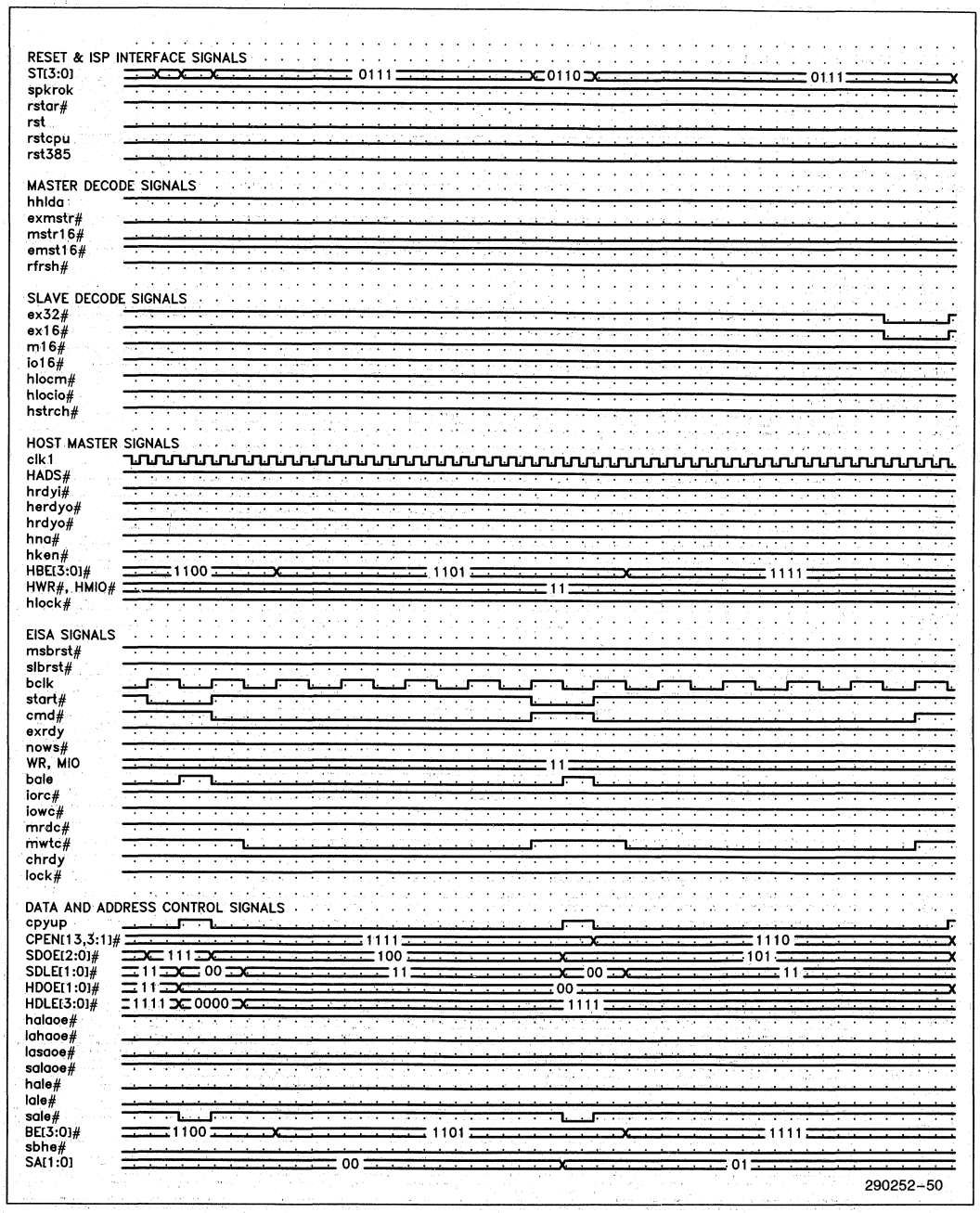


Figure 3-30. 16-Bit EISA Master to 8-Bit ISA Memory Slave Disassembly Write Cycle

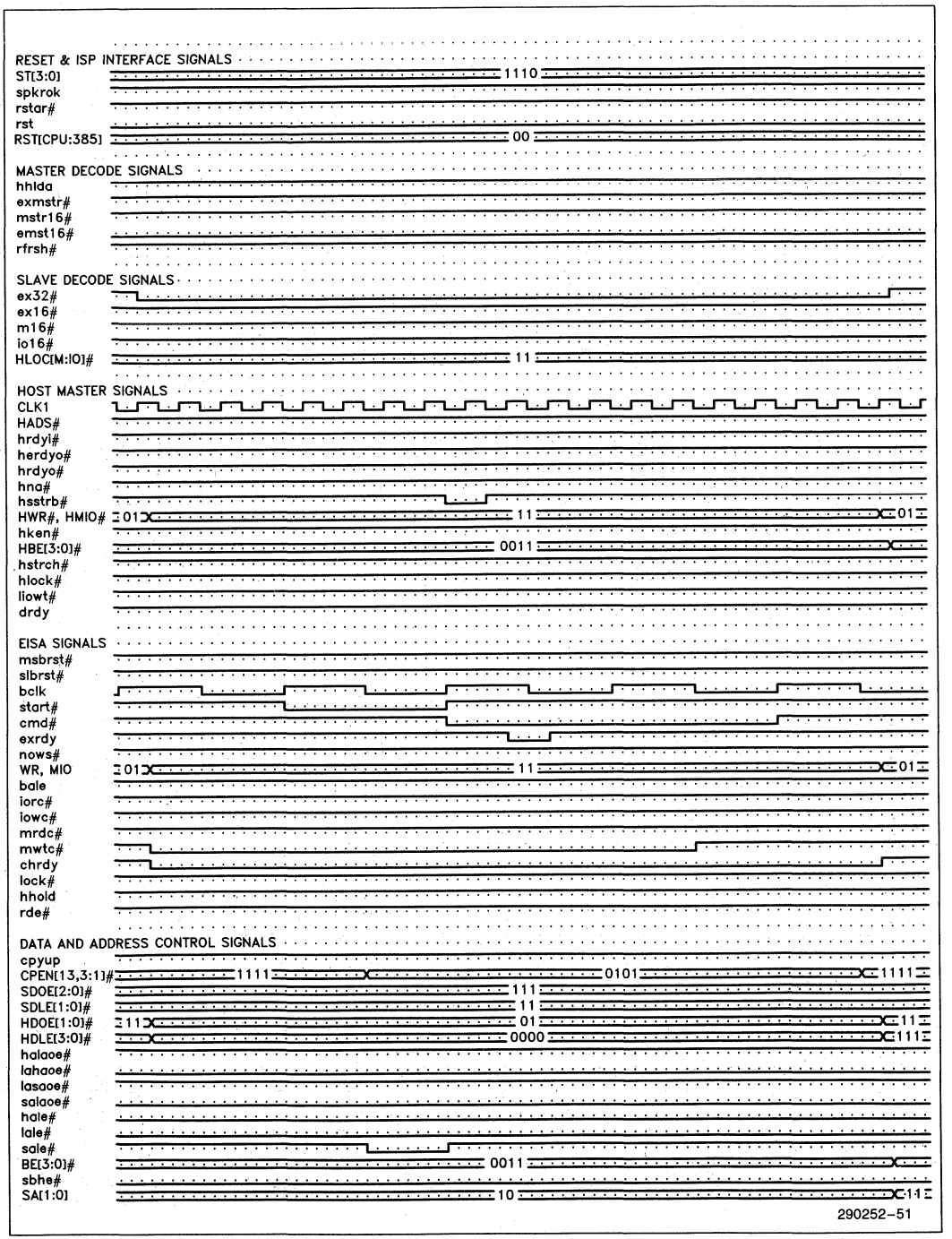


Figure 3-31. 16-Bit ISA Master to 32-Bit EISA Memory Slave 1 Wait State Write Cycle

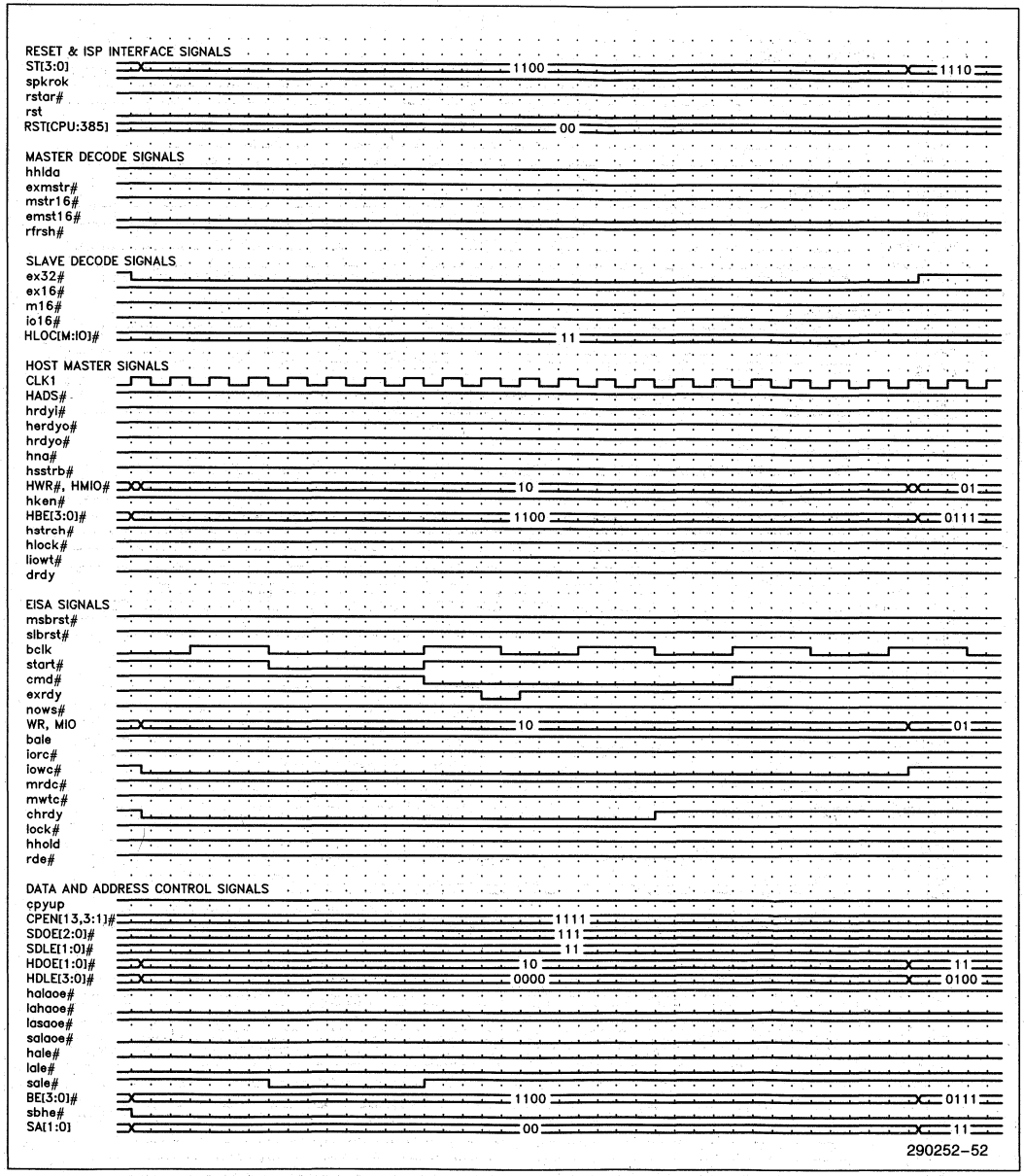


Figure 3-32. 16-Bit ISA Master to 32-Bit EISA I/O Slave Assembly Write Cycle with Wait States

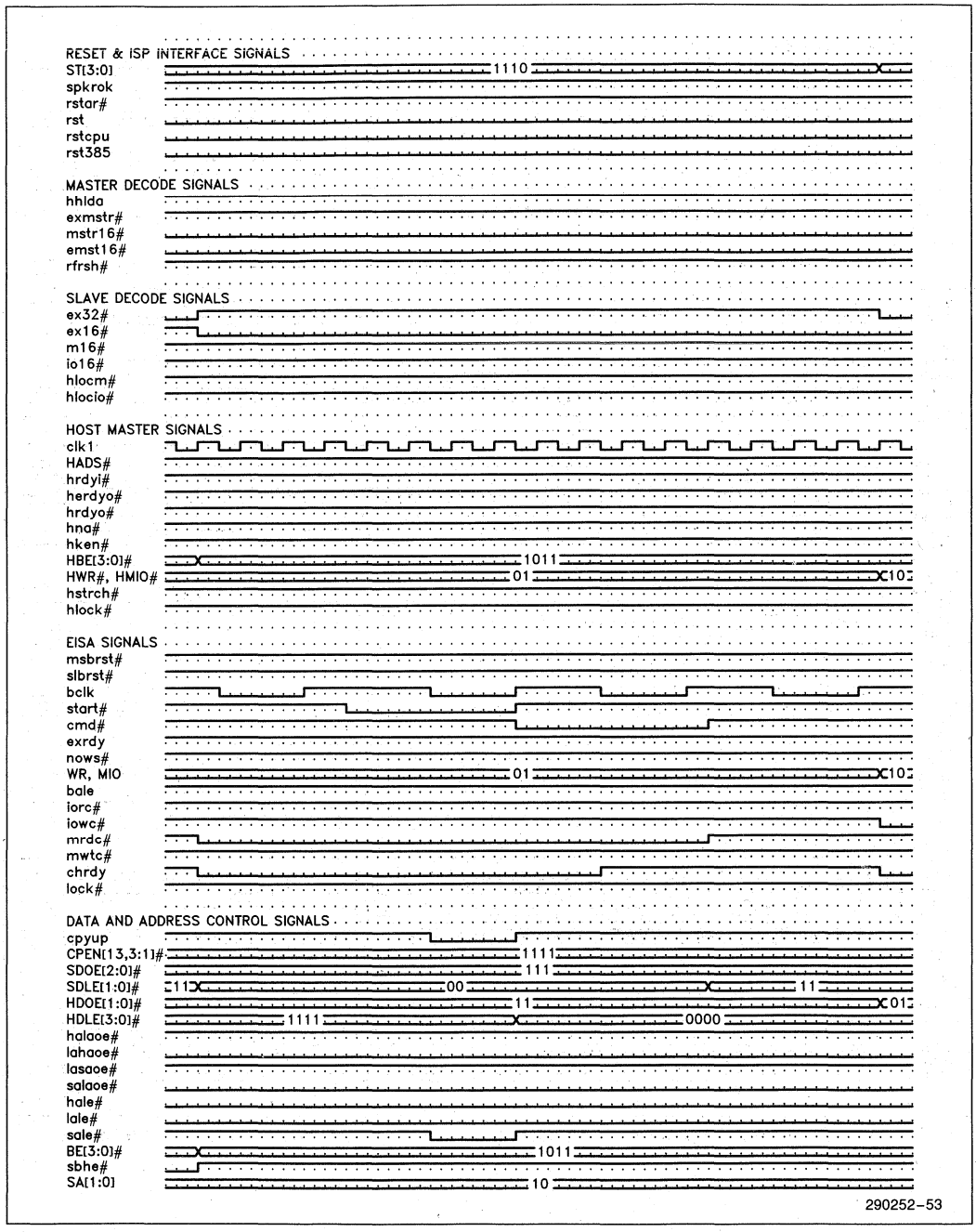


Figure 3-33. 16-Bit ISA Master to 16-Bit EISA Memory Slave BCLK Stretched Read Cycle

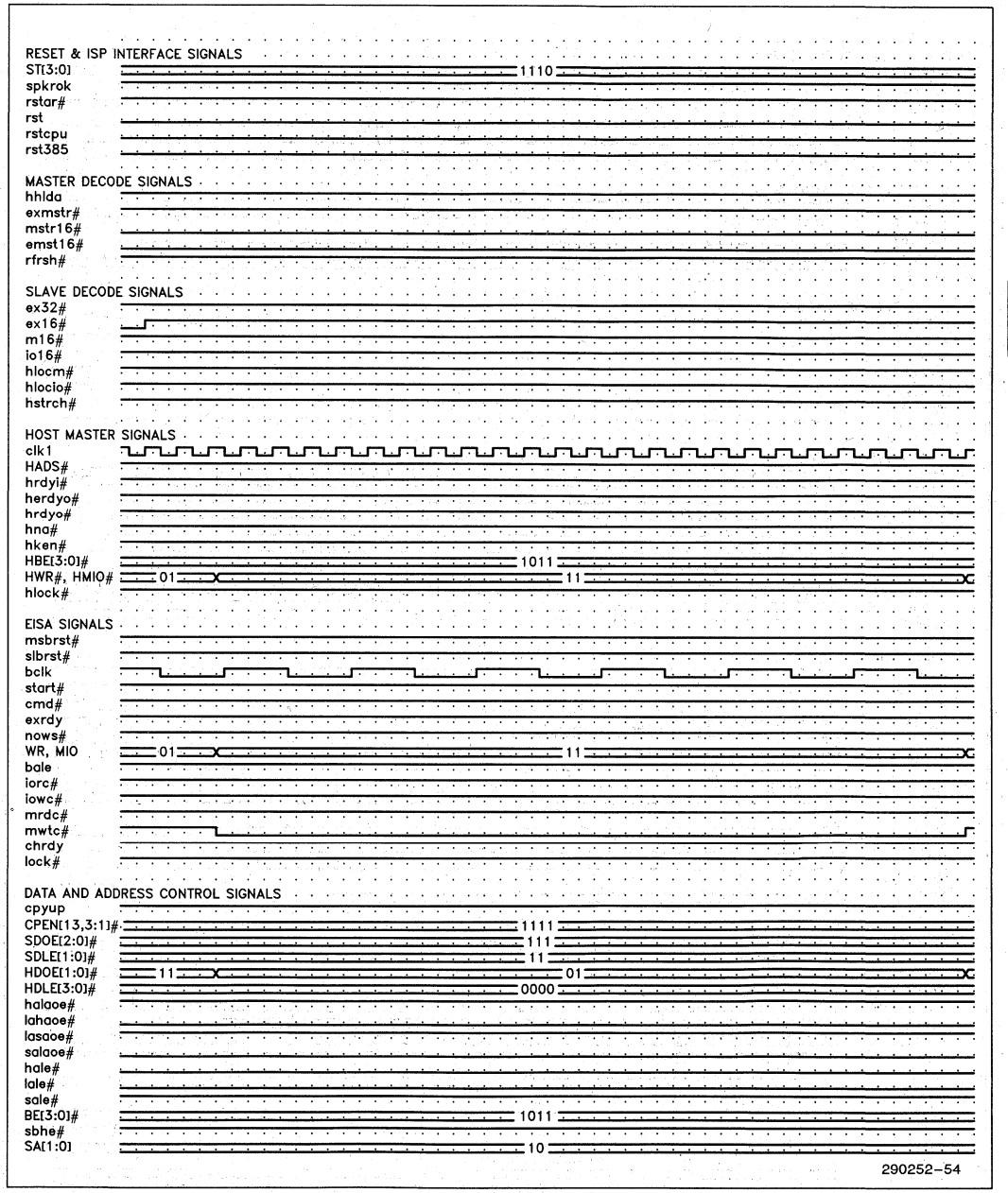


Figure 3-34. 16-Bit ISA Master to 8-Bit ISA Memory Slave Write Cycle

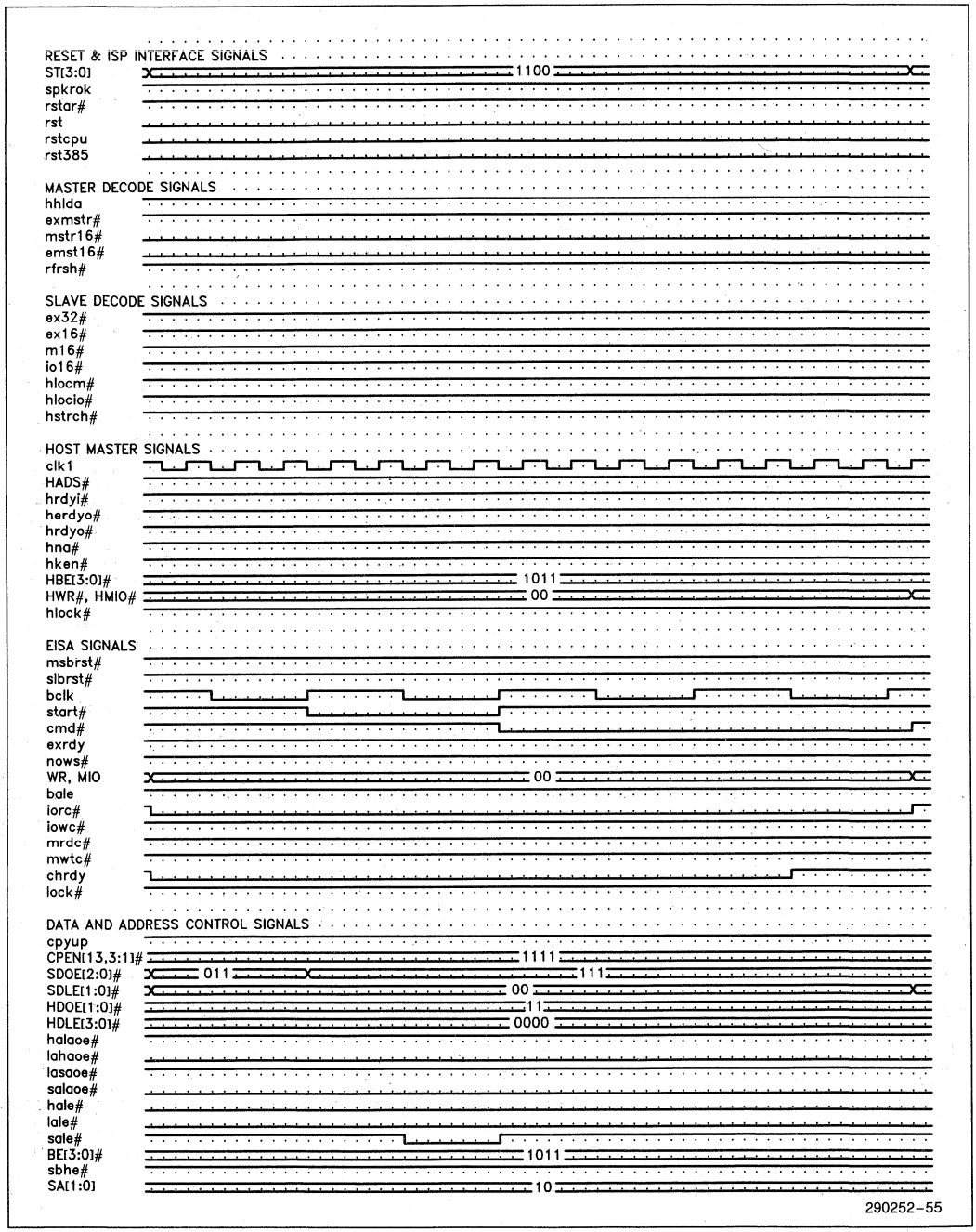


Figure 3-35. 16-Bit ISA to 8-Bit ISA I/O Slave Standard Read Cycle

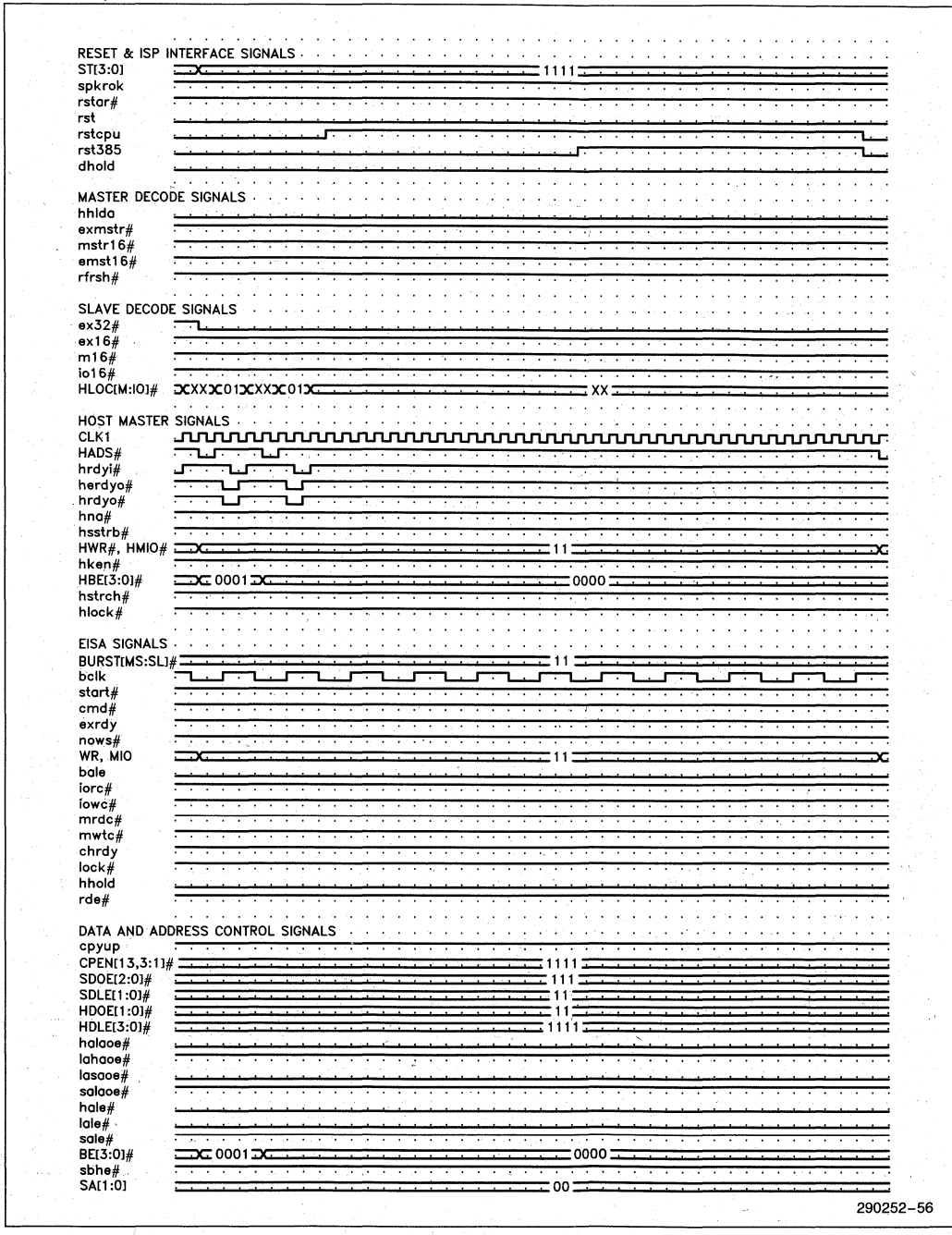


Figure 3-36. Halt/Shutdown Cycle

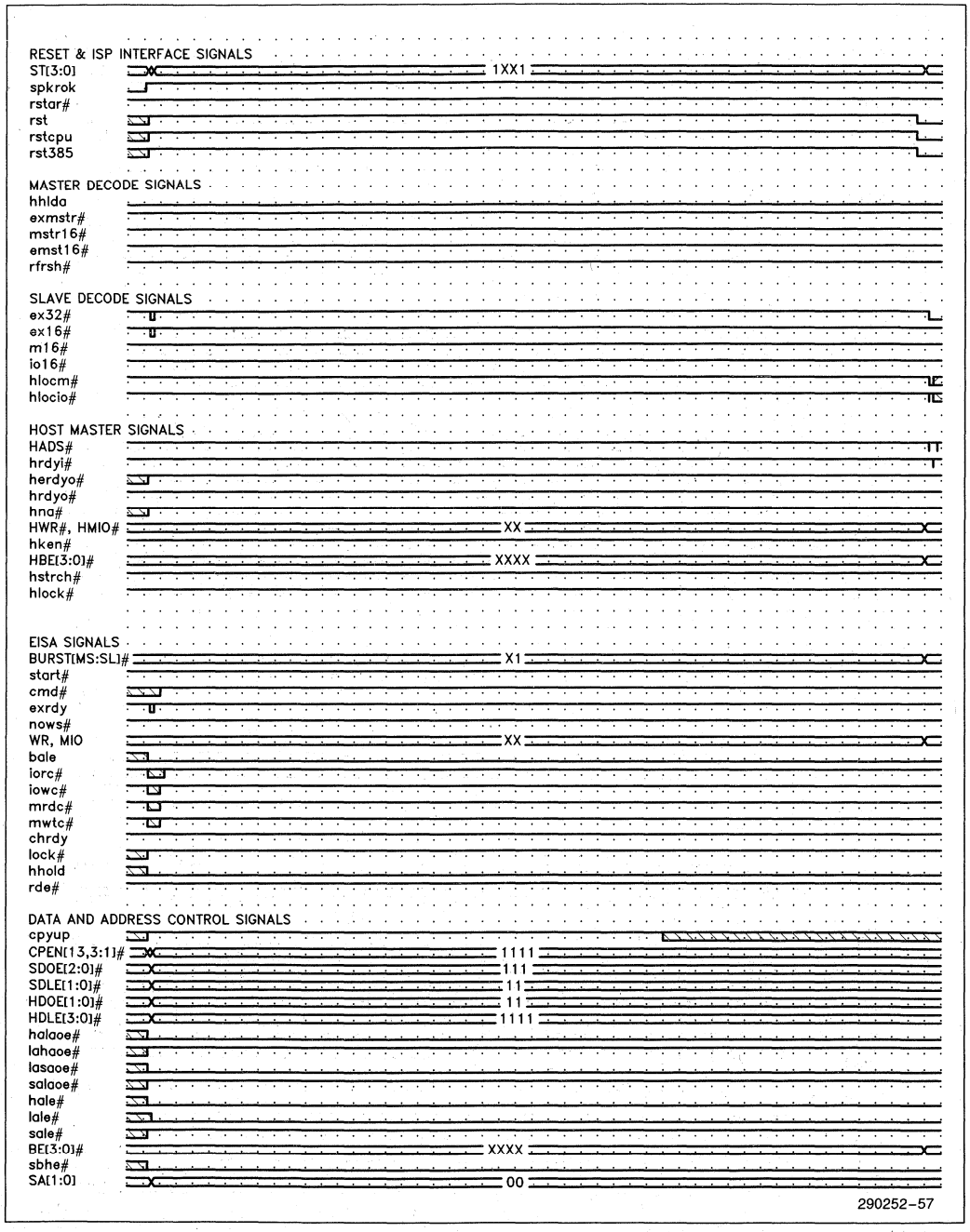


Figure 3-37. SPWROK

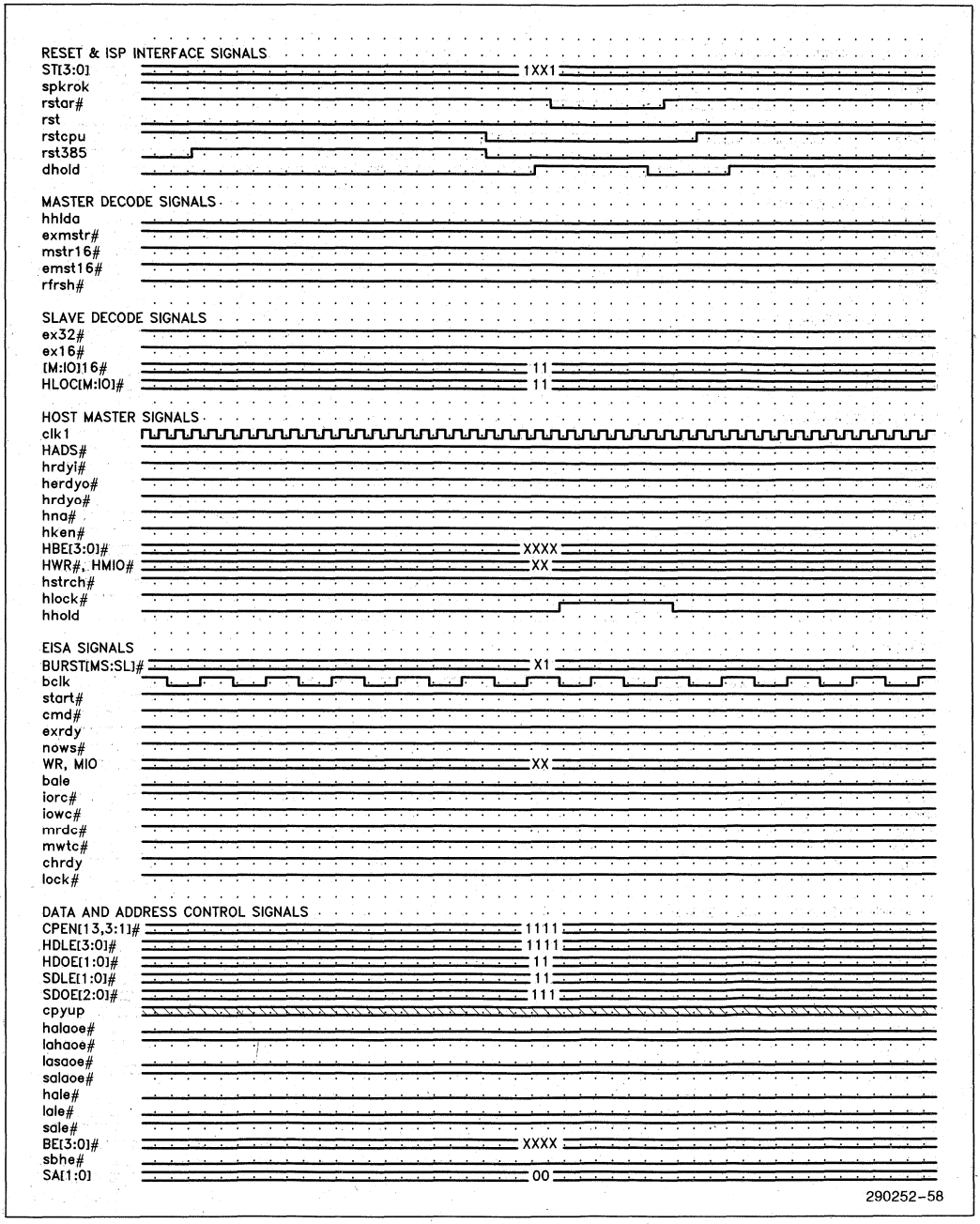


Figure 3-38. RSTAR# < 32 Clocks

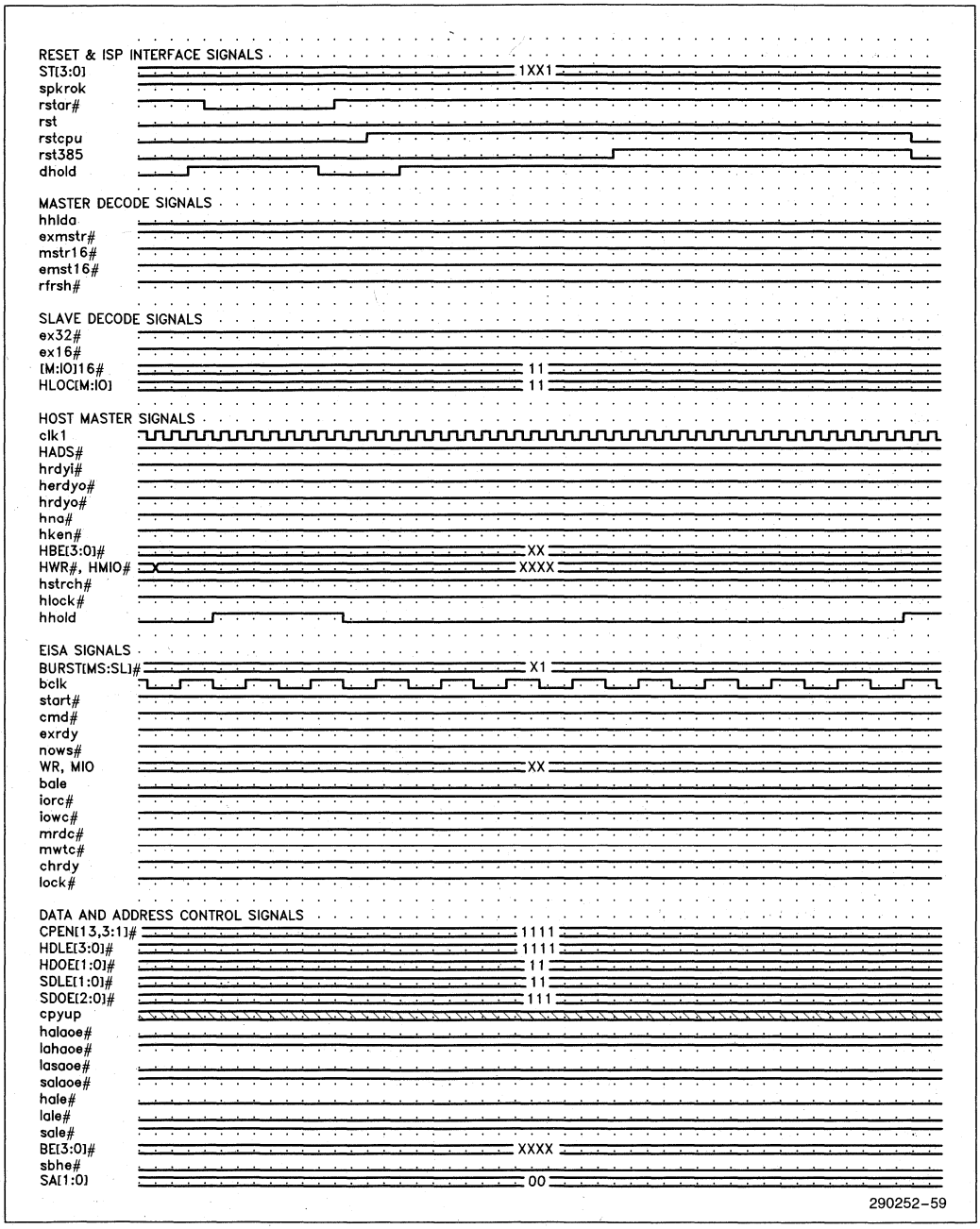
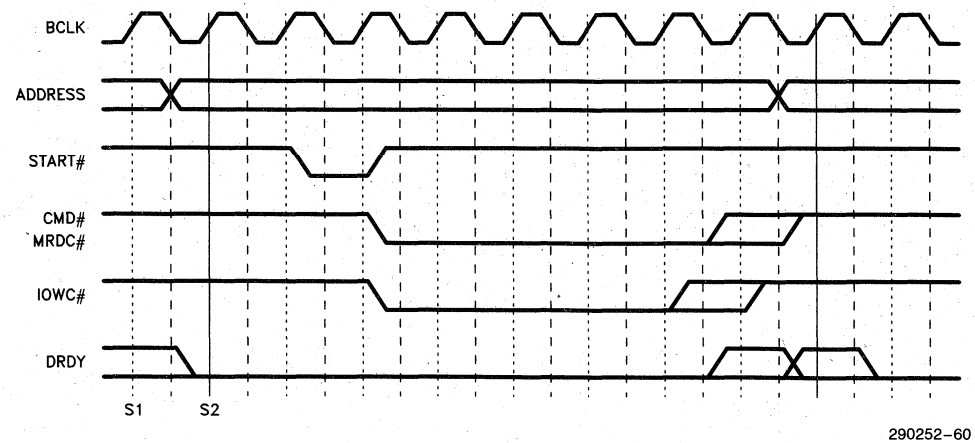
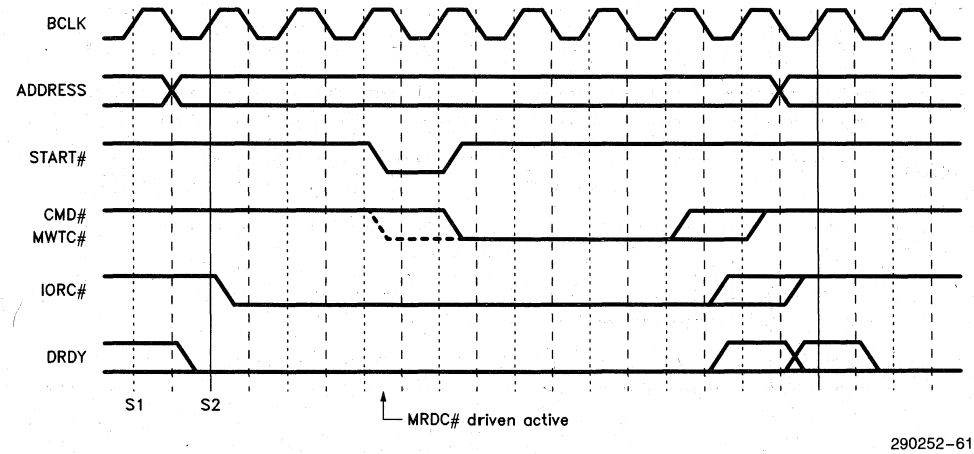


Figure 3-39. RSTCPU/HHOLD Signal Interlock



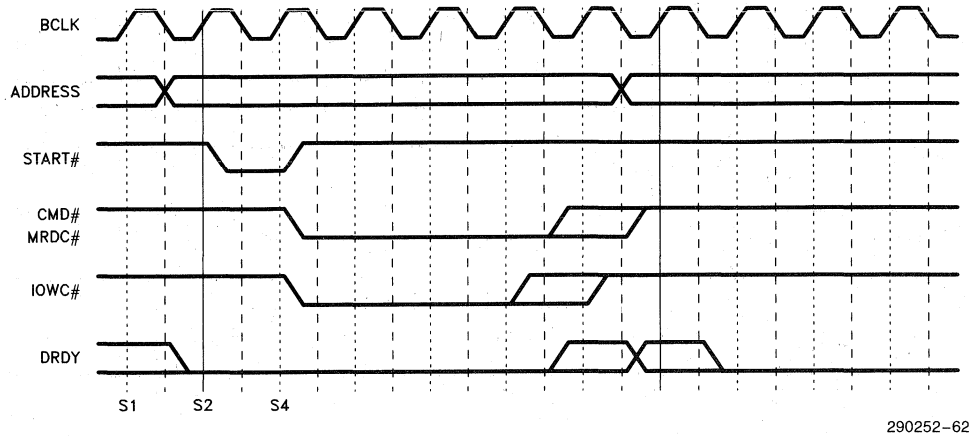
S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0] W-R

Figure 3-40. Compatible DMA Read Cycle (Memory Size .GE. I/O Size)



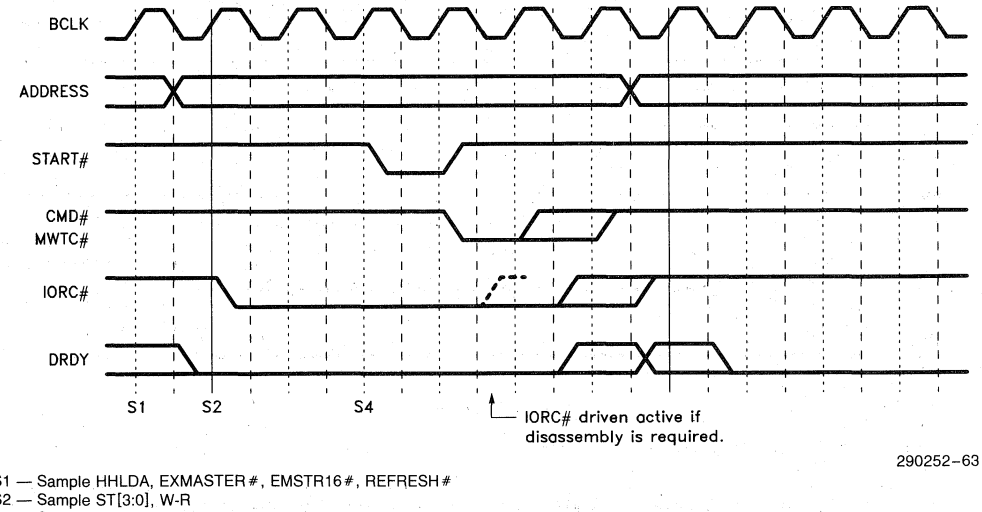
S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R

Figure 3-41. Compatible DMA Write Cycle (Memory Size .GE. I/O Size)



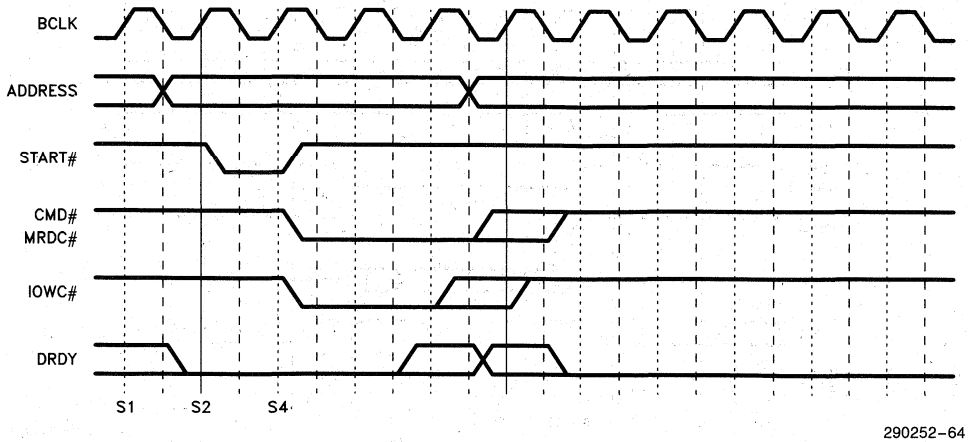
S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R
S4 — Sample EX32#, EX16#, M16#, HLOCMEM#

Figure 3-42. Type A DMA Read Cycle (Memory Size .GE. I/O Size)



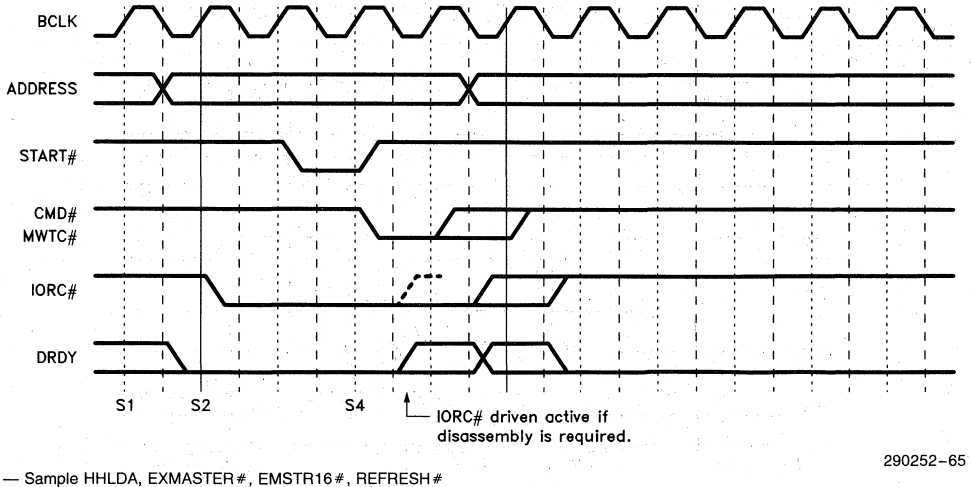
S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R
S4 — Sample EX32#, EX16#, M16#, HLOCMEM#

Figure 3-43. Type A DMA Write Cycle (Memory Size .GE. I/O Size)



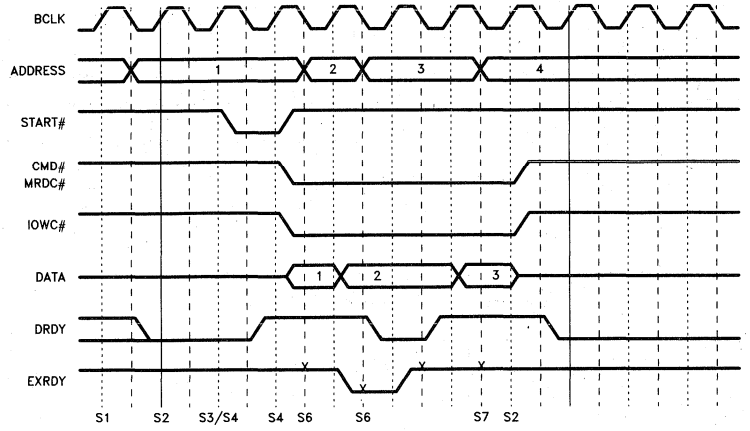
S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R
S4 — Sample EX32#, EX16#, M16#, HLOCMEM#

Figure 3-44. Type B DMA Read Cycle (Memory Size .GE. I/O Size)



S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R
S4 — Sample EX32#, EX16#, M16#, HLOCMEM#

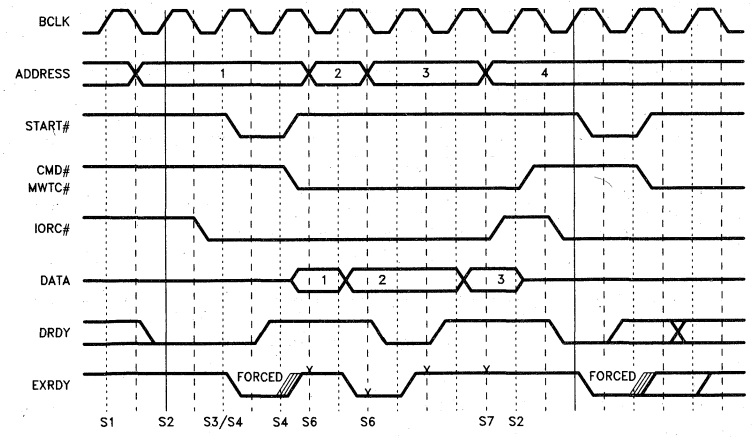
Figure 3-45. Type B DMA Write Cycle (Memory Size .GE. I/O Size)



290252-66

S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R
S3 — Sample SLBURST
S4 — Sample EX32#, EX16#, M16#, HLOCMEM#
S6 — No Change in Status detected
S7 — Change in Status detected (IORC# de-asserted on BCLK's falling edge)
X = EBC sampling points

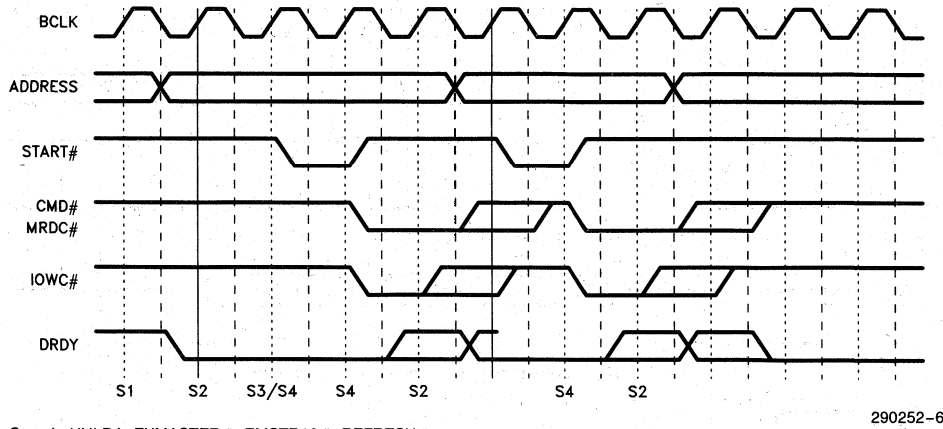
Figure 3-46. Burst DMA Read Cycle (Memory Size .GE. I/O Size)



290252-67

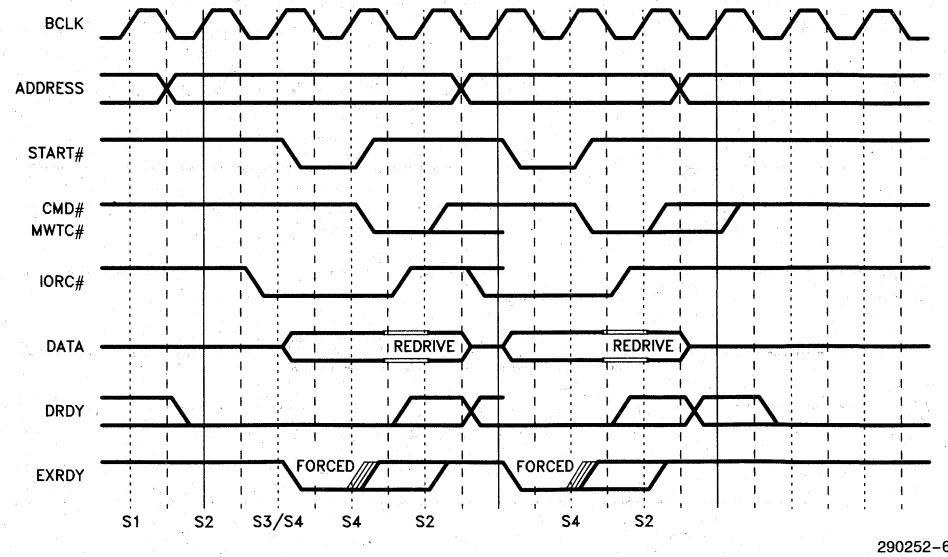
S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R
S3 — Sample SLBURST
S4 — Sample EX32#, EX16#, M16#, HLOCMEM#
S6 — No Change in Status detected
S7 — Change in Status detected (IORC# de-asserted on BCLK's falling edge)
X = EBC sampling points

Figure 3-47. Burst DMA Write Cycle (Memory Size .GE. I/O Size)



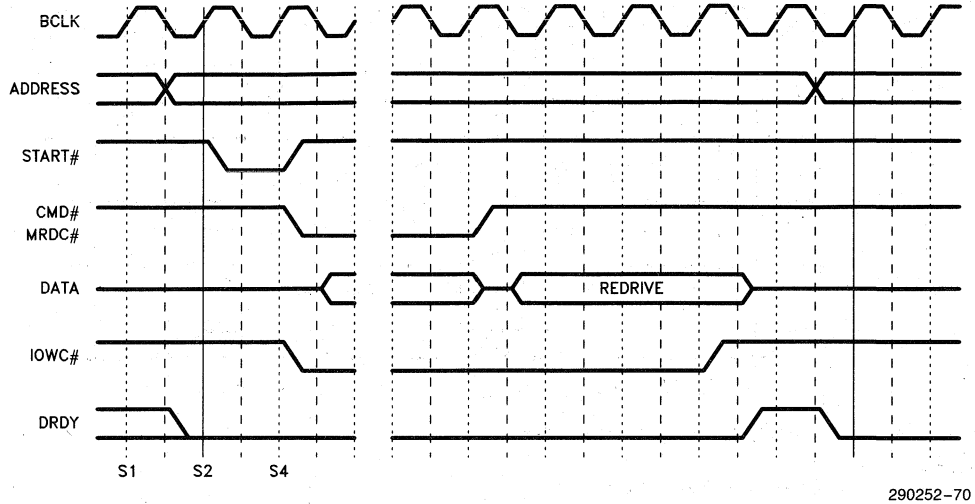
S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R
S3 — Sample SLBURST
S4 — Sample EX32#, EX16#, M16#, HLOCMEM#

Figure 3-48. Burst DMA Read Cycle to Non-Burst Memory (Memory Size .GE. I/O Size)



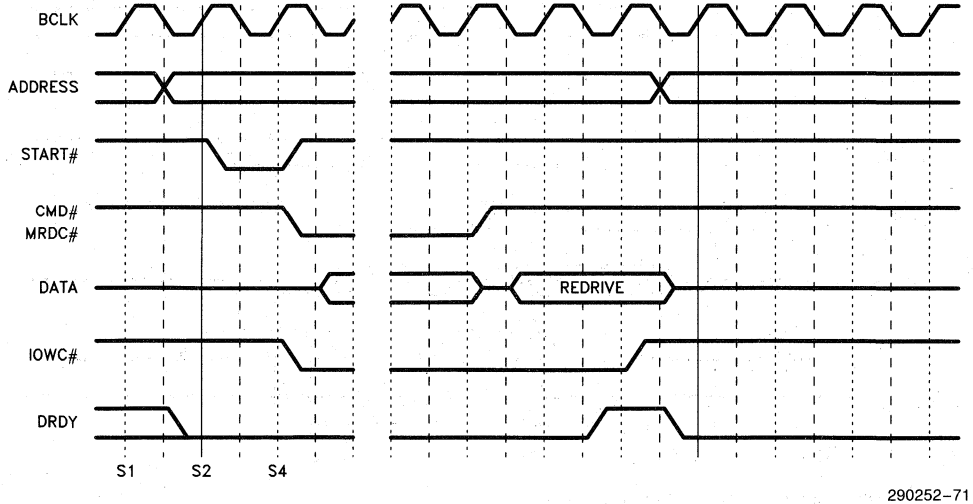
S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R
S3 — Sample SLBURST
S4 — Sample EX32#, EX16#, M16#, HLOCMEM#

Figure 3-49. Burst DMA Write Cycle to Non-Burst Memory (Memory Size .GE. I/O Size) Burst DMA Write (Memory Size .LT. I/O Size)



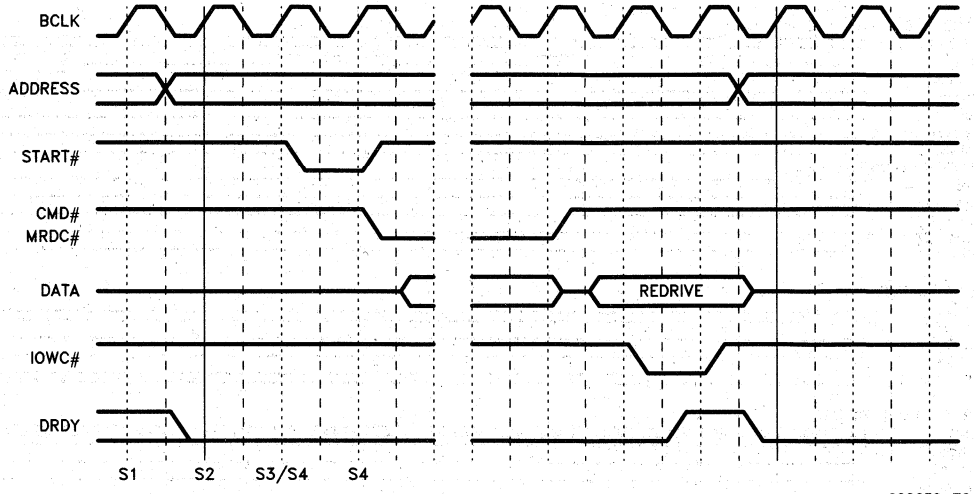
S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R
S4 — Sample EX32#, EX16#, M16#, HLOCMEM#

Figure 3-50. Type A DMA Read Cycle (Memory Size .LT. I/O Size)



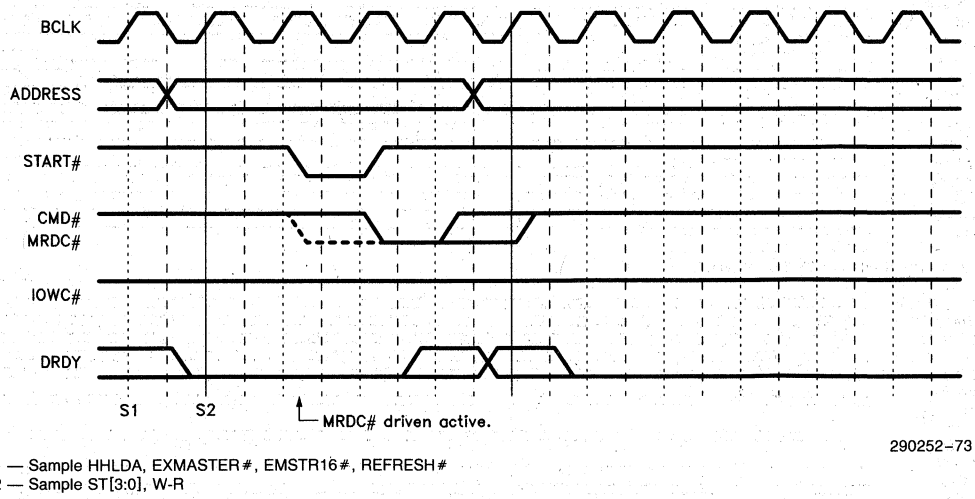
S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R
S4 — Sample EX32#, EX16#, M16#, HLOCMEM#

Figure 3-51. Type B Read Cycle (Memory Size .LT. I/O Size)



S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R
S3 — Sample SLBURST
S4 — Sample EX32#, EX16#, M16#, HLOCMEM# (Assembly Required detected)

Figure 3-52. Burst DMA Read with Assembly Required (Memory Size .LT. I/O Size)



S1 — Sample HHLDA, EXMASTER#, EMSTR16#, REFRESH#
S2 — Sample ST[3:0], W-R

Figure 3-53. Refresh Cycle

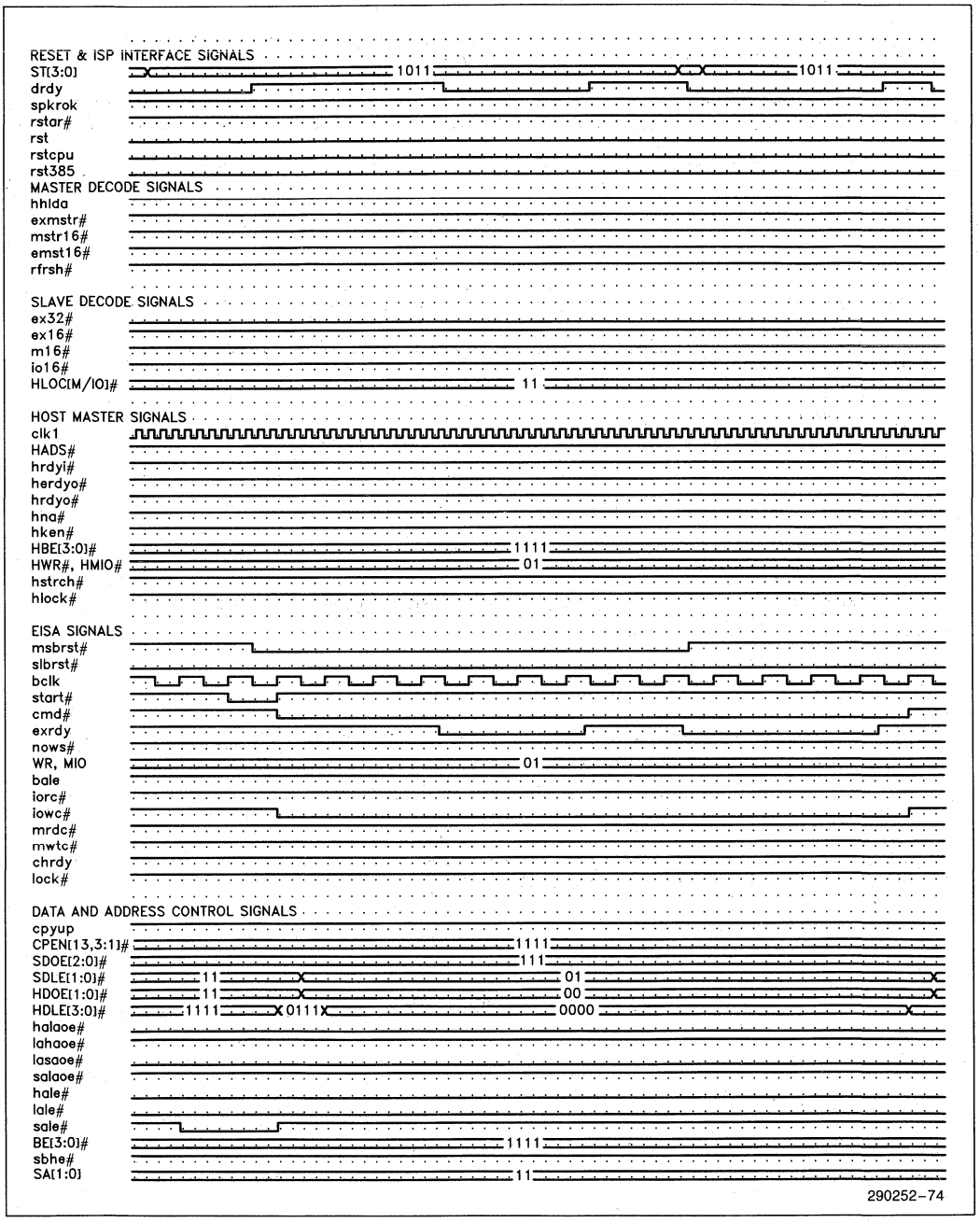


Figure 3-54. 32-Bit I/O to 32-Bit Memory Burst Read DMA Cycle

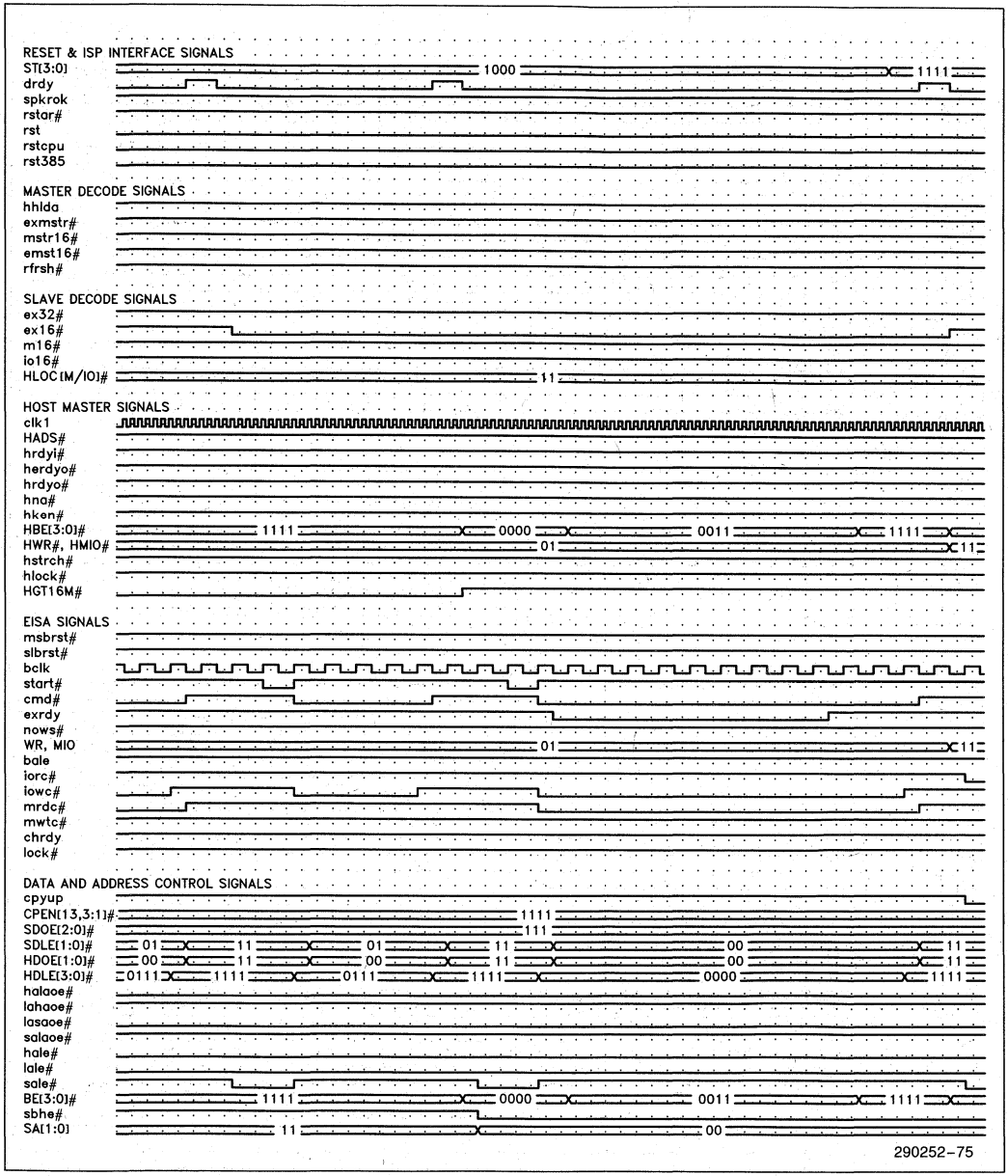


Figure 3-55. 32-Bit I/O to 16-Bit Memory Compatible Read DMA Cycle

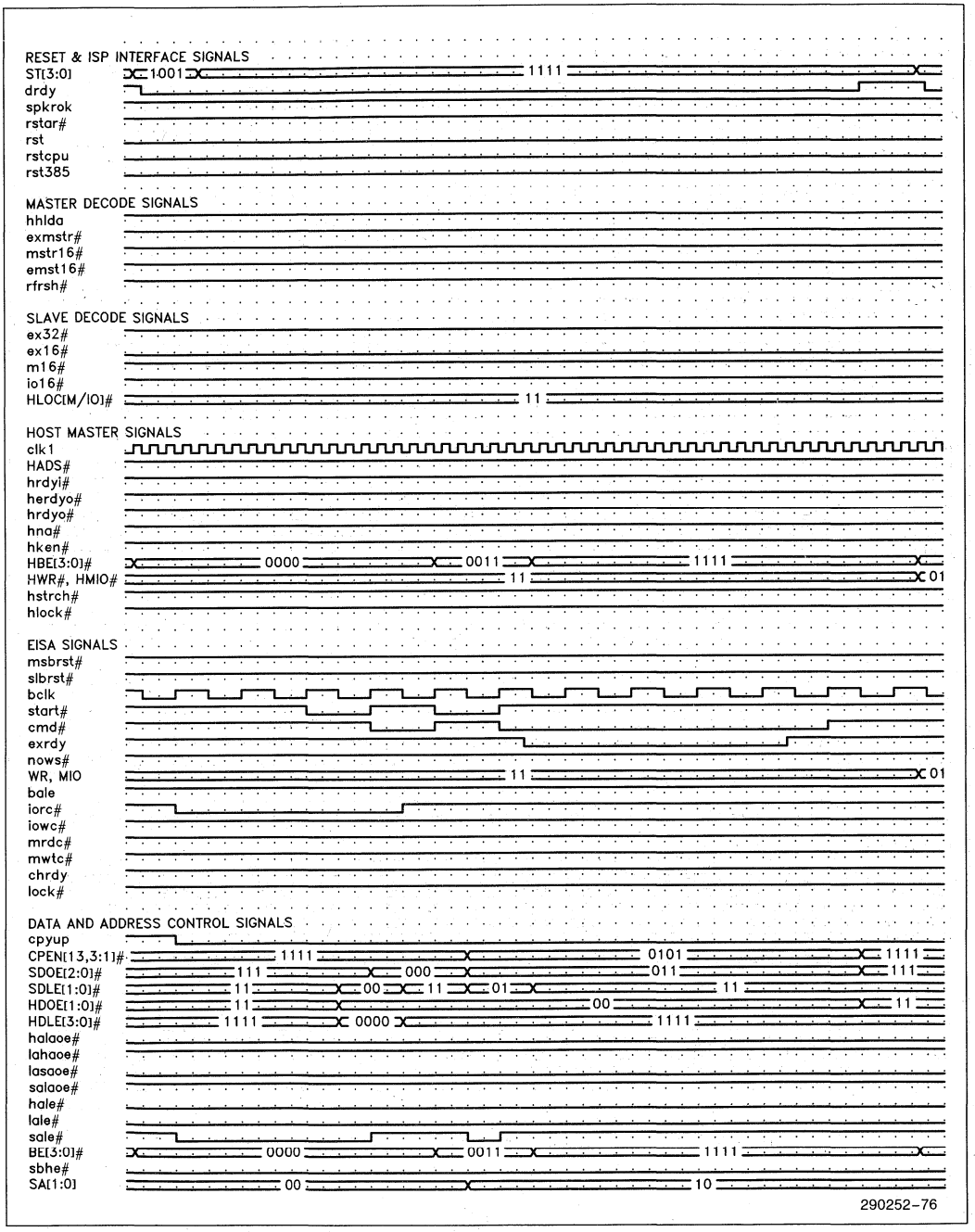


Figure 3-56. 32-Bit I/O to 16-Bit Memory Type A Write DMA Cycle

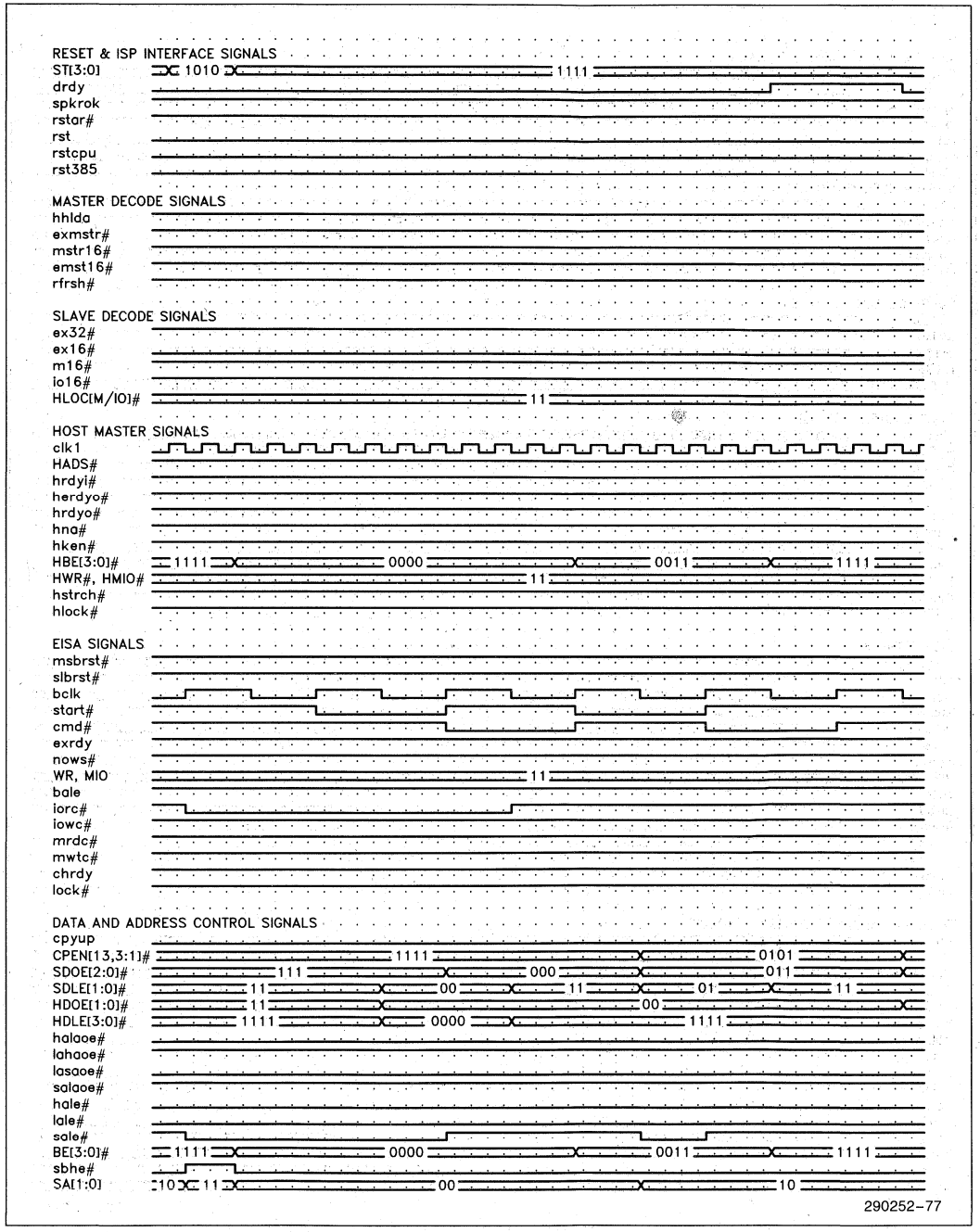


Figure 3-57. 32-Bit I/O to 16-Bit Memory Type B Write DMA Cycle

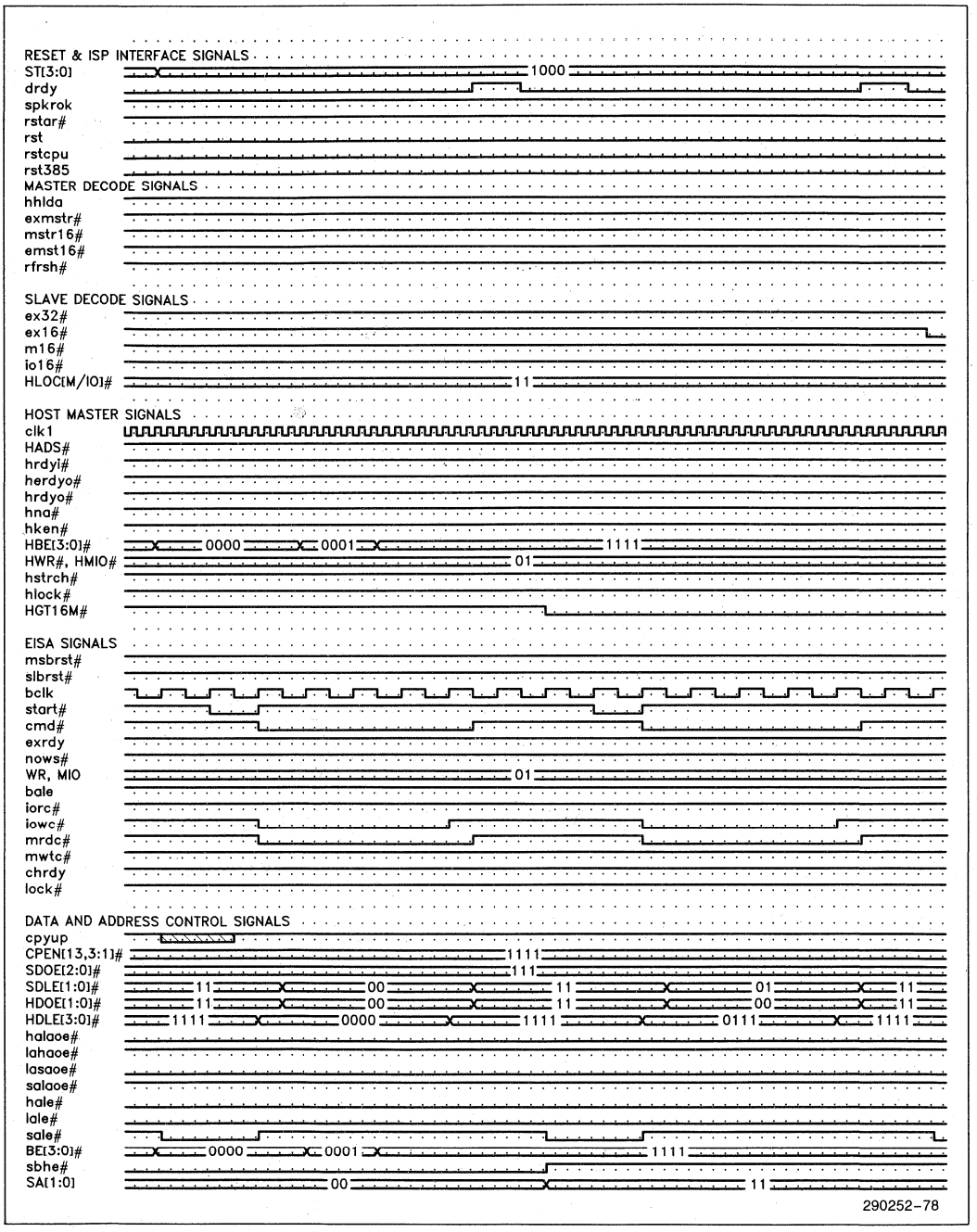


Figure 3-58. 32-Bit I/O to 8-Bit Memory Compatible Read DMA Cycle

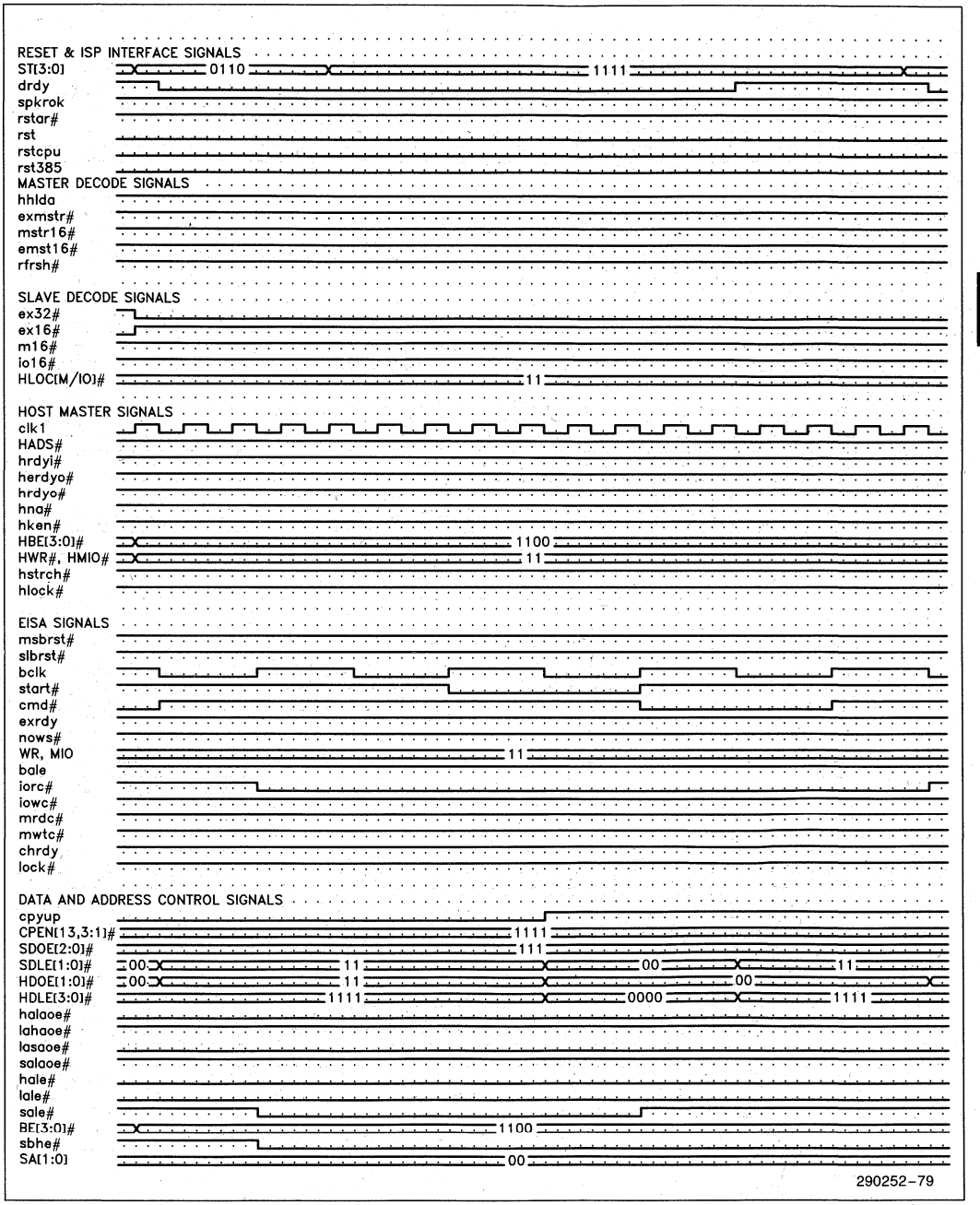


Figure 3-59. 16-Bit I/O to 32-Bit Memory Type B Write DMA Cycle

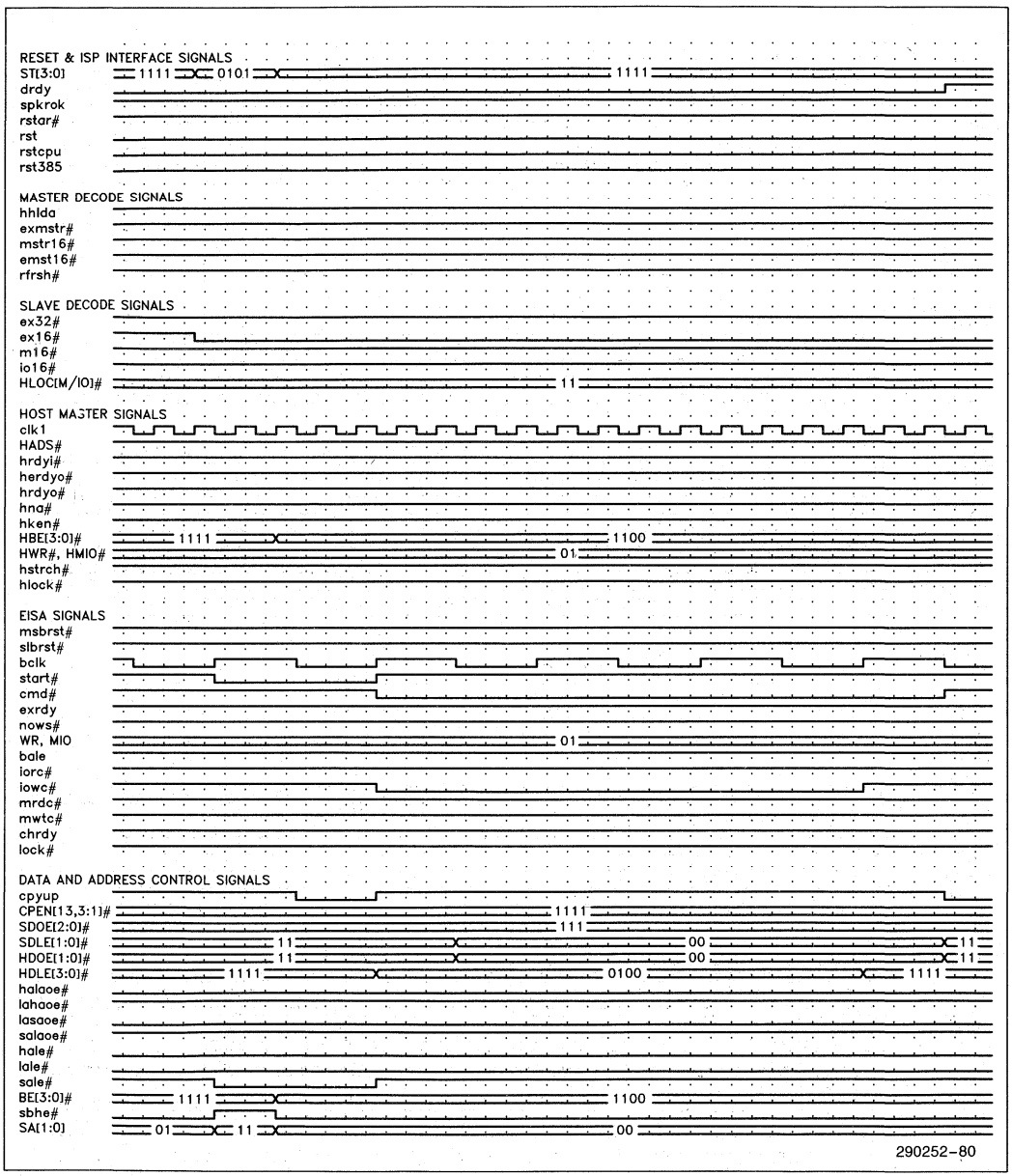


Figure 3-60. 16-Bit I/O to 16-Bit Memory Type A Read DMA Cycle

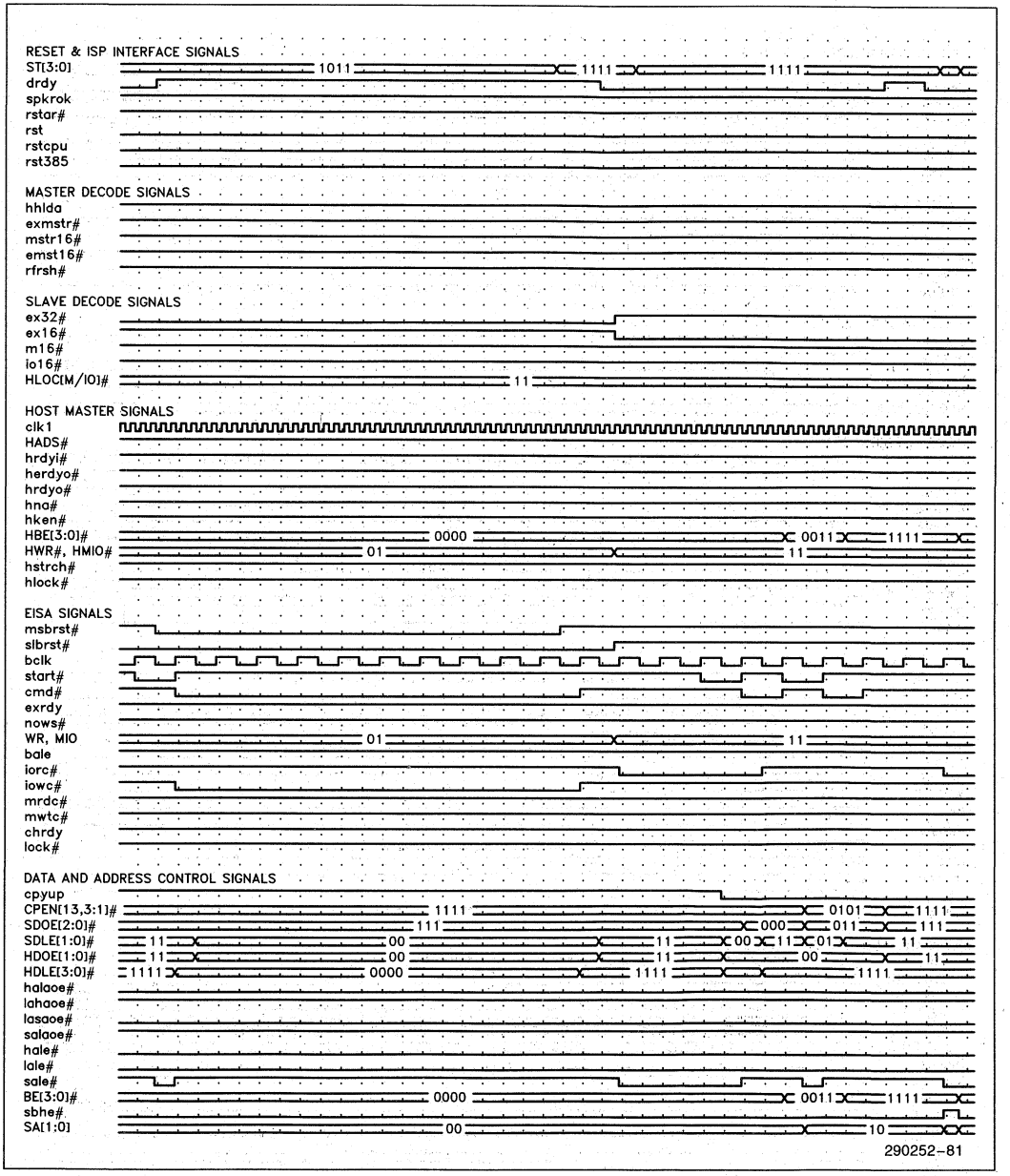


Figure 3-61. 16-Bit I/O to 32-Bit Memory Burst Read DMA Cycle

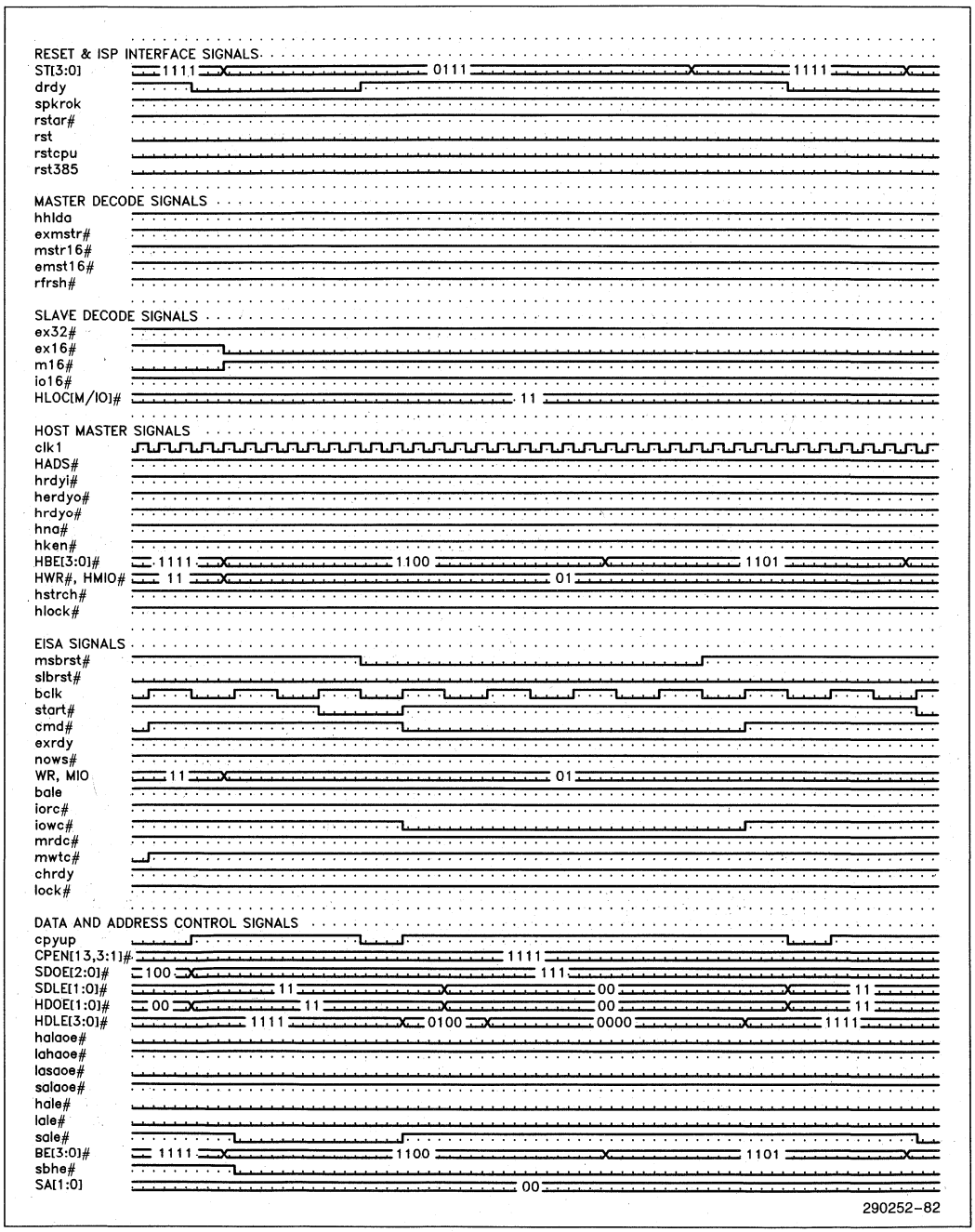


Figure 3-62. 16-Bit I/O to 16-Bit Memory Burst Read DMA Cycle

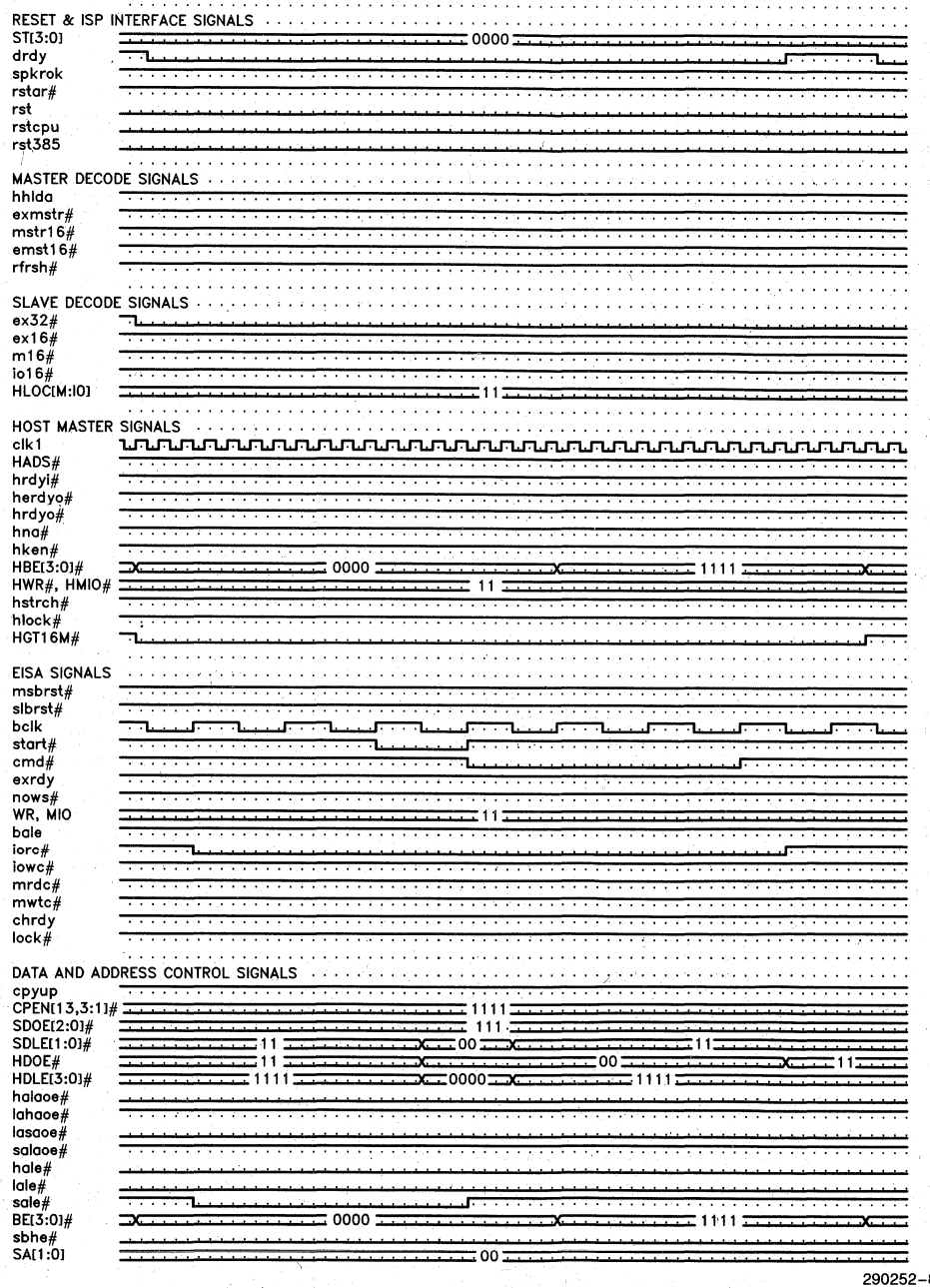


Figure 3-63. 8-Bit I/O to 32-Bit Memory Compatible Write DMA Cycle

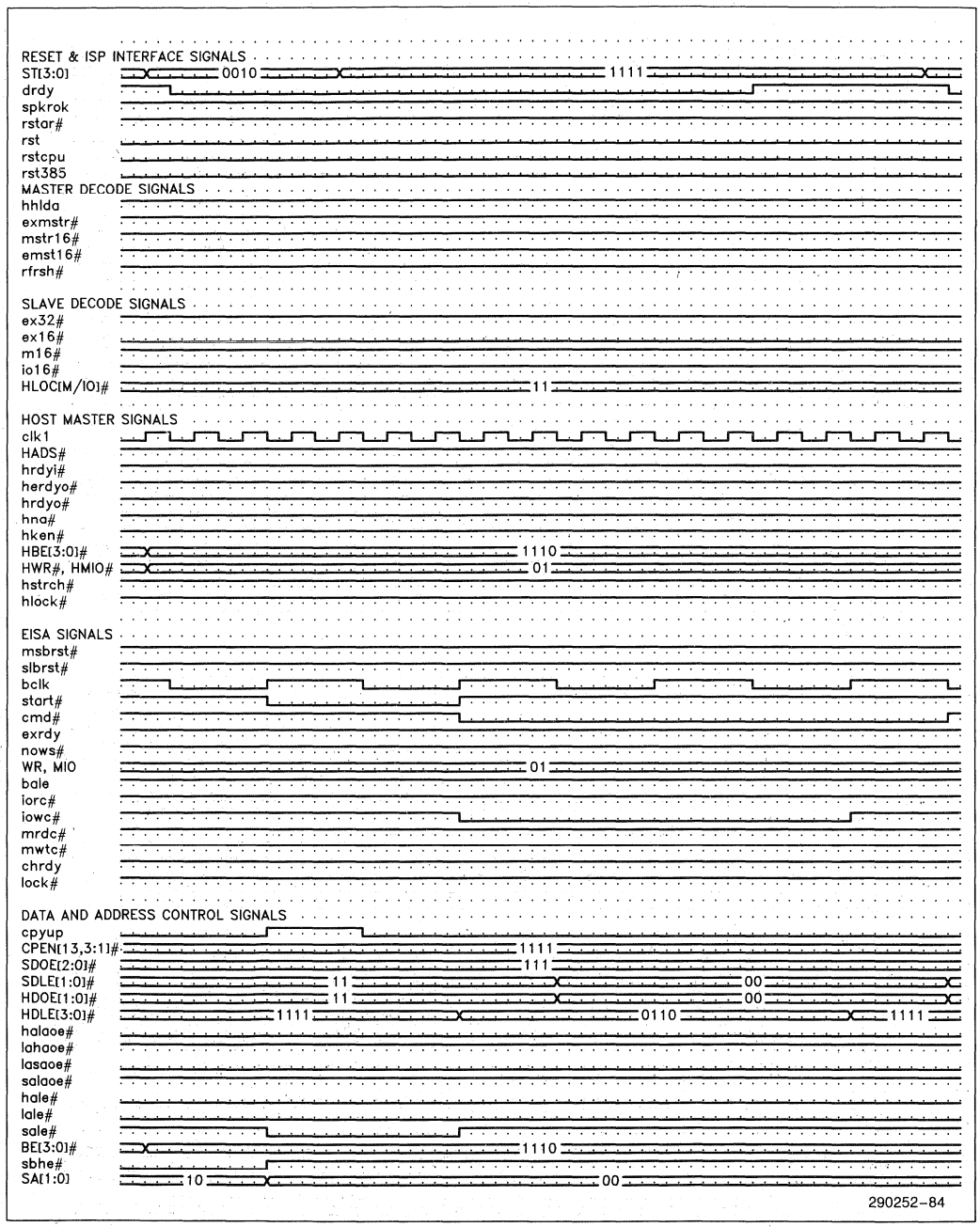


Figure 3-64. 8-Bit I/O to 16-Bit Memory Read Type B DMA Cycle

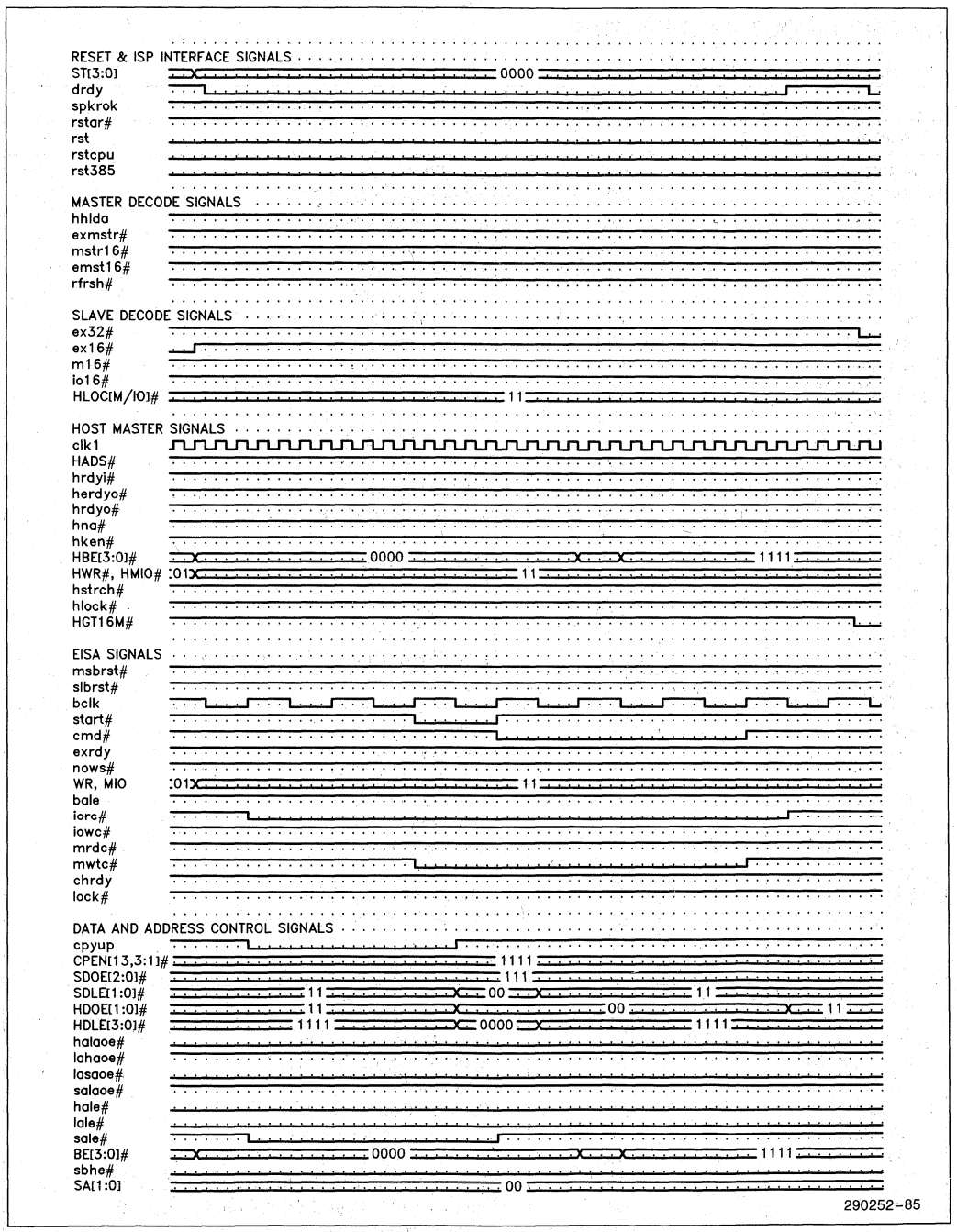


Figure 3-65. 8-Bit I/O to 8-Bit Memory Write Compatible DMA Cycle

4.0 EBC PINOUT INFORMATION

EBC Bus Controller (132 Pins)

Name	Type	Pin	Description
Host Bus Signals (23 Pins)			
HCLKCPU	I	89	Host CPU Clock
HADS0#	I	72	Host Address Status
HADS1#	I	128	Host Address Status
HBE3#	B	118	Host Byte Enables
HBE(2:0)#	B	120:122	Host Byte Enables
HNA#	O	106	Host Next Address
HD/C#	B	108	Host Data or Code
HM/IO#	B	109	Host Memory or I/O
HW/R#	B	110	Host Write or READ
HLOCMEM#	I	102	Host Local Memory (MB DRAM)
HLOCIO#	I	103	Host Local I/O (387)
HSTRETCH#	I	104	Host Bus Stretch (BCLK)
HRDYI#	I	127	Host Bus Ready Input
HRDYO#	O	112	Host Bus Ready Output
HERDYO#	O	114	Host Bus Early Ready Output
HHOLD	O	115	Host Hold Request
HHLDA	I	129	Host Acknowledge
HLOCK#	I	130	Host Bus Lock
HSSTRB#	O	116	Host Snoop Strobe
HKEN#	I	126	Host Cache Enable
HGT16M#	I	131	Host Greater than 16 Megabytes
EISA EBC Signals (14 Pins)			
BE(3:0)#	B	63:60	Bytes Enables
M-IO	B	73	Memory or I/O Cycle
W-R	B	64	Write or Read Cycle
START#	B	31	Start of EISA Cycle
CMD#	O	41	Command
MSBURST#	B	58	Master Burst
SLBURST#	I	57	Slave Burst
EX32#	BC	56	EISA 32-Bit Slave Size
EX16#	BC	55	EISA 16-Bit Slave Size
EXRDY	BC	53	EISA Ready
LOCK#	O	36	Locked Cycle
ISA EBC Signals (17 Pins)			
BALE	O	39	Bus Address Latch Enable
MASTER16#	I	66	16-Bit Master (EISA or ISA)
BCLK	O	86	Bus Clock
IORC#	B	52	I/O Read Control Strobe
IOWC#	B	50	I/O Write Control Strobe
IO16#	BC	49	16-Bit I/O Slave Size
MRDC#	B	48	Memory Read Control Strobe (under 16 MEG)
MWTC#	B	47	Memory Write Control Strobe (under 16 MEG)
M16#	I	46	16-Bit Memory Slave Size
SMRDC#	O	38	Standard Memory Read Control Strobe (under 1 MEG)

Name = Pin Name, **Type** = I-Input, O-Output, B-Input/Output BC-Input/Output Open Collector, NC-No Connect, V-Power
Pin = Pin Location

4.0 EBC PINOUT INFORMATION (Continued)

EBC Bus Controller (132 Pins) (Continued)

Name	Type	Pin	Description
ISA EBC Signals (17 Pins) (Continued)			
SMWTC#	O	37	Standard Memory Write Control Strobe (under 1 MEG)
CHRDY	BC	65	Channel Ready
NOWS#	I	32	Zero Wait States
SA(1:0)	B	44, 43	System A1 & A0
SBHE#	B	42	System Byte High Enable
REFRESH#	I	33	Refresh Cycle
ISP EBC Signals (9 Pins)			
DHOLD	I	105	ISP Hold Request
DRDY	B	77	ISP Ready
GT1M#	I	71	Greater than 1 Megabyte
ST(3:0)	B	82:79	DMA Status (8-,16-,32-Bit Device & Type A,B,Burst(C) & Compatible Timings)
EXMASTER#	I	70	EISA Master
EMSTR16#	I	69	Early Indication of 16-Bit ISA Master
Data Buffer Control (All Signals are Outputs) (16 Pins)			
SDCPYEN01#	O	4	Copy Enable between Bytes 0 and 1
SDCPYEN02#	O	5	Copy Enable between Bytes 0 and 2
SDCPYEN03#	O	6	Copy Enable between Bytes 0 and 3
SDCPYEN13#	O	7	Copy Enable between Bytes 1 and 3
SDCPYUP	O	8	Copy Up (Low Byte to High Byte)
SDHDLE(3:0)#	O	10:13	System (EISA) Data to Host Data Latch Enables
SDOE(2:0)#	O	14, 16, 17	System (EISA) Data Output Enable
HDSDLE1#	O	18	Host Data to (EISA) Data Latch Enable
HDOE(1:0)#	O	20, 22	Host Data Output Enable
Address Buffer Control (7 Pins)			
HALAOE#	O	23	Host Address to EISA LA Output Enable
HALE#	O	24	Host Address Latch Enable
LASAOE#	O	25	EISA LA to EISA SA Output Enable
LAHAOE#	O	26	EISA LA to Host Address Output Enable
LALE#	O	28	LA Latch Enable
SALAOE#	O	29	EISA SA to EISA LA Output Enable
SALE#	O	30	SA Latch Enable

Name = Pin Name, **Type** = I-Input, O-Output, B-Input/Output BC-Input/Output Open Collector, NC-No Connect, V-Power
Pin = Pin Location

4.0 EBC PINOUT INFORMATION (Continued)

EBC Bus Controller (132 Pins) (Continued)

Name	Type	Pin	Description
Reset Signals (5 Pins)			
SPWROK	I	101	Synchronous Power OK
RST	O	90	System Reset
RSTAR#	I	98	Restart
RSTCPU	O	93	Reset CPU Only
RST385	O	91	Reset 385 Only
Configuration Signals (5 Pins)			
CPU(3:0)	I	97:94	CPU Type and Frequency Indicator
RDE#	I	111	1 CLK1 HERDYO# & HRDYO# Delay for Host to EISA Reads
Miscellaneous Signals (7 Pins)			
CLKKB	O	75	Keyboard Controller Clock
LIOWAIT#	I	3	Long Wait between I/O Cycles
AENLE#	O	76	AEN Latch Enable
TEST1#	I	2	Tri State All Outputs of the EBC
BCLKIN	I	83	For BCLK Delays
Reserved	O	19	NC
Reserved	I	123	Tie High
Reserved	I	124	Tie Low
VCC and GND Signals (29 Pins)			
VCC	V	9,15,21,34,40,54,67,74,85,87,92,99,107,119,132	
VSS	V	1,27,35,45,51,59,68,78,84,88,100,113,117,125	
Total of 103 Control Pins and 29 Power Pins			

Name = Pin Name, **Type** = I-Input, O-Output, B-Input/Output BC-Input/Output Open Collector, NC-No Connect, V-Power
Pin = Pin Location .

4.1 EBC Device Pin List

Device Pinout—132-Lead PQFP

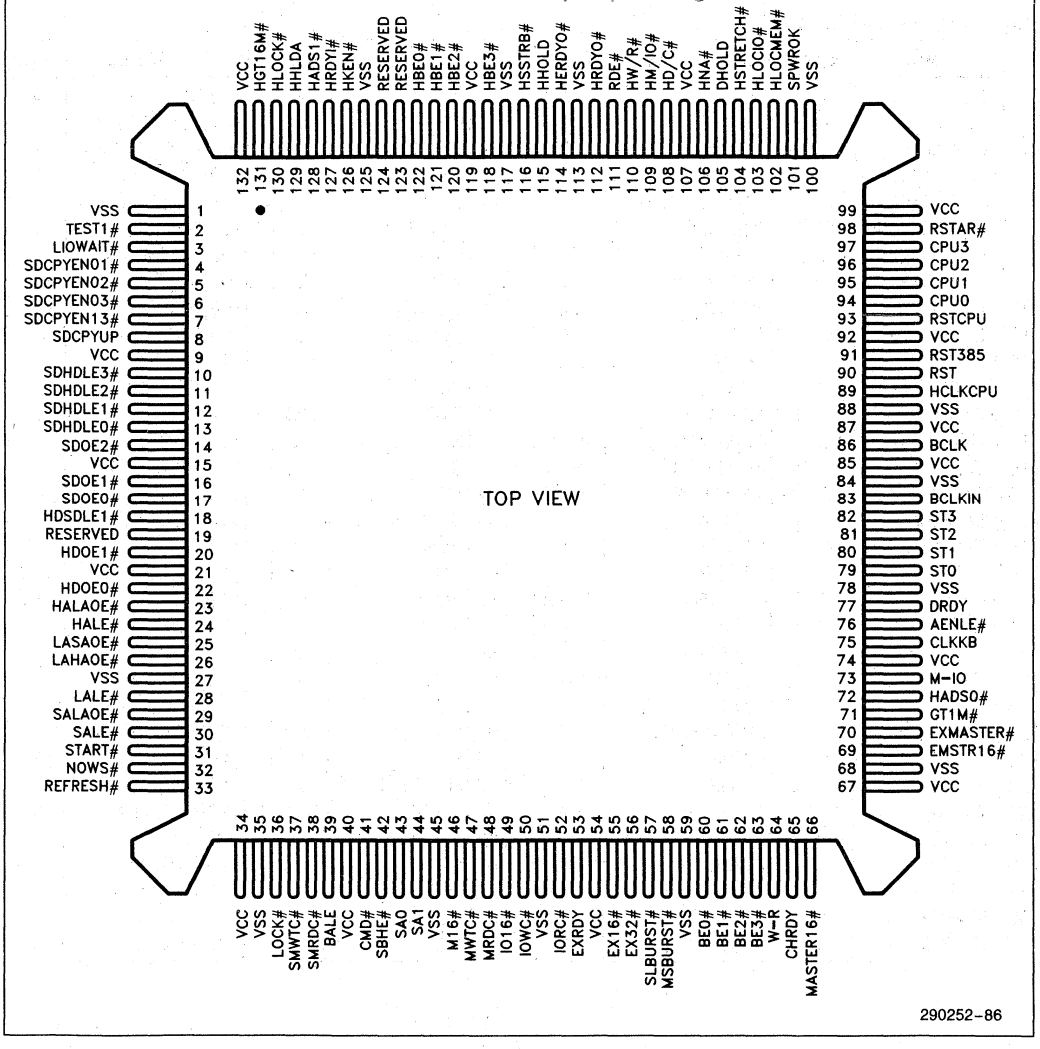
Pin	Name	Type
1	VSS	V
2	TEST1#	I
3	LIOWAIT#	I
4	SDCPYEN01#	O
5	SDCPYEN02#	O
6	SDCPYEN03#	O
7	SDCPYEN13#	O
8	SDCPYUP	O
9	VCC	V
10	SDHDLE3#	O
11	SDHDLE2#	O
12	SDHDLE1#	O
13	SDHDLE0#	O
14	SDOE2#	O
15	VCC	V
16	SDOE1#	O
17	SDOE0#	O
18	HDSDLE1#	O
19	RESERVED	
20	HDOE1#	O
21	VCC	V
22	HDOE0#	O
23	HALAOE#	O
24	HALE#	O
25	LASAOE#	O
26	LAHAOE#	O
27	VSS	V
28	LALE#	O
29	SALAOE#	O
30	SALE#	O
31	START#	I/O
32	NOWS#	I
33	REFRESH#	I

Pin	Name	Type
34	VCC	V
35	VSS	V
36	LOCK#	O
37	SMWTC#	O
38	SMRDC#	O
39	BALE	O
40	VCC	V
41	CMD#	O
42	SBHE#	I/O
43	SA0	I/O
44	SA1	I/O
45	VSS	V
46	M16#	I
47	MWTC#	I/O
48	MRDC#	I/O
49	IO16#	I/O OC
50	IOWC#	I/O
51	VSS	V
52	IORC#	I/O
53	EXRDY	I/O OC
54	VCC	V
55	EX16#	I/O OC
56	EX32#	I/O OC
57	SLBURST#	I
58	MSBURST#	I/O
59	VSS	V
60	BE0#	I/O
61	BE1#	I/O
62	BE2#	I/O
63	BE3#	I/O
64	W-R	I/O
65	CHRDY	I/O OC
66	MASTER16#	I

Pin	Name	Type
67	VCC	V
68	VSS	V
69	EMSTR16#	I
70	EXMASTER#	I
71	GTIM#	I
72	HADS0#	I
73	M-IO	I/O
74	VCC	V
75	CLKKB	O
76	AENLE#	O
77	DRDY	I/O
78	VSS	V
79	ST0	I/O
80	ST1	I/O
81	ST2	I/O
82	ST3	I/O
83	BCLKIN	I
84	VSS	V
85	VCC	V
86	BCLK	O
87	VCC	V
88	VSS	V
89	HCLKCPU	I
90	RST	O
91	RST385	O
92	VCC	V
93	RSTCPU	O
94	CPU0	I
95	CPU1	I
96	CPU2	I
97	CPU3	I
98	RSTAR#	I
99	VCC	V

Pin	Name	Type
100	VSS	V
101	SPWROK	I
102	HLOCMEM#	I
103	HLOCIO#	I
104	HSTRETCH#	I
105	DHOLD	I
106	HNA#	O
107	VCC	V
108	HD/C#	I/O
109	HM/IO#	I/O
110	HW/R#	I/O
111	RDE#	I
112	HRDYO#	O
113	VSS	V
114	HERDYO#	O
115	HHOLD	O
116	HSSTRB#	O
117	VSS	V
118	HBE3#	I/O
119	VCC	V
120	HBE2#	I/O
121	HBE1#	I/O
122	HBE0#	I/O
123	RESERVED	
124	RESERVED	
125	VSS	V
126	HKEN#	I
127	HRDYI#	I
128	HADS1#	I
129	HHLDA	I
130	HLOCK#	I
131	HGT16M#	I
132	VCC	V

82358 EISA Bus Controller (EBC) Pin Diagram



290252-86

5.0 D.C. SPECIFICATIONS

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

5.1 Maximum Ratings

Case Temperature Under Bias . . . −65°C to +110°C
Storage Temperature . . . . . −65°C to +150°C
Supply Voltages with
Respect to Ground . . . . . −0.5V to V_{CC} +6.5V
Voltage on any Pin . . . . . −0.5V to V_{CC} +0.5V

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

1

5.2 D.C. Specifications for the EISA Bus Controller (EBC)

T_{CASE} = 0°C to +70°C, V_{CC} = 5V ±5%

Symbol	Parameter	Min	Max	Units	Conditions
VIL	Input Low Voltage	−0.5	0.8	V	
VIH	Input High Voltage	2.0	VCC + 0.5	V	
VIHC	Input High Voltage	3.0	VCC + 0.5	V	For HCLKCPU
VOLC	BCLK Output Low		0.4	V	IOL = 24 mA
VOHC	BCLK Output High	VCC − 0.5		V	IOH = −4 mA
VOL1	Output Low Voltage		0.45	V	IOL = 5 mA(1)
VOH1	Output High Voltage	2.4		V	IOH = −1 mA(1)
VOL2	Output Low Voltage		0.45	V	IOL = 24 mA(1)
VOH2	Output High Voltage	2.4		V	IOH = −4 mA(1)
VOL3	Output Low Voltage		0.45	V	IOL = 5 mA(1)
VOH3	Output High Voltage	2.4		V	IOH = −250 μA(1)
ILI1	Input Leakage		±15	μA	0V < Vin < VCC
ILO	Output Leakage		±15	μA	0.45 < Vin < VCC
ICC	VCC Supply Current		200	mA	HCLK = 66 MHz
Cap	Cap. In, Out, I/O		20	pF	@1 MHz(2)
C_{CLK}	BCLK Cap.		20	pF	@1 MHz(2)

NOTES:
1. VOL2 and VOH2 apply only to the signals that directly drive the EISA bus and are not slot specific. These are: BE<3:0>#, M-IO, W-R, START#, CMD# MSBURST#, EX32#, EX16#, EXRDY, LOCK#, BALE, IORC#, IOWC#, IO16#, MRDC#, MWTC#, SMRDC#, SMWTC#, CHRDY, SA<1:0>, SBHE#.
VOL3 and VOH3 apply only to the signals that do not drive onto buses in the system. These are:
ST<3:0>, SDCPYENx#, SDCPYUP, SDHDLE<3:0>#, SDOE<2:0>#, HDSDLE1#, HDOE<1:0>#, HALAOE#, HALE#, LASAOE#, LAHAOE#, LALE#, SALAOE#, SALE#, RSTCPU, RST385, CLKKB, AENLE#.
All other outputs use VOL1 and VOH1.
2. Sampled only.

6.0 A.C. SPECIFICATIONS

A.C. Specifications for the EISA Bus Controller (EBC)

T_{CASE} = 0°C to +70°C, V_{CC} = 5V ±5%, $T_{ambient}$ = 0°C to +55°C

Symbol	Parameter	25 MHz		33 MHz		Units	Notes
		Min	Max	Min	Max		
Host Interface Signals							
	HCLKCPU						
t1	Period (2x Mode)	20		15		ns	386 Only
t1a	High, Low Time (2x Mode)	8		6.25		ns	386 Only
t1b	Period (1x Mode)	40		30		ns	i486 only
t1c	High, Low Time (1x Mode)	16		13		ns	i486 Only
t1d	Rise, Fall Time		4		3	ns	60
t1e	Duty Cycle	16/40		NA			67
	HKEN#						
t2	Setup Time	20		15		ns	i486 Only, 7c
t2a	Hold Time	1 CLK1		1 CLK1		ns	i486 Only, 7c
	HADS0#						
t3	Setup Time	17		12		ns	H, 7
t3a	Hold Time	3		3		ns	H, 7
	HADS1#						
t4	Setup Time	17		10.5		ns	H, 7
t4a	Hold Time	3		3		ns	H, 7
	HBE<3:0>#						
t5	Setup Time	12		10		ns	H, 2, 7, 41
t5a	Hold Time	1 CLK1		1 CLK1		ns	H, 2, 7, 41
t5b	Propagation Delay from BE<3:0># or SA<1:0>, SBHE		25		25	ns	D, E, I, R
	HNA#						
t6	Active Edge Valid Delay	3	20	3	16	ns	H, 7
t6a	Float Delay		20		15	ns	H, 7, 60
t6b	Inactive Edge Valid Delay		20		20	ns	H, 7
	HD/C#, HW/R#, HM/IO#						
t7	Setup Time	10		6		ns	H, 51
t7a	Hold Time	4		2		ns	H, 51
	HD/C#, HM/IO#						
t8	Float Delay from HHLDA		25		25	ns	60
	HW/R#						
t9	Propagation Delay from W-R		30		30	ns	E
t9a	Propagation Delay from IOWC#, MWTC#		40		40	ns	I, 33
t9b	Setup Time	20		20		ns	D, R, 5a

6.0 A.C. SPECIFICATIONS (Continued)

A.C. Specifications for the EISA Bus Controller (EBC) (Continued)

Symbol	Parameter	25 MHz		33 MHz		Units	Notes
		Min	Max	Min	Max		
Host Interface Signals (Continued)							
	HLOCMEM#, HLOCIO#						
t10	Setup Time	24		21		ns	H, 2, 7
t10a	Hold Time	2		2		ns	H, 2, 7
t10b	Setup Time	40		40		ns	D, E, 5, 34
t10c	Hold Time	15		15		ns	D, E, 6
	HSTRETCH#						
t11	Setup Time	20		15		ns	HS, 3, 7, 59
t11a	Hold Time	5		5		ns	HS, 3, 7, 59
	HRDYI#						
t12	Setup Time	20		16.5		ns	H, 7
t12a	Hold Time	3.5		3.5		ns	H, 7
	HERDYO#, HRDYO#						
t13	Valid Delay	2	21	2	21	ns	H, 7
t13a	HRDYO# Float Delay(1)		20		15	ns	H, 7, 60
	HHOLD						
t14	Valid Delay	2	22	2	17	ns	7
	HHLDA						
t15	Setup Time	12		10		ns	7
	HLOCK#						
t16	Setup Time	10		7		ns	H, 7
t16a	Hold Time	4.5		4.5		ns	H, 7
	HSSTRB#						
t17	Valid Delay	4	23	4	23	ns	A, 7, 10
	HGT16M#						
t18	Setup Time	60		60		ns	D, 5, 36
t18a	Hold Time	60		60		ns	D, 5, 36
EISA Bus Interface Signals							
	BE<3:0>#						
t19	Valid Delay	1	75	1	75	ns	H, DA, 6
t19a	Valid Delay	1	50	1	50	ns	DA, 5
t19b	Setup Time	25		25		ns	D, E, 6a
t19c	Hold Time	½ BCLK		½ BCLK			D, E, 5, 36
t19d	Propagation Delay from SA<1:0>, SBHE#		60		60	ns	I
t19e	Setup Time (Burst)	30		30		ns	D, E, 6
t19f	Hold Time (Burst)	2		2		ns	D, E, 6
t19g	Propagation Delay from HKEN#		50		40	ns	H, i486 only, 31
t19h	Propagation Delay from HBE#		40		30	ns	H

NOTE:
1. HERDYO# does not tristate.

6.0 A.C. SPECIFICATIONS (Continued)

A.C. Specifications for the EISA Bus Controller (EBC) (Continued)

Symbol	Parameter	25 MHz		33 MHz		Units	Notes
		Min	Max	Min	Max		
EISA Bus Interface Signals (Continued)							
	M-IO						
t20	Propagation Delay from IORC#, IOWC#		40		40	ns	I, 12
t20a	Setup Time	80		80		ns	E, 5, 36
t20b	Hold time	½ BCLK		½ BCLK		ns	E, 5, 36
	W-R						
t21	Propagation Delay from IOWC#, MWTC#		60		60	ns	I
t21a	Propagation Delay from HW/R#		50		50	ns	D, H, 68
t21b	Valid Delay	1	50	1	50	ns	D, H, 6
t21c	Setup Time	25		25		ns	E, 6
t21d	Hold Time	½ BCLK		½ BCLK		ns	E, 5, 36
	START#						
t22	Valid Delay from BCLK Rising	1	30	1	30	ns	AS, D, H, I, R, 5
t22a	Valid Delay from HCLK CPU	1	32	1	32	ns	7a
t22b	Pulse Width	$T_{per}-5$		$T_{per}-5$		ns	
t22c	Setup Time	25		25		ns	E, 6
t22d	Hold Time	0		0		ns	E, 5
	CMD#						
t23	Valid Delay	1	30	1	30	ns	D, E, H, R 5, 6, 62
t23a	Prop Delay from ISA CMD	0	30	0	30	ns	I, 61
t23b	Pulse Width (Standard)	$T_{per}-5$		$T_{per}-5$		ns	
	MSBURST#						
t24	Valid Delay	1	35	1	35	ns	D, 6
t24a	Setup Time	15		15		ns	E, 5
t24b	Hold Time	1 CLK1 + 5		1 CLK1 + 15		ns	E, 5
	SLBURST#						
t25	Setup Time	15		15		ns	D, 5
t25a	Hold Time	25		25		ns	D, 5
	EX32#, EX16#						
t26	Valid Delay	1	35	1	35	ns	E, AS, 6, 31
t26a	Float Delay		19		19	ns	E, AS, 6, 31a, 60
t26b	Propagation Delay from HLOCMEM#/HLOCIO#		30		30	ns	D, E, 31, 8
t26c	Float Delay from HLOCMEM#/HLOCIO# Inactive		18.5		18.5	ns	D, E, 31a, 60, 8
t26d	Setup Time	25		25		ns	D, E, H, 5, 36
t26e	Hold Time	55		55		ns	D, E, H, 5, 36
	EXRDY						
t27	Valid Delay	1	35	1	35	ns	D1, 5
t27a	Setup Time	15		15		ns	A, 6
t27b	Hold Time	3		3		ns	A, 6
t27c	Float Delay		30		30	ns	D1, 6, 60
	LOCK#						
t28	Valid Delay	1	24	1	24	ns	H, 50, 5A
	BALE						
t29	Valid Delay	1	20	1	20	ns	E, H, 5, 6, 15

6.0 A.C. SPECIFICATIONS (Continued)

A.C. Specifications for the EISA Bus Controller (EBC) (Continued)

Symbol	Parameter	25 MHz		33 MHz		Units	Notes
		Min	Max	Min	Max		
EISA Bus Interface Signals (Continued)							
	MASTER16#						
t30	Setup Time	20		20		ns	E, 5
t30a	Hold Time	0		0		ns	E, 5
	BCLK						
t31	Valid Delay	1	15	1	15	ns	7b
t31a	High, Low Time	55		55		ns	
t31b	Rise, Fall Time		5		5	ns	60
	IORC#, IOWC#, MRDC# and MWTC#						D, E, H, R
t32	Valid Delay	3.5	30	3.5	30	ns	5, 6, 16
	SMRDC#, SMWTC#						
t33	Propagation Delay from MRDC#, MWTC#		40		40	ns	A, 31
t33a	Propagation Delay from GT1M# Inactive Edge		40		40	ns	I, 31a
	M16#						
t34	Setup Time	18		18		ns	D, E, H, 5
t34a	Hold Time	25		25		ns	D, E, H, 5
	IO16#						
t35	Setup Time	20		20		ns	E, H, 6
t35a	Hold Time	20		20		ns	E, H, 6
t35b	Propagation Delay from HLOCIO#		70		70	ns	I
	CHRDY						
t36	Propagation Delay from ISA Command		60		60	ns	I, 19
t36a	Float Delay		30		30	ns	I, 6, 19, 60
t36b	Input Pulse Width	10		10		ns	
t36c	Inactive Setup Time	10		10		ns	A, 6
t36d	Active Setup Time	10		10		ns	A, 5
	NOWS#						
t37	Setup Time	10		10		ns	E, I, H, 6
t37a	Hold Time	20		20		ns	A, 6
t37b	Setup Time	35		35		ns	D, 6
	SA<1:0>, SBHE#						
t38	Propagation Delay from BE<>#		45		45	ns	H, 30, 64
t38a	Valid Delay		33		33	ns	DA, E, H, 6, 30, 65
	REFRESH#						
t39	Setup Time	15		15		ns	D, 5, 20
t39a	Hold Time	0		0		ns	D, 5, 20

6.0 A.C. SPECIFICATIONS (Continued)

A.C. Specifications for the EISA Bus Controller (EBC) (Continued)

Symbol	Parameter	25 MHz		33 MHz		Units	Notes
		Min	Max	Min	Max		
ISP Interface Signals							
	ST<3:0>						
t42	ST0 Valid Delay	1	23	1	23	ns	H, 7
t42a	ST0 Prop Delay from START#		25		25	ns	E, 40
t42c	ST2 Propagation Delay from HD/C#, HM/IO#, HW/R#		40		40	ns	H
t42d	ST2 Valid Delay (Pipelined)	1	40	1	40	ns	H, 6, 21
	EXMASTER#						
t43	Setup Time	15		15		ns	5
	EMSTR16#						
t44	Setup Time	15		15		ns	5
Data Buffer Control Signals(69)							
	SDCPYEN01#, SDCPYEN02# SDCPYEN03#, SDCPYEN013#						D, E, H, I, W
t45	Valid Delay Leading Edge	3	26	3	26	ns	6, 49, 66
t45a	Prop Delay from IO16#		25		25	ns	14
t45b	Valid Delay Trailing Edge	1	25	1	25	ns	B, D, E, 5, 53
t45c	Valid Delay Leading Edge	1	15	1	15	ns	B, D, E, 6, 53
t45d	Valid Delay		21		21	ns	5, 63, 31
t45e	SCPYUP Setup to SDCPYEN01#	1		1		ns	63
	SDCPYUP						D, E, H, I
t46	Valid Delay	1	25	1	25	ns	5, 6, 49, 54
	SDHDLE<3:0>#						
t47	Valid Delay	3	25	3	25	ns	5, 6
t47a	Negated Before CMD# Negated	−2		−2		ns	H, AS, 9
47b	Negated Before SDCPYEN<># Negated	2		2		ns	H, AS, 9
47c	Prop Delay from MWTC#/IOWC#		25		25	ns	I
	SDOE<2:0>#						
t48	Valid Delay	3	15	3	15	ns	5, 6, 45
t48a	Valid Delay	8	42	8	42	ns	5, 42, 31
t48b	Valid Delay	1	25	1	25	ns	6, 46
t48c	Prop Delay from IORC#/MRDC#		54		54	ns	I, 31
t48d	Valid Delay	1	30	1	30	ns	5, HS, 31
	HDSDLE1#						
t49	Valid Delay	3	15	3	15	ns	5, 6
t49a	Valid Delay	1	25	1	25	ns	H, 7
t49b	Prop Delay from Active Edge MRDC#/IORC#		30		30	ns	I, 31
	HDOE<1:0>#						
t50	Valid Delay	4	25	4	25	ns	5, 6,7
t50a	Prop Delay from MWTC#/IOWC#		25		25	ns	I

6.0 A.C. SPECIFICATIONS (Continued)

A.C. Specifications for the EISA Bus Controller (EBC) (Continued)

Symbol	Parameter	25 MHz		33 MHz		Units	Notes
		Min	Max	Min	Max		
Address Buffer Control Signals							
t51	HALAOE# Propagation Delay		35		35	ns	A, 47
t52	LASAOE# Propagation Delay		35		35	ns	A, 47
t53	LAHAOE# Propagation Delay		35		35	ns	A, 47
t54	LALE# Valid Delay	1	30	1	30	ns	A, 5, 6, 55
t55	SALAOE# Propagation Delay		35		35	ns	A, 47
	SALE#						
t56	Valid Delay	1	15	1	15	ns	A, 5, 6, 56
t56a	Valid Delay	1	38	1	38	ns	5, 57
Miscellaneous Signals							
	SPWROK						
t57	Setup Time	7		7		ns	7, 11
t57a	Hold Time	2		2		ns	7, 11
	RST						
t58	Valid Delay	1	21	1	21	ns	7
	RSTCPU, RST385						
t59	Valid Delay	2	21	2	21	ns	7, 25
	CLKKB						
t60	Period	90	120	90	120	ns	
t60a	High, Low Time	25	80	25	80	ns	
t60b	Valid Delay	1	40	1	40	ns	7b
t60c	Rise, Fall Time		10		10	ns	60
	LIOWAIT#						
t61	Setup Time	35		35		ns	5, 23
t61a	Hold Time	2		2		ns	5
	AENLE#						
t62	Valid Delay	1	30	1	30	ns	H, E, I, D, 6

NOTES:
A. All cases
AS. Assembly cycle
B. Burst Read cycle
D. DMA cycle
D1. Burst DMA write cycle
DA. EISA master or DMA disassembly cycle
E. EISA master cycle
H. Host cycle
HS. Host slave
I. ISA master cycle
R. Refresh cycle
W. Write cycle
ISA CMD = MRDC#, MWTC#, IORC# or IOWC#

NOTES: (Continued)
2. 1 CLK1 period after HADS# sampled active by the EBC on internal CLK1 falling edge.
3. Setup to CLK1 where BCLK would go high at end of CMD.
5. Referenced to BCLK rising edge
5a. Referenced to BCLK rising edge at START# active.
6. Referenced to BCLK falling edge
6a. Referenced to BCLK falling edge at START# active.
7. Referenced to HCLKCPU. 386 systems: HCLKCPU = rising edge of 386 CLK2. 486 systems: HCLKCPU = rising edge of 486 CLK1.
7a. When there is no BCLK edge present (due to stretching), HCLKCPU rising edge is used as the reference point. The trailing edge of START# is always referenced to BCLK rising edge.
7b. Referenced to HCLKCPU: rising edge for even divisors, both rising and falling edges for odd divisors.
7c. Referenced to HCLKCPU rising at START# active.
8. Applicable to EX32# only.
9. Read Cycle only.
10. HSSTRB# min delay assumes a 35 pF minimum load.
11. SPWROK is sampled asynchronously in i486 CPU based systems.
12. M-IO# is driven by EBB except in ISA master case.
14. Valid for all master cycles during I/O reads only.
15. BALE = START# & BCLK low. For the normal case, BALE is a delay from BCLK falling edge during START#. For the BCLK stretching case, BALE is a delay from either START# or BCLK falling, whichever is worse. In either case, BALE is guaranteed to be valid 20 ns from the middle of START#.
16. IORC# and IOWC# leading edge is generated from BCLK falling edge. MRDC# and MWTC# leading edge is generated from BCLK rising for 16 bit case, and from BCLK falling for 8 bit case. The trailing edge of these signals is generated from BCLK rising edge, except for DMA case, which uses BCLK falling edge for IORC# and MRDC#.
19. The leading edge of CHRDY is propagated from the ISA command signals. The trailing edge of CHRDY is a float delay from BCLK falling edge. This signal uses an open collector driver. CHRDY is asynchronous when it is an input.
20. REFRESH# is sampled on BCLK rising edge before ST<3:0> pins are sampled. During an idle state, REFRESH# is sampled at the same time as the ST<3:0> pins.
21. ST<3:0> are inputs to the EBC when the ISP is the bus master. Otherwise, these signals are outputs of the EBC.
23. LIOWAIT# is setup to BCLK rising at the end of an ISA I/O command.
25. RSTCPU and RST385 must be synchronized by an external PAL at all frequencies.
30. SA <1:0>, SBHE# output delay is a function of either BCLK (where SALE# is active) (t38a); or prop delay from BE <3:0># (t38), whichever occurs last.
31. Active edge only.
31a. Inactive edge only.
33. This case is for access to ISP registers.
34. HLOCMEM# and HLOCIO# are set up to BCLK rising edge at the end of START#.
36. This signal is latched internally during BCLK rising at START# driven inactive.
40. For EISA masters, ST0 is a combinatorial delay from START#.
41. HBE<3:0># are referenced to HCLKCPU rising edge at the beginning of START# when no BCLK edge is present. They are referenced to BCLK rising when this edge is present at START#.
42. This specification is for write cycles requiring EISA master back-off (i.e. the first cycle of a disassembly).
45. This specification is for a read redrive cycle.
46. This specification is for a write disassembly, except the first cycle of EISA or DMA bus masters.
47. HALAOE# prop delay is from HHLDA#, EXMASTER#, EMSTR16# or REFRESH#.
    LAHAOE# prop delay is the same as HALAOE# (except REFRESH#).
    LASAOE# prop delay is from EMSTR16# or REFRESH#.
    SALAOE# prop delay is the same as LASAOE#.
49. Assumes tracking between the SDCPYEN#, SDCPYUP outputs and the MWTC#, SMWTC# outputs (specifically, a 25 ns output delay for the copy enables assumes a minimum delay on MWTC#, SMWTC# of 7 ns).
50. LOCK# is generated one CLK1 period after START# assertion.
51. See the following waveforms which show the EBC setup and hold times relative to HCLKCPU for the sampling points for HD/C#, HW/R#, HM/IO#.
53. For a burst read cycle, SDCPYEN<># change during BCLK low time only (unless wait states are added). For burst write cycles, SDCPYEN<># are always active.
54. SDCPYUP is related to BCLK falling edge in all cases except 16-bit EISA master, or DMA cycles to 8-bit ISA slaves, in which case it is related to BCLK rising edge.
55. LALE# rising edge is referenced to BCLK rising edge when BCLK is not stretched, and to START# when BCLK is stretched. LALE# falling edge is always referenced to BCLK falling edge.
56. SALE# = START# & BCLK LOW for this case (same as BALE).
57. This case is for DMA cycles that do not require assembly. SALE# leading edge is related to BCLK rising edge where ST<3:0> is sampled. SALE# trailing edge is related to BCLK rising edge at the end of START#.
59. HSTRETCH# can not be held low for more than 400 ns.
60. Sampled, but not 100% tested.
61. Inactive edge during ISA Master Read cycle.
62. CMD# is always negated from BCLKs rising edge, except during some types of DMA cycles, and ISA bus master cycles.
63. 16-bit EISA master to 8-bit ISA slave write cycle referenced to BCLK rising edge at end of START#.

NOTES: (Continued)
64. Host master cycle with BCLK stretching.
65. Host master cycle without BCLK stretching.
66. This spec applies to write cycles only, read cycles are not specified because system timings guarantee adequate timing margins for the copy enable signal paths.
67. Minimum of 16 ns on high or low time out of 40 ns period.
68. This spec applies to the first cycle that follows an EISA idle cycle.
69. Data Buffer Control Signal Notes:
    LE = Leading edge
    TE = Trailing edge

Signal	Case	Edge	Reference
HDOE#	CPU Read	L.E. T.E.	BLCK Rising @ CMD# Active Edge CLK1 (where HRDYI Sampled Active)
	Write Disassembly	L.E. T.E.	BCLK Falling during START# Active Edge BCLK Falling @ CMD Inactive Edge +1/2 BCLK
	Read Assembly	L.E. T.E.	BCLK Falling during CMD# Active Edge BCLK Falling @ CMD# Inactive Edge +1/2 BCLK
	Host Memory Write	L.E. T.E.	BCLK Falling during START# Active Edge BCLK Falling @ CMD# Inactive Edge +1/2 BCLK
	ISA Master		Combinatorial from MWTC#/IOWC#
HDSDLE#	CPU Write	Both	Single CLK1 Pulse during START# Active Edge
	Read Assembly	L.E. T.E.	BCLK Falling during CMD# Active Edge BCLK Falling @ CMD# Inactive Edge +1/2 BCLK
	Write Disassembly	L.E. T.E.	BCLK Falling during START# Active Edge BCLK Falling during CMD# Active Edge
	Host Memory	L.E. T.E.	BCLK Rising at CMD# Active Edge BCLK Rising at CMD# Inactive Edge
	ISA Master		Combinatorial from MRDC#, IORC#
SDOE#	CPU Write	L.E. T.E.	BCLK Falling during START# Active Edge or START# & BCLK Low (Stretch Case) BCLK Falling at CMD# Inactive Edge +1/2 BCLK
	Read Assembly	Both	BCLK Low Time during Redrive (BCLK Rising and Falling Edge)
	Write Redrive Master Back-Off Sub-Cycles	L.E. L.E.	BCLK Rising at CMD# Active Edge BCLK Falling during START# Active Edge
	ISA Master		Combinatorial from IORC#/MRDC#
	Host Memory Read (Burst) or Non-Burst)	Both	BCLK Rising Edge (Looks the Same as CMD#)
SDHDLE#	Read (Burst or Non-Burst)	Both	BCLK Rising Edge (Looks the Same as CMD#)
	Write	Both	BCLK Falling Edge (Delayed START#)
	Burst Write	L.E. T.E.	Same as Normal Write Case until Burst Begins. Then Goes Active from BCLK Rising Edge. BCLK Rising at CMD# Inactive Edge
	ISA Master		Active for Entire ISA Command

Signal	Case	Edge	Reference
SDCPYEN#	Burst Read	L.E.	BCLK Falling Edge
	Read	Both	Propagation Delay from CMD#
	Write	L.E.	BCLK Falling during START# Active Edge
		T.E.	BCLK Falling @ CMD# Inactive Edge +1/2 BCLK
SDCPYUP	Read or Write	L.E.	BCLK Falling during START# Active Edge
	16-Bit EISA Master to 8-Bit ISA Slave	L.E.	BCLK Rising @ End of START# Active Edge

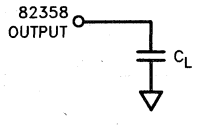
A.C. Test Loads

C_L = 50 pF on HHOLD, SDCPYEN01#, SDCPYEN02#, SDCPYEN03#, SDCPYEN13#, SDHDLE<3:0>#, SDOE<2:0>#, HDSDLE<1:0>#, HDOE<1:0>#, HALAOE#, HALE#, LASAOE#, LAHAOE#, LALE#, SALAOE#, SALE#, HERDYO#, HHOLD, DRDY, ST<3:0>, RST, RSTCPU, RST385, AENLE, HNA#.

C_L = 80 pF on SDCPYUP, CLKKB

C_L = 120 pF on HM/IO#, HW/R#, HD/C#, HRDYO#, HBE<3:0>#, HSSTRB#

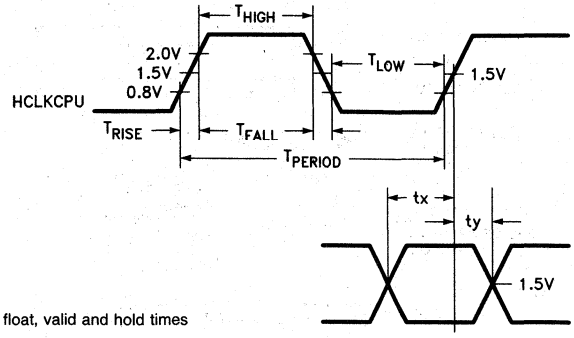
C_L = 240 pF on BE<3:0>#, M-IO, W-R, START#, CMD#, MSBURST#, SLBURST#, EX32#, EX16#, EXRDY, LOCK#, BALE, BCLK, IORC#, IOWC#, MRDC#, MWTC#, SMRDC#, SMWTC#, IO16#, M16# CHRDY, SA<1:0>, SBHE#.



290252-87

C_L includes all parasitic capacitances.

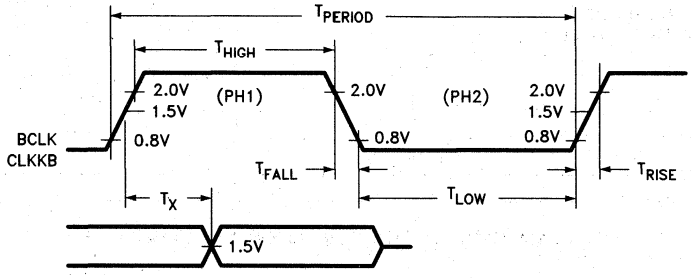
CLK Waveforms



T_{HIGH} = t1a, t1c
T_{LOW} = t1a, t1c
T_{RISE} = t1d
T_{FALL} = t1d
T_{PERIOD} = t1b, t1
tx = input setup times
ty = input hold times, output float, valid and hold times

290252-88

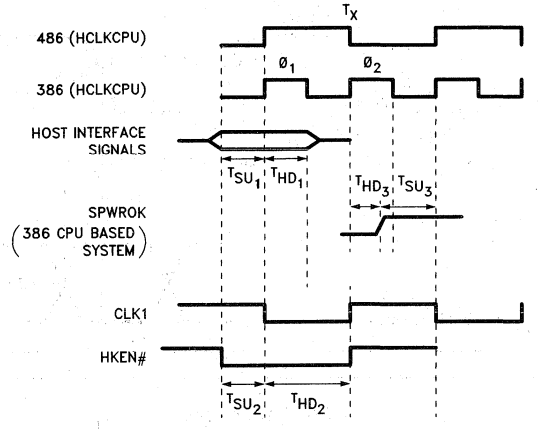
CLK Waveforms



290252-89

T_{HIGH} = t31a, t60a
T_{LOW} = t31a, t60a
T_{RISE} = t31b
T_{FALL} = t31b
T_{PERIOD} = t60
T_X = Output Float or Valid delay times

EBC Signal Setup and Hold Times Relative to HCLKCPU

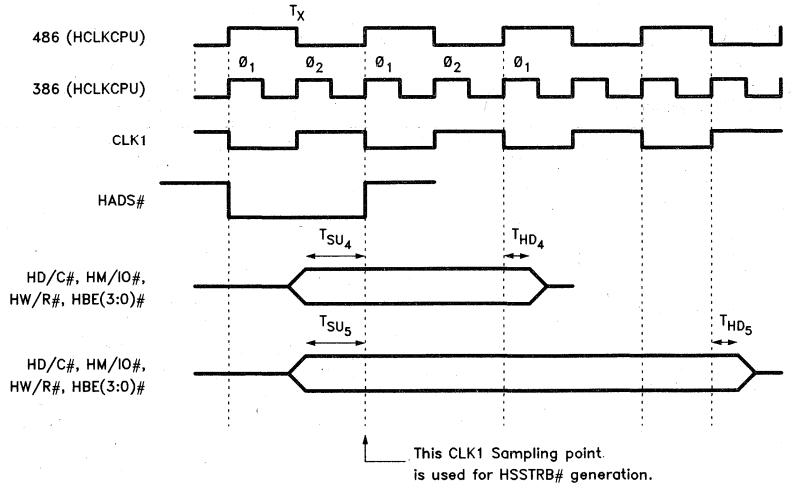


290252-90

T_{SU_1} = t3, t4, t10, t11, t12, t15, t16
T_{HD_1} = t3a, t4a, t10a, t11a, t12a, t16a
T_{SU_2} = t2
T_{HD_2} = t2a
T_{SU_3} = 57
T_{HD_3} = 57a

Host Interface Signals: HADS0#, HADS1#, HLOCMEM#, HLOCIO#, HSTRETCH#, HRDYI#, HHLDA#, HLOCK#

EBC Signal Setup and Hold Times Relative to HCLKCPU

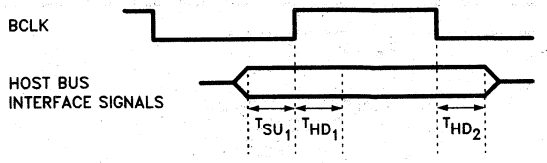


290252-B2

Host Master to Host Slave
T_{SU_4} = t5, t7
T_{HD_4} = t5a, t7a

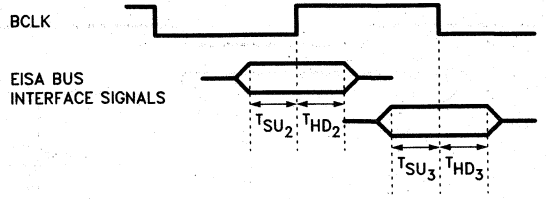
Host Master to Any Non-Host Slave
T_{SU_5} = t5, t7
T_{HD_5} = t5a, t7a

EBC Signal Setup and Hold Times Relative to BCLK



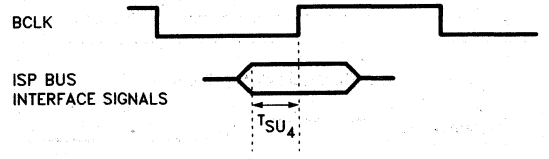
290252-91

T_{SU_1} = t9b, t10b, t18
T_{HD_1} = t18a
T_{HD_2} = t10c



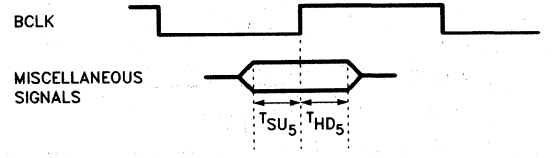
290252-92

T_{SU_2} = t20a, t24a, t25, t26d, t30, t34, t36d, t39
T_{HD_2} = t20b, t21d, t22d, t24b, t25a, t26e, t30a, t34a, t39a
T_{SU_3} = t19b, t19e, t21c, t22c, t27a, t35, t36c, t37
T_{HD_3} = t19c, t19f, t27b, t35a, t37a, t37b



290252-93

T_{SU_4} = t43, t44

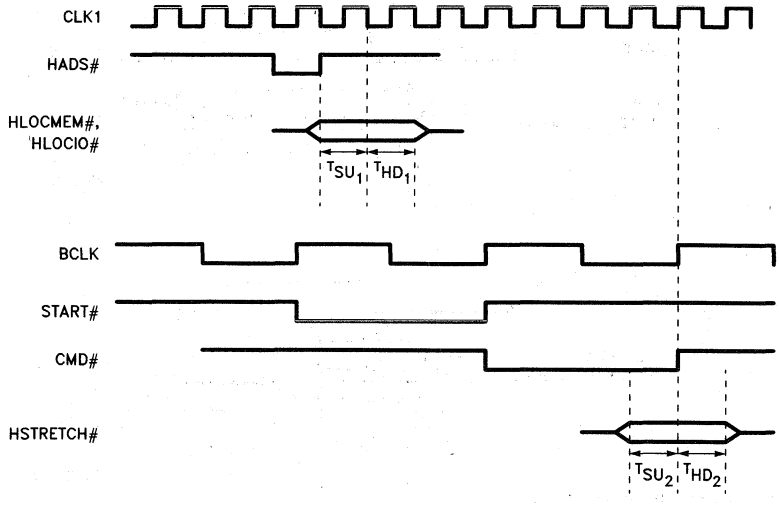


290252-94

T_{SU_5} – T_{61}
T_{HD_5} – T_{61a}

Host Bus Interface Signals — HW/R#, HLOCMEM#, HLOCIO#, HGT16M#,
EISA Bus Interface Signals — BE<3:0>#, M-IO, W-R, MSBURST#, SLBURST#, EX16#, EX32#, EXRDY, MASTER16#, M16#, IO16#, CHRDY, NOWS#, REFRESH#
ISP Bus Interface Signals — DRDY , EXMASTER#, EMSTR16#
Miscellaneous Signals — LIOWAIT#

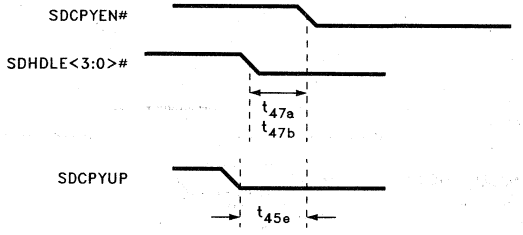
EBC Setup and Hold Times Relative to CLK1



290252-95

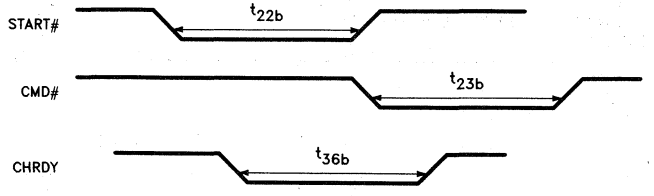
T_{SU_1} = t10
T_{HD_1} = t10a
T_{SU_2} = t11
T_{HD_2} = t11a

Setup and Hold Times



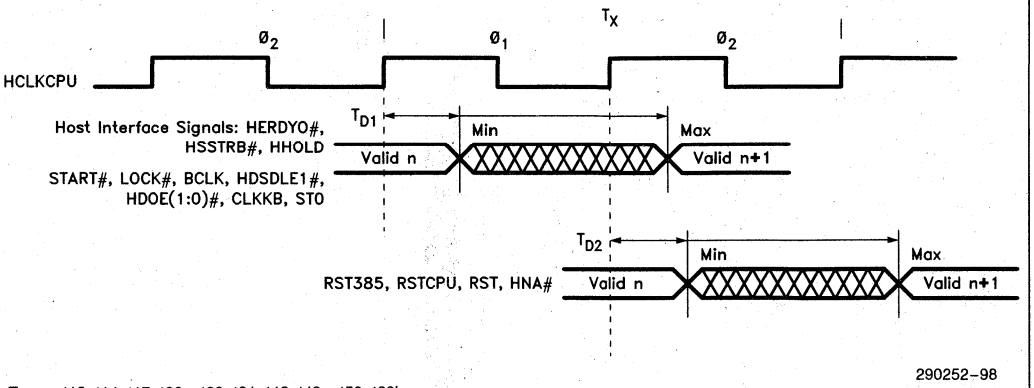
290252-96

Pulse Width Timings



290252-97

Output Valid Delay Timings Relative to HCLKCPU

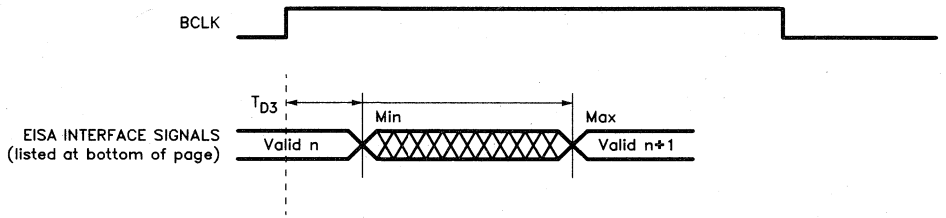


290252-98

T_{D_1} = t13, t14, t17, t22a, t28, t31, t42, t49a, t50, t60b
T_{D_2} = t6, t6b, t58, t59

1

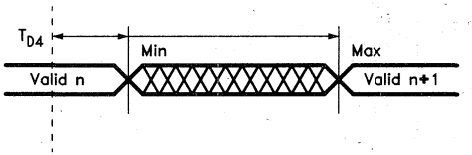
Output Valid Delay Timings Relative to BCLK



290252-A0

CMD#, W-R, EXRDY, MSBURST#, MRDC#, MWTC#, IORC#, IOWC#, EX16#, EX32#, BALE, BE <3:0>#, SA<1:0>, SBHE#

ISP INTERFACE SIGNALS
ST2



290252-A1

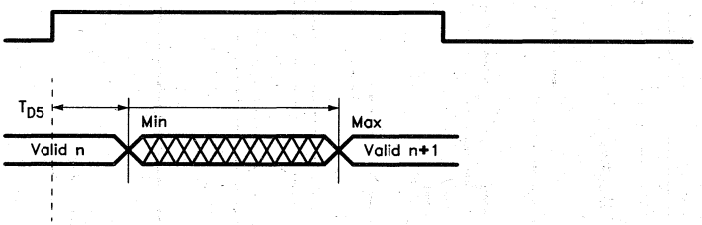
T_{D_3} = t19a, t22, t23, t27, t28, t29, t32
T_{D_4} = t19, t21b, t23, t24, t26, t29, t32, t42d, t23, t38a

EISA Interface Signals: BE <3:0>#, CMD#, START# EXRDY, BALE, IORC#, IOWC#, MRDC#, MWTC#, LOCK#

Output Valid Delay Timings Relative to BCLK (Continued)

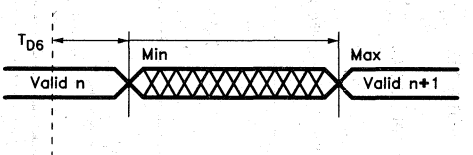
Data Buffer Control Signals
SDCPYEN01#,
SDCPYEN02#,
SDCPYEN03#,
SDCPYEN13#, SDCPYUP,
SDHDLE <3:0>#,
SDOE<2:0>#, HDSDLE1#,
HDOE<1:0>#

Address Buffer Control Signals
SALE#, LALE#



290252-A2

Data Buffer Control Signals
SDCPYEN01#, SDCPYEN02#,
SDCPYEN03#, SDCPYEN13#,
SDHDLE <3:0> #,
SDOE<2:0>#, HDSDLE1 #,
HDOE<1:0>#

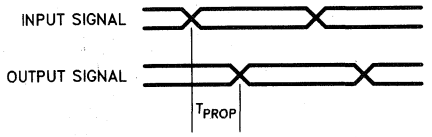


290252-A3

Address Buffer Control Signals
SALE#

Miscellaneous Signals
AENLE#
T_{D5} = t45b, t45d, t46, t47, t48, t48a, t48d, t49, t50, t54, t56, t56a
T_{D6} = t45, t45c, t46, t47, t48, t48b, t49, t50, t56, t62

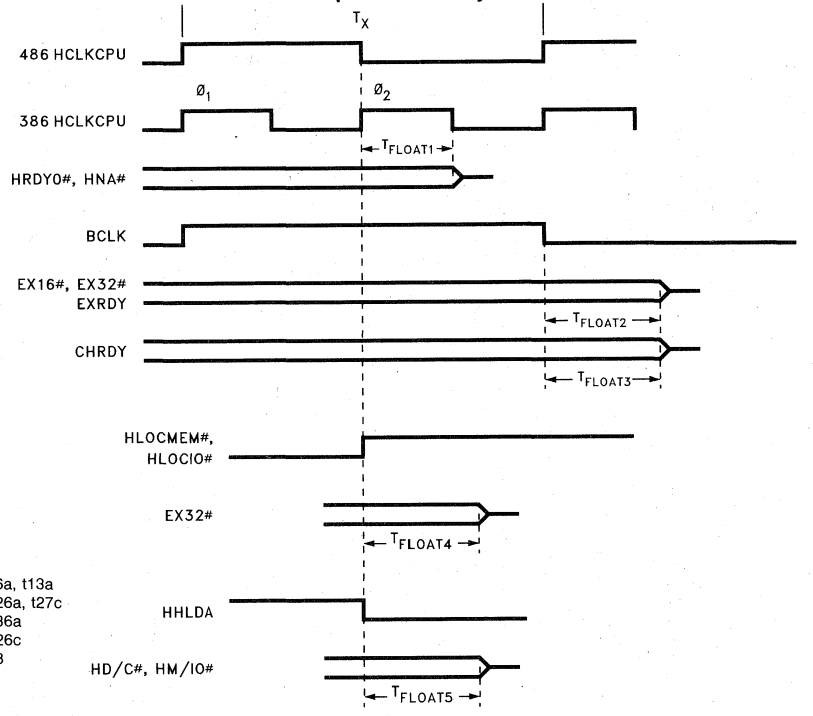
Propagation Delay



290252-A4

T_{PROP} = t5b, t9, t9a, t19d, t19g, t19h, t20, t21, t21a, t23a, t26b, t33, t33a, t35b, t36, t38, t42a, t42c, t45a, t47c, t48c, t49b, t50a, t51, t52, t53, t55

Output Float Delays



T_{FLOAT_1} = t6a, t13a
T_{FLOAT_2} = t26a, t27c
T_{FLOAT_3} = t36a
T_{FLOAT_4} = t26c
T_{FLOAT_5} = t8

290252-A6

82352
EISA BUS BUFFER (EBB)

- **Designed Specifically for EISA Bus Requirements**
- **Provides Three Modes of Operation**
  - **— Data Latch and Swap Functions Allow Swapping and Assembly of Data between the Host and EISA/ISA Buses on a Byte by Byte Basis (Mode 0)**
  - **— Provides a Buffered Path with Parity Generation/Check between the Host Data Bus and DRAM (Mode 1)**
  - **— Address Latch Functions Provide Latching between the Host and EISA/ISA Buses (LA and SA Addresses) (Mode 3)**
- **Similar in Function to Discrete Implementation Using 74F543s/544, 74180s, and 74ALS245s**
- **Replaces 19 Discrete Components**
  - **— Three 82352s are Used Per EISA System**
- **The 82352 Interfaces Easily to the System**
  - **— Buffer Control for the 32-Bit Mode W/O Parity and the EISA Address Mode is Provided by the 82358 (EISA Bus Controller)**
- **120-Pin Quad Flat Pack (QFP)**

The 82352 design allows it to replace the multiple address and data latch-buffer/driver ICs used in EISA applications. The EBB provides three modes of operation: a 32-bit mode without parity to replace the EISA data swap buffers, a 32-bit mode with parity to replace the EISA DRAM data parity buffers, and an EISA address mode to replace the host to EISA/ISA address buffers. Mode 2 on the EBB is reserved. The same chip is strapped in three different ways to obtain the three configurations. These three chips as a group, integrate the logic and buffers that normally require approximately 19 chips.

Manufactured and tested for Intel by Texas Instruments and IC Solutions in accordance with their respective internal standards.

October 1990
Order Number: 290254-003

82352 EISA BUS BUFFER (EBB)

CONTENTS

PAGE

1.0 INTRODUCTION 1-392
1.1 Bus Description/EISA System Interface 1-392
1.2 System Block Diagram (Figure 1) 1-393

2.0 FUNCTIONAL DESCRIPTION 1-393
2.1 Mode 0: 32-Bit Data Mode without Parity 1-393
2.1.1 Description 1-393
2.1.2 Block Diagram (Figure 2) 1-394
2.2 Mode 1: 32-Bit Data Mode with Parity 1-395
2.2.1 Description 1-395
2.2.2 Block Diagram (Figure 3) 1-395
2.3 Mode 3: EISA Address Mode 1-396
2.3.1 Description 1-396
2.3.1.1 EBB–EISA Bus Relationships 1-396
2.3.1.2 Host System as Bus Master 1-396
2.3.1.3 ISA Bus Master 1-396
2.3.1.4 EISA Bus Master 1-396
2.3.2 Block Diagram (Figure 4) 1-397

3.0 MODE 0: 32-BIT DATA MODE WITHOUT PARITY DETAILED DESCRIPTION 1-398
3.1 Detailed Pin Description 1-398
3.2 D.C. Specifications 1-400
3.2.1 Maximum Ratings 1-400
3.2.2 D.C. Specification Table 1-400
3.3 A.C. Specifications 1-401
3.3.1 A.C. Specifications Table 1-401
3.3.2 A.C. Characteristic Waveforms (Figure 5) 1-402
3.4 Pin Summary 1-403
3.5 120-Pin Package Pinout (Figure 6) 1-403

4.0 MODE 1: 32-BIT DATA MODE WITH PARITY DETAILED PIN DESCRIPTION 1-404
4.1 Detailed Pin Description 1-404
4.2 D.C. Specifications 1-407
4.2.1 Maximum Ratings 1-407
4.2.2 D.C. Specification Table 1-407
4.3 A.C. Specifications 1-408
4.3.1 A.C. Specification Table 1-408
4.3.2 A.C. Characteristic Waveforms (Figure 7) 1-409
4.4 Pin Summary 1-410
4.5 120-Pin Package Pinout (Figure 8) 1-411

CONTENTS

PAGE

5.0 MODE 3: EISA ADDRESS MODE 1-412
5.1 Detailed Pin Description 1-412
5.2 D.C. Specifications 1-414
5.2.1 Maximum Ratings 1-414
5.2.2 D.C. Specification Table 1-414
5.3 A.C. Specification 1-414
5.3.1 A.C. Specification Table 1-414
5.3.2 A.C. Characteristic Waveforms (Figure 9) 1-415
5.4 Pin Summary 1-417
5.5 120-Pin Package Pinout (Figure 10) 1-417

6.0 82352 CRITICAL FEATURES 1-418
6.1 A-BUS Output Buffer Contention (16-Bit Host Application) 1-418
6.2 B-BUS Output Buffer Contention (B-BUS to B-BUS Re-drive) 1-418

7.0 82352 DEVICE PINOUT 1-419

8.0 82352 PACKAGE DIMENSIONS (Figure 12) 1-422

9.0 82352 PACKAGE THERMAL CHARACTERISTICS 1-422

10.0 EISA BUS MODEL (Figure 13) 1-423

11.0 FREQUENTLY USED ABBREVIATIONS 1-423

1.0 INTRODUCTION

1.1 Bus Description/EISA System Interface

The EBB supports up to three buses when used in an EISA system. Each bus is broken down into groups; usually one byte in width. Each group is identified by a letter/number combination which identifies the associated bus and relative group location in the bus. The following identifies the various bus/group relationships on a per mode basis.

Mode 0: 32-Bit Data Mode without Parity

A-BUS =A-BUS<31..0> = **A0** + **A1** + **A2** + **A3**

A0 =A0<7..0> = HOST DATA BUS <7..0>

A1 =A1<7..0> = HOST DATA BUS <15..8>

A2 =A2<7..0> = HOST DATA BUS <23..16>

A3 =A3<7..0> = HOST DATA BUS <31..24>

B-BUS B-BUS<31..0> = **B0** + **B1** + **B2** + **B3**

B0 =B0<7..0> = EISA/ISA DATA BUS <7..0>

B1 =B1<7..0> = EISA/ISA DATA BUS <15..8>

B2 =B2<7..0> = EISA/ISA DATA BUS <23..16>

B3 =B3<7..0> = EISA/ISA DATA BUS <31..24>

Mode 1: 32-Bit Data Mode with Parity

A-BUS =A-BUS<31..0> = **A0** + **A1** + **A2** + **A3**

A0 =A0<7..0> = DRAM DATA BUS <7..0>

A1 =A1<7..0> = DRAM DATA BUS <15..8>

A2 =A2<7..0> = DRAM DATA BUS <23..16>

A3 =A3<7..0> = DRAM DATA BUS <31..24>

B-BUS B-BUS<31..0> = **B0** + **B1** + **B2** + **B3**

B0 =B0<7..0> = HOST DATA BUS <7..0>

B1 =B1<7..0> = HOST DATA BUS <15..8>

B2 =B2<7..0> = HOST DATA BUS <23..16>

B3 =B3<7..0> = HOST DATA BUS <31..24>

Mode 3: EISA Address Mode

A-BUS =A-BUS<31..2> = **A0** + **A1** + **A2** + **A3**

A0 =A0<6..0> = HOST ADDRESS BUS <8..2>

A1 =A1<7..0> = HOST ADDRESS BUS <16..9>

A2 =A2<6..0> = HOST ADDRESS BUS <23..17>

A2 =A2<7> = HOST (HM/IO#)

A3 =A3<7..0> = HOST ADDRESS BUS <31..24>

B-BUS =B-BUS<31..2> = **B0** + **B1** + **B2** + **B3#**

B0 =B0<6..0> = EISA ADDRESS BUS LA <8..2>

B1 =B1<7..0> = EISA ADDRESS BUS LA <16..9>

B2 =B2<6..0> = EISA/ISA ADDRESS BUS LA <23..17>

B3 =B3#<7..0> = EISA ADDRESS BUS B-BUS<31..24>

S-BUS = B-BUS<19..2> = **S0** + **S1** + **S2**

S0 S0<6..0> = ISA ADDRESS BUS SA <8..2>

S1 S1<7..0> = ISA ADDRESS BUS SA <16..9>

S2 S2<2..0> = ISA ADDRESS BUS SA <19..17>

NOTE:

The **B3#** (LA#<31..24>) bus group is allowed to float high during ISA bus master operation. These high bits are inverted by the 74F544 style latch/buffer and driven onto **A3** as zeros. This allows correct 32-bit addresses to be transferred to the system during ISA bus master operation.

SUMMARY:

Mode 0 is connected to the EISA System as follows:

AX— Host Data Bus

BX— EISA/ISA Data Bus

Mode 1 is connected to the EISA System as follows:

AX— DRAM Memory lines

BX— Host Data Bus

Mode 2 is reserved.

Mode 3 is connected to the EISA System as follows:

AX— Host Address Bus

BX— LA Address Bus (EISA/ISA Bus)

SX— SA Address Bus (ISA Bus)

2.0 FUNCTIONAL DESCRIPTION

The EBB's functional description is broken down into three discrete descriptions along with one integrated description. Each of the three discrete descriptions view the EBB's functional logic in terms of the ICs it is replacing.

Unless otherwise noted, each functional description assumes the Master Output Enable (MOE#) and the **A-BUS** Latch Enable (A_LE#) signals are asserted (low).

NOTE:
Mode 2 is reserved.

2.1 Mode 0: 32-Bit Data Mode without Parity

2.1.1 MODE 0 DESCRIPTION

Functionally, this 32-bit data mode is similar to a discrete implementation using 74F543s and 74ALS245s. Each of the four latch/buffer modules in Figure 2 can be replaced by a 74F543 with their "chip enable" lines tied to ground. A 74ALS245 can replace each bidirectional transceiver module also found in Figure 2.

The latch/buffer modules, when operated together, will allow bidirectional data transfers and latching between the 32-bit **A-BUS** and 32-bit **B-BUS**. Control of the buses on both 16-bit and 8-bit increments is also possible within the defined constraints of the control signals (REFER TO SECTIONS 3.1, 4.1 and 5.1).

The bidirectional transceiver modules allow 8-bit data transfers to occur between the four bytes comprising the **B-BUS**. If the **B-BUS** is viewed as a four-byte data word, these buffers will allow the lowest order byte, **B0**, to transfer data to or from the remaining three bytes. Also 16-bit data transfers can be supported between the low order bytes (**B0, B1**) and the high order bytes (**B2, B3**). The **Ax** D-Latch inputs will also have access to the data during transfer.

The bidirectional transceiver modules are enabled by asserting the corresponding BxxCPYE# line low. The direction of the copy is determined by the logic level of the CPY_DN# line.

The A_LE# control signal is OR'ed with all the individual Ax latch enable control signals. If all Ax lines are asserted, this line will act as a global **A-BUS** latch enable.

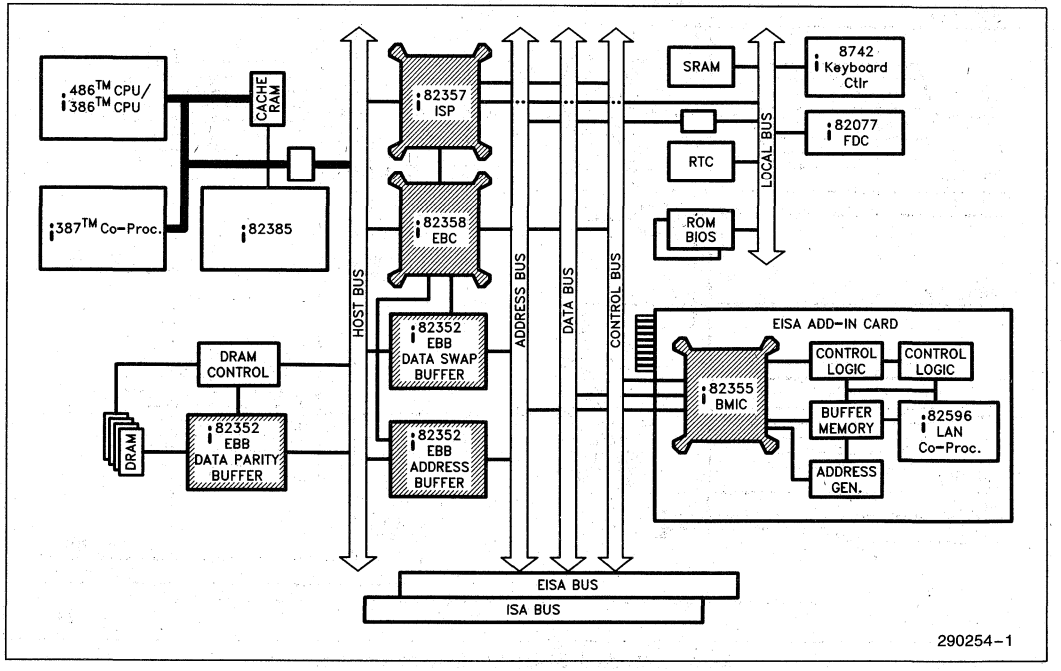


Figure 1. 82350 EISA Chip Set

2.1.2 MODE 0: BLOCK DIAGRAM

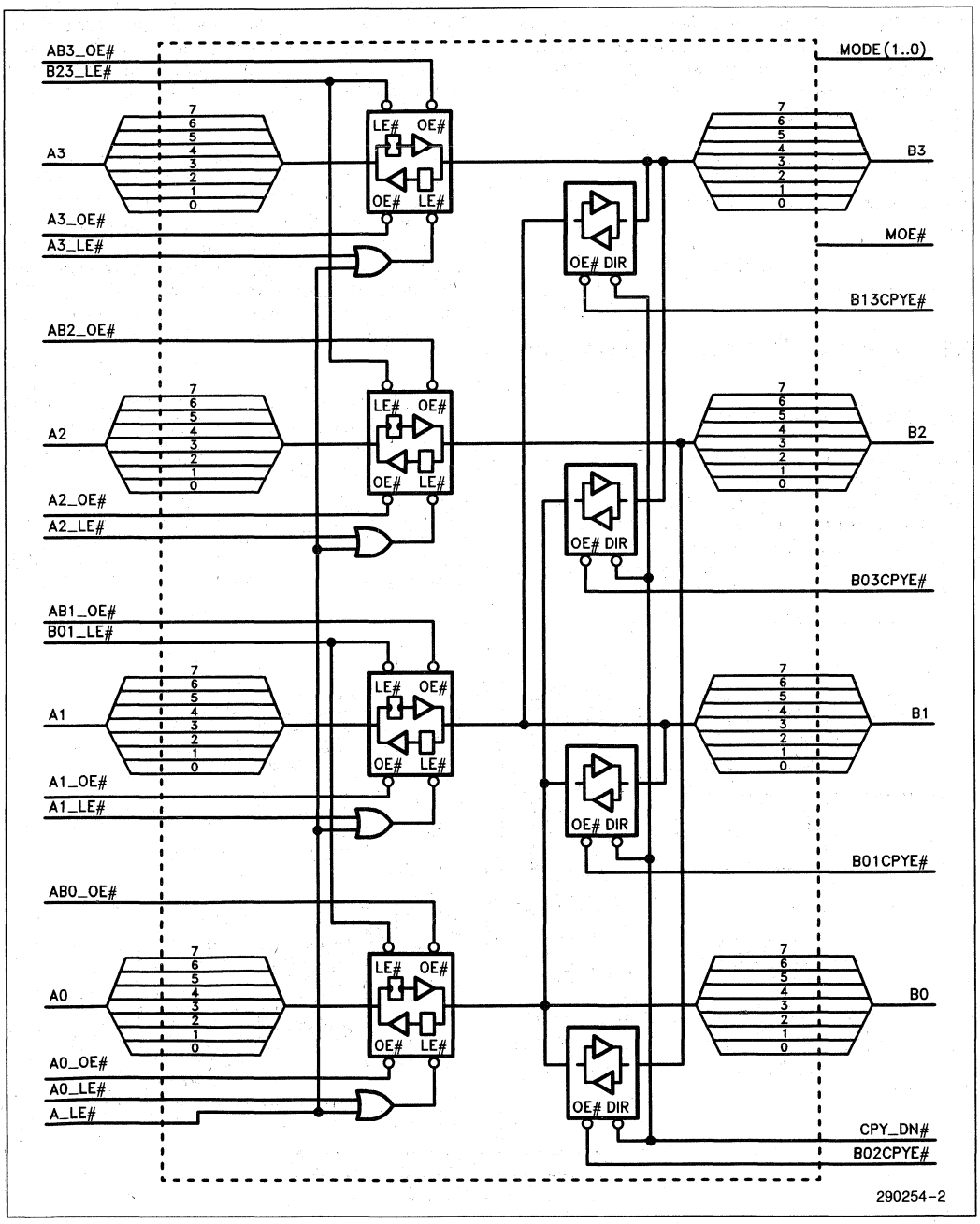


Figure 2. Mode 0: 32-Bit Mode without Parity

2.2 Mode 1: 32-Bit Data Mode with Parity

2.2.1 MODE 1 DESCRIPTION

This data mode is similar to the mode discussed in 2.1 except for the addition of even-parity generation/detection circuits (see Figure 3). Parity support is available for each byte in both the **A-BUS** and **B-BUS**.

The parity generation and detection is functionally similar to the 74180. Each **Ax** and **Bx** is serviced by a separate parity module.

During data transfers from the **B-BUS** to the **A-BUS**, four independent even-parity signals are generated. These parity signals are routed through the same 74F543 style latch/buffer modules that supports the byte used in their generation. This will allow each parity signal to be latched with its associated **Bx** byte.

2.2.2 MODE 1: BLOCK DIAGRAM

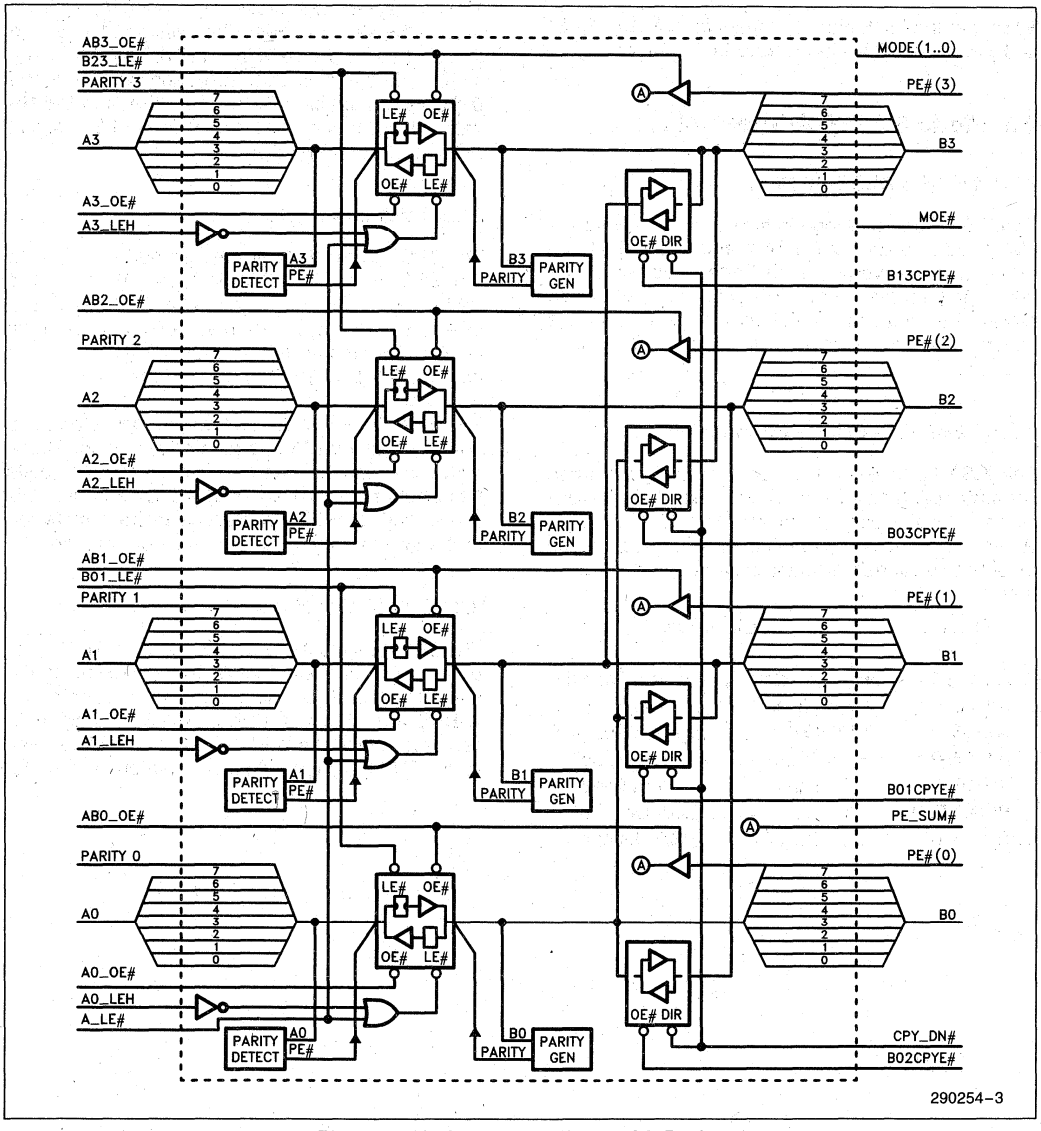


Figure 3. Mode 1: 32-Bit Mode with Parity

During data transfers from the **A-BUS** to the **B-BUS**, incoming **Ax** byte parity is compared with the parity generated in the parity detection modules. If the parities differ and if the output enable for that byte is active, the parity error signal for that byte, along with the parity-sum signal will go active.

The parity-sum signal will not respond to errors for bytes that do not have asserted output enables.

Parity support is not available for any **Bx** to **Bx** data transfer using the bidirectional transceiver modules.

Note that the latch enable signals for the **A0**,**A1**, **A2** and **A3** bus groups are inverted in this mode.

2.3 Mode 3: EISA Address Mode

2.3.1 MODE 3 DESCRIPTION

The EISA Address Mode is similar to a discrete implementation using six 74F543s and a 74F544. Each of the six non-inverting latch/buffer modules in Figure 4 can be replaced by a 74F543. The 74F544 can replace the single inverting latch/buffer. Each of the seven chips would have their "chip enable" lines tied low.

2.3.1.1 EBB–EISA Bus Relationships

The EISA Address Mode is required to support three separate address buses (**A-BUS**, **B-BUS**, **S-BUS**) along with two control signals (HM/IO#, M-IO#). When this mode is operating in an EISA system, the following relationships must exist for proper operation:

EBB Bus	EISA Bus	
A3<7..0>	HA<31..24>	System Specific
A2<6..0>	HA<23..17>	System Specific
A2<7>	HM/IO#	System Specific
A1<7..0>	HA<16..9>	System Specific
A0<6..0>	HA<8..2>	System Specific
B3#<7..0>	LA#<31..24>	EISA Defined
B2<6..0>	LA<23..17>	EISA Defined
B2<7>	M-IO#	EISA Defined
B1<7..0>	LA<16..9>	EISA Defined
B0<6..0>	LA<8..2>	EISA Defined
S2<2..0>	SA<19..17>	EISA Defined
S1<7..0>	SA<16..9>	EISA Defined
S0<6..0>	SA<8..2>	EISA Defined

2.3.1.2 Host System as Bus Master

The host system will present system (HA) address data along with the HM/IO# command signal to the EBB using the **A-BUS**. This data will immediately update the **B-BUS** (LA) and M-IO# if the **B-BUS** latch enable is asserted. When the **B-BUS** data is considered valid, the S-BUS__LE# signal will unlatch the **S-BUS** (SA).

2.3.1.3 ISA Bus Master

The ISA bus master will present address data to the EBB using the **S1, S0 (SA<16..2>) and B2**(LA<23..17>) bus groups. If both A-BUS__LE# and S-BUS__OE# are asserted, this data will immediately update the **A2**, **A1**, **A0** (HA<23..2>) and **B1**, **B0** (LA<16..2>) bus groups.

The **B3**# (LA#<31..24>) bus group is allowed to float high during ISA bus master operation. These high bits are inverted by the 74F544 style latch/buffer and driven onto **A3** as zeroes. This allows correct 32-bit addresses to be transferred to the system during ISA bus master operation.

2.3.1.4 EISA Bus Master

The EISA bus master presents the LA<31..2> address data along with M-IO# to the EBB using the **B-BUS**. This data will immediately update the **A-BUS** and HM/IO# if the $\overline{\text{A-BUS\_\_LE}}$ signal is asserted. When the **B-BUS** data is considered valid, the S-BUS__LE# signal will unlatch the **S-BUS**.

2.3.2 MODE 3: BLOCK DIAGRAM

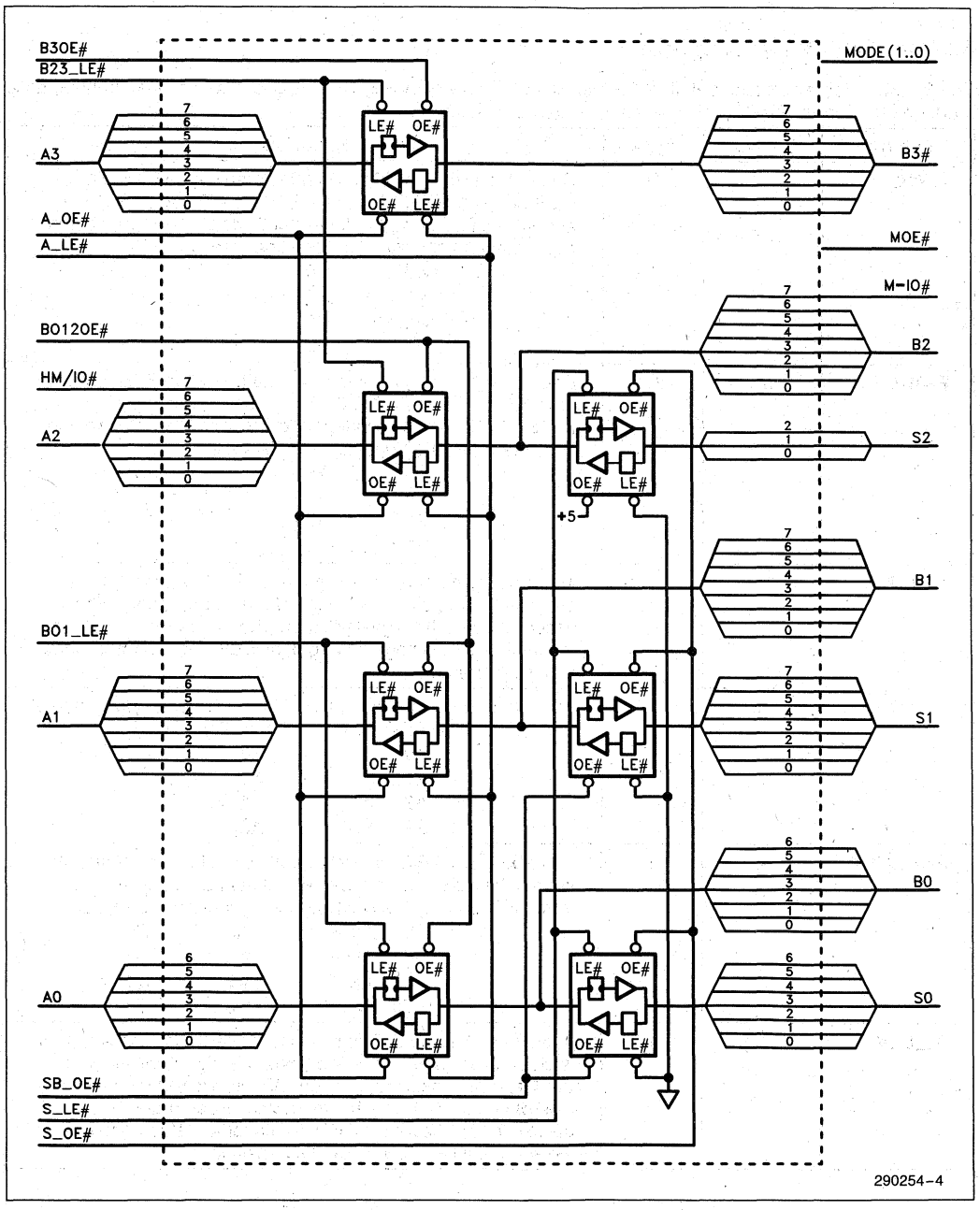


Figure 4 Mode 3: EISA Address Mode

3.0 MODE 0: 32-BIT DATA MODE WITHOUT PARITY DETAILED DESCRIPTION

3.1 Detailed Pin Description

The following table contains detailed descriptions of each signal while the EBB is in its non-parity 32-bit data mode. Note: For all "N/C" pins, use an 8.2K pullup resistor.

Symbol	Pin No.	Type	Function
A-BUS	32	B	One of the two 32-bit buses manipulated by the EBB. Each of the four bytes in **A-BUS** (A0<7..0>, A1<7..0>, A2<7..0> and A3<7..0>) can be independently controlled. During non-copy transfers, the signal associations are: **A0** **B0** **A1** **B1** **A2** **B2** **A3** **B3**
B-BUS	32	HCB	One of the two 32-bit buses manipulated by the EBB. Each of the four bytes in **B-BUS** (B0<7..0>, B1<7..0>, B2<7..0> and B3<7..0>) can be independently controlled. During non-copy transfers, the signal associations are: **B0** **A0** **B1** **A1** **B2** **A2** **B3** **A3**
MOE#	1	I	Master Output Enable. This signal is ORed together with all other output enables used in this mode of operation. When asserted (low), this signal will allow the other output enables to function. When negated (high), all EBB outputs will go into their High-Z states.
Ax__OE#	4	I	**Ax** Output Enables. When MOE# is low, these signals enable the output of the **Ax**. The signal associations are: A0__OE# **A0** A1__OE# **A1** A2__OE# **A2** A3__OE# **A3**
ABx__OE#	4	I	**Ax** to **Bx** Output Enables. When MOE# is low, these signals enable the outputs of the **Bx** buffers during data transfers from the **A-BUS** to the **B-BUS**. The signal associations are: AB0__OE# **A0** **B0** AB1__OE# **A1** **B1** AB2__OE# **A2** **B2** AB3__OE# **A3** **B3**
A__LE#	1	I	**A-BUS** Latch Enable. This signal is "ORed" with the Ax__LE# latch enables. When asserted (low), this signal will allow all the **Ax** latch enables to function. When negated (high), all the **A-BUS** latches will latch the input data present at the time of the low to high transition if the individual **Ax** latch enable is asserted (low) at the time of this transition.

Symbol	Pin No.	Type	Function
Ax__LE#	4	I	**Ax** Latch Enables. When A__LE# is low, these signals can be used to individually control the latching of **B-BUS** data to the **A-BUS**. For non-copy transfers, the signal associations are: A0__LE# **B0** **A0** A1__LE# **B1** **A1** A2__LE# **B2** **A2** A3__LE# **B3** **A3**
B01__LE#	1	I	**B01** and **B1** Latch Enable. This signal is used to control the latching of **A0** and **A1** bus groups to the **B0** and **B1** bus groups respectively. On the asserted to negated (low to high) transition, data present at the latch inputs will be latched onto the latch output.
B23__LE#	1	I	**B2** and **B3** Latch Enable. This signal is used to control the latching of **A2** and **A3** bus groups to the **B2** and **B3** bus groups respectively. On the asserted to negated (low to high) transition, data present at the latch inputs will be latched onto the latch output.
CPY__DN#	1	I	Bus Group Copy Down. This signal determines the direction of data flow for data appearing on the four **Bx** copy transceivers. When asserted (low), the transceivers will allow the following potential data transfers: Transceiver-1: From **B1** to **B0**, **A0** D-Latch Transceiver-2: From **B2** to **B0**, **A0** D-Latch Transceiver-3: From **B3** to **B0**, **A0** D-Latch Transceiver-4: From **B3** to **B1**, **A0** D-Latch When negated (high), the transceivers will allow the following potential data transfers: Transceiver-1: From **B0** to **B1**, **A1** D-Latch Transceiver-2: From **B0** to **B2**, **A2** D-Latch Transceiver-3: From **B0** to **B3**, **A3** D-Latch Transceiver-4: From **B1** to **B3**, **A3** D-Latch
B01CPYE#	1	I	**B-BUS** Copy Enable 01. When MOE# is low, this signal enables copy Transceiver-1 to transfer data between **B0** and **B1**. The data direction is determined by the logic value of CPY__DN#.
B02CPYE#	1	I	**B-BUS** Copy Enable 02. When MOE# is low, this signal enables copy Transceiver-2 to transfer data between **B0** and **B2**. The data direction is determined by the logic value of CPY__DN#.
B03CPYE#	1	I	**B-BUS** Copy Enable 03. When MOE# is low, this signal enables copy Transceiver-3 to transfer data between **B0** and **B3**. The data direction is determined by the logic value of CPY__DN#.
B13CPYE#	1	I	**B-BUS** Copy Enable 13. When MOE# is low, this signal enables copy Transceiver-4 to transfer data between **B1** and **B3**. The data direction is determined by the logic value of CPY__DN#.
SLEW	1	I	Output Buffer Slew Rate Control. This signal will provide slew rate control for all external output buffers used in the EBB. When this signal is negated (low), all output buffers will switch at their non-compensated slew rate. When this signal is asserted (high), the output buffers will switch at a slower (> 2 ns) slew rate. **NOTE:** When the Slew signal is asserted (high), add 1.5 ns to all A.C. timings.

Symbol	Pin No.	Type	Function
MODE<1..0>	2	I	Operating Mode Pins. These signals determine the mode of operation for the EBB. They are intended to be hardwired to the proper value. Operating mode determination: Mode<1> Mode<0> Operating Mode #L L 32-Bit Wide Data Mode w/o Parity L H 32-Bit Wide Data Mode with Parity H L RESERVED H H EISA Address Mode # Mode described in this section

3.2 D.C. Specifications

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

3.2.1 MAXIMUM RATINGS

Temperature Under Bias .....0° to +70°C

Storage Temperature .....−65°C to +150°C

Supply Voltage with Respect to Ground.....−0.5V to +7V

Voltage on Any Pin .....−0.5V to +7V

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

3.2.2 D.C. SPECIFICATION TABLE

The following is a table of load capacitances, input voltage levels, current levels, and input leakage currents for the 32-bit non-parity mode:
T_A = 0°C to +70°C, V_{CC} = 5V ±5%, I_{CC} Max = 160 mA

Signal Name	Pins	Type	I_{OL} Max (mA)	I_{OH} Max (mA)	I_I Max µA	V_I Max (V)	V_H Min (V)	C_{IN} (pF)	C_{LOAD} (pF)
A-BUS<31..0>	8	B	12	−3	10	0.8/0.5	2.0/2.4	27	100
B-BUS<31..0>	8	HCB	24	−3	10	0.8/0.5	2.0/2.4	27	240
MOE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
Ax_OE#	4	I	N/A	N/A	10	0.8	2.0	27	N/A
ABx_OE#	4	I	N/A	N/A	10	0.8	2.0	27	N/A
A_LE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
Ax_LE#	4	I	N/A	N/A	10	0.8	2.0	27	N/A
B01_LE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
B23_LE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
CPY_DN#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
B01CPYE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
B02CPYE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
B03CPYE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
B13CPYE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
MODE<1..0>	2	I	N/A	N/A	10	0.8	2.0	27	N/A

NOTES:
1. I_I Max = I_{LI} for inputs, I_{LO} for outputs (or I/O).
2. V_I Max = V_{IL} for inputs, V_{IL}/V_{OL} for I/O.
3. V_H Min = V_{IH} for inputs, V_{IH}/V_{OH} for I/O.
4. V_{IN} for leakage = 0.45V to V_{CC}.
5. V_{OL} is tested while sinking I_{OL} (24 mA).
6. V_{OH} is tested while sourcing I_{OH} (−3 mA).

3.3 A.C. Specifications

3.3.1 A.C. SPECIFICATION TABLE

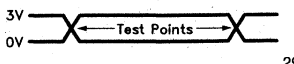
Parameter	From	To	Waveform	Parameter Value	C_L Loading
t_{PLH} or t_{PHL}	A-BUS	B-BUS	1	15.00 ns Max	240 pF
t_{PLH} or t_{PHL}	B-BUS	A-BUS	1	15.00 ns Max	100 pF
t_{PLH} or t_{PHL}	B-BUS	B-BUS	1	15.00 ns Max	240 pF
t_{PLH} or t_{PHL}	BxxCPYE#	B-BUS	1, 2	15.00 ns Max	*240 pF
t_{PLH} or T_{PHL}	BxxCPYE#	A-BUS	1, 2	17.00 ns Max	**100 pF
t_{PLH} or t_{PHL}	CPY__DN#	B-BUS	1, 2	15.00 ns Max	*240 pF
t_{PLH} or T_{QHL}	CPY__DN#	A-BUS	1, 2	17.00 ns Max	**100 pF
t_{PLH} or t_{PHL}	LE#/A__LE#	A-BUS	1, 2	15.00 ns Max	100 pF
t_{PLH} or t_{PHL}	LE#	B-BUS	1, 2	15.00 ns Max	240 pF
t_{PZH} or t_{PZL}	OE# or MOE#	A-BUS	3, 4	15.00 ns Max	100 pF
t_{PHZ} or t_{PLZ}	OE# or MOE#	A-BUS	3, 4	15.00 ns Max	100 pF
t_{PZH} or t_{PZL}	OE# or MOE#	B-BUS	3, 4	15.00 ns Max	240 pF
t_{PHZ} or t_{PLZ}	OE# or MOE#	B-BUS	3, 4	15.00 ns Max	240 pF
t_{SU}	A or B-BUS Data before LE		5	3.50 ns Min	
t_H			5	3.50 ns Min	
t_W	LE# Pulse Width (Low)		5	4.00 ns Min	

*Data originating from the A-Bus or B-Bus going to the B-Bus.
**Data originating from the B-Bus going to the A-Bus.

NOTES:
1. Increase the above A.C. timings by a maximum of 1.5 ns when the slew rate control pin is asserted high.
2. The EBB outputs driving the EISA bus, identified by a 240 pF load, are guaranteed using a distributed load (refer to Section 10.0).

3.3.2 A.C. CHARACTERISTIC WAVEFORMS

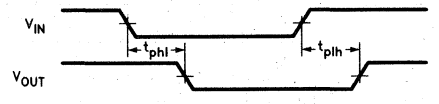
A.C. TESTING INPUT, OUTPUT WAVEFORM



290254-5

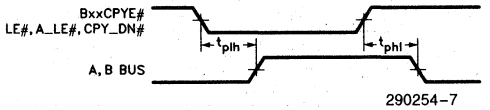
A.C. Testing: Inputs are driven at 3V for a logic "1" and 0V for a logic "0". A.C. Timings are tested at 1.5V.

NOTE:
1. The input waveforms have tr ≤2.5 ns from 0V to 3V.



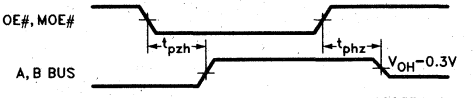
290254-6

Waveform 1. Delay for Outputs



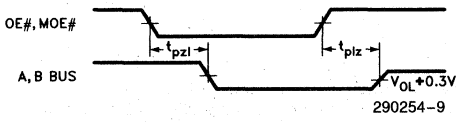
290254-7

Waveform 2. Delay for Outputs



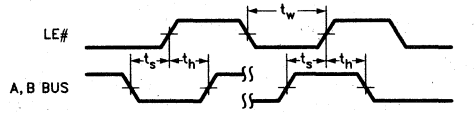
290254-8

Waveform 3. 3-State Output Enable Time to High Level and Output Disable Time from High Level



290254-9

Waveform 4. 3-State Output Enable Time to Low Level and Output Disable Time from Low Level



290254-10

Waveform 5. Data Setup and Hold Times

NOTE:
All timings are referenced to 1.5V unless otherwise stated.

Figure 5. Mode 0

3.4 Pin Summary

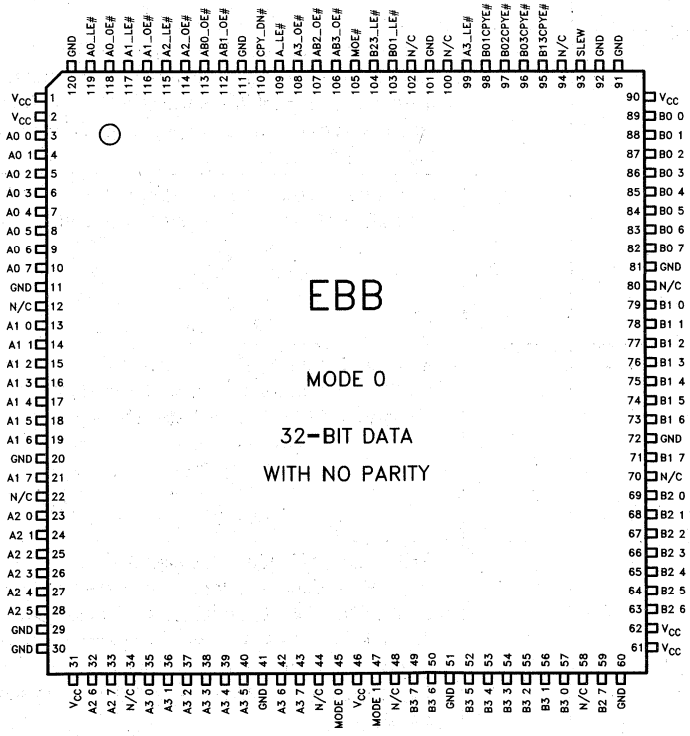
The following table indentifies the signals required for the EBB to operate in its non-parity 32-bit data mode:

Signal Name	No. of Pins	Signal Type
A-BUS	32	B
B-BUS	32	HCB
MOE#	1	I
Ax__OE#	4	I
ABx__OE#	4	I
A__LE#	1	I
Ax__LE#	4	I
B01__LE#	1	I
B23__LE#	1	I
CPY__DN#	1	I
B01CPYE#	1	I
B02CPYE#	1	I

Signal Name	No. of Pins	Signal Type
B03CPYE#	1	I
B13CPYE#	1	I
SLEW	1	I
MODE<1..0>	2	I
GND	14	Ground
V_{CC}	7	Power
N/C	11	
	120	
Pin Summary:	88 21 11	Signals Power & Ground N/C

1

3.5 Mode 0: 120-Pin Package Pinout



NOTE:
For all "N/C" pins, use an 8.2k pullup resistor.

Figure 6. 120-Pin Package Pinout

4.0 MODE 1: 32-BIT DATA MODE WITH PARITY DETAILED PIN DESCRIPTION

4.1 Detailed Pin Description

The following table contains detailed descriptions of each signal while the EBB is in its 32-bit data mode with parity support. Note: For all "N/C" pins, use an 8.2K pullup resistor.

Symbol	Pin No.	Type	Function
A-BUS	32	B	One of the two 32-bit buses manipulated by the EBB. Each of the four bytes in **A-BUS** (A0<7..0>, A1<7..0>, A2<7..0> and A3<7..0>) can be independently controlled. During non-copy transfers, the signal associations are: **A0** **B0** **A1** **B1** **A2** **B2** **A3** **B3**
B-BUS	32	HCB	One of the two 32-bit buses manipulated by the EBB. Each of the four bytes in **B-BUS** (B0<7..0>, B1<7..0>, B2<7..0> and B3<7..0>) can be independently controlled. During non-copy transfers, the signal associations are: **B0** **A0** **B1** **A1** **B2** **A2** **B3** **A3**
MOE#	1	I	Master Output Enable. This signal is "ORed" with all other output enables used in this mode of operation. When asserted (low), this signal will allow the other output enables to function. When negated (high), all EBB outputs will go into their High-Z states.
Ax__OE#	4	I	**Ax** Output Enables. When MOE# is low, these signals enable the output of the **Ax**. The signal associations are: A0__OE# **A0** A1__OE# **A1** A2__OE# **A2** A3__OE# **A3**
ABx__OE#	4	I	**Ax** to **Bx** Output Enables. When MOE# is low, these signals enable the outputs of the **Bx** buffers during data transfers from the **A-BUS** to the **B-BUS**. The signal associations are: AB0__OE# **A0** **B0** AB1__OE# **A1** **B1** AB2__OE# **A2** **B2** AB3__OE# **A3** **B3**
A__LE#	1	I	**A-BUS** Latch Enable. This signal is "ORed" with the Ax__LEH# latch enables. When asserted (low), this signal will allow all the **Ax** latch enables to function. When negated (high), all the **A-BUS** latches will latch the input data present at the time of the low to high transition if the individual **Ax__LEH** latch enable is asserted (high) at the time of this transition. **NOTE:** The Ax latch enables are inverted in this mode. (Refer to Figure 2.)

Symbol	Pin No.	Type	Function
Ax__LEH	4	I	**Ax** Latch Enables. When A__LE# is low, these signals are used to control the latching of **B-BUS** data to the **A-BUS**. For non-copy transfers, the signal associations are: **NOTE:** The Ax latch enables are inverted in this mode. (Refer to Figure 2.) A0__LEH **B0** **A0** A1__LEH **B1** **A1** A2__LEH **B2** **A2** A3__LEH **B3** **A3**
B01__LE#	1	I	**B01** and **B1** Latch Enable. This signal is used to control the latching of **A0** and **A1** bus groups to the **B0** and **B1** bus groups respectively. On the asserted to negated (low to high) transition, data present at the latch inputs will be latched onto the latch output.
B23__LE#	1	I	**B2** and **B3** Latch Enable. This signal is used to control the latching of **A2** and **A3** bus groups to the **B2** and **B3** bus groups respectively. On the asserted to negated (low to high) transition, data present at the latch inputs will be latched onto the latch output.
CPY__DN#	1	I	Bus Group Copy Down. This signal determines the direction of data flow for data appearing on the four **Bx** copy transceivers. When asserted (low), the transceivers will allow the following potential data transfers: Transceiver-1: From **B1** to **B0**, **A0** D-Latch Transceiver-2: From **B2** to **B0**, **A0** D-Latch Transceiver-3: From **B3** to **B0**, **A0** D-Latch Transceiver-4: From **B3** to **B1**, **A1** D-Latch When negated (high), the transceivers will allow the following potential data transfers: Transceiver-1: From **B0** to **B1**, **A1** D-Latch Transceiver-2: From **B0** to **B2**, **A2** D-Latch Transceiver-3: From **B0** to **B3**, **A3** D-Latch Transceiver-4: From **B1** to **B3**, **A3** D-Latch
B01CPYE#	1	I	**B-BUS** Copy Enable 01. When MOE# is low, this signal enables copy Transceiver-1 to transfer data between **B0** and **B1**. The data direction is determined by the logic value of CPY__DN#.
B02CPYE#	1	I	**B-BUS** Copy Enable 02. When MOE# is low, this signal enables copy Transceiver-2 to transfer data between **B0** and **B2**. The data direction is determined by the logic value of CPY__DN#.
B03CPYE#	1	I	**B-BUS** Copy Enable 03. When MOE# is low, this signal enables copy Transceiver-3 to transfer data between **B0** and **B3**. The data direction is determined by the logic value of CPY__DN#.
B13CPYE#	1	I	**B-BUS** Copy Enable 13. When MOE# is low, this signal enables copy Transceiver-4 to transfer data between **B1** and **B3**. The data direction is determined by the logic value of CPY__DN#.
SLEW	1	I	Output Buffer Slew Rate Control. This signal will provide slew rate control for all external output buffers used in the EBB. When this signal is negated (low), all output buffers will switch at their non-compensated slew rate. When this signal is asserted (high), the output buffers will switch at a slower (>2 ns) slew rate. **NOTE:** When the Slew signal is asserted (high), add 1.5 ns to all A.C. timings.

Symbol	Pin No.	Type	Function
MODE<1..0>	2	I	Operating Mode Pins. These signals determine the mode of operation for the EBB. They are intended to be hardwired to the proper value. Operating mode determination: Mode<1> Mode<0> Operating Mode L L 32-Bit Wide Data Mode w/o Parity #L H 32-Bit Wide Data Mode with Parity H L RESERVED H H EISA Address Mode # Mode described in this section
PARITY<3..0>	4	HCB	During data transfers from the **B-BUS** to the **A-BUS**, these signals are generated within the EBB to provide even byte parity information. During data transfers from the **A-BUS** to the **B-BUS**, these signals are EBB inputs and are used to determine if even byte parity is received by the EBB. During non-copy transfers, the signal associations are: PARITY<0> **A0** **B0** PARITY<1> **A1** **B1** PARITY<2> **A2** **B2** PARITY<3> **A3** **B3** **NOTE:** Parity is not supported during **Bx** to **Bx** byte copies.
PE#<3..0>	4	O	Parity Error. When active (low), these signals will identify incorrect parity for any of the **Ax** bus groups. PE#<3..0> are valid when the associated ABx_OE# are active. The signal associations are: PE#<0> **A0** AB0_OE# PE#<1> **A1** AB1_OE# PE#<2> **A2** AB2_OE# PE#<3> **A3** AB3_OE#
PE_SUM#	1	O	Parity Error Sum. This signal will go active (low) when any valid PE#<3..0> signals goes active (low).

4.2 D.C. Specifications

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

4.2.1 MAXIMUM RATINGS

Temperature Under Bias ..... 0° to +70°C
Storage Temperature ..... −65°C to +150°C
Supply Voltage
with Respect to Ground..... −0.5V to +7V
Voltage on Any Pin ..... −0.5V to +7V

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

4.2.2 D.C. SPECIFICATION TABLE

The following is a table of load capacitances, input voltage levels, current levels, and input leakage currents for the 32-bit parity mode:

T_A = 0°C to +70°C, V_{CC} = 5V ±5%, I_{CC} Max = 190 mA

Signal Name	Pins	Type	I_{OL} Max (mA)	I_{OH} Max (mA)	I_I Max µA	V_I Max (V)	V_H Min (V)	C_{IN} (pF)	C_{LOAD} (pF)
A-BUS<31..0>	8	B	12	−3	10	0.8/0.5	2.0/2.4	27	150
B-BUS<31..0>	8	HCB	24	−3	10	0.8/0.5	2.0/2.4	27	100
MOE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
Ax__OE#	4	I	N/A	N/A	10	0.8	2.0	27	N/A
ABx__OE#	4	I	N/A	N/A	10	0.8	2.0	27	N/A
A__LE#	1	I	N/A	N/A	N/A	0.8	2.0	27	N/A
Ax__LEH	4	I	N/A	N/A	10	0.8	2.0	27	N/A
B01__LE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
B23__LE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
CPY__DN#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
B01CPYE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
B02CPYE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
B03CPYE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
B13CPYE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
MODE<1..0>	2	I	N/A	N/A	10	0.8	2.0	27	N/A
PARITY<3..0>	4	B	12	−3	10	0.8/0.5	2.0/2.4	27	150
PE#<3..0>	4	O	12	−3	10	0.5	2.4	N/A	100
PE__SUM#	1	O	12	−3	10	0.5	2.4	N/A	100

NOTES:
1. I_I Max = I_{LI} for inputs, I_{LO} for outputs (or I/O).
2. V_I Max = V_{IL} for inputs, V_{IL}/V_{OL} for I/O.
3. V_H Min = V_{IH} for inputs, V_{IH}/V_{OH} for I/O.
4. V_{IN} for leakage = 0.45V to V_{CC}.
5. V_{OL} is tested while sinking I_{OL} (24 mA).
6. V_{OH} is tested while sourcing I_{OH} (−3 mA).

4.3 A.C. Specifications

4.3.1 A.C. SPECIFICATION TABLE

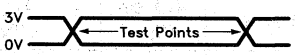
Parameter	From	To	Waveform	Parameter Value	C_L Loading
t_{PLH} or t_{PHL}	A-BUS	B-BUS	1	12.5 ns Max	100 pF
t_{PLH} or t_{PHL}	B-BUS	A-BUS	1	16.00 ns Max	150 pF
t_{PLH} or t_{PHL}	B-BUS	B-BUS	1	16.00 ns Max	100 pF
t_{PLH} or t_{PHL}	BxxCPYE#	B-BUS	1, 2	16.00 ns Max	*100 pF
T_{PLH} or T_{PHL}	BxxCPYE#	A-BUS	1, 2	18.00 ns Max	**150 pF
t_{PLH} or t_{PHL}	CPY__DN#	B-BUS	1, 2	16.00 ns Max	*100 pF
T_{PLH} or T_{PHL}	CPY__DN#	A-BUS	1, 2	18.00 ns Max	**150 pF
t_{PLH} or t_{PHL}	LEH/A__LE#	A-BUS	1, 2	15.00 ns Max	150 pF
t_{PLH} or t_{PHL}	LE#	B-BUS	1, 2	15.00 ns Max	100 pF
t_{PLH} or t_{PHL}	A-BUS	PE#	3	16.00 ns Max	100 pF
t_{PLH} or t_{PHL}	A-BUS	PE__SUM#	3	20.00 ns Max	100 pF
t_{PLH} or t_{PHL}	B-BUS	PARITY	3	19.00 ns Max	150 pF
t_{PZH} or t_{PZL}	OE# or MOE#	A-BUS	4, 5	16.00 ns Max	150 pF
t_{PHZ} or t_{PLZ}	OE# or MOE#	A-BUS	4, 5	16.00 ns Max	150 pF
t_{PZH} or t_{PZL}	OE# or MOE#	B-BUS	4, 5	15.00 ns Max	100 pF
t_{PHZ} or t_{PLZ}	OE# or MOE#	B-BUS	4, 5	15.00 ns Max	100 pF
t_{SU}	A or B-BUS Data before LE		6	3.50 ns Min	
t_{SU}-PE#	Ax Data before LE		6	3.50 ns Min	
t_H			6	3.50 ns Min	
t_W	LE# Pulse Width (Low)		6	4.00 ns Min	

*Data originating from the A-Bus or B-Bus going to the B-Bus.
**Data originating from the B-Bus going to the A-Bus.

NOTE:
1. Increase the above A.C timings by a maximum of 1.5 ns when slew rate control pin is asserted high.

4.3.2 A.C. CHARACTERISTIC WAVEFORMS

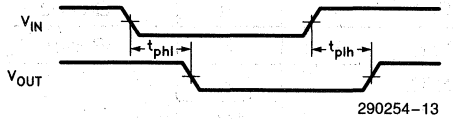
A.C. TESTING INPUT, OUTPUT WAVEFORM



290254-12

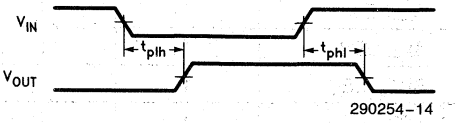
A.C. Testing: Inputs are driven at 3V for a logic "1" and 0V for a logic "0". A.C. Timings are tested at 1.5V.

NOTE:
1. The input waveforms have tr ≤2.5 ns from 0V to 3V.



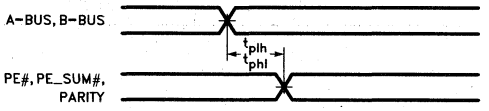
290254-13

Waveform 1. Delay for Outputs



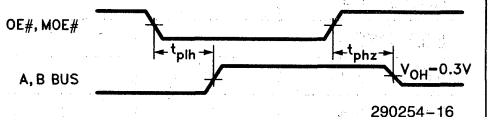
290254-14

Waveform 2. Delay for Outputs



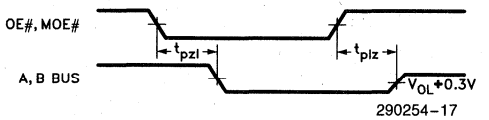
290254-15

Waveform 3. PE*, PE__SUM*, and Parity Output Delays



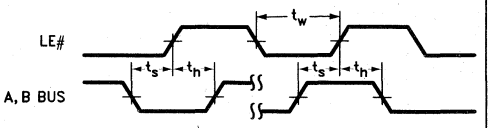
290254-16

Waveform 4. 3-State Output Enable Time to High Level and Output Disable Time from High Level



290254-17

Waveform 5. 3-State Output Enable Time to Low Level and Output Disable Time from Low Level



290254-18

Waveform 6. Data Setup and Hold Times

NOTE:
All timings are referenced to 1.5V unless otherwise stated.

Figure 7. Mode 1

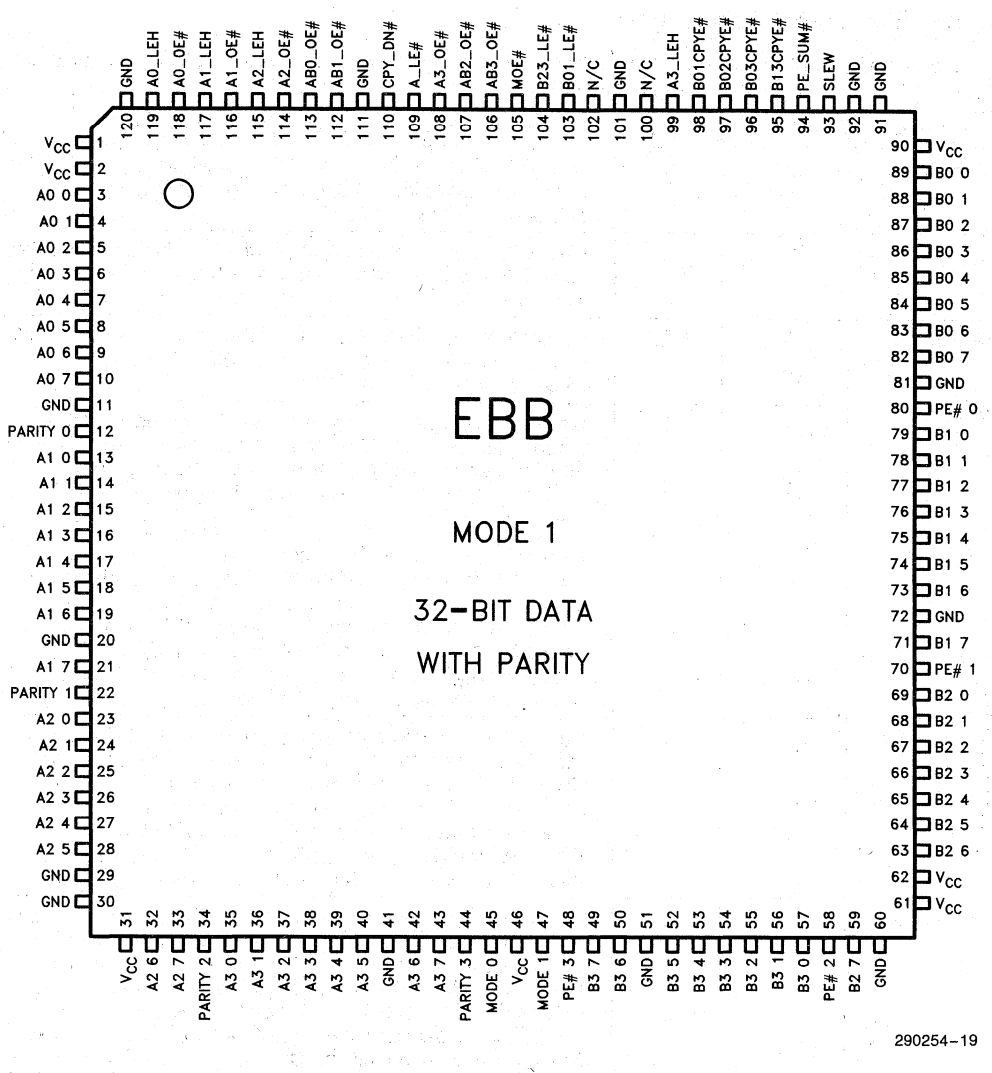
4.4 Pin Summary

The following table indentifies the signals required for the EBB to operate in its 32-bit data mode with parity support:

Signal Name	No. of Pins	Signal Type
A-BUS	32	B
B-BUS	32	HCB
MOE#	1	I
Ax_OE#	4	I
ABx_OE#	4	I
A_LE#	1	I
Ax_LEH	4	I
B01_LE#	1	I
B23_LE#	1	I
CPY_DN#	1	I
B01CPYE#	1	I
B02CPYE#	1	I
B03CPYE#	1	I
B13CPYE#	1	I

Signal Name	No. of Pins	Signal Type
SLEW	1	I
MODE<1..0>	2	I
PARITY<3..0>	4	B
PE#<3..0>	4	O
PE_SUM#	1	O
GND	14	Ground
V_{CC}	7	Power
N/C	2	
	120	
Pin Summary:	97	Signals
	21	Power & Ground
	2	N/C

4.5 Mode 1: 120-Pin Package Pinout



NOTE:
For all "N/C" pins, use an 8.2k pullup resistor.

Figure 8. 120-Pin Package Pinout

5.0 MODE 3 DETAILED DESCRIPTION

5.1 Detailed Signal Description

The following table contains detailed descriptions of each signal while the EBB is in its EISA Address operating mode. Note: For all "N/C" pins, use an 8.2k pullup resistor.

Symbol	Pin No.	Type	Function
A-BUS	31	B	One of three buses manipulated by the EBB in this mode. This bus is intended to interface with the host system bus as discussed in Section 2.3.1.
B-BUS	31	HCB	One of three buses manipulated by the EBB in this mode. This bus is intended to interface with the EISA LA bus as discussed in Section 2.3.1.
S-BUS	18	HCB	One of three buses manipulated by the EBB in this mode. This bus is intended to interface with the EISA SA bus as discussed in Section 2.3.1.
MOE#	1	I	Master Output Enable. This signal is "ORed" with all other output enables used in this mode of operation. When asserted (low), this signal will allow the other output enables to function. When negated (high), all EBB outputs will go into their high-Z states.
A__OE#	1	I	**A-BUS** Output Enable. This signal is used with MOE# to transfer address data onto the Host HA<31..2> address lines. When both MOE# and A__OE# are asserted (low), data present on the outputs of the **A-BUS** D-Latches is driven onto the external **A-BUS** pins. When either MOE# or A__OE# switch to their negated (high) state, the **A-BUS** drivers will transition to a high-Z state.
AB3__OE#	1	I	**A3** to **B3**# Output Enable. This signal is used with MOE# to transfer inverted Host HA<31..24> address data onto the EISA LA#<31..24> address bus. When both MOE# and AB3__OE# are asserted (low), **A3** data present on the **B3**# D-Latch outputs is driven onto the external **B3**# pins. When either MOE# or AB3__OE# switch to their negated (high) state, the **B3**# drivers will transition to a high-Z state.
B012OE#	1	I	**A0, A1** and **A2** to **B0, B1** and **B2** Output Enable. This signal is used with MOE# to: (1) transfer Host HA<23..2> address data onto the EISA LA <23..2> address pins and (2) make the HA<19..2> address data available to the inputs of the EISA SA<19..2> D-Latches. When both MOE# and B012OE# are asserted (low), the **B0–B2** output buffers will drive data present on the **B0–B2** D-Latch outputs onto the external **B0–B2** pins. Simultaneously, the same data is presented to the **S-BUS** D-Latch inputs. When either MOE# or B012OE# are negated, **B0–B2** buffer data is not available on the **B0–B2** pins but **B0–B2** external data will remain on the **S-BUS** latch inputs unless SB__OE# is asserted. (The external pins will typically assume a high-Z state.)

Symbol	Pin No.	Type	Function
S__OE#	1	I	**S-BUS** Output Enable. This signal is used with MOE# to transfer address data onto the EISA SA<19..2> address pins. When both MOE# and S__OE# are asserted (low), **S-BUS** output buffers will drive **S-BUS** D-Latch data to the respective **S-BUS** pins. When either MOE# or S__OE# switch to their deasserted (high) state, the **S-BUS** drivers will transition to a high-Z state.
SB__OE#	1	I	**S-BUS** to **B-BUS** Output Enable. This signal is used with MOE# to: (1) transfer **S-BUS** address data onto the EISA LA<19..2> address pins and (2) make the **S-BUS** available to the inputs of the Host HA<19..2> D-Latches. When both MOE# and SB__OE# are asserted (low), the data present on the S-BUS is driven onto the B-BUS<19..2> external pins. Simultaneously, the same data is presented to the A-BUS<19..2> D-Latch inputs. When either MOE# or SB__OE# are negated, **S-BUS** data is not available on the B-BUS<19..2> pins and **B-BUS** data will appear on the **A-BUS** latch inputs unless B0123OE# is asserted. (The external pins will typically assume a high-Z state.)
A__LE#	1	I	**A-BUS** Latch Enable. This signal is used to latch EISA LA<31..2> address data into the Host HA D-Latches. When A__LE# transitions from asserted to negated (low to high), **B-BUS** data is latched into the **A-BUS** D-Latches. This data will remain latched until A__LE# is reasserted.
B01__LE#	1	I	**B01** and **B1** Latch Enable. This signal is used to control the latching of **A0** and **A1** bus groups to the **B0** and **B1** bus groups respectively. On the asserted to negated (low to high) transition, data present at the latch inputs will be latched onto the latch output.
B23__LE#	1	I	**B2** and **B3** Latch Enable. This signal is used to control the latching of **A2** and **A3** bus groups to the **B2** and **B3** bus groups respectively. On the asserted to negated (low to high) transition, data present at the latch inputs will be latched onto the latch output.
S__LE#	1	I	**S-BUS** Latch Enable. This signal is used to latch EISA LA<19..2> address data into the EISA SA D-Latches. When S__LE# transitions from asserted to negated (low to high), **B-BUS** data is latched into the **S-BUS** D-Latches. This data will remain latched until S__LE# is reasserted.
SLEW	1	I	Output Buffer Slew Rate Control. This signal will provide slew rate control for all external output buffers used in the EBB. When this signal is negated (low), all output buffers will switch at their non-compensated slew rate. When this signal is asserted (high), the output buffers will switch at a slower (>2 ns) slew rate. **NOTE:** When the Slew signal is asserted (high), add 1.5 ns to all A.C. timings.
MODE<1..0>	2	I	Operating Mode Pins. These signals determine the mode of operation for the EBB. They are intended to be hardwired to the proper value. Operating mode determination: Mode<1> Mode<0> Operating Mode L L 32-Bit Wide Data Mode w/o Parity L H 32-Bit Wide Data Mode with Parity H L RESERVED #H H EISA Address Mode # Mode described in this section

5.2 D.C. Specifications

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

5.2.1 MAXIMUM RATINGS

Temperature Under Bias .....0° to +70°C
Storage Temperature .....−65°C to +150°C
Supply Voltage
with Respect to Ground.....−0.5V to +7V
Voltage on Any Pin .....−0.5V to +7V

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

5.2.2 D.C. SPECIFICATION TABLE

The following is a table of load capacitances, input voltage levels, current levels, and input leakage currents.

T_A = 0°C to +70°C, V_{CC} = 5V ±5%, I_{CC} Max = 115 mA

Signal Name	Pins	Type	I_{OL} Max (mA)	I_{OH} Max (mA)	I_I Max µA	V_I Max (V)	V_H Min (V)	C_{IN} (pF)	C_{LOAD} (pF)
A-BUS	30	B	12	−3	10	0.8/0.5	2.0/2.4	27	100
B-BUS	30	HCB	24	−3	10	0.8/0.5	2.0/2.4	27	240
S-BUS	18	HCB	24	−3	10	0.8/0.5	2.0/2.4	27	240
MOE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
A__OE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
AB3__OE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
AB012OE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
SB__OE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
S__OE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
B01__LE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
B23__LE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
A__LE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
S__LE#	1	I	N/A	N/A	10	0.8	2.0	27	N/A
MODE<1..0>	2	I	N/A	N/A	10	0.8	2.0	27	N/A

NOTES:
1. I_I Max = I_{LI} for inputs, I_{LO} for outputs (or I/O).
2. V_I Max = V_{IL} for inputs, V_{IL}/V_{OL} for I/O.
3. V_H Min = V_{IH} for inputs, V_{IH}/V_{OH} for I/O.
4. V_{IN} for leakage = 0.45V to V_{CC}.
5. V_{OL} is tested while sinking I_{OL} (24 mA).
6. V_{OH} is tested while sourcing I_{OH} (−3 mA).

5.3 A.C. Specifications

5.3.1 A.C. SPECIFICATION TABLE

Parameter	From	To	Waveform	Parameter Value	C_L Loading
t_{PLH} or t_{PHL}	A-BUS	B-BUS	1, 2	15.00 ns Max	240 pF
t_{PLH} or t_{PHL}	A-BUS	S-BUS	2	15.00 ns Max	240 pF
t_{PLH} or t_{PHL}	B-BUS	A-BUS	1, 2	15.00 ns Max	100 pF
t_{PLH} or t_{PHL}	B-BUS	S-BUS	2	15.00 ns Max	240 pF
t_{PLH} or t_{PHL}	S-BUS	A-BUS	2	15.00 ns Max	100 pF
t_{PLH} or t_{PHL}	S-BUS	B-BUS	2	15.00 ns Max	240 pF

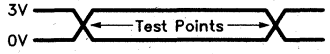
5.3.1 A.C. SPECIFICATION TABLE (Continued)

Parameter	From	To	Waveform	Parameter Value	C_L Loading
t_{PLH} or t_{PHL}	LE#	A-BUS	1, 2	15.00 ns Max	100 pF
t_{PLH} or t_{PHL}	LE#	B-BUS	1, 2	15.00 ns Max	240 pF
t_{PLH} or t_{PHL}	LE#	S-BUS	1, 2	15.00 ns Max	240 pF
t_{PZH} or t_{PZL}	OE#/MOE#	A-BUS	3, 4	15.00 ns Max	100 pF
t_{PHZ} or t_{PLZ}	OE#/MOE#	A-BUS	3, 4	15.00 ns Max	100 pF
t_{PZH} or t_{PZL}	OE#/MOE#	B-BUS	3, 4	15.00 ns Max	240 pF
t_{PHZ} or t_{PLZ}	OE#/MOE#	B-BUS	3, 4	15.00 ns Max	240 pF
t_{PZH} or t_{PZL}	OE#/MOE#	S-BUS	3, 4	15.00 ns Max	240 pF
t_{PHZ} or t_{PLZ}	OE#/MOE#	S-BUS	3, 4	15.00 ns Max	240 pF
t_{SU}	A/B/S-BUS Data before $\overline{LE}$		5	3.50 ns Min	
t_H			5	3.50 ns Min	
t_W	All LE# Pulse Width (Low)		5	4.00 ns Min	

NOTES:
1. Increase the above A.C timings by a maximum of 1.5 ns when the slew rate control pin is asserted high.
2. The EBB outputs driving the EISA bus, identified by a 240 pF load, are guaranteed using a distributed load (refer to Section 10.0).

5.3.2 A.C. CHARACTERISTIC WAVEFORMS

A.C. TESTING INPUT, OUTPUT WAVEFORM



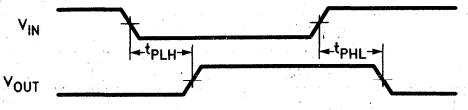
290254-20

A.C. Testing: Inputs are driven at 3V for a logic "1" and 0V for a logic "0". A.C. Timings are tested at 1.5V.

NOTE:
1. The input waveforms have tr ≤2.5 ns from 0V to 3V.

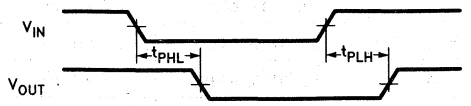
5.3.2 A.C. CHARACTERISTIC WAVEFORMS (Continued)

A.C. TESTING INPUT, OUTPUT WAVEFORM (Continued)



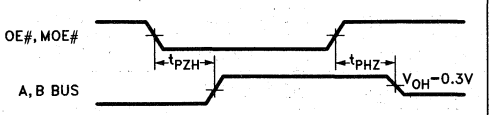
290254-21

Waveform 1. Delay for Inverting Outputs



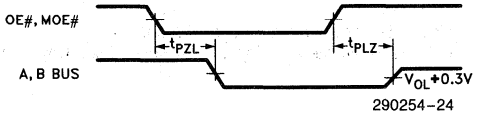
290254-22

Waveform 2. Delay for Non-Inverting Outputs



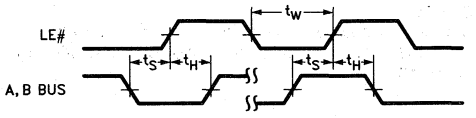
290254-23

Waveform 3. 3-State Output Enable Time to High Level and Output Disable Time from High Level



290254-24

Waveform 4. 3-State Output Enable Time to Low Level and Output Disable Time from Low Level



290254-25

Waveform 5. Data Setup and Hold Times

NOTE:
All A.C. timings are referenced to 1.5V unless otherwise stated.

Figure 9. Mode 3

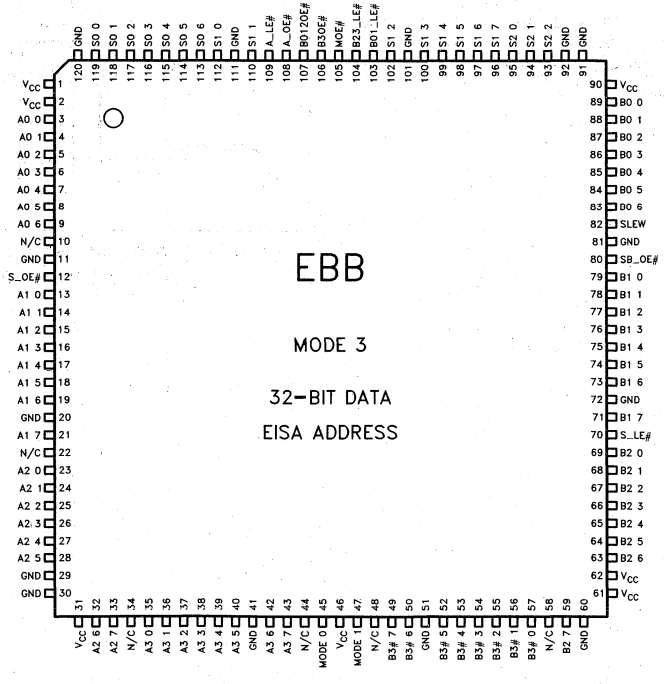
5.4 Pin Summary

The following table indentifies the signals required for the EBB to operate in its EISA Address Mode of operation.

Signal Name	No. of Pins	Signal Type
A-BUS	31	I/O
B-BUS	31	I/O
S-BUS	18	I/O
MOE#	1	I
A_OE#	1	I
AB3_OE#	1	I
B012OE#	1	I
S_OE#	1	I
SB_OE#	1	I
A_LE#	1	I
B01_LE#	1	I
B23_LE#	1	I

Signal Name	No. of Pins	Signal Type
S_LE#	1	I
SLEW	1	I
MODE<1..0>	2	I
GND	14	Ground
V_{CC}	7	Power
N/C	6	
	120	
Pin Summary:	91	Signals
	21	Power & Ground
	8	Reserved

5.5 Mode 3: 120-Pin Package Pinout



NOTE:
For all "N/C" pins, use an 8.2k pullup resistor.

Figure 10. 120-Pin Package Pinout

6.0 82352 CRITICAL FEATURES

6.1 A-BUS Output Buffer Contention (16-Bit Host Application)

The **A-BUS** is capable of continuous operation where **A0** is wired to **A2**, and **A1** is wired to **A3**. When alternating between driving from **A0** and **A1** and from **A2** and **A3** respectively, one output enable is allowed to assert 5 ns before the other output enable negates, at a 4 MHz rate.

6.2 B-BUS Output Buffer Contention (B-Bus to B-Bus re-drive)

The B-Bus is capable of continuous execution of the data-latch and re-drive cycles. For re-drive cycles, the EBB is capable of securing data from the B-Bus and then re-driving it back onto the B-Bus without glitching, while the original data is still driving the B-Bus. This capture and re-drive is accomplished by allowing the data to flow from the B-Bus to the A-Bus and, once the A-Bus has settled, open the A to B latch and assert the A to B output enable. In continuous operation the overlap time with the original data driver will have a 17% duty cycle.

For copy cases, the data is latched in the source byte lane and the copy enable is asserted both before and after the re-drive. No latching is done in the destination byte lanes.

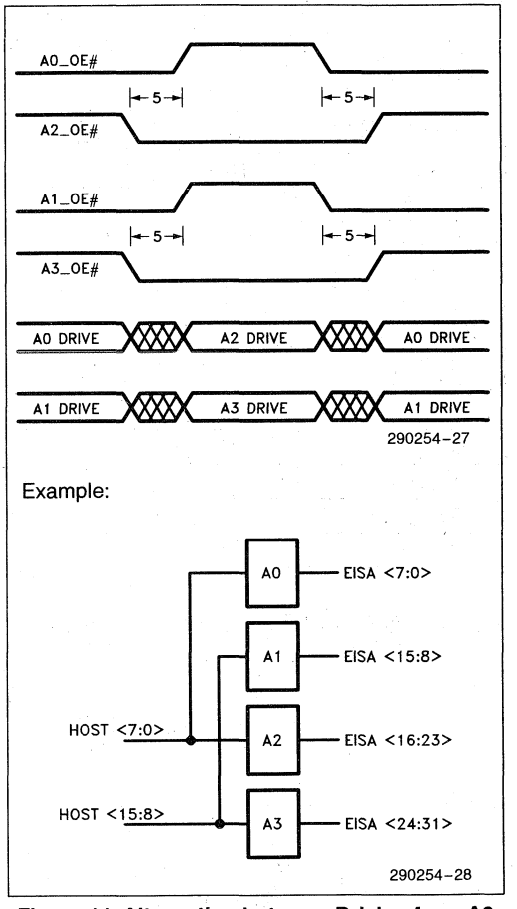


Figure 11. Alternating between Driving from A0 and A1 and from A2 and A3

7.0 82352 DEVICE PINOUT

Pin No.	Mode 0 32-Bit Data w/o Parity	Mode 1 32-Bit Data with Parity	Mode 3 Address
1	V_{CC}	V_{CC}	V_{CC}
2	V_{CC}	V_{CC}	V_{CC}
3	A0 0	A0 0	A0 0
4	A0 1	A0 1	A0 1
5	A0 2	A0 2	A0 2
6	A0 3	A0 3	A0 3
7	A0 4	A0 4	A0 4
8	A0 5	A0 5	A0 5
9	A0 6	A0 6	A0 6
10	A0 7	A0 7	N/C
11	GND	GND	GND
12	N/C	PARITY 0	S_OE#
13	A1 0	A1 0	A1 0
14	A1 1	A1 1	A1 1
15	A1 2	A1 2	A1 2
16	A1 3	A1 3	A1 3
17	A1 4	A1 4	A1 4
18	A1 5	A1 5	A1 5
19	A1 6	A1 6	A1 6
20	GND	GND	GND
21	A1 7	A1 7	A1 7
22	N/C	PARITY 1	N/C
23	A2 0	A2 0	A2 0
24	A2 1	A2 1	A2 1
25	A2 2	A2 2	A2 2
26	A2 3	A2 3	A2 3
27	A2 4	A2 4	A2 4
28	A2 5	A2 5	A2 5
29	GND	GND	GND
30	GND	GND	GND
31	V_{CC}	V_{CC}	V_{CC}
32	A2 6	A2 6	A2 6
33	A2 7	A2 7	A2 7
34	N/C	PARITY 2	N/C
35	A3 0	A3 0	A3 0
36	A3 1	A3 1	A3 1
37	A3 2	A3 2	A3 2
38	A3 3	A3 3	A3 3
39	A3 4	A3 4	A3 4
40	A3 5	A3 5	A3 5
41	GND	GND	GND
42	A3 6	A3 6	A3 6
43	A3 7	A3 7	A3 7
44	N/C	PARITY 3	N/C
45	MODE 0	MODE 0	MODE 0
46	V_{CC}	V_{CC}	V_{CC}

7.0 82352 DEVICE PINOUT (Continued)

Pin No.	Mode 0 32-Bit Data w/o Parity	Mode 1 32-Bit Data with Parity	Mode 3 Address
47	MODE 1	MODE 1	MODE 1
48	N/C	PE# 3	N/C
49	B3 7	B3 7	B3 7
50	B3 6	B3 6	B3 6
51	GND	GND	GND
52	B3 5	B3 5	B3 5
53	B3 4	B3 4	B3 4
54	B3 3	B3 3	B3 3
55	B3 2	B3 2	B3 2
56	B3 1	B3 1	B3 1
57	B3 0	B3 0	B3 0
58	N/C	PE# 2	N/C
59	B2 7	B2 7	B2 7
60	GND	GND	GND
61	V_{CC}	V_{CC}	V_{CC}
62	V_{CC}	V_{CC}	V_{CC}
63	B2 6	B2 6	B2 6
64	B2 5	B2 5	B2 5
65	B2 4	B2 4	B2 4
66	B2 3	B2 3	B2 3
67	B2 2	B2 2	B2 2
68	B2 1	B2 1	B2 1
69	B2 0	B2 0	B2 0
70	N/C	PE# 1	S_LE#
71	B1 7	B1 7	B1 7
72	GND	GND	GND
73	B1 6	B1 6	B1 6
74	B1 5	B1 5	B1 5
75	B1 4	B1 4	B1 4
76	B1 3	B1 3	B1 3
77	B1 2	B1 2	B1 2
78	B1 1	B1 1	B1 1
79	B1 0	B1 0	B1 0
80	N/C	PE# 0	SB_OE#
81	GND	GND	GND
82	B0 7	B0 7	SLEW
83	B0 6	B0 6	B0 6
84	B0 5	B0 5	B0 5
85	B0 4	B0 4	B0 4
86	B0 3	B0 3	B0 3
87	B0 2	B0 2	B0 2
88	B0 1	B0 1	B0 1
89	B0 0	B0 0	B0 0
90	V_{CC}	V_{CC}	V_{CC}

7.0 82352 DEVICE PINOUT (Continued)

Pin No.	Mode 0 32-Bit Data w/o Parity	Mode 1 32-Bit Data with Parity	Mode 3 Address
91	GND	GND	GND
92	GND	GND	GND
93	SLEW	SLEW	S2 2
94	N/C	PE_SUM#	S2 1
95	B13CPYE#	B13CPYE#	S2 0
96	B03CPYE#	B03CPYE#	S1 7
97	B02CPYE#	B02CPYE#	S1 6
98	B01CPYE#	B01CPYE#	S1 5
99	A3_LE	A3_LEH	S1 4
100	N/C	N/C	S1 3
101	GND	GND	GND
102	N/C	N/C	S1 2
103	B01_LE#	B01_LE#	B01_LE#
104	B23_LE#	B23_LE#	B23_LE#
105	MOE#	MOE#	MOE#
106	AB3_OE#	AB3_OE#	AB3_OE#
107	AB2_OE#	AB2_OE#	B012OE#
108	A3_OE#	A3_OE#	A_OE#
109	A_LE#	A_LE#	A_LE#
110	CPY_DN#	CPY_DN#	S1 1
111	GND	GND	GND
112	AB1_OE#	AB1_OE#	S1 0
113	AB0_OE#	AB0_OE#	S0 6
114	A2_OE#	A2_OE#	S0 5
115	A2_LE#	A2_LEH	S0 4
116	A1_OE#	A1_OE#	S0 3
117	A1_LE#	A1_LEH	S0 2
118	A0_OE#	A0_OE#	S0 1
119	A0_LE#	A0_LEH	S0 0
120	GND	GND	GND

NOTE:
For all "N/C" pins, use an 8.2k pullup resistor.

8.0 82352 PACKAGE DIMENSIONS

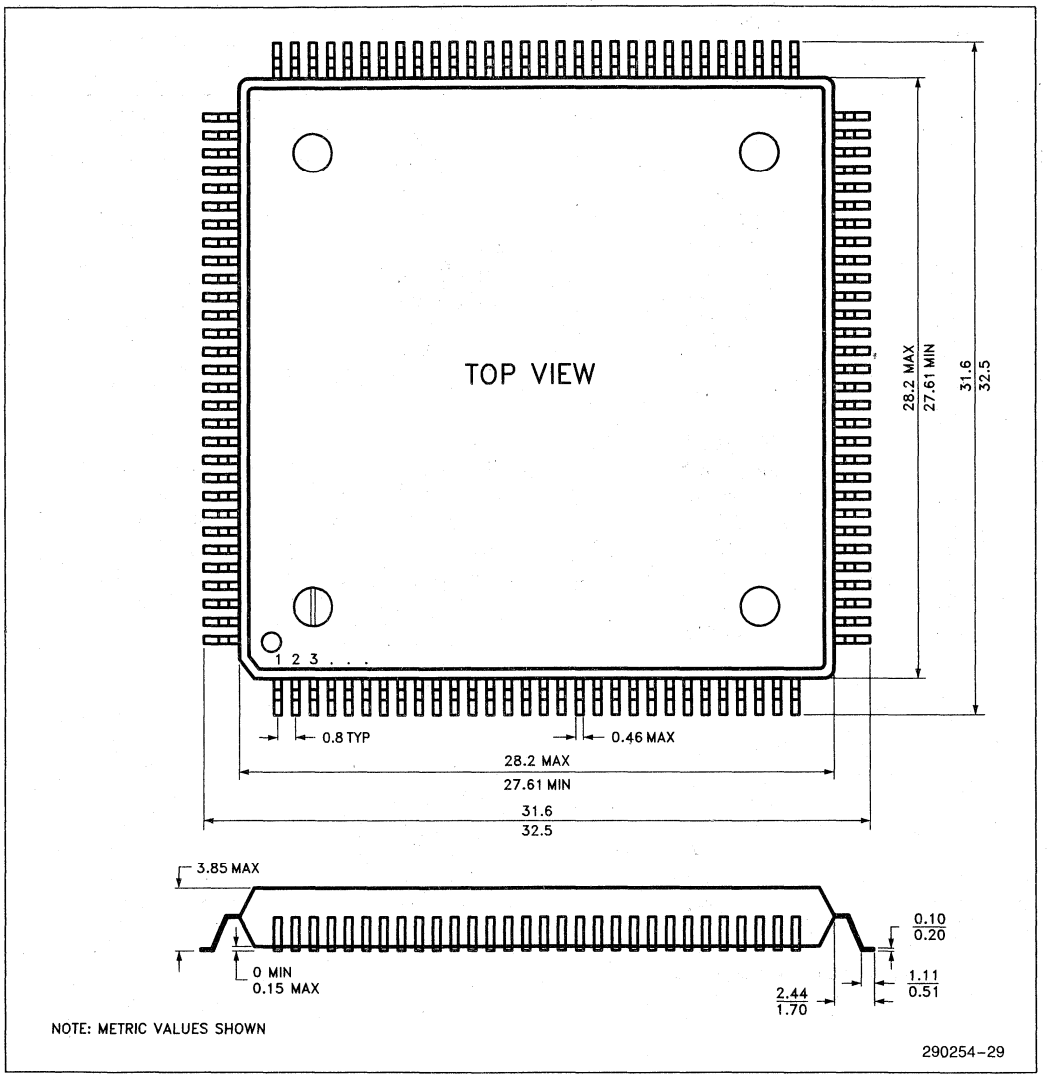
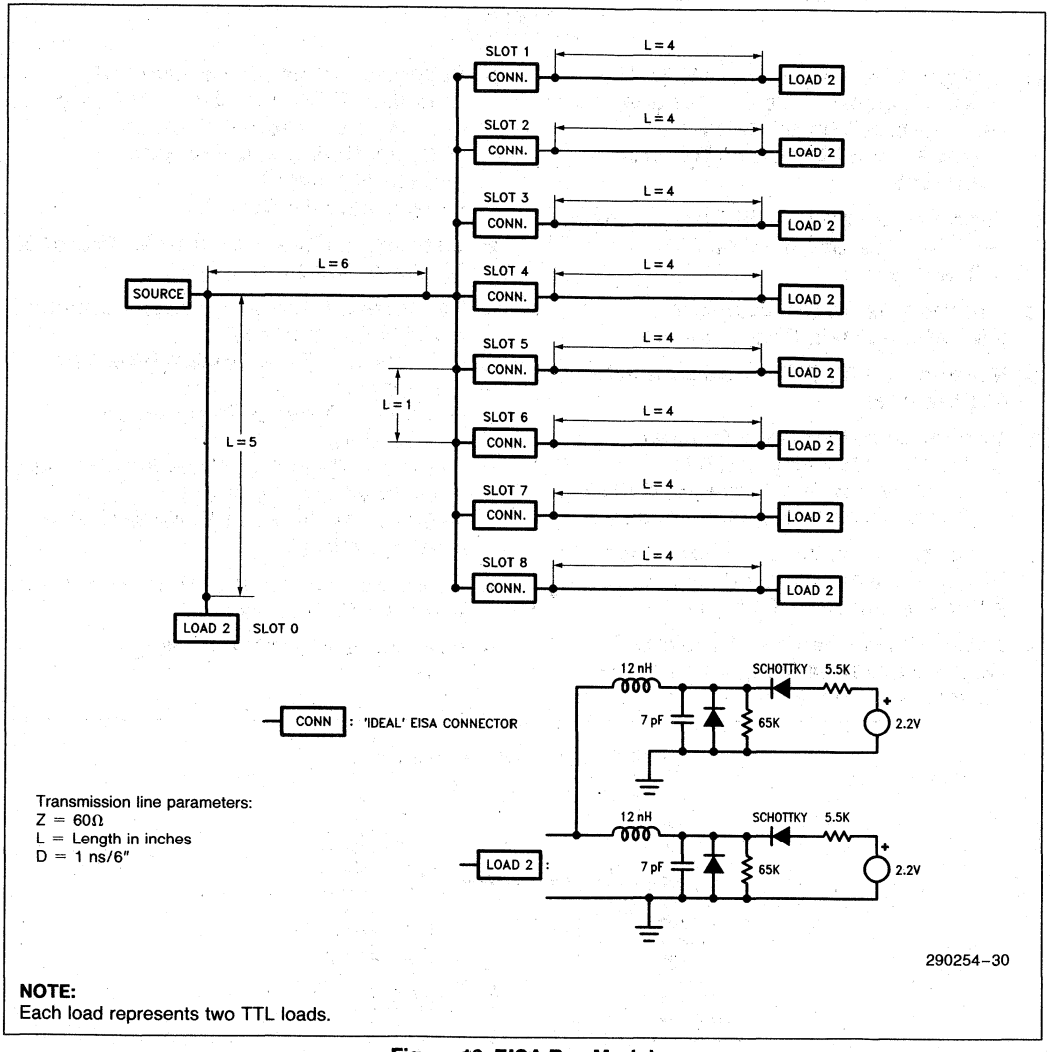


Figure 12. 82352 Package Dimensions

9.0 82352 PACKAGE THERMAL CHARACTERISTICS

Parameter	Air Flow Rate (Ft/Min)			
	0	100	250	500
θ_{JA}	63	48.9	41.9	34.3

10.0 EISA BUS MODEL



NOTE:
Each load represents two TTL loads.

Figure 13. EISA Bus Model

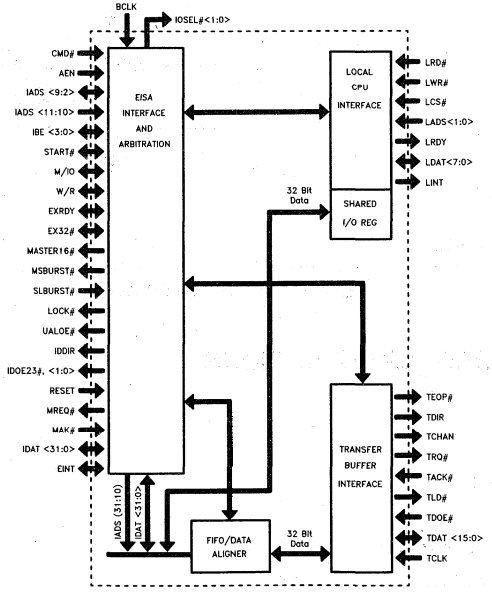
11.0 FREQUENTLY USED ABBREVIATIONS

I = Input Pin
O = Output Pin
B = Bidirectional Pin
HCB = High Current Bidirectional Pin
EBB = EISA Bus Buffer
ISA = Industry Standard Architecture
EISA = Enhanced Industry Standard Architecture
QFP = Quad Flat Pack

82355
BUS MASTER INTERFACE CONTROLLER (BMIC)

- Designed for use in 32-Bit EISA Bus Master Expansion Board Designs
  - — Integrates Three Interfaces (EISA, Local CPU, and Transfer Buffer)
- Supports 16- and 32-Bit Burst Transfers
  - — 33 Mbytes/Sec Maximum Data Transfers
- Supports 32-Bit Non-Burst and Mismatched Data Size Transfers
- Supports 32-Bit EISA Addressability (4 Gigabyte)
- Two independent Data Transfer Channels with 24-Byte FIFOs
  - — Expansion Board Timing and EISA Timing Operate Asynchronously
- Supports Peek/Poke Operation with the Ability to Access Individual Locations in EISA Memory or I/O space
- Automatically Handles Misaligned Doubleword Data Transfers with No Performance Penalty
- Supports Automatic Handling of Complete EISA Bus Master Protocol
  - — EISA Arbitration/Preemption
  - — Cycle Timing and Execution
  - — Byte Alignment
  - — 1K Boundary Detection
- Supports Local Data Transfer Protocol Similar to Traditional DMA
- Supports a General Purpose Command and Status Interface
  - — Local and EISA System Interrupt Support
  - — General Purpose Information Transfers
  - — Set-and-Test-Functions in I/O Space (Semaphore Function)
  - — Supports the EISA Expansion Board ID Function
- Supports Decode of Slot Specific and General I/O Addresses
- 132-Pin JEDEC PQFP Package

82355 Internal Block Diagram



290255-1

October 1990
Order Number: 290255-003

82355 BUS MASTER INTERFACE CONTROLLER (BMIC)

CONTENTS

PAGE

1.0 BMIC TERMINOLOGY/DEFINITIONS 1-429

2.0 BMIC INTERFACE ILLUSTRATION 1-430

3.0 FUNCTIONAL OVERVIEW 1-431
- 3.1 EISA Master and EISA Slave Operations 1-431
- 3.2 82355 Internal Architecture Description 1-431
  - 3.2.1 EISA Interface Block 1-431
  - 3.2.2 Transfer Buffer Interface Block 1-431
  - 3.2.3 FIFO/Data Aligner Block 1-432
  - 3.2.4 Local Processor Interface Block 1-432
  - 3.2.5 Data Transfer Types 1-432
  - 3.2.6 Register Accessing 1-433
    - 3.2.6.1 Register Accessing through the Local Processor Interface 1-433
    - 3.2.6.2 Register Accessing through the EISA Interface 1-433
  - 3.2.7 Interrupts 1-433
    - 3.2.7.1 Interrupt Sources 1-434
    - 3.2.7.2 Interrupt Handling 1-434

4.0 EISA INTERFACE 1-434
- 4.1 EISA Interface Signals 1-434
- 4.2 Transfer Channels 1-435
  - 4.2.1 Burst and Non-Burst Modes of Operation 1-435
  - 4.2.2 1K Page Address Boundary Detection 1-436
- 4.3 Peek/Poke, Locked Exchange Transfers 1-436
- 4.4 Arbitration 1-437
  - 4.4.1 EISA/BMIC Arbitration 1-437
  - 4.4.2 BMIC Preempt Timer 1-437
- 4.5 EISA Address Incrementer 1-440
- 4.6 EISA Byte Decrementer 1-440
- 4.7 EISA Address Incrementer/Byte Decrementer Illustration 1-441
- 4.8 I/O Address Range Decode Support 1-442

5.0 TRANSFER BUFFER INTERFACE 1-443
- 5.1 Transfer Buffer Interface Signals 1-443
- 5.2 External Transfer Buffer Logic 1-444
- 5.3 Transfer Interface Start Address Generation 1-444
- 5.4 Transfer Buffer Interface Timing Example 1-445

CONTENTS

PAGE

6.0 FIFO/DATA ALIGNER 1-447
6.1 FIFO/Data Aligner 1-447
6.2 FIFOs 1-447
6.3 Data Aligner 1-447
6.3.1 EISA Byte Alignment 1-447
6.3.2 Data Assembly/Dissassembly 1-448

7.0 LOCAL PROCESSOR INTERFACE 1-448
7.1 Shared Registers—Status/Command Support 1-448
7.1.1 Semaphore Ports 1-449
7.1.2 Mailbox Registers 1-449
7.1.3 Doorbell Registers 1-449
7.2 Local Processor Recommendations 1-450
7.3 Requirements for No Local Processor 1-450
7.4 EISA ID Function Support/Registers 1-450

8.0 REGISTER DESCRIPTION 1-451
8.1 Shared Register Description 1-451
8.1.1 Command/Status Support Registers 1-451
8.1.1.1 Semaphore Ports 1-451
8.1.1.2 Mailbox Registers 1-451
8.1.1.3 Doorbell Registers 1-452
8.1.1.3.1 Local Doorbell Interrupt/Status Register 1-452
8.1.1.3.2 Local Doorbell Enable Register 1-452
8.1.1.3.3 EISA System Doorbell Interrupt/Status Register 1-452
8.1.1.3.4 EISA System Doorbell Enable Register 1-453
8.1.1.3.5 System Interrupt Enable/Control Register 1-453
8.1.2 Global Configuration Register 1-453
8.1.3 ID Registers 1-453
8.2 Local Processor Only Registers 1-455
8.2.1 Index Registers 1-456
8.2.1.1 Local Index Register 1-456
8.2.1.2 Local Data Register 1-456
8.2.2 Local Status/Control Register 1-456
8.2.3 Data Channel Transfer Registers 1-456
8.2.3.1 Channel 0 and 1 Transfer Base Address Register 1-457
8.2.3.2 Channel 0 and 1 Transfer Current Address Register 1-457
8.2.3.3 Channel 0 and 1 Transfer Base Count Register 1-457
8.2.3.4 Channel 0 and 1 Transfer Current Count Register 1-457

CONTENTS

PAGE

8.2.4 Data Transfer Status/Control Registers 1-458
8.2.4.1 Channel 0 and 1 Transfer Strobe Register 1-458
8.2.4.2 Channel 0 and 1 Transfer Channel Configuration Registers 1-458
8.2.4.3 Channel 0 and 1 Transfer Channel Status Register 1-459
8.2.5 Peek/Poke Registers 1-460
8.2.5.1 Peek/Poke Registers 1-460
8.2.5.2 Peek/Poke Data Registers 1-460
8.2.5.3 Peek/Poke Control Register 1-460
8.2.6 I/O Range Decode Registers 1-461
8.2.6.1 Range 0 and 1 I/O Decode Base Address Register 1-461
8.2.6.2 Range 0 and 1 I/O Decode Control Registers 1-461
8.2.7 Transfer Buffer Interface (TBI) Registers 1-462
8.2.7.1 Channel 0 and 1 TBI Base Address Registers 1-462
8.2.7.2 Channel 0 and 1 TBI Current Address Registers 1-462

9.0 DETAILED PIN DESCRIPTION 1-463
9.1 EISA Interface Signals 1-463
9.2 EISA Buffer Control Signals 1-466
9.3 Address Decode Signals 1-466
9.4 Transfer Buffer Interface Signals 1-467
9.5 Local Processor Interface Signals 1-468
9.6 Power Supplies 1-468

10.0 BASIC FUNCTION TIMING DIAGRAMS 1-469

11.0 D.C. SPECIFICATIONS 1-481
11.1 Maximum Ratings 1-481
11.2 D.C. Characteristics Table 1-481

12.0 A.C. SPECIFICATIONS 1-482
12.1 A.C. Characteristics Tables 1-482
12.2 A.C. Characteristics Waveforms 1-488

13.0 BMIC PIN AND PACKAGE INFORMATION 1-497
13.1 Signal Overview 1-497
13.2 Device Pinout 1-499
13.3 132-Pin PQFP Package Pinout 1-500
13.4 Package Dimensions and Datums 1-502
13.5 Package Thermal Specification 1-504

14.0 BMIC REGISTER ADDRESS MAP 1-504
14.1 Index Register Set 1-504
14.2 Shared Register Set 1-505
14.3 Processor Only Register Set 1-506

The 82355 Bus Master Interface Controller (BMIC) is a highly integrated Bus Master designed for use in 32-Bit EISA Bus Master expansion board designs and supports all of the enhancements defined in the EISA specifications required for EISA bus master applications. The BMIC provides a simple, yet, powerful and flexible interface between the functions on the expansion board and the EISA bus. With the help of external buffer devices, the BMIC provides all EISA control, address, and data signals necessary to interface to the EISA bus.

The primary function of the 82355 is to support 16- and 32-bit burst data transfers between functions on the EISA expansion board and the EISA bus. Data transfer rates of up to 33 Mbytes/sec are supported (the fastest transfer rate available on an EISA bus). The following logic on the BMIC supports efficient burst transfers:

- Arbitration logic, for gaining control of the EISA bus
- Two transfer-address and byte counters
- Two data FIFOs, which allow expansion board and EISA bus timing to operate asynchronously
- Data shifters, which align data to specific byte boundaries
- A transfer buffer interface, for the data transfers on the expansion board
- General-purpose command and status interface logic
- Local processor interface, to allow programming by an on-board processor
- EISA slave interface, to allow communication with the EISA system

The BMIC greatly simplifies the design of EISA expansion boards. With the 82355, a board can be implemented with simple logic similar to that used in traditional ISA DMA designs. The EISA standard allows designs with 32-bit data and address buses, burst transfers, and automatic handling of the full EISA bus master protocol.

To maximize system throughput, the 82355 BMIC incorporates three fully concurrent interfaces: EISA interface, Transfer Buffer interface, and Local Processor interface. The EISA interface incorporates two 24-byte FIFOs, and implements the full EISA protocol. The Transfer Buffer interface is optimized for high speed static RAM buffers, and can operate at a maximum frequency of 20 MHz. The Local Processor interface supports a generic slave interface, and allows the local processor to fully program the BMIC for operation. Local processors are supported with the ability to access individual locations in system memory or I/O space; this peek-and-poke feature allows the expansion board to communicate easily with other devices in the system. All three interfaces can operate simultaneously, thus maximizing overall system performance.

Address-generation support for the data transfer buffer logic on the expansion board is provided on-chip. The transfer logic on the expansion board can use a high-speed asynchronous transfer clock. The BMIC handles all synchronization with the EISA bus. A FIFO within the BMIC eliminates performance degradation on burst transfers caused by synchronization delays. The BMIC also provides a set of programmable address comparators that drive external chip selects on the expansion board to assist local devices in decoding I/O address ranges.

1.0 BMIC TERMINOLOGY/ DEFINITIONS

EISA BUS MASTER—A 32- or 16-bit device that uses the extended part of the EISA bus to generate memory or I/O cycles.

Downshifting Bus Master—A "downshifting" master is a 32-bit master which can convert to a 16-bit master "on the fly". The BMIC will only downshift from a 32-bit master to a 16-bit master if programmed for burst mode (refer to Section 4.3).

EISA READ—A data transfer (burst, non-burst (two BCLK), or mismatched) from system to the expansion board across one of the two transfer channels.

EISA WRITE—A data transfer (burst, non-burst (two BCLK), or mismatched) from the expansion board to system memory across one of the two transfer channels.

I/O ADDRESS DECODE SUPPORT—Refers to slot specific or general I/O address decoding.

Slot Specific Address Decoding—Refers to the decoding of unique addresses allocated to EISA slot specific expansion boards. These addresses are: X000h–X0FFh, X400h–X4FFh, X800h–X8FFh, and XC00h–XCFFh, where X represents the EISA slot number. EISA slot number "0" is reserved for the EISA system board.

General I/O Address Decoding—Refers to the decoding of addresses allocated to ISA expansion boards. These addresses are: 0100h–03FFh.

LOCAL PROCESSOR—A processor located on the expansion board.

SYSTEM CPU—Processor located on the motherboard.

SYSTEM MEMORY—MEMORY located on the EISA bus or motherboard.

TRANSFER INTERRUPTION—A transfer interruption is defined as an occurrence resulting in a break in a transfer caused by one of the following conditions: A FIFO pause, a FIFO stall, a channel preemption, a channel clear or suspension, a 1K page break, a transfer complete (EOP), or a transfer termination due to the external logic driving TEOP# and TACK# low simultaneously.

FIFO Pause—This is a condition where the EISA bus does not provide or take data at a rate fast enough to keep up with the expansion board transfer buffer logic. During an EISA read, this condition is defined as an empty FIFO. During an EISA write, this condition is defined as a full FIFO. A FIFO pause is considered a preferred condition and under normal operations should occur frequently. A FIFO pause will result in the BMIC negating TRQ# until the FIFO becomes not full during an EISA write or not empty during an EISA read.

FIFO Stall—This is a condition where the transfer buffer logic on the expansion board does not provide or take data at a rate fast enough to keep up with the EISA bus. During an EISA read, this condition is defined as a full FIFO. During an EISA write, this condition is defined as an empty FIFO. Under normal operations, a FIFO stall is expected to be a rare and exceptional event. For additional information regarding a FIFO stall, refer to Section 6.2.

Channel Clear—A channel clear results in the immediate termination of the current transfer and the flushing of the channel's corresponding FIFO. A channel clear is initiated by setting the CFGCL bit in the corresponding channel's Configuration register to a 1. For additional information regarding channel clear, refer to Section 8.2.4.2.

Channel Suspension—This temporarily prevents a channel from proceeding with a transfer. A transfer can be temporarily suspended by setting the CFGSU bit in the corresponding channel's Configuration register to a 1.

Channel Preemption—The BMIC can be preempted from the EISA bus by the 82357 (ISP). The 82357 negates MAK#, indicating to the BMIC that it must finish the current bus cycle and relinquish control of the EISA bus by negating MREQ# within 64 BCLK periods. The BMIC is programmable to relinquish the bus within 0, 32, or 64 BCLKs from the negation of MAK# (refer to Section 4.4.2).

1K Page Break—The temporary termination of a burst, non-burst (two BCLK), or mismatched data transfer due to a 1K page address boundary crossing (refer to Section 4.2.2).

Transfer Complete (EOP)—End of process due to the transfer byte count being exhausted or a channel being cleared (channel clear). A transfer complete (EOP) will result in the BMIC asserting TEOP# with the last cycle (refer to Section 5.4).

TRANSFER BUFFER LOGIC—Logic located on the expansion board used to support the transfer and storage of data during BMIC EISA master mode transfers between the expansion board and system memory.

The transfer buffer logic interfaces to the Transfer Buffer interface of the BMIC. Refer to Section 5.2 for additional information regarding transfer buffer logic.

2.0 BMIC INTERFACE ILLUSTRATION

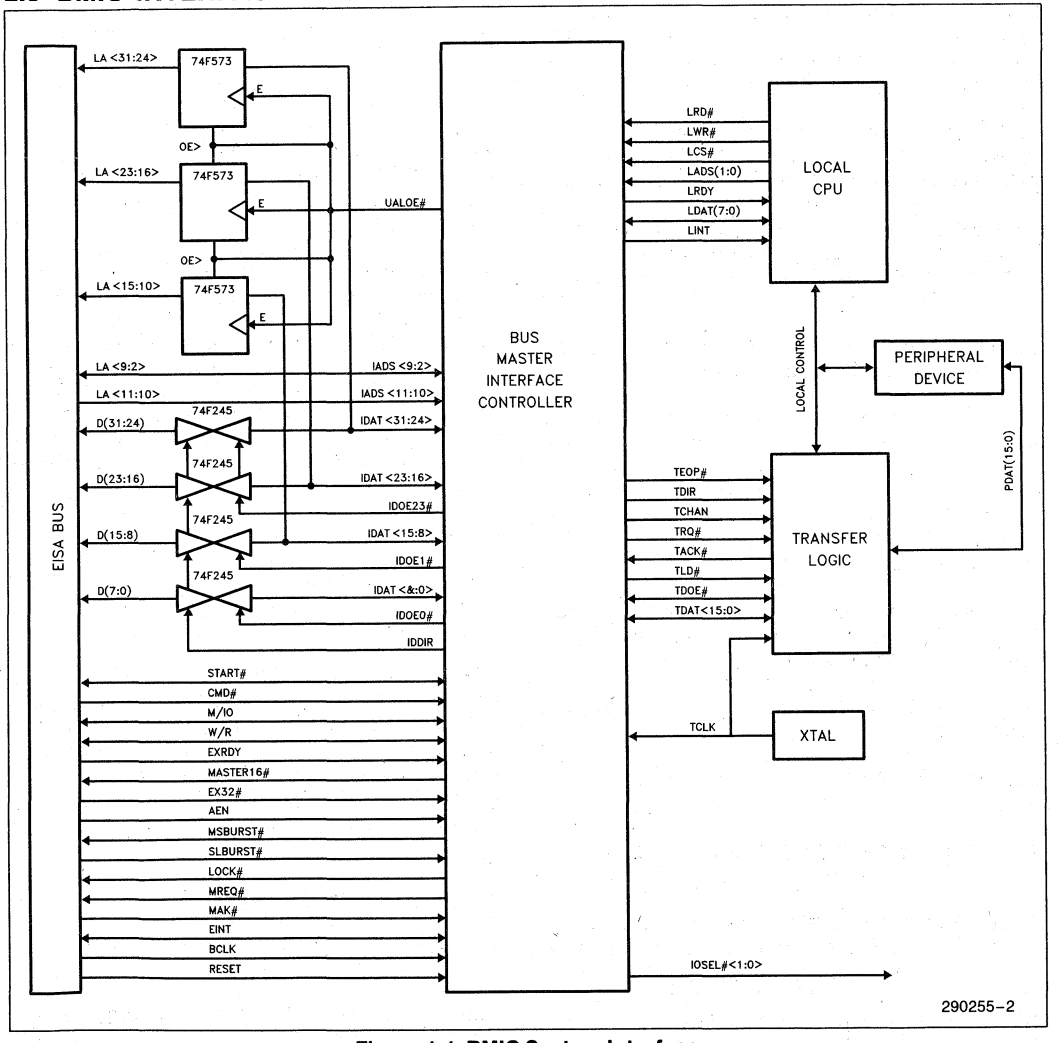


Figure 1-1. BMIC System Interface

3.0 FUNCTIONAL OVERVIEW

The following is a brief discussion of the functional blocks and features of the 82355. The EISA interface, Transfer Buffer interface, FIFO/Data Aligner, and Local interface each have a corresponding detailed section later in this data sheet.

3.1 EISA Master and EISA Slave Operations

In EISA slave mode, the 82355 monitors the EISA address lines <11:2> for general I/O address decoding, slot-specific address decoding, and Shared register accessing. During slave mode operations, all internal registers are accessible through the Local Processor interface, and all Shared registers are accessible through either the Local Processor interface or the EISA interface of the BMIC.

In EISA master mode, the 82355 becomes the master of the EISA bus. It may perform burst, non-burst (two BCLK), mismatched, or Peek/Poke data transfers at this time. During master mode operations, all internal registers are accessible through the Local Processor interface of the BMIC.

The arbiter portion of the BMIC determines which mode the device is in, performs the EISA arbitration, and provides the control signals necessary to regulate the slave and master activities internal to the chip. In slave mode, the arbiter also mediates between the EISA side and the local side during Shared register accesses.

The following is a table of the functions that can be performed during master and slave operations:

	Shared Reg. Accessing	Local CPU Only Reg. Accessing	EISA I/O Address Decoding	Data Transfers
EISA Slave Mode	YES (1, 2)	YES	YES	NO
EISA Master Mode	YES (2)	YES	NO	YES

NOTE:
Shared Reg. Accessing refers to the registers that are accessible through either the EISA interface or Local Processor interface.
Local Processor Only Reg. Accessing refers to the registers that are accessible through the Local Processor interface only.
EISA I/O Address Decoding refers to either general or slot specific I/O decoding support for the expansion board.
Data Transfers refer to either burst, non-burst (two BCLK), mismatched, or peek/poke data transfers.
YES = Can Be Performed
NO = Can Not Be Performed
1 = EISA interface
2 = Local interface

3.2 82355 Internal Architecture Description

The 82355 contains four blocks of control logic. The EISA interface block, Transfer Buffer interface block, FIFO/Data Aligner block, and the Local Processor interface block.

3.2.1 EISA INTERFACE BLOCK

The EISA interface block provides the following functions:

— generates the 32-bit EISA address for burst, non-burst (two BCLK), and peek/poke data transfers
— generates the EISA control signals necessary to implement an EISA 16-bit or 32-bit bus master, and a 32-bit EISA slave
— generates the control signals necessary to enable and disable the external buffer devices
— performs the EISA arbitration and provides the internal control signals required to regulate the slave and master activities of the BMIC
— integrates the registers necessary for the above operations as well as the registers required to provide the configuration and status of the data transfers between the EISA bus and the memory buffer on the expansion board

The EISA memory address range of the 82355 covers the 4 Gigabytes and supports the detection of 1K page address boundaries during burst, non-burst (two BCLK), and mismatched data cycles to and from system memory.

During slave mode, the EISA interface also supports slot specific and general I/O address decode necessary for Shared registers accesses and general decode as required by the expansion board. The shared register addresses are mapped into the slot specific I/O range (C80h-C9Fh).

The EISA interface block contains 43 registers necessary to execute the above functions. A detailed description of the registers and their functions can be found under Register Description (Section 8.2).

3.2.2 TRANSFER BUFFER INTERFACE BLOCK

The Transfer Buffer interface block provides the group of signals that are required to perform 16-bit data transfers to and from the memory buffer on the expansion board. The protocol used is similar to that found in standard DMA designs. The interface includes a 16-bit data bus (TDAT), seven control signals and a transfer clock (TCLK). The transfer clock can run completely asynchronous to the EISA BCLK signal.

The Transfer Buffer interface block also provides a 16-bit transfer start address which is generated at the beginning of all new data transfers to and from the memory buffer on the expansion board. The 16 TDAT data lines are used to transfer the address.

The Transfer Buffer interface block contains eight registers. A detailed description of the registers and their functions can be found under Register Description (Section 8.2).

3.2.3 FIFO/DATA ALIGNER BLOCK

The FIFO/Data Aligner block is used to isolate and simplify the timing relationships between the EISA bus and the bus master expansion board. This allows the transfer buffer logic and EISA bus timing to operate asynchronously. The FIFO provides the data channel between the EISA bus and the expansion board during BMIC master data transfers and the Data Aligner provides the byte alignment and assembly necessary for the EISA bus.

There are two dual-port six doubleword wide (24 byte) FIFOs on-board, one per transfer channel. The data is written into the FIFO from either the EISA bus side or the expansion board side, depending on the direction of the transfer. The transfer direction is controlled by a bit in the Transfer Base Count register set.

3.2.4 LOCAL PROCESSOR INTERFACE BLOCK

The Local Processor interface block provides the interface between the BMIC and the local processor. If a local processor is not present, the processor interface can be connected to the ISA bus. The Local Processor interface block is based on an 8086 style slave mode and provides an 8-bit data path for BMIC programming. All of the BMICs internal registers are accessible through this interface.

The Local Processor interface block a group of Shared registers used to support general-purpose command and status interactions between the system CPU or EISA bus master and the local processor. In addition to the command/status registers, the CPU interface includes a set of ID registers for EISA expansion board ID support, and a set of Peek/Poke data registers used to hold the data during peek/poke operations.

The local interface portion of the BMIC also contains three 8-bit registers which are used by the local processor to access all of the BMICs internal registers. These registers are mapped into the local processor interface and include a local status register, local data register, and a local index register (refer to Section 3.2.6.1).

The Local Processor block contains 31 registers. A detailed description of the registers and their functions can be found under Register Description (Sections 8.1 and 8.2).

3.2.5 DATA TRANSFER TYPES

The BMIC supports four types of data transfers on the EISA bus: Burst, non-burst (two BCLK), peek/poke or locked exchange, and mismatched. For all of the above transfer types, the addressed slave device can negate EXRDY if wait state timing is required (each wait state is one BCLK).

The primary function of the BMIC is to support 16- and 32-bit burst data transfers between functions on the expansion board and the EISA memory. If the addressed memory is not capable of supporting burst transfers, the BMIC will run either 32-bit non-burst (two BCLK) cycles or, with the support of the 82358 EISA bus controller, run mismatched data cycles.

The Burst cycle type provides a continuous sequence of one BCLK read or write cycles to and from 16- or 32-bit EISA memory. Burst cycles can not be used with I/O devices or ISA devices (slave or masters) (refer to Section 4.3).

The non-burst cycle type provides a continuous sequence of two BCLK read or write transfers to and from 32-bit EISA memory. The BMIC will only respond as a 32-bit master when configured for two BCLK transfers (refer to Section 4.3).

The peek/poke and locked exchange feature allows local processor accesses to and from individual I/O space or system memory locations on the EISA bus. The BMIC responds as a 32-bit master and generates two BCLK cycles when configured for peek/poke transfers (refer to Section 4.4). A locked exchange transfer consists of six BCLKs (peek followed by a poke). A peek/poke data transfer has the same timings as a non-burst (two BCLK) data transfer.

The mismatched cycle type provides a means of communicating with 8- or 16-bit EISA or ISA devices. In the event the I/O or memory slave device that has been addressed requires a data size translation, the BMIC will back-off the bus and allow the 82358 EISA Bus Controller to perform the necessary data size translations (refer to Section 4.3). The BMIC will generate mismatched cycles as required for all data transfers (burst, non-burst, peek, poke, or locked-exchange).

The following table identifies the BMIC cycle types, master sizes, slave types accessable (memory-I/O), and BCLKs per cycle.

Transfer Type	BMIC Master Size		Slave Type Accessible		BCLKs per Cycle
	16-Bit	32-Bit	I/O	Memory	
Burst	X	X		X	1
Mismatched		X		X	*
Non-Burst		X		X	2
Mismatched		X		X	*
Peek/Poke		X	X	X	2
Mismatched		X	X	X	*
Locked Exchange		X	X	X	6
Mismatched		X	X	X	*

*Depends on slave type/size (EISA/ISA, I/O/Memory, 8-bit/16-bit)

For all of the above transfer types, the addressed slave device can negate EXRDY if wait-state timing is required (each wait-state is one BCLK).

3.2.6 REGISTER ACCESSING

The BMIC provides three distinct groups of registers; the Shared register set, the Local Processor Only register set, and the Index register set. The Shared register set is used by the system CPU or EISA bus master and the local processor for general-purpose command and status interactions and expansion board ID support. The Local Processor only registers are used by the local processor to program the BMIC and provide status for data transfers across the EISA bus and Transfer Buffer interface. The Local Processor Only register set also provides address range decode support for slot specific and general I/O address ranges of interest to the expansion board. The Index register set is used by the local processor as a means of accessing all of the above registers through an indexing scheme.

The Shared register set is accessible through either the EISA interface or the Local Processor interface, the remaining two register sets are accessible through the Local Processor interface only. In the case of contention between the EISA bus and the local processor accessing a Shared register simultaneously, the local processor on the expansion board will have priority.

3.2.6.1 Register Accessing through the Local Processor Interface

Register accessing on the local side of the BMIC is accomplished using an indexing procedure. The local interface portion of the BMIC contains two 8-bit registers which are used by the local processor to access all of the BMIC's internal registers. These registers are mapped into the Local Processor interface and include a local data register and a local index register. The registers are selected using the two local address lines (LADS<1:0>). The BMIC's internal register set is read by writing the address of the register to be accessed into the local index register. The register contents are then read through the Local Data register. To write to one of the BMIC's internal registers, the local processor must first write the address of the register to be accessed into the local index register, same as a read, then write the new data value to the Local Data register.

An optional auto-increment mode is supported by the BMIC, which automatically increments the index register after each register read or write. This allows for efficient programming of the register set by using byte string moves. If the Local Index register is given a local index address with bit (7) set high, the local index address will automatically increment every time the Local Data register is read or written.

The Local Status/Control register is directly mapped into the Local Processor interface and is also accessible using the two address lines (LADS<1:0>).

3.2.6.2 Register Accessing through the EISA Interface

The shared registers are mapped directly into the EISA slot-specific I/O space XC80–XC9F. The EISA address lines <11:2> and the byte enables <3:0> are used for decode during shared register accesses.

A standard slave read or write access to the BMIC consists of two BCLKs + one wait-state (one wait-state = one BCLK period). During a slave cycle where the EISA access loses the internal register access through arbitration to the local processor, the cycle will consist of two BCLKs + two wait-states. The BMIC will negate EXRDY for one BCLK for each wait-state required.

3.2.7 INTERRUPTS

The BMIC provides two interrupt request lines, one for the EISA side (EINT), and one for the local side (LINT). The EISA interrupt (EINT) can be programmed for either edge or level-triggered operations. During edge-triggered operations the EINT signal will transition from a low level to a high level. in level-triggered mode, the EISA interrupt signal is an active low open collector output. The local interrupt signal (LINT) can be programmed for either active low or active high level operations and will default to active low operation upon reset. The LINT signal is not an open collector output during active low operations and will require external logic if interrupts need to be tied together on the local side. The EINT and LINT modes of operation are programmed through the Global Configuration register.

3.2.7.1 Interrupt Sources

Several events can trigger each of the two interrupt request signals, and the events can be enabled or disabled on an individual or global basis (refer to Sections 8.1.1.3 and 8.2.2). The system CPU or EISA bus master can only be interrupted by an I/O write from the local processor to the BMIC EISA System Doorbell register. However, the local processor can be interrupted by several sources which are listed below:

— An I/O write from the system CPU or EISA bus master to the BMIC Local Doorbell register.

— The completion of a data transfer on one of the transfer channels.

— The termination of a data transfer by an external device which asserted the end-of-process (TEOP#) and TACK# simultaneously.

3.2.7.2 Interrupt Handling

To prevent the BMIC from allowing undetected interrupts from occurring, when servicing an interrupt initiated by the BMIC, all additional interrupts must be disabled prior to reading the Local or EISA System Doorbell Status registers. The interrupts are disabled by writing to the Local or EISA System Doorbell Enable registers, depending on the source of the interrupt.

This is required due to the nature of the interrupt mechanism of the BMIC. All interrupt sources have an edge triggered nature internal to the BMIC, with each event being 'OR'ed together. Additional interrupt sources occurring after the first interrupt will set their appropriate bit in the Status register, but they will not generate an external interrupt until the initial event has been cleared. Thus if the Status register was read first, and another interrupt occurred after this read, the second interrupt would remain undetected in the status register until another event occurred. Disabling of the interrupts prior to reading the status register will prevent this from occurring.

4.0 EISA INTERFACE

4.1 EISA Interface Signals

The BMIC provides a complete interface to the EISA bus and supplies all of the control signals, data lines, and address lines necessary to implement a 16- or 32-bit EISA bus master and a 32-bit EISA slave. This includes a 32-bit data path, a 32-bit address path, and 20 EISA control signals. The BMIC also provides five control signals used to enable and disable the external data buffers and address latches, as shown in Figure 2-1.

The BMIC uses four 74F245 external bidirectional buffers to drive and receive the 32 EISA data and three 74F573 external latches to latch and drive the upper 22 EISA address lines. The external data buffers and address latches should be comprised of "F" or "AS" type logic to meet EISA speed requirements.

The upper 22 EISA addresses are multiplexed through the 22 upper EISA data lines of the BMIC. They are latched externally by the 74F573's. EISA address lines <11:2> and byte enable lines <3:0> are tied directly to the EISA bus. Address lines 10 and 11 are input directly to the BMIC for slave mode address decode. During EISA master operation, lines 10 and 11 are driven indirectly through the external latches.

As a slave, the BMIC receives address lines IADS<11:2> and byte enable lines IBE<3:0># for I/O address decode. Address lines <11:2> are used for slot specific decode, and address lines <9:2> are used for general I/O address decode. Address lines <11:2> along with BE#<3:0> are used by the BMIC during Shared register accesses. Address lines <31:12> are not used by the BMIC in slave mode.

The following address lines are used during I/O decoding as shown:
Slot specific I/O address decoding (expansion board)—IADS<11:2>
Slot specific I/O address decoding (shared registers)—IADS<11:2>/IBE<3:0>#
General I/O address decoding (expansion board)—IADS<9:2>

All of the BMIC EISA control signals function as defined in the EISA bus specification. The signals are used to support the following cycles:

BMIC as a Master

	(Cycle Type Performed)			
Master Type	**Burst**	**Non-Burst**	**Mismatched**	**Peek/Poke/Locked Exchange**
32-Bit	X	X	X	X
16-Bit	X			

BMIC as a Slave

1. Responds to EISA shared register accesses as 32-bit slave.
2. Responds to slot specific and general I/O accesses (refer to Section 4.8).

4.2 Transfer Channels

The BMIC contains two identical independent transfer channels which are configurable to run either burst or non-burst (two BCLK) cycles to and from system memory. The BMIC will automatically run non-burst (two BCLK) or mismatched cycles if the memory the BMIC has addressed cannot run burst cycles. Mismatched cycles will be run if data size translation is required.

Each channel has three sets of registers to regulate data transfers. These are the Base register group, the Current register group, and the Data Status/Control register group. This implementation of a triple register set allows a processor to begin programming the next transfer on the channel while the current transfer is being executed.

The Base register set contains seven 8-bit registers. These registers are programmed by the local processor when a transfer is required across one of the channels. Four Transfer Channel Base Address registers are combined to form the starting 32-bit EISA address to be used during the transfer. The remaining three registers are the Transfer Channel Base Count Registers. The Base Count registers are combined to determine the number of transfers (in bytes) to be performed. The number of bytes which can be transferred ranges from 1 byte to 4 Mbytes. The most significant bit of the Transfer Channel Base Count register group is used to control the start of the transfer and the second most significant bit is used to control the direction of the transfer (refer to Section 8.2.3.3).

The Current register set contains seven registers each of which correspond to a Base registers. These registers are loaded from the Base registers. The Transfer Channel Current Address registers contain the 32-bit real-time EISA memory address. The Transfer Channel Current Count registers contain the number of bytes remaining to be transferred on the channel. The current register set is readable by the local processor. However, there are possible coherency problems involved with reading multiple bytes while the current registers are being updated during a transfer. To avoid these problems, a channel's transfer should be temporarily suspended (using the channel's Configuration Register) before trying to read the channel's current register set.

The Status/Control register set contains three registers: the Transfer Channel Strobe register, Transfer Channel Configuration register, and the Transfer Channel Status register. The Transfer Channel Strobe register is used to initiate the transfer of data from the Base register set to the associated Current register set. A transfer request for that channel will be generated following the Current register load. The Transfer Channel Configuration register is used to program the mode of the transfer. The Transfer Channel Status register provides current FIFO and transfer channel status.

To initialize a transfer over either of the two transfer channels, the following steps must be completed:

1. Verify that the Base registers for the desired transfer channel are available.

   The Transfer Channel Base and Count registers must be available before they can be programmed. This is determined by the status of bits 0 and 1 in the Local Status/Control register. A "1" in either of the two bits indicates that the corresponding channel is currently running a transfer and the Base registers are busy. A "0" indicates that the Base registers are free and available for programming. In the event that the Base registers are not available, the local processor must wait until the data transfer executing on the requested channel has completed, at which time bits "0" or "1" (depending on which channel was programmed) in the Local Status/Control registers will be reset to 0. Programming the Base registers during a Base register Busy state, is illegal and will corrupt the Base register data of the pending transfer. Programming the Transfer Configuration register during a cycle in progress may cause the termination of the transfer, depending on which bit in the register was changed.
2. Program the transfer channel's associated Transfer Base register set with the desired transfer information (Base registers must be available).
3. Initiate the Base register to Current register load and schedule a transfer request by writing to the channel's Transfer Strobe register.

   If a transfer is in progress on the requested channel and a write to the associated channel's Strobe register is done, the Base to Current register load will take place immediately after the date transfer on the requested channel has completed.

4.2.1 BURST AND NON-BURST MODES OF OPERATION

The BMIC can be programmed for burst or non-burst (two BCLK) data transfers to and from EISA memory. This is determined by a write to the Channel Configuration Register.

If burst mode is enabled, the BMIC will look for the SLBURST# signal at the beginning of the transfer to determine if the slave device that was addressed is

capable of running burst cycles. If the slave device does not respond with an active SLBURST# signal, the BMIC will not activate the MSBURST# signal and will proceed with either non-burst (two BCLK) bus cycles or mismatched cycles.

In burst mode, the BMIC can respond as a 16- or 32-bit master. The BMIC informs the system of this capability by driving MASTER16# low from the same BCLK rising edge that START# is asserted. MASTER16# will remain low for one BCLK. The BMIC will automatically "downshift" from a 32- to a 16-bit master if the EX32# signal is sampled inactive and the SLBURST# signal is sampled active at the beginning of a transfer. If EX32# and SLBURST# are sampled active at the beginning of the transfer, the BMIC will proceed with a 32-bit burst transfer.

In non-burst mode, the BMIC will respond as a 32-bit master. The BMIC will look for the EX32# signal at the beginning of the transfer to determine if the system memory it has addressed has the same bus width. If the EX32# signal is not returned (mismatched cycle indicated), the BMIC will "back-off" the bus by floating START#, IBE#<3:0>, and IDAT<31:0> to allow the 82358 EISA Bus Controller to take control of the transfer. The EISA Bus Controller will then proceed to assemble or disassemble the data as needed. The EISA Bus Controller will return the EX32# signal after the mismatched cycle is complete, indicating to the BMIC that a new address can be placed on the bus. If the EX32# signal is sampled active at the beginning of the transfer, the BMIC will proceed with a 32-bit non-burst (two BCLK) transfer.

4.2.2 1K PAGE ADDRESS BOUNDARY DETECTION

During burst, non-burst (two BCLK), and mismatched data cycles, the BMIC provides the support to detect 1K page address boundary crossings. If the BMIC detects that the current cycle is about to cross a 1K page boundary, the transfer will be terminated on the next cycle. The BMIC will then arbitrate between restarting the transfer on the current channel, selecting the second channel, doing a peek/poke cycle, or preempting the channel (refer to Section 4.4 for information regarding BMIC arbitration).

Example: Transfer = 32-bit transfer and page address boundary is at location 400h = 1024

1. The BMIC detects that the current cycle is about to cross a 1K page address boundary—current address (3FCh = 1020).
2. Address after BMIC has executed the current cycle (400h = 1024).
3. Transfer is terminated.
4. BMIC will now arbitrate between restarting the transfer on a new page, selecting the second channel, doing a peek/poke cycle, or preempting the channel.

4.3 Peek/Poke, Locked Exchange Transfers

To allow the local processor to communicate with other devices in the main system, the BMIC allows the local processor to execute individual I/O or memory cycles over the EISA bus. These cycles can be thought of as being similar to "peek" and "poke" statements in the Basic programming language. These cycles may be reads, writes, or locked exchanges in 8-, 16-, 24-, or 32-bit values. All cycles must be contained within a single doubleword.

The Peek/Poke operation requires the following set of registers: Four 8-bit Peek/Poke Address registers which are combined to provide the 32-bit Peek/Poke address; One 8-bit Peek/Poke Control register which contains the bits defining whether the cycle is I/O or memory, peek (read)/poke (write) or locked exchange, and which byte enables are to be active during the cycle; and four 8-bit Peek/Poke Data registers which are used to hold the data for the Peek/Poke cycle. During all peek/poke or locked exchange cycles, byte enables IBE<3:0># are derived from bits 0–3 in the Peek/Poke Control register set. The lower two bits of address register are ignored. Peek, poke, or locked exchange cycles will not be generated for illegal combinations of byte enables (i.e., 1111, 1010, 0110, 0101, 0100, 0010).

To do an individual write cycle (poke), the local processor must first write to the Peek/Poke Address register set to specify the 32-bit memory address or the 16-bit I/O address. It must then write the data to be transferred into the Peek/Poke Data register set. The data must be placed in the appropriate byte positions in the Data register set so that it goes out on the correct byte lanes during a 32-bit bus master transfer.

Once the appropriate data and address have been programmed, the local processor must write to the Peek/Poke Control register to specify the cycle type and initiate the cycle. After this write to the Peek/Poke Control register, bit 2 in the Local Status/Control register will be set to a 1 by the BMIC to indicate that a peek/poke request is pending and that the peek/poke registers are busy. When the poke cycle has finished executing on the EISA bus, the Peek/Poke status bit 2 in the Local Status/Control register will return to normal (0).

To do an individual read cycle (peek), the local processor must write to the Peek/Poke Address register, then to the Peek/Poke control register to initiate the read cycle. The peek/poke status bit 2 in the Local Status/Control register will be set high by the BMIC and remain active until the peek cycle finishes on the EISA bus. The local processor can then read the data from the peek/poke data register.

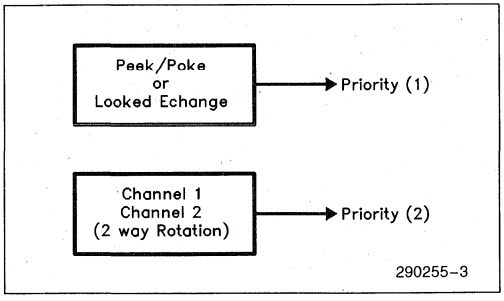
NOTE:
When running consecutive peek transfers, the data must be read from the Peek/Poke data registers before each new peek transfer is generated. The BMIC will read the data off the EISA bus from all four byte lanes regardless of which Byte enables (BE<3:0>#) are active. (Although all bytes are read, the value of the byte enables are important to the system and must be programmed for the peek transfer).

When a locked exchange cycle is requested by the local processor, a peek cycle is scheduled first and then immediately followed by a poke cycle. The LOCK# signal is active during the locked exchange cycle to indicate to the system that no othe accesses to the addressed location can be made.

4.4 Arbitration

4.4.1 EISA/BMIC ARBITRATION

The BMIC will begin master mode operation any time a transfer request is pending. If more than one transfer request is pending, the BMIC will service them in the following order. Peek/poke cycles have the highest priority access to the EISA bus followed by the two data channels. Once the BMIC has gained control of the EISA bus, the BMIC will first perform any peek, poke, or locked exchange transfers that may be pending. If there are no peek, poke, or locked exchange transfers pending, the BMIC will run data transfers initiated by either of the two transfer channels. The two transfer channels have equal priority with respect to each other and are serviced in an alternating fashion. The priorities and assignments are as follows:



The BMIC will maintain ownership of the EISA bus until it has serviced all outstanding data transfer requests or it is preempted from the bus by the removal of the MAK# signal. The BMIC can be configured to relinquish the EISA bus immediately, 4 μs, or 8 μs after a preempt is received. If the BMIC has completed all outstanding data transfer requests prior to the time-out of the preempt timer, it will give up the bus. If the BMIC finishes one task prior to the time-out of the preempt timer, it will start on the next pending transfer request unless the request is a peek, poke, or locked exchange cycle. The BMIC will not start a set of peek, poke, or locked exchange cycles after the MAK# signal has been removed. If a transfer is cut-off due to a preempt timer time-out, the BMIC, upon regaining access to the EISA bus and following its internal arbitration priority scheme, will continue the transfer that was preempted at the point the transfer was cut-off.

When a channel is interrupted for any reason, 1K page break, FIFO stall, channel clear, transfer suspended, transfer complete, or transfer terminated, the BMIC may immediately relinquish the EISA bus depending on the state of the CFGFF bit in the Channel Configuration register set.

NOTE:
During a FIFO pause, the CFGFF bit in the associated Channel's Configuration register is ignored. The function of the CFGFF bit, as related to the above channel interruptions, is as follows:

If the CFGFF bit = 1, the BMIC will immediately relinquish control of the EISA bus upon the detection of any of the above interruptions. This will occur regardless if there are additional data transfer requests pending. If there are additional data transfer requests pending, the BMIC will reassert MREQ# a minimum of two BCLKs later to reacquire the EISA bus. The BMIC will follow the arbitration priority scheme outlined above when servicing a data transfer request after a transfer interruption has occurred.

If the CFGFF bit = 0, the BMIC will retain ownership of the EISA bus upon detection of any of the above interruptions as long as there are data requests pending and a preempt timer time-out has not occurred. If there are no additional data requests pending, the BMIC will relinquish ownership of the EISA bus after the current transfer interruption has been serviced and completed.

4.4.2 BMIC PREEMPT TIMER

The BMIC can be preempted from the EISA bus by the 82357 (ISP). The 82357 negates MAK#, indicat-

ing to the BMIC that it must finish the current bus cycle and relinquish control of the EISA bus by negating MREQ# within 64 BCLK periods (8 μs).

The BMIC provides a programmable preempt timer which can be programmed to relinquish the bus within 3, 32, or 64 BCLKs. The preempt timer is programmable through the Global Configuration register.

The following diagrams illustrate the latest the BMIC will start a new transfer after MAK# has been negated.

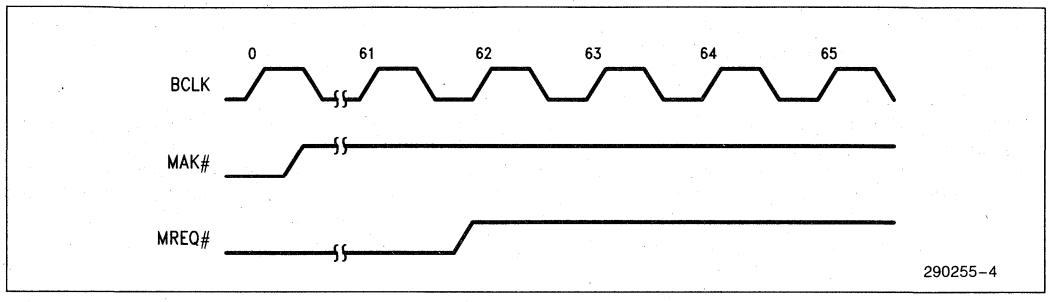
Depending on the type of transfer started, the BMIC will respond as follows:

Assumptions:

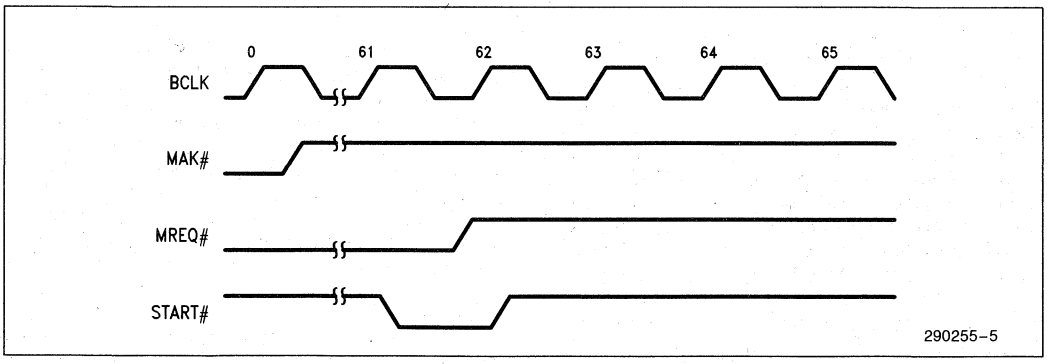
1. The 82357 has negated the MAK# signal at BCLK zero.
2. The preempt timer is programmed to relinquish the EISA bus within 64 BCLKs after the negation of MAK#.
3. Let X = programmed value of preempt delay (in BCLKs).

BMIC Response:

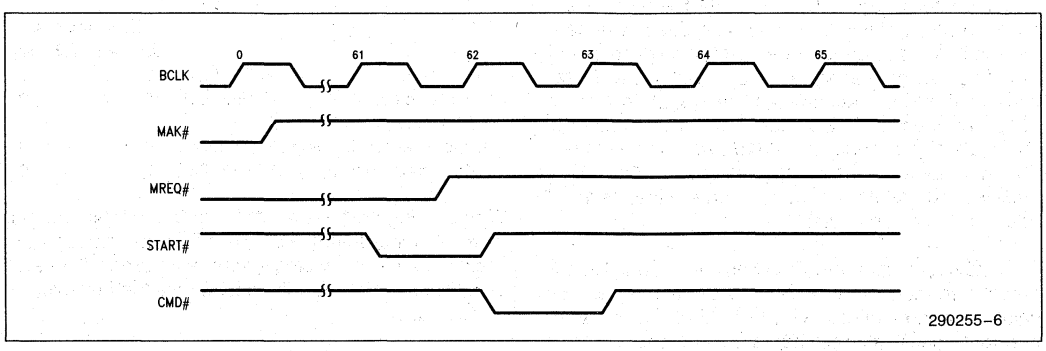
For all transfers, the BMIC will negate MREQ# within (X-2.5) BCLK periods following the MAK# transition to an inactive state (BCLK 61.5).



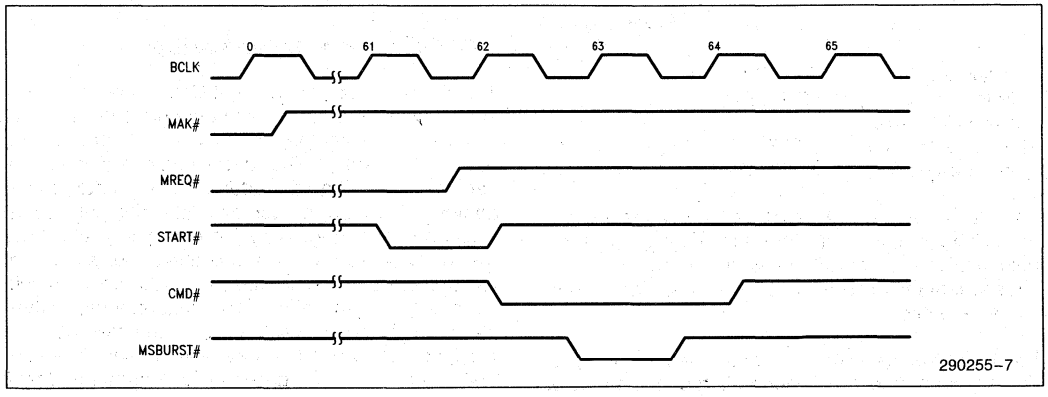
For all transfers, the BMIC may assert START# on any of the first X-3 rising edges of BCLK following the MAK# transition to an inactive state (BCLK 61).



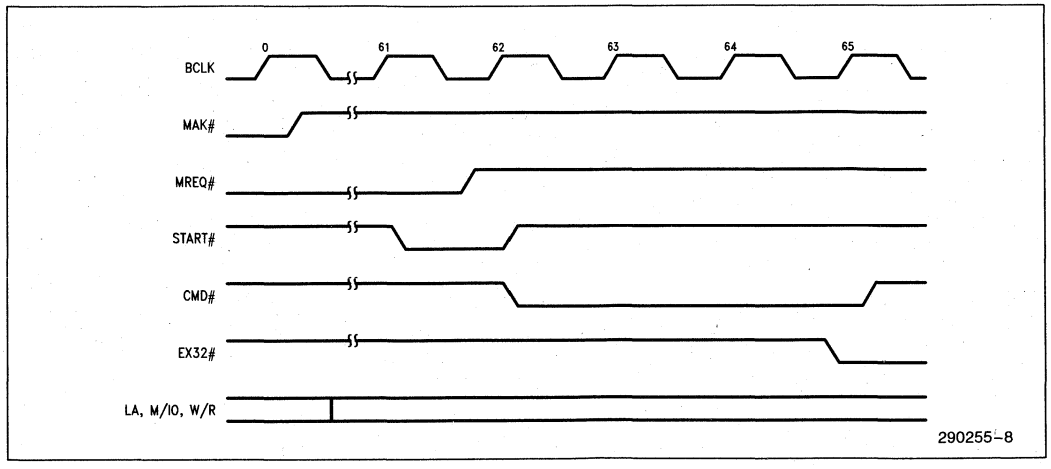
If the last cycle is a non-burst two BCLK cycle, CMD# will become inactive within (X-1) BCLK periods from the inactive transition of MAK# (BCLK 63), this is assuming that EXRDY is active.



If the last cycle is a burst EISA cycle, the BMIC will negate MSBURST# within (X-0.5) BCLK periods from the inactive transition of MAK# (BCLK 63.5). The last CMD# will go inactive within X BCLK periods from the deassertion of MAK# (BCLK 64). This is assuming EXRDY is active.



If the last cycle is a mismatched, cycle completion will be controlled by the system. The BMIC will drive the LA address, M/IO, and W/R signals until the falling edge of BCLK after the last CMD# inactive transition.



4.5 EISA Address Incrementer

The Transfer Channel Current Address register set for each channel functions as an address incrementer and is used to generate and track the address of the data during transfers. The register set increments the address according to the number of bytes being transferred during that cycle. The transfer is automatically aligned on doubleword boundaries. The two least significant bits of the starting 32-bit address (A0 and A1) are used to determine the initial address increment value.

For 32-bit transfers, the BMIC provides an initial address increment of 1, 2, 3 or 4 depending on the value of address lines A<1:0>. After the initial increment, the BMIC increments the address by 4 until the last cycle is detected.

The following example illustrates the BMIC address incrementer during a 32-bit master mode transfer.

		EISA Address			
		A3	A2	A1	A0
Start Address	FFFFF001h	0	0	0	1
Initial Increment (Incremented by 3)	FFFFF004h	0	1	0	0
All Increments Following (Incremented by 4)	FFFFF008h	1	0	0	0
	FFFFF00Ch	1	1	0	0

The starting address A<1:0> is 01, this means that the initial increment must be 3 in order to align the next increments on doubleword boundaries. The subsequent increments will be by 4 until the last cycle is detected.

For 16-bit transfers, the BMIC provides an initial address increment of 1 or 2 depending on the status of address lines A<1:0>. After the initial increment, the BMIC increments the address by two until the last cycle is detected.

The following example illustrates the BMIC address incrementer during a 16-bit master mode transfer.

		EISA Address			
		A3	A2	A1	A0
Start Address	FFFFF001h	0	0	0	1
Initial Increment (Incremented by 1)	FFFFF002h	0	0	1	0
All Increments Following (Incremented by 2)	FFFFF004h	0	1	0	0
	FFFFF006h	0	1	1	0

The starting address A<1:0> is 01, this means that the initial increment must be 1 in order to align the next increments on singleword boundaries. The subsequent increments will be by 2 until the last cycle is detected.

NOTE:
The BMIC internally assembles 32-bit dwords. When a 16-bit burst transfer is preempted, the transfer will stop on a doubleword boundary.

4.6 EISA Byte Decrementer

The Transfer Channel Current Count register set for each channel contains the intermediate value of the byte count during the transfer and is used as the byte decrementer. The decrementer's function is partially based upon the address incrementer. In the above 32-bit incrementer example, the byte count would be decremented by 3 on the first cycle. After the initial decrement, the channel's Current Count register set is decremented by 4 until the last cycle is detected. In the above 16-bit incrementer example, the byte count would be decremented by 1 on the first cycle. After the initial decrement, the channel's Current Count register set is decremented by 2 until the last cycle is detected.

4.7 EISA Address Incrementer/Byte Decrementer Illustration

The following table illustrates the various states of (A0, A1) vs the transfer byte-count and the initial address during a 32-bit transfer.

Byte Count	Starting Address	Next Address	Initial Increment	Number of Bytes Left	Last Cycle	Number of Cycles Left
1	XXX 0000	NA	NA	0	Yes	0
	XXX 0001	NA	NA	0	Yes	0
	XXX 0010	NA	NA	0	Yes	0
	XXX 0011	NA	NA	0	Yes	0
2	XXX 0000	NA	NA	0	Yes	0
	XXX 0001	NA	NA	0	Yes	0
	XXX 0010	NA	NA	0	Yes	0
	XXX 0011	XXX 0100	1	1	No	1
3	XXX 0000	NA	NA	0	Yes	0
	XXX 0001	NA	NA	0	Yes	0
	XXX 0010	XXX 0100	2	1	No	1
	XXX 0011	XXX 0100	1	2	No	1
4	XXX 0000	NA	NA	0	Yes	0
	XXX 0001	XXX 0100	3	1	No	1
	XXX 0010	XXX 0100	2	2	No	1
	XXX 0011	XXX 0100	1	3	No	1
5	XXX 0000	XXX 0100	4	1	No	1
	XXX 0001	XXX 0100	3	2	No	1
	XXX 0010	XXX 0100	2	3	No	1
	XXX 0011	XXX 0100	1	4	No	1
6	XXX 0000	XXX 0100	4	2	No	1
	XXX 0001	XXX 0100	3	3	No	1
	XXX 0010	XXX 0100	2	4	No	1
	XXX 0011	XXX 0100	1	5	No	2
7	XXX 0000	XXX 0100	4	3	No	1
	XXX 0001	XXX 0100	3	4	No	1
	XXX 0010	XXX 0100	2	5	No	2
	XXX 0011	XXX 0100	1	6	No	2
8	XXX 0000	XXX 0100	4	4	No	1
	XXX 0001	XXX 0100	3	5	No	2
	XXX 0010	XXX 0100	2	6	No	2
	XXX 0011	XXX 0100	1	7	No	2
9	XXX 0000	XXX 0100	4	5	No	2
	XXX 0001	XXX 0100	3	6	No	2
	XXX 0010	XXX 0100	2	7	No	2
	XXX 0011	XXX 0100	1	8	No	2
10	XXX 0000	XXX 0100	4	6	No	2
	XXX 0001	XXX 0100	3	7	No	2
	XXX 0010	XXX 0100	2	8	No	2
	XXX 0011	XXX 0100	1	9	No	3

NOTES:
1. "X" = Don't Care
2. If the "byte count" is less than or equal to the "initial increment", then the current cycle = the first cycle = the last cycle.
3. If the number of bytes left is less than or equal to 4, then the next cycle = the last cycle.
4. For information regarding byte alignment, refer to Section 6.3.1.

4.8 I/O Address Range Decode Support

The BMIC provides on-board decoder logic, two I/O select pins (IOSEL<1:0>#), and a set of 8-bit I/O Decode Range registers to support both general I/O decode and expansion board slot specific I/O decode and expansion board slot specific I/O decode. The BMIC also uses the AEN signal when decoding I/O locations.

The set of I/O Decode registers include two I/O Decode Range Base address registers and two I/O Decode Range Control registers (refer to Section 8.2.6). The I/O Decode registers are used to define the address ranges of interest to the bus master expansion board. Each IOSEL#<1:0> pin has an associated Control and Base register along with an associated address range as defined by the I/O Decode register set.

Through the I/O Decode Range Control register set, the BMIC can be programmed to respond to a select I/O address range as either an 8-bit or 32-bit EISA device. The only control signal provided by the BMIC to the EISA bus during an I/O decode is the EX32# signal. The output state of the EX32# pin on the BMIC will indicate the elected response (low = 32-bit EISA, high = 8-bit EISA). The Control register set controls the size of the I/O decode range, the I/O decode type (slot specific or general I/O), and the I/O decode address latching. The I/O address can be latched by the CMD# signal (de-pipelined) or merely decoded. By latching the I/O address, the associated IOSEL# line will remain active a minimum of 5 ns from the rising edge of CMD#.

The I/O decode range size depends on the value of bits <4:0> in the Control register. Each of these bits mask a corresponding address comparison bit in the Base register. If no bits are masked in the Control register, the BMIC will decode a doubleword address. The bits are masked as follows:

I/O Control Register	I/O Base Register Bit Masked	EISA Address Bit Masked
Bit 0	Bit 0	IADS2
Bit 1	Bit 1	IADS3
Bit 2	Bit 2	IADS4
Bit 3	Bit 3	IADS5
Bit 4	Bit 4, 5	IADS<7:6>

The I/O Decode Range Base Address register contains the address range that is used during the I/O decode address comparison. The following table gives the bits in the I/O Base Address Register and the EISA Address that are used during the comparison:

I/O Base Address Register	(EISA Address Bits)	
	Slot Specific	General I/O
Bit 0	IADS2	IADS2
Bit 1	IADS3	IADS3
Bit 2	IADS4	IADS4
Bit 3	IADS5	IADS5
Bit 4	IADS6	IADS6
Bit 5	IADS7	IADS7
Bit 6	IADS10	IADS8
Bit 7	IADS11	IADS9

If bit 6 in the I/O Decode Range Control register is programmed for General I/O decode, and the two most significant bits in the I/O Decode Range Base Address register are programmed to 0 (IADS<9:8>), I/O decoding for that range will be disabled. This is done to insure that the I/O address does not conflict with the slot specific address range or the EISA system board address range. The following table summarizes the EISA system I/O address mapping:

I/O Address Range (HEX)	I/O Range Reserved for
0000–00FF	EISA/ISA System Board
0100–03FF	General I/O (ISA Expansion Board)
0400–04FF	ISP (82358)
0500–07FF	General I/O (Alias of 0100h–03FFh)
0800–08FF	EISA System Board
0900–0BFF	General I/O (Alias of 0100h–03FFh)
0C00–0CFF	EISA System Board
0D00–0FFF	General I/O (Alias of 0100h–03FFh)
Slot Specific Range where X = Slot Number	
X000–X0FF	Slot (X)
X100–X3FF	General I/O (Alias of 0100h–03FFh)
X400–X4FF	Slot (X)
X500–X7FF	General I/O (Alias of 0100h–03FFh)
X800–X8FF	Slot (X)
X900–XBFF	General I/O (Alias of 0100h–03FFh)
XC00–XCFF	Slot (X) (BMIC Registers 0C80h–0CAFh)
XD00–XFFF	General I/O (Alias of 0100h–03FFh)

The following is an example of the BMIC programmed for slot specific decode:

I/O Decode Range 0 Control register programmed for (ECh)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(ECh)	1	1	1	0	1	1	1	1

— Bit 7—Respond as a 32-bit EISA slave
— Bit 6—Slot specific decode enabled
— Bit 5—Slot specific address latched by CMD#
— Bit 4—Compare I/O Decode Range 0 Base Address Bits (5) and (4) with EISA address signals IADS7 and IADS6 respectively
— Bit 3—Mask I/O Decode Range 0 Base Address Bit (3)
— Bit 2—Mask I/O Decode Range 0 Base Address Bit (2)
— Bit 1—Mask Decode Range 0 Base Address Bit (1)
— Bit 0—Mask I/O Decode Range 0 Base Address Bit (0)

I/O Decode Range 0 Base Address register programmed for (E3h)

	IADS11	IADS10	IADS7	IADS6	IADS5	IADS4	IADS3	IADS2
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(E3h)	0	0	1	0	—	—	—	—

EISA slot specific address range decoded—X080h through X0BFh where X represents the expansion board slot number

```
        (X080h)                                  (X0BFh)
xxxx000010(00)(00)00throughxxxx000010(11)(11)11
              ↑  ↑                          ↑  ↑
                    EISA Address
                     Bits Masked
                  (IADS4 and IADS5)
```

Byte enables IBE<3:0># and EISA address lines IADS<1:0> are not used during either slot specific or general I/O decode. During slot specific I/O decode, EISA address lines IADS<9:8> must be 0 to insure that the I/O address does not conflict with the ISA general I/O address range (0100h–03FFh).

IOSEL0# and EX32# will be driven low by the BMIC if addresses XA8Ch through XABFh are present on the EISA bus.

AEN is used as part of the decode and must be negated low when a response from the BMIC is required.

5.0 TRANSFER BUFFER INTERFACE

5.1 Transfer Buffer Interface Signals

The Transfer Buffer Interface portion of the BMIC provides the signal essential for interfacing to the expansion board as required for EISA-to-expansion board and expansion board-to-EISA burst data transfers. The Transfer Buffer Interface is designed to interface to a high speed transfer buffer and simple logic similar to that used in traditional DMA designs. This interface includes a 16-bit data bus, one clock input, and seven control signals.

The 16-bit data lines TDAT<15:0> are used by the BMIC to transfer the data to and from the transfer buffer logic on the expansion board during transfers. The data is word aligned. The BMIC automatically assembles the words received from the expansion board into 32-bit dwords for 32-bit transfers over the EISA bus. The data lines are also used by the BMIC to transport internally generated transfer start and real-time addresses to the external logic for use during data transfers (refer to Section 5.3).

The clock input (TCLK) controls the transfer rate between the BMIC and the external transfer buffer logic. The TCLK can be asynchronous to the BCLK.

The seven control signals include:

— Transfer Request (TRQ#); an output to request data transfers over the Transfer Buffer interface.
— Transfer Acknowledge (TACK#); An input to acknowledge data transfers. The TACK# signal may be used by the transfer buffer logic to add wait states to the data cycle and to terminate the transfer when used in conjunction with TEOP#.
— Data Transfer Direction (TDIR); An output to inform the external transfer buffer logic as to the direction of the current transfer (EISA read or EISA write).
— Transfer Channel Select (TCHAN); An output to indicate which of the two channels is currently active.
— Transfer Address Counter Load (TLD#); An output to load the current transfer start address to an external address counter, depending on the expansion board application.
— Transfer Data Output Enable (TDOE#); An input that unconditionally disables the BMIC from driving the TDAT<15:0> lines. With this signal, the BMIC can be prevented from driving the TDAT<15:0> lines while the local processor accesses the transfer buffer logic on the expansion board. No handshaking is required, so throughput is increased.

— Transfer End-of-Process I/O signal (TEOP#); TEOP# is both a status output pin, and an input pin for causing the termination of transfers, depending on the requirements of the external hardware.

NOTE:
Refer to Section 9.4 for additional information regarding the above signals.

5.2 External Transfer Buffer Logic

The Transfer Buffer interface is designed for high speed devices, such as SRAM based designs, or FIFOs. The Transfer Buffer interface data path is 16 bits wide. This requires the transfer clock (TCLK) to run at a speed of 16 MHz to 20 MHz to maintain the EISA maximum data rate of 33 Mbytes/sec. The fast cycle times required on the data Transfer Buffer interface can be implemented in the controlled environment found locally on the expansion board. If two BCLK transfers are used on the EISA side (16 Mbytes/sec), the timing requirements for the transfer buffer can be relaxed, and lower cost implementations can be utilized.

If the transfer buffer controller does dynamic arbitration for the transfer buffer between the BMIC and the peripheral device(s) on the expansion board, the peripheral device accesses should be short enough so that the BMIC's data FIFO can handle the interruption to its data flow, without stalling the EISA transfer.

Examples of transfer buffer architecture implementations that could be interfaced to the BMIC include:

— A FIFO implementation which is large enough to buffer the difference in throughput rates between the peripheral device on the expansion board and the EISA Bus. See Section 5.2.1.

— A small high-speed DMA like device that generates addressing for a SRAM based transfer buffer.

— A controller implementation for dual-ported SRAM for high transfer buffer bandwidth.

— A page or nibble-mode dynamic-RAM controller implementation for large, low cost transfer buffers.

— For graphics systems, the frame buffer itself can be used for the transfer buffer with a non-linear address generator for transferring windows in the screen image.

5.2.1 FIFO IMPLEMENTATION

During EISA writes, the BMIC will overread (read data beyond the number of bytes to be transferred) the transfer buffer by a maximum of 28 bytes. These overread bytes may contain valid data (back to back transfers) which will be lost. The data loss can be avoided through software or hardware. The software solution avoids back to back transfers. This implies that there is data for only one transfer in the FIFO at any given time.

The hardware solution requires an external 22-bit Byte Counter and a Flip-Flop. The terminal count of the Byte Counter is used to SET the Flip-Flop which disables BMIC reads to the FIFO. The BMIC will continue to read (overread) "stale" data. The BMIC TEOP# output signal is used to RESET the Flip-Flop which enables the BMIC reads to the FIFO.

5.3 Transfer Interface Start Address Generation

The BMIC provides four 8-bit Transfer Buffer Interface (TBI) registers, 2 Base, and two Current registers, which can be programmed with 16-bit transfer start addresses. Each transfer channel has an associated Base and Current register pair. The Base registers contain the start address and the Current registers provide the real-time address used to track the current transfer. The Current registers will increment by one each time a 16-bit word is transferred across the Transfer Buffer interface.

The 16-bit start address is transferred across the TDAT<15:0> lines to the transfer buffer logic at the beginning of all new data transfers (i.e., each time the TBI Base register set contents are transferred to the TBI Current register set). The contents of the TBI Base registers are transferred to the TBI Current registers after a write to the associated channel's Transfer Strobe register is completed (refer to Section 4.2). The BMIC provides a load signal (TLD#) which can be used to latch the start address into an external address counter for use by the transfer buffer logic.

The BMIC can also be programmed to generate the transfer address each time the associated channel regains the bus, in which case, the address will be the real-time address. By programming the CFGEA bit in the Channel Configuration register to a "1", the start address will be transferred to the transfer buffer logic at the beginning of all new transfers and the real-time address will be transferred each time the associated channel regains the bus. If the CFGEA bit is set to a "0", the transfer start address will be transferred at the beginning of all new transfers and the real-time address will not be transferred.

NOTE:
The TBI Current register set is readable by the local processor. However, there are possible coherency problems involved with reading multiple bytes while the current registers are being updated during a transfer. To avoid these problems, the channel's transfer should be temporarily suspended (using the channel's Configuration Register) before trying to read the channel's TBI Current register set.

5.4 Transfer Buffer Interface Timing Example

Figures 5-1 and 5-2 illustrate the start up and conclusion of a transfer cycle across the Transfer Buffer interface and should be used as a reference when reading the following text.

1. At the start of a data transfer TCHAN and TDIR change to their new values prior to the falling edge of TLD# to set up the cycle. TCHAN and TDIR will not change states as long as TRQ# is asserted.
2. TLD# is asserted until acknowledged by TACK#. The transfer address is transferred to the external logic each time the TBI Base register contents are transferred to the TBI Current register set (new transfer) and, if programmed, each time the current channel regains the bus.
3. The new address is loaded using the TDAT bus during TLD# at point (A). The TDAT bus should be turned on by asserting TDOE# during TLD# if the internal start address is required. Once the external channel address and direction are set up, the data transfer can begin.
4. Data transfer requests are signaled by TRQ# being asserted (low). TRQ# will remain active until the data transfer is completed or a transfer interruption occurs (refer to Section 1.0) followed by TACK# active. During an EISA write, there will be a one TCLK delay between TLD# deasserting and TRQ# asserting as denoted by point (D) in Figure 5-2. This is to allow time for the external buffers to change direction after the TLD# has been completed.
5. Each word transfer to or from the BMIC is acknowledged by the TACK# signal. If TACK# is active at the rising edge of TCLK, one word will be transferred. If TACK# is not active at the rising edge of TCLK, the word that is currently being transferred will be inhibited and a wait state will

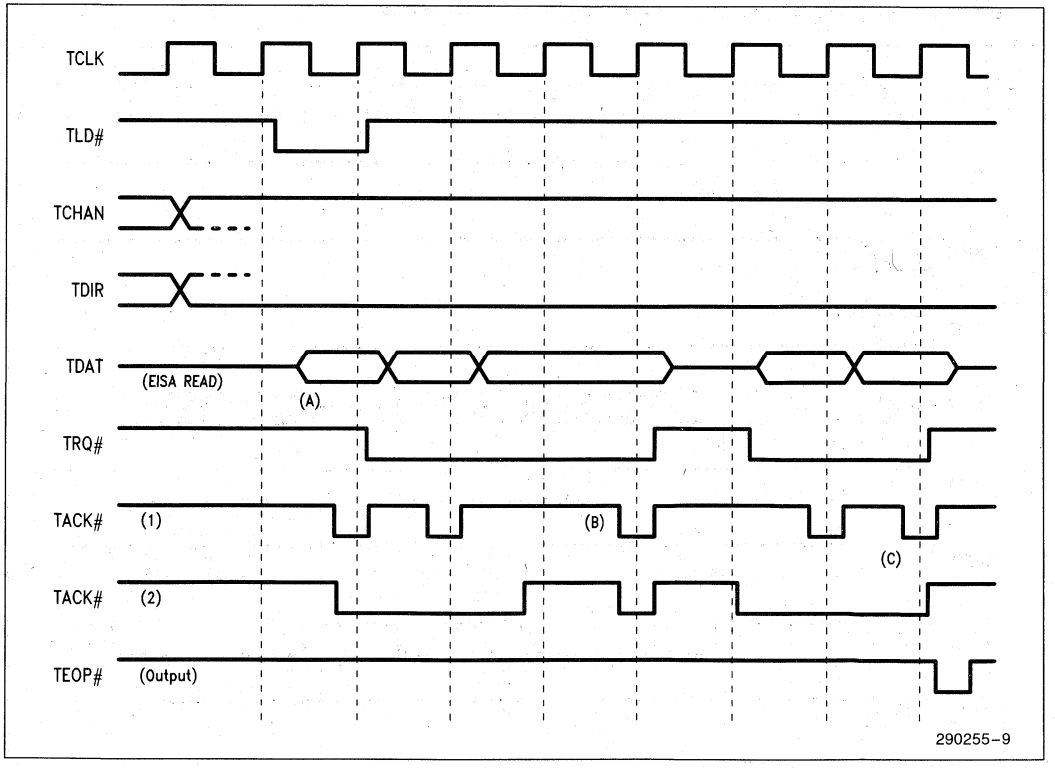


Figure 5-1. Transfer Buffer Interface Timing (EISA READ)

be inserted. This is shown at point (B) in Figure 5-1. Such a wait may be needed when the external transfer buffer logic is arbitrating between the BMIC and the I/O subsystem on the expansion board. Wait states may also be inserted by stretching TCLK as shown at point (C) in both of the figures. Clock stretching is possible as long as the one to one ratio of TCLK to BCLK is not violated.

NOTE:
A long TCLK stretch time will hang the Transfer Buffer interface. Also, TCLK must be running during the time TRQ# is inactive in order for the Transfer Buffer interface to function properly.

As indicated above, TACK# must be stable at the rising edge of TCLK. However, TACK# can assume any convenient pattern at other times. As shown by the first pattern, TACK# (1) pules low at the TCLK edge that data is transferred. This pattern is particularly useful when one TCLK wait-states are desired as indicated at point (B) in Figure 5-1. The alternate pattern (TACK#2) is useful during TCLK stretching since TACK# is always low during TRQ# as shown at point (C). This is effective since the transfer clock edge timing is controlled by the amount TCLK is stretched.

6. TEOP# is asserted at the end of a transfer by the BMIC.

   The BMIC will indicate end-of-process by asserting TEOP# shortly after the negation of the last CMD# in the transfer. During an EISA write transfer, the BMIC will assert TEOP# a maximum of two TCLKs after CMD# is negated. During an EISA read transfer, the BMIC will assert TEOP# a maximum of 20 TCLKs (typically eight TCLKs) after the negation of CMD#. In either case (EISA read or EISA write), the TEOP# signal is delayed from the rising edge of TCLK.

NOTE:
The BMIC will assert the expansion board interrupt signal (LINT) at the end of a transfer, if so programmed in the Transfer Channel Configuration register.

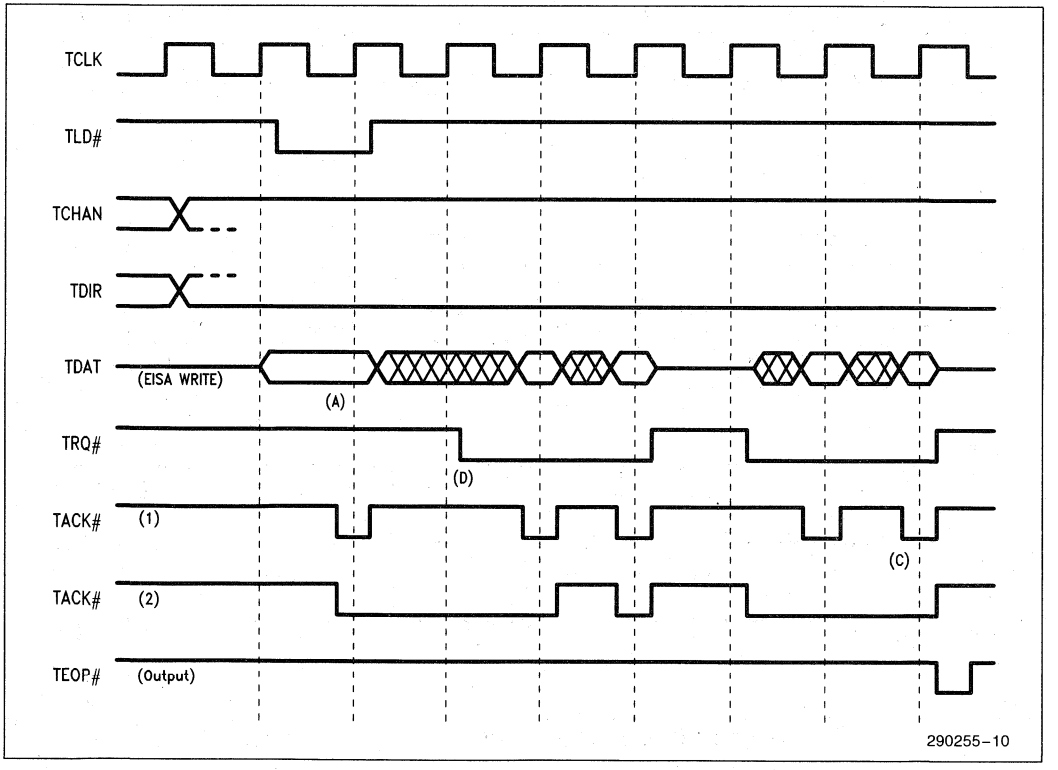


Figure 5-2. Transfer Buffer Interface Timing (EISA WRITE)

6.0 FIFO/DATA ALIGNER

6.1 FIFO/Data Aligner

The BMIC uses two identical FIFOs, one per transfer channel, and a common data aligner for data transfers between system memory and the bus master expansion board. The primary function of the FIFO/Data Aligner Unit is to help isolate and simplify the timing relationships between the EISA bus and the devices on the expansion board.

The FIFO allows the timing on the expansion Board side of the BMIC to be based on a locally generated clock signal. This transfer clock (TCLK) can be independent of the EISA BCLK signal that governs EISA bus timing. The FIFO also provides latency protection for wait states generated on either the EISA bus or expansion board.

The Data Aligner arranges the 16-bit data from the external transfer buffer to any arbitrary byte alignment in system memory. The data aligner also performs the assembly and disassembly of the EISA data during the transfer. The TDAT data assembly and disassembly is done by the Transfer Buffer interface portion of the BMIC.

6.2 FIFOs

Each FIFO on-board the BMIC is 24 bytes in size. The transfer data is written into the FIFOs from either the expansion board or the EISA bus side, depending on the direction of transfer. The data is written into the FIFO as doublewords during the transfer. However, if the data is not doubleword aligned, partial FIFO loads will be done at the beginning or end of a transfer depending on the byte count, address programmed and the direction of the transfer.

The condition of the FIFOs can be determined by a read to the Transfer Channel Status register set. A read to this register will indicate whether the FIFOs are stalled or active. A FIFO stall is defined as a FIFO that is full during an EISA read or empty during an EISA write. In either case, the transfer buffer logic is unable to keep up with the EISA device. If a FIFO stall occurs, the transfer will be stopped and the BMIC will either service the transfer request with the highest priority or relinquish the EISA bus to the system. The BMIC will relinquish the bus to the system if the CFGFF bit in the channel's corresponding Configuration register is set to a 1.

6.3 Data Aligner

6.3.1 EISA BYTE ALIGNMENT

The BMIC automatically handles the byte alignment for the EISA bus in the case of misaligned doubleword boundaries and assumes no performance penalty. The BMIC will do any partial doubleword transfers as required at the beginning and the end of all transfers. The two least significant bits of the 32-bit transfer start address (A1 and A0) are used to provide the byte alignment for both EISA read and EISA write transfers. The following tables illustate the BMIC's byte alignment approach during 32- and 16-bit transfers:

In the following tables "—" represents no data transferred and the digits represent the data items being transferred. The byte alignment for an EISA read is identical to that of an EISA write.

EISA Write (32-bit/12-byte Transfer) and (16-bit/6-byte Transfer)

		(32-Bit)				(16-Bit)			
A1	**A0**	**Output Data to EISA Bus**				**Output Data to EISA Bus**			
Byte Lane	**→**	**3**	**2**	**1**	**0**	**3**	**2**	**1**	**0**
0	0	03	02	01	00	—	—	01	00
		07	06	05	04	—	—	03	02
		11	10	09	08	—	—	05	04
0	1	02	01	00	—	—	—	00	—
		06	05	04	03	—	—	02	01
		10	09	08	07	—	—	04	03
		—	—	—	11	—	—	—	05
1	0	01	00	—	—	—	—	01	00
		05	04	03	02	—	—	03	02
		09	08	07	06	—	—	05	04
		—	—	11	10	—	—	—	—
1	1	00	—	—	—	—	—	00	—
		04	03	02	01	—	—	02	01
		08	07	06	05	—	—	04	03
		—	11	10	09	—	—	—	05

6.3.2 DATA ASSEMBLY/DISASSEMBLY

Before being placed on either the TDAT or IDAT data buses during an EISA read or EISA write, the data will be assembled or disassembled as required. The IDAT data is assembled and disassembled by the FIFO/data aligner portion of the BMIC and the TDAT data is assembled and disassembled by the Transfer Buffer interface portion of the BMIC. The following paragraphs illustrate the BMIC's assembly and disassembly approach during 32- and 16-bit transfers. The illustration assumes that byte alignment is not required.

During 32-bit EISA read transfers, the 32-bit doublewords are removed from the EISA bus and placed into the FIFO. After flowing through the FIFO, the 32-bit doublewords are copied-down to 16-bit words and then placed on the TDAT bus.

During 32-bit EISA write transfers, the 16-bit words are removed from the TDAT lines, assembled into 32-bit doublewords, and then placed into the FIFO. After flowing through the FIFO, the 32-bit data is placed on the EISA bus. No further assembly or disassembly is required after the FIFO as the data is already in 32-bit doubleword form.

During 16-bit EISA read burst transfers, the 16-bit words are removed from the EISA bus, assembled into 32-bit doublewords, and then placed into the FIFO. After flowing through the FIFO, the 32-bit data is copied-down to 16-bit words and then placed on the TDAT bus.

During 16-bit EISA write burst transfers, the 16-bit words are removed from the TDAT EISA bus, assembled into 32-bit doublewords, and then placed into the FIFO. After flowing through the FIFO, the 32-bit data is copied-down to 16-bit words and then placed on the EISA bus.

7.0 LOCAL PROCESSOR INTERFACE

The BMIC's Local Processor interface is based on an asynchronous, 8-bit interface. All of the slave signals required for a local processor to program the BMIC are provided through this interface. These signals include (LCS#, LRD#, LWR#); two address lines (LADS0 and LADS1) for addressing internal registers; an 8-bit data path (LDAT); an Interrupt signal (LINT); and a ready signal (LRDY). LINT allows the BMIC to interrupt the local processor and the ready signal (LRDY) indicates when valid data is available on the LDAT lines (shared register accesses only, see below). If a local processor is not used, the Local Processor interface can be connected to the 8-bit ISA bus (refer to Section 7.3). The choice of the local microprocessor or microcontroller used depends upon the specific application and the degree of performance and data processing needed (refer to Section 7.2).

The Local Processor interface portion of the BMIC contains two 8-bit registers which are used by the local processor to access all of the BMIC's internal registers. These registers are mapped into the Local Processor interface and include a Local Data register and a Local Index register. These registers are selected using the Local Processor interface's two address lines. The Local Status/Control register is also directly mapped into the Local Processor interface and is used to provide the local processor with the interrupt, peek/poke, and Base register status.

The BMIC allows the local processor and the EISA bus to communicate with each other through a set of Command/Status registers. The Command/Status registers are referred to as shared registers and include a set of Mailbox registers, Semaphore ports, and doorbell registers. The mailbox registers are used to pass messages to and from the local processor and the EISA bus and are controlled by the Semaphore ports. The Doorbell register set is used to inform the respective processor of new messages. Also part of the shared register set, are the ID registers, which are used to support the EISA expansion board ID function.

The BMIC allows the local processor access to individual locations in system memory or I/O space using the Peek/Poke feature. The local processor can also initiate BMIC burst and non-burst (two BCLK) data transfers to and from system memory.

7.1 Shared Registers—Status/ Command Support

As data transfer rates increase, it is critical that an efficient command and status passing mechanism be implemented so that command and status exchange does not become a new bottleneck to system performance. The BMIC utilizes a high-performance command/status interface between the main system and the local processor to minimize command/status overhead.

The Shared registers are a group of registers accessable by the system CPU or EISA bus master and the local processor for general-purpose command and status interactions and EISA expansion board ID function support. The features of the BMIC command/status support include a pair of semaphore ports, a pair of interrupt ports ("doorbell registers"), and a set of mailbox registers. With these functions, many different types of high-performance communication protocols can be defined between the system and the expansion board. Three additional registers, the Global Configuration register,

the System Interrupt/Enable register, and the ID registers are also part of the shared register set.

7.1.1 SEMAPHORE PORTS

The two semaphore ports are specifically designed to allow set-and-test functions in I/O space. Specifically, the ports are used to lock access to the mailbox registers and to lock access to links in main memory. Each of the semaphore ports consists of two parts: the semaphore flag bit and the semaphore test bit.

When a write occurs to the semaphore flag bit through either the EISA interface or the Local Processor interface, the old value of the semaphore flag bit is copied to the appropriate semaphore test bit. The old value of the semaphore flag bit is then available in the test bit to be read back by the processor. If the value read back from the semaphore test bit is a "1", the requested resource is unavailable for use. If the value read back is a "0", the requested resource is available for use and is now locked by the requesting processor or bus master. In this manner, set-and-test algorithms can be implemented without using the EISA bus lock function. The processor or EISA bus master unlocks the semaphore by simply writing a "0" to the associated semaphore flag bit.

NOTE:
The Semaphore ports and resources are locking only in a software sense, as in any semaphore in main memory. The Semaphore ports are identical and are not associated with either interface (EISA or Local). The protocol for the semaphores and the effect they have on other shared registers, like the Mailbox registers, is strictly a matter of how the system software chooses to use them.

Implementing the semaphore in the BMIC instead of main memory eliminates the need for the BMIC to arbitrate for the EISA bus every time it wishes to update or test the semaphore. Note that the semaphore scheme described here is functional only when a single device on the EISA is communicating with the BMIC; the semaphore coordinates "locks" between the single device and the local processor. In the case that multiple masters attempt to lock access to the BMIC, the masters must first agree amongst themselves which one has the privilege to use the BMIC semaphore port(s).

7.1.2 MAILBOX REGISTERS

A set of 16 8-bit general-purpose mailbox registers are used to pass information between the bus master expansion board and the EISA system. The 16 registers are mapped contiguously in EISA slot-specific I/O space, so they can be accessed as bytes, words, or doublewords. These registers can be used to directly pass command and status information, or they can be used as pointers to larger command blocks in memory.

The mailbox registers can be read or written at any time from either the EISA bus or the Local Processor interface. An internal arbitration is implemented in such a way that if there is a simultaneous read and write from both sides of a mailbox register, then the read operation will not contain indeterminate bits. In other words, when a read operation is done on a mailbox register at the same time as a write operation to that register, the bit pattern that is read will be either the old bit pattern in the mailbox, or the new bit pattern being written, but never some transitory, invalid bit pattern.

7.1.3 DOORBELL REGISTERS

There are two 8-bit doorbell Interrupt/Status registers in the BMIC, one assigned to the EISA side and one assigned to the expansion board side. The EISA System Doorbell register is used by the local processor to request service from the EISA side and the Local Doorbell register is used by the device on the EISA side to send an interrupt request to the local processor on the bus master expansion board. The doorbell Interrupt/Status registers are implemented with "sticky" bits, so that individual bits in the register can be set by the interrupting device or reset by the servicing device without knowledge of the states of the other bits in the register. The eight bits in each doorbell register allow up to eight separate devices or events in each direction to have interrupt requests pending simultaneously. The interrupt requests pending in either of the two Doorbell registers are ORed with the other interrupt sources from within the BMIC, and the result is sent out over one of the two interrupt pins: LINT or EINT.

Each doorbell register has an associated 8-bit Interrupt Enable register used to enable or disable the interrupts on an individual basis. The BMIC also includes a System Interrupt Enable/Control register and a Local Status/Control register used to disable the system (EINT) and local (LINT) interrupts and to verify the status of the system and local interrupts on a global basis (refer to Sections 8.1.1.3.3 and 8.2.2).

The following paragraphs describe the operation of the Local Doorbell Interrupt/Status register. The EISA System Doorbell Interrupt/Status register is similar, but operates in the opposite direction.

Each device or event that can interrupt the bus master expansion board can be assigned a bit position within the BMIC's Local Interrupt/Status Doorbell

register. When the device on the EISA bus wants to send an interrupt request to the bus master expansion board, it writes to the Local Interrupt/Status Doorbell register (from the EISA side) with that device's assigned bit position set active. This will set that bit in the Local Interrupt/Status Doorbell register, but leave the other bits in the register unaffected. If that bit position is not disabled, then the interrupt signal to the local processor will be asserted.

When the local processor services the interrupt, it checks the Local Status/Control Register to determine the source of the interrupt. If the control register indicates that the Local Doorbell register is one of the active interrupt sources, then the local processor can read the Local Doorbell register to determine which bits are active and requesting interrupts. If the local processor decides to service one of the requests from the Local Doorbell register, it can write to the Local Doorbell register with that bit's position set. This action will cause that bit in the Local Doorbell register to reset, but the other bits will remain unaffected. Thus, each bit in the Local Doorbell register is like a set-reset flip-flop, with the EISA bus controlling the "set" input, and the Local Processor interface controlling the "reset" input.

7.2 Local Processor Recommendations

The Local Processor interface to the BMIC will support numerous processors, from the 8088 microprocessor to the 376™ embedded processor.

The 80186, 80C186, 80188, and 80C188 family of processors provide a clean interface to the BMIC's Local Processor interface and eliminate the need for additional logic. An on-board programmable wait-state generator eliminates the need for external wait-state generation logic between the processor and the BMIC during non-shared register accesses.

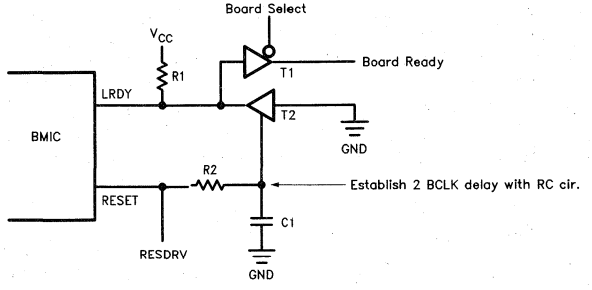
7.3 Requirements for No Local Processor

The BMIC allows for expansion board designs that do not require a local processor. To support the programming of the BMIC in a no local processor board design, the Local Processor interface must be connected to the ISA bus. However, when the ISA bus is used, the BMIC must be informed that there is no local processor and that it must change its function slightly (refer to next section). To inform the BMIC that no local processor is present, LRDY must be driven low during RESET and remain low a minimum of two BCLKs after RESET is negated.

The following circuit can be used to establish the proper LRDY/RESET timing as required for a no local processor design:

7.4 EISA ID Function Support/ Registers

The BMIC provides support for the EISA expansion board ID function. The primary ID implementation takes advantage of the local processor. Upon reset, the local processor executes a routine from its ROM that writes the product identifier for the expansion board to the four 8-bit ID registers in the BMIC. The registers have accessed through the Local Processor interface and are located at local index addresses 00h–03h. On the EISA side, these registers are mapped into the EISA slot specific ID address range XC80h–XC83h.



290255-11

R1 = 8.2K
R2 = Depends on frequency of BCLK and application implemented.
C1 = Depends on frequency of BCLK and application implemented.
T1 = Active low tri-state buffer
T2 = Active high tri-state buffer

If the host CPU accesses the ID registers in the BMIC before the local processor has programmed them, the BMIC will return the setup delay ID code 0111XXXXh in the byte 0 ID register located at EISA slot specific I/O address XC80h. The byte 0 ID register should be programmed last by the local processor.

If a local processor is not used, external registers will have to be implemented on the expansion board to hold the expansion board ID value. The BMIC will automatically set its I/O decode range 0 registers to decode 8-bit EISA ID addresses. The IOSEL0# output signal can then be used to trigger external logic on the expansion board to enable ID data onto the IDAT<7:0> data lines. The external lines connected to the IDAT<7:0> lines should be connected to the bus between the BMIC and the external F245 data buffers. The BMIC will enable the external data buffers to drive byte lane 0 of the EISA bus upon detection of the ID address. The external logic should monitor SA1 and SA0 on the ISA bus to determine which data byte to drive.

8.0 REGISTER DESCRIPTION

8.1 Shared Register Description

The following is a table of the Shared register group listing the number of registers, register type (read/write) as related to the local and EISA side, register name, and register size:

Number	EISA	Local Type	Register Name Type	Active Bits per Register
2	R/W	R/W	Semaphore Register	2 Bits
16	R/W	R/W	Mailbox Register	8 Bits
1	R/W	R/W	Local Doorbell Interrupt/Status Register	8 Bits
1	R	R/W	Local Doorbell Enable Register	8 Bits
1	R/W	R	EISA System Doorbell Interrupt/Status Register	8 Bits
1	R/W	R	EISA System Doorbell Enable Register	8 Bits
1	R/W	R	System Interrupt Enable/Control Register	8 Bits
1	R	R/W	Global Configuration Register	8 Bits
4	R	R/W	ID Register	8 Bits

8.1.1 COMMAND/STATUS SUPPORT REGISTERS

8.1.1.1 Semaphore Ports (Read/Write)

The BMIC contains two Semaphore ports which can be used to software lock resources between the EISA bus and the local processor. Each semaphore port controls a 1-bit semaphore flag. Upon reset, the Semaphore ports are reset to 0.

Semaphore Port 0 EISA Address—XC8Ah
Semaphore Port 0 Local Index Address—0Ah

Semaphore Port 1 EISA Address—XC8Bh
Semaphore Port 1 Local Index Address—0Bh

—	—	—	—	—	—	Bit 1	Bit 0

Bit 7–2 —Reserved, set to 0
Bit 1 —Semaphore Test bit (Read Only)
Bit 0 —Semaphore Flag bit (Read/Write)

Bit (0) reflects the actual value of the semaphore at any given instant. Whenever a write is done to the Semaphore Flag bit (0), its previous value is simultaneously copied to the Semaphore test bit (1). Internal to the BMIC, there are two test bits for each semaphore port: one for the EISA interface and one for the Local Processor interface. To do a test-and-set function, write to the semaphore port with the desired semaphore value in the flag bit. After a write has been completed, read the semaphore port and check the test bit to verify that a collision did not occur.

8.1.1.2 Mailbox Registers (Read/Write)

The mailbox registers are sixteen 8-bit, general purpose registers. The format of the contents of the mailbox registers is user-defined. The Mailbox register set is not initialized to a fixed value upon reset.

EISA Address—XC90h through XC9Fh
Local Index Address—10h through 1Fh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

8.1.1.3 Doorbell Registers

8.1.1.3.1 Local Doorbell Interrupt/Status Register (Read/Write)

This register is implemented with "sticky" bits (refer to Section 7.1.3). The Local Doorbell Interrupt/Status register is used by the EISA bus to send an interrupt request to the expansion board. When read from, this register indicates the status of pending interrupt events. Upon reset, the Doorbell Interrupt/Status register is reset to 0.

EISA Address—XC8Dh
Local Index Address—0Dh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7–0 1 = Doorbell interrupt pending (local CPU read)
Set Doorbell bit (EISA write)
Reset Doorbell bit (Local CPU write)
0 = No doorbell interrupt pending (Local CPU read)
No action (EISA or local CPU write)

Bits 0–7 allow up to eight events or devices on the EISA side to interrupt the local side of the BMIC. The above bits can only be reset by the servicing processor on the local side.

8.1.1.3.2 Local Doorbell Enable Register (Read/Write)

The Local Doorbell Enable register is used by the local processor to enable or disable interrupt requests to the local expansion board. This register is read only from the EISA side. Upon reset, the Doorbell Enable register is set to 0.

EISA Address—XC8Ch
Local Index Address—0Ch

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7–0 1 = Enable doorbell interrupt for corresponding bit position
0 = Disable doorbell interrupt for corresponding bit position
No action (EISA or local CPU write)

Bits 0 through 7 act as interrupt enables for bits 0 through 7 in the Local Interrupt/Status register respectively.

8.1.1.3.3 EISA System Doorbell Interrupt/Status Register (Read/Write)

This register is implemented with "sticky" bits (refer to Section 7.1.3). The EISA System Doorbell Interrupt/Status register is used by the expansion board to send an interrupt request to the EISA bus. When read from, this register indicates the status of pending interrupt events. Upon reset, the EISA System Doorbell Interrupt/Status register is reset to 0.

EISA Address—XC8Fh
Local Index Address—0Fh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7–0 1 = Doorbell interrupt pending (EISA read)
Set Doorbell bit (Local CPU write)
Reset Doorbell bit (EISA write)
0 = No doorbell interrupt pending

Bits 0–7 allow up to eight events or devices on the expansion board to send interrupts to the EISA bus. The above bits can only be reset by the servicing processor on the EISA side.

8.1.1.3.4 EISA System Doorbell Enable Register (Read/Write)

The EISA System Doorbell Enable register is used by the EISA processor to enable or disable interrupt requests to the EISA side. This register is read only from the local side. Upon reset, the EISA System Doorbell Enable register is reset to 0.

EISA Address—XC8Eh
Local Index Address—0Eh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7–0 1 = Enable doorbell interrupt for corresponding bit position
0 = Disable doorbell interrupt for corresponding bit position

Bits 0 through 7 act as interrupt enables for bits 0 through 7 in the EISA System Doorbell Interrupt/Status register respectively.

8.1.1.3.5 System Interrupt Enable/Control Register (Read/Write)

This register is used by the processor on the EISA side to disable the EINT signal. The EISA processor also can read this register to determine whether there are any pending interrupt requests in the EISA System Doorbell Interrupt/Status register. This register is read only from the local side. Upon reset, this register is reset to 0.

EISA Address—XC89h
Local Index Address—09h

—	—	—	—	—	—	Bit 1	Bit 0

Bit 7–2 — Reserved, set to 0

Bit 1 — (read-only bit)
1 = Enabled interrupts are pending in EISA System Doorbell Interrupt/Status register
0 = No enabled interrupts are pending in EISA System Doorbell Interrupt/Status register

Bit 0 —
1 = Enable interrupts from System Doorbell register (EISA write)
0 = Disable interrupts from System Doorbell register (EISA write)

8.1.2 GLOBAL CONFIGURATION REGISTER (READ/WRITE)

This register is used to program the type of protocol, edge or level-triggered, that will be used with the EINT and LINT interrupt signals. The Global Configuration register is also used to program the preempt timer and provide four bits for a BMIC hardware revision number. This register is read only from the EISA side. Upon reset, Bits 0–3 are reset to 0.

EISA Address—XC88h
Local Index Address—08h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bits 7–4 (read-only)
Hardware revision number of the BMIC

Bit 3 1 = System interrupt pin (EINT) uses edge-triggered protocol (Active high)
0 = System interrupt pin (EINT) uses level-triggered protocol (Active low open collector)

Bit 2 1 = Local interrupt pin (EINT) is set for active high operation
0 = Local interrupt pin (LINT) is set for active low operation

Bit 1 Delay to give up bus after preempt
00 = 3 BCLKs
01 = 32 BCLKs
10 = 64 BCLKs
11 = reserved

8.1.3 ID REGISTERS

The ID register set consists of four 8-bit registers. These registers are programmed at initialization time with the product identifier for the expansion board which contains the BMIC. The registers are mapped at read-only into the EISA ID I/O address range. Upon reset, the ID byte 0 register will contain the value 0111XXXXh, which is the EISA ID delay value. The local processor should program byte 0 last. If the external ID support scheme is selected, then these registers are disabled. The bit definitions defined below have significance for the EISA ID protocol, but not for any BMIC hardware functionality. Upon reset, ID bytes 1–3 are reset to 0.

EISA Address—XC80h through XC83h (bytes 0–3)
Local Index Address—0h through 3h (bytes 0–3)

ID Register Bytes 0–3:

	7	6	5	4	3	2	1	0
Byte 0	—	MCC14	MCC13	MCC12	MCC11	MCC10	MCC24	MCC23
Byte 1	MCC22	MCC21	MCC20	MCC34	MCC33	MCC32	MCC31	MCC30
Byte 2	MCC43	MCC42	MCC41	MCC40	MCC53	MCC52	MCC51	MCC50
Byte 3	MCC63	MCC62	MCC61	MCC60	MCC73	MCC73	MCC71	MCC70

ID Register Byte 0:

Bit 7	—	Reserved
Bits 6–2	MCC1<4:0>	First character of manufacturer's code
Bits 1, 0	MCC2<4:3>	First portion of Second character of manufacturer's code

ID Register Byte 1:

Bits 7–5	MCC2<2:0>	Second portion of Second character of manufacturer's code
Bits 4–0	MCC3<4:0>	Third character of manufacturer's code

ID Register Byte 2:

Bits 7–4	MCC4<3:0>	First hex digit of product number
Bits 3–0	MCC5<3:0>	Second hex digit of product number

ID Register Byte 3:

Bits 7–4	MCC6<3:0>	First hex digit of product revision
Bits 4–0	MCC7<3:0>	Second hex digit of product revision

8.2 Local Processor Only Registers

The following is a table of the Local Processor Only register group listing the number of registers, register type (read/write) as related to the local side, register name, and register size:

Number	Local Type	Register Name	Active Bits per Register
INDEX REGISTERS			
1	R/W	Local Index Register	8 Bits
1	R/W	Local Data Register	8 Bits
1	R/W	Local Status/Control Register	8 Bits
DATA CHANNEL TRANSFER REGISTERS			
4	R/W	Data Transfer Channel 0 Base Address Register	8 Bits
4	R/W	Data Transfer Channel 1 Base Address Register	8 Bits
4	R	Data Transfer Channel 0 Current Address Register	8 Bits
4	R	Data Transfer Channel 1 Current Address Register	8 Bits
3	R/W	Data Transfer Channel 0 Base Count Register	8 Bits
3	R/W	Data Transfer Channel 1 Base Count Register	8 Bits
3	R	Data Transfer Channel 0 Current Count Register	8 Bits
3	R	Data Transfer Channel 1 Current Count Register	8 Bits
DATA TRANSFER CONTROL/STATUS REGISTERS			
1	W	Channel 0 Transfer Strobe Register	0
1	W	Channel 1 Transfer Strobe Register	0
1	R/W	Channel 0 Configuration Register	8 Bits
1	R/W	Channel 1 Configuration Register	8 Bits
1	R	Channel 0 Status Register	6 Bits
1	R	Channel 1 Status Register	6 Bits
PEEK/POKE REGISTER			
4	R/W	Peek/Poke Address Register	8 Bits
4	R/W	Peek/Poke Data Register	8 Bits
1	R/W	Peek/Poke Control Register	8 Bits
I/O DECODE REGISTERS			
1	R/W	I/O Decode Range 0 Base Address Register	8 Bits
1	R/W	I/O Decode Range 1 Base Address Register	8 Bits
1	R/W	I/O Decode Range 0 Control Register	8 Bits
1	R/W	I/O Decode Range 1 Control Register	8 Bits
TRANSFER BUFFER INTERFACE (TBI) REGISTERS			
2	R/W	TBI Channel 0 Base Address Register	8 Bits
2	R/W	TBI Channel 1 Base Address Register	8 Bits
2	R	TBI Channel 0 Current Address Register	8 Bits
2	R	TBI Channel 1 Current Address Register	8 Bits

8.2.1 INDEX REGISTERS

The BMIC's register set is accessed using the local Index and Local Data register set (refer to Section 3.2.6.1). The Local Index and Local Data registers are mapped directly into the Local Processor interface of the BMIC.

8.2.1.1 Local Index Register (Read/Write)

The Local Index register contains the address of the BMIC register that is currently being accessed. An optional auto-increment mode is supported through this register, which automatically increments the index register after each Local Data register read or write. Upon reset, the Local Index register is set to 0.

Local Address—1h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7 — 1 = Autoincrement local index register after access to local data register

0 = Do not autoincrement

Bits 6–0 — Local index address

8.2.1.2 Local Data Register (Read/Write)

During a BMIC local register access, the value of the register being accessed is passed through this register.

Local Address—0h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

8.2.2 LOCAL STATUS/CONTROL REGISTER (READ/WRITE)

The Local Status/Control register is directly mapped into the Local Processor interface and is accessible using the two address lines (LADS<1:0>). This register provides current local doorbell interrupt status, current Channel 0 and Channel 1 interrupt and Base register status, and current peek/poke cycle status. This register is also used by the local processor on the expansion board to disable and provide the current status of the LINT signal (active or inactive). Bit 4 in this register is read/write and the remaining bits are read only. Upon reset, the Local Status/Control register is reset to 0.

Local Address—2h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7 — R
- 1 = Enabled interrupts are pending in Local Doorbell register
- 0 = No enabled interrupts are pending in Local Doorbell register

Bit 6 — R
- 1 = Enabled interrupts are pending from channel 1 events
- 0 = No enabled interrupts are pending from channel 1 events

Bit 5 — R
- 1 = Enabled interrupts are pending from channel 0 events
- 0 = No enabled interrupts are pending from channel 0 events

Bit 4 — R/W
- 1 = Local interrupts enabled
- 0 = All local interrupts disabled

Bit 3 — R
- 1 = Local interrupt signal (LINT) is currently active
- 0 = LINT signal is currently inactive

Bit 2 — R
- 1 = Most recent peek/poke command is still pending
- 0 = Most recent peek/poke command is complete

Bit 1 — R
- 1 = Base register set for channel 1 is busy
- 0 = Base register set channel 1 is available

Bit 0 — R
- 1 = Base register set for channel 0 is busy
- 0 = Base register set for channel 0 is available

8.2.3 DATA CHANNEL TRANSFER REGISTERS

The Data Channel Transfer register set is used to control burst and standard EISA data transfers. Each transfer channel has a set of Base and current registers, and also a Transfer Strobe, Configuration, and Status register.

NOTE:
The Base register set and the Transfer Strobe register must be initialized before a transfer can take place. They are not initialized to a fixed value upon reset.

8.2.3.1 Channel 0 and 1 Transfer Base Address Registers (Read/Write)

Each Channel has an associated Base Address register set. The Transfer Base Address registers are programmed with the 32-bit starting address to be used during the data transfer. After the Base registers have been programmed, they should not be programmed again until the contents of the Base registers have been transferred to the Current registers. The Base Address registers are not initialized to a fixed value upon reset.

Channel 0 Local Index Address—43h through 46h (bytes 0 through 3)
Channel 1 Local Index Address—63h through 66h (bytes 0 through 3)

BYTE 3	BYTE 2	BYTE 1	BYTE 0

8.2.3.2 Channel 0 and 1 Transfer Current Address Registers (Read Only)

Each Channel has an associated Current Address register set. The Transfer Current Address registers contain the real-time status of the 32-bit transfer address. The Current Address registers are not initialized to a fixed value upon reset.

NOTE:
The current register set is readable by the local processor. However, there are possible coherency problems involved with reading multiple bytes while the current registers are being updated during a transfer. To avoid these problems, a channel's transfer should be temporarily suspended (using the channel's Configuration Register) before trying to read the channel's current register set.

Channel 0 Local Index Address—53h through 56h (bytes 0 through 3)
Channel 1 Local Index Address—73h through 76h (bytes 0 through 3)

BYTE 3	BYTE 2	BYTE 1	BYTE 0

8.2.3.3 Channel 0 and 1 Transfer Base Count Registers (Read/Write)

Each Channel has an associated Base Count register set. The Transfer Base Count registers are programmed with the number of bytes to be transferred. Each Channel has 22 bits of counter space for a maximum transfer block size of 4 Mbytes. Bits 22 and 23 are used for channel control. The Base Count registers are not initialized to a fixed value upon reset.

Channel 0 Local Address—40h through 42h (bytes 0 through 2)
Channel 1 Local Address—60h through 62h (bytes 0 through 2)

BYTE 2			BYTE 1	BYTE 0
Bit 23	Bit 22	Bit 16–21	Bits 8–15	Bits 0–7

Bit 23 — R/W
- 1 = Start transfer as soon as base register set is copied to current register set
- 0 = Hold transfer after current register set is loaded Wait for transfer suspend bit 0 to be reset

Bit 22 — W
- 1 = Transfer from bus master expansion board to EISA bus (EISA write)
- 0 = Transfer from EISA bus to bus master expansion board (EISA write)

Bits 0–21 — R/W
Transfer byte count

If bit 23 in the Base Count register is not set to a 1, the channel suspend bit (CFGSU) in that channel's corresponding configuration register is automatically set to a 1. The bit will be set during the Base register to Current register transfer. This insures that a channel request for that channel is not generated. When the local processor resets the channel suspension bit to 0 in the corresponding Configuration register, a transfer request will be generated.

NOTE:
If the initial byte count is programmed to be "0", no transfer request will be generated and no transfer will occur.

8.2.3.4 Channel 0 and 1 Transfer Current Count Registers (Read Only)

Each Channel has an associated Current Count register set. The Transfer Current Count registers contain the 22-bit value representing the number of bytes remaining to be transferred on the channel. This value can be from byte to 4 Mbytes. Bit 23 is reserved. Bit 22 is used to indicate the direction of the transfer. Upon reset, the Current Count registers are not initialized.

Channel 0 Local Index Address—50h through 52h (bytes 0 through 2)
Channel 1 Local Index Address—70h through 72h (bytes 0 through 2)

BYTE 2			BYTE 1	BYTE 0
Bit 23	Bit 22	Bit 16–21	Bits 8–15	Bits 0–7

Bit 23 — Reserved

Bit 22 — 1 = Current transfer is from bus master expansion board to EISA bus

0 = Current transfer is from EISA bus to bus master expansion board

Bits 0–21 — Current transfer byte count

8.2.4 DATA TRANSFER STATUS/CONTROL REGISTERS

8.2.4.1 Channel 0 and 1 Transfer Strobe Registers (Write Only)

Each channel has an associated Transfer Channel Strobe register. The Strobe register is used to initiate the transfer of information from the Base register set to the Current register set. The act of writing to this register will initiate the Base to Current transfer. There are no bits to this register, the data written to this register is ignored and the register cannot be read.

If bit 23 in the Transfer Base Count Register is set to a 1, the data transfer will be requested immediately. Otherwise, the transfer will wait until the transfer suspend bit CFGSU for the corresponding channel is reset. The transfer suspend bit is located in the Configuration register.

Channel 0 Local Index Address—49h
Channel 1 Local Index Address—69h

8.2.4.2 Channel 0 and 1 Transfer Channel Configuration Registers (Read/Write)

Each channel has an associated Transfer Configuration register. Upon reset, the Configuration register set is reset to 0. The Configuration register set is used to configure the channels as follows:

Channel 0 Local Index Address—48h
Channel 1 Local Index Address—68h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CFGEA	CFGIE	CFGIT	CFGFF	CFGBR	CFGCL	CFGEI	CFGSU

Bit 7 — CFGEA 1 = Enable real-time address transfer to transfer buffer logic

0 = Disable real-time address transfer to transfer buffer logic

Bit 6 — CFGIE 1 = Enable interrupt on external TEOP

0 = Disable interrupt on external TEOP

Bit 5 — CFGIT 1 = Enable interrupt on transfer complete

0 = Disable interrupt on transfer complete

Bit 4 — CFGFF 1 = Give up ownership of EISA bus if a transfer interruption occurs on this channel

0 = Retain ownership of EISA bus if a transfer interruption occurs on this channel

Bit 3 — CFGBR 1 = Enable EISA burst transfer

0 = Disable burst transfers (channel uses non-burst (2 BCLK) cycle transfers)

Bit 2 — CFGCL 1 = Clear channel

Stop any transfers and flush the data FIFO

0 = No operation

Always returns a 0 when read

Bit 1 — CFGEI 1 = Disable loading of current registers from bases if Status Register Bit (TSTET) on this channel is set

Allow loading of current registers when (TSTET) is reset (or not set)

0 = Always load current register from base registers

Bit 0 — CFGSU 1 = Temporarily suspend transfer

0 = Allow transfer to proceed

The CFGEA Bit enables the real-time address transfer to the transfer buffer logic. If the CFGEA bit is set to a 1, the transfer buffer real-time address for the active channel is transferred to the transfer buffer logic each time that channel regains the bus and the start address is transferred each time the Base register contents are loaded into the corresponding current registers. If the CFGEA bit is set to 0, the address load signal (TLD#) is activated only when the Base is loaded into the Current register (refer to Section 5.3).

The CFGIE Bit enables an interrupt on the external transfer end signal TEOP#. An interrupt will be generated only if the BMIC detects TEOP# and TACK# active at the same time before the byte count expires.

If TEOP# and TACK# are active on an EISA read before the byte count reaches zero, then an interrupt will be generated immediately. ON an EISA write, the effect of the TEOP# will not be seen until the FIFO is emptied, so an interrupt will happen only if the FIFO is emptied before the byte count expires.

The CFGIT Bit enables an interrupt on transfer complete (EOP).

The CFGFF Bit controls whether EISA bus ownership is relinquished after a transfer interruption or maintained. When a channel is interrupted for any reason, (1K page break, FIFO stall, channel clear, transfer suspended, transfer complete, or transfer terminated), the BMIC may relinquish the EISA bus depending on the state of the CFGFF bit in the above register. The function of the CFGFF bit, as related to the above channel interruptions, is as follows:

If the CFGFF bit = 1, the BMIC will relinquish control of the EISA bus upon the detection of any of the above interruptions. This will occur regardless if there are additional data transfer requests pending. If there are additional data transfer requests pending, the BMIC will reassert MREQ# two BCLK later to reacquire the EISA bus.

If the CFGFF bit = 0, the BMIC will retain ownership of the EISA bus as long as there are data requests pending and a preempt timer time-out has not occurred. If there are no additional data requests pending, the BMIC will relinquish ownership of the EISA bus upon the detection of a transfer interruption. The BMIC will follow the arbitration priority scheme outlined in the arbitration section when servicing the data transfer requests after a transfer interrupt has occurred.

NOTE:
During a FIFO pause, CFGFF is ignored.

The CFGBR Bit defines the type of transfer cycles (burst or non-burst) that can be requested on the transfer channel. If burst cycles have been selected and system memory is unable to run burst cycles, the BMIC will default to non-burst (two BCLK) or mismatched cycles.

The CFGCL Bit is used to generate a channel clear. A channel clear terminates the current transfer and flushes the associated FIFO. The FIFO is reset during the next Base to Current register copy.

If a channel is enabled for a transfer during a Channel clear, the BMIC will generate an end of process by asserting (TEOP#). If the channel is not enabled for a transfer, a TEOP# will not be generated. The channel clear will be active for at least two complete BCLK cycles.

The CFGEI Bit is used to disable the reloading of the Current register set from the Base register set during a transfer termination due to the transfer buffer logic asserting TEOP# and TACK# simultaneously. This allows the local processor to read the contents of Current registers before the Current registers are cleared by the reloading of new values from the base Registers. The local processor can enable reloading of the Current from the Base, either by clearing bit 1 in that channel's Transfer Status register (i.e., servicing the interrupt due to TEOP#) or by resetting the CFGEI bit to a "0".

The CFGSU Bit is used to temporarily suspend the data transfer.

8.2.4.3 Channel 0 and 1 Transfer Channel Status Registers (Read/Write)

Each channel has an associated Transfer Channel Status register. Bits 2 through 4 are read only and bits 5 through 7 are reserved. Upon reset, the Channel Status register set is reset to 0.

Channel 0 Local Index Address—4Ah
Channel 1 Local Index Address—6Ah

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	TST1K	TSTFF	TSTEN	TSTIP	TSTET	TSTTC

Bit 7, 6, 5 —W — Reserved. 0 should be written into these bits during writes. Ignore any data on these bits during register reads

Bit 4 — R
- 1 = The FIFO is currently stalled
- 0 = The FIFO is active

Bit 3 — R
- 1 = The transfer channel is enabled for transfer (transfer in progress)
- 0 = The transfer channel is not enabled for transfer (transfer not in progress)

Bit 2 — R
- 1 = A transfer is in progress on this channel (Channel request is active)
- 0 = No transfer is in progress on this channel (Channel request is not active)

Bit 1 — R/W 1 = The transfer was terminated by TEOP# (read)
Reset this bit (write)
0 = No TEOP# termination detected (read)

Bit 0 — R/W 1 = Transfer completed on this channel (read)
Reset this bit (write)
0 = No transfer completion on this channel (read)
No action (write)

Bits (7), (6), and the TST1K Bit are reserved. Any data read from these bits should be ignored.

The TSTFF Bit is read only and indicates whether the FIFO for the corresponding channel stalled.

The TSTEN and TSTIP Bits are read only and indicate whether the corresponding channel is requesting a transfer or whether the channel's transfer is currently in progress.

The TSTET Bit is read/write and is used to indicate the current external TEOP# termination status of the transfer. If TEOP# and TACK are asserted simultaneously by the transfer buffer logic, the BMIC will set bit (1) to a 1 and generate an interrupt to the local processor. The BMIC will not generate the interrupt if the CFGIE bit in the channel's corresponding Transfer Configuration register is set to a 0.

The TSTTC Bit is read/write and is used to indicate the current end-of-process status of the transfer. If an EOP occurs, the BMIC will set bit (0) to a 1 and generate an interrupt to the local processor. The BMIC will not generate the interrupt if the CFGIT bit in the channel's corresponding Transfer Configuration register is set to a 0.

NOTE:
The TSTTC and TSTET Bits are implemented with sticky bits. These bits can be reset by the local processor without affecting the status of the other bits in the register.

8.2.5 PEEK/POKE REGISTERS

The Peek/Poke register set consists of four 8-bit Address registers, four 8-bit Data registers and one Peek/Poke control register. The Address and Data registers are used to define the 32-bit address and data that will be used during the peek/poke cycles, and the Control register is used to request and define the type of cycle that will be generated (peek, poke, or locked exchange). The peek/poke or locked exchange cycle is initiated by write to the Peek/Poke control register. During Reset, the Peek/Poke Address registers and the Control register are reset to 0.

8.2.5.1 Peek/Poke Address Registers (Read/Write)

The four 8-bit Peek/Poke Address registers contain the 32-bit peek/poke address. Only the lower 16 bits are used for I/O cycles. Address bits 0 and 1 are ignored. Upon reset, this register is reset to 0.

Local Index Address—34h through 37h (bytes 0 through 3)

BYTE 3	BYTE 2	BYTE 1	BYTE 0
Bits 24–31	Bits 16–23	Bits 8–15	Bits 2–7

8.2.5.2 Peek/Poke Data Registers (Read/Write)

The four 8-bit peek/poke data registers hold the data for the peek/poke cycle. Each peek/poke data register is associated with one byte lane. During peek transfers, only those peek/poke data registers whose corresponding byte enable bit is set in the peek/poke control register contain valid data. During poke transfers, the data must be placed in the appropriate register as determined by the corresponding byte enable bit.

Local Index Address—30h through 33h (bytes 0 through 3)

BYTE 3	BYTE 2	BYTE 1	BYTE 0
Bits 24–31	Bits 16–23	Bits 8–15	Bits 0–7

The Shared register timings (t85, t93, t96–t98) are used when accessing the Peek/Poke Data registers.

8.2.5.3 Peek/Poke Control Register (Read/Write)

The Peek/Poke Control register is written to by the local processor when a peek/poke transfer is desired over the EISA bus. Upon reset, this register is reset to 0.

Local Index Address—38h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7 — Reserved. Set to 0

Bit 6 — 10 = Do read cycle (peek)

Bit 5 — 01 = Do write cycle (poke)
11 = Do locked exchange cycle (peek/poke)
00 = Do Nothing (Nop)

Bit 4 — 1 = Do memory cycle
0 = Do I/O cycle

Bit 3, Bit 2, Bit 1, Bit 0 — 1 = Byte enable for given byte lane
0 = Byte disable for given byte lane

Bits (6) and (5) are used to define the type of cycle requested (peek, poke or locked exchange).

Bit (4) defines whether the cycle is memory or I/O.

Bits (3–0) are used to define the byte enables for the doubleword data written to or read from the Peek/Poke data register. Peek/Poke cycles will not be generated for illegal combinations of byte enables.

ILLEGAL COMBINATIONS OF BYTE ENABLES:

Bits 3–0	IBE# <3:0>
0000	1111
0101	1010
1001	0110
1010	0101
1011	0110
1101	0010

NOTE:
Bits 3–0 in the above register are active high whereas the EISA byte enables (IBE# <3–0>) are active low.

8.2.6 I/O RANGE DECODE REGISTERS

The I/O Decode Range register set consists of two I/O Decode Range Base Address registers and two I/O Decode Range Control Registers. The Address registers are used to define the address range of interest to the expansion board and the Control registers are used to define the decode range size, type of decode (slot specific or general), and the response of the local I/O (32-bit EISA or 8-bit EISA). The I/O decode register set controls the two IOSEL# pins on the BMIC. Each pin has an associated Address and Control register.

Upon reset, the I/O Decode Range registers are initialized according to the following table:

Local Processor	*Local Processor Not Present		*Local Processor Present	
Range	**Rng 0**	**Rng 1**	**Rng 0**	**Rng 1**
Control Registers	60h	60h	20h	20h
Address Registers	E0h	00h	00h	00h

*Refer to Section 7.3 for information regarding "local processor present" or "local processor not present".

8.2.6.1 Range 0 and 1 I/O Decode Base Address Register (Read/Write)

Each Decode range and IOSEL# pin has an associated I/O Decode Range Base Address register.

Range 0 Local Index Address—39h
Range 1 Local Index Address—3Bh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

During general I/O decode, bits 0–7 are used to compare against EISA address lines LA<9:2>. During slot specific decode, bits 6 and 7 are compared against EISA address lines LA<11:10> and bits 0–5 are compared against EISA address lines LA<7:2> (refer to Section 4.8).

8.2.6.2 Range 0 and 1 I/O Decode Control Registers (Read/Write)

Each Decode range and IOSEL# pin has an associated I/O Decode Range Control register.

Range 0 Local Index Address—3Ah
Range 1 Local Index Address—3Ch

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7 — 1 = Respond as a 32-bit EISA I/O device
0 = Respond as an 8-bit EISA I/O device

Bit 6 — 1 = Slot Specific I/O Decode
0 = General I/O Decode

Bit 5 — 1 = IOSEL# held during CMD# active
0 = IOSEL# follows I/O address changes

Bit 4 — 1 = Do not compare I/O Range Base Register and corresponding EISA address bit (Mask)
Bit 3 —

Bit 2 — 0 = Compare I/O Range Base Register with corresponding EISA address bit
Bit 1 —
Bit 0 —

Refer to Section 4.7 for a complete description of the I/O Decode Range Control registers and the BMIC decode function in general.

8.2.7 TRANSFER BUFFER INTERFACE (TBI) REGISTERS (READ/WRITE)

The TBI registers are programmed to provide the 16-bit start address of the data transfer for use by the transfer buffer logic (refer to Section 5.3). Each transfer channel has a corresponding TBI Base and Current Address register set. The contents of the TBI Base Address registers are transferred to the TBI Current Address registers during a write to the channel's corresponding Transfer Channel Strobe Register.

8.2.7.1 Channel 0 and 1 TBI Base Address Registers (Read/Write)

The BMIC provides two 8-bit TBI Base Address registers per channel. The registers are programmed with the 16-bit start address of the data transfer in the transfer Buffer memory space. The TBI Base Address register set is not initialized to a fixed value upon reset.

Channel 0 Local Index Address—4Bh and 4Ch (byte 0 and 1)
Channel 1 Local Index Address—6Bh and 6Ch (byte 0 and 1)

Byte 1	Byte 0

TBI Base Address Register (Bytes 0–1 (Read/Write)

8.2.7.2 Channel 0 and 1 TBI Current Address Registers (Read Only)

The BMIC provides two 8-bit TBI Current Address registers per channel. The TBI Current Address registers contain the 16-bit real-time address of the data transfer. The contents of the Current register set are transferred to the external buffer logic at the beginning of every new data block transfer. The BMIC may also be programmed to transfer the contents of the Current Address register each time the corresponding channel regains control of the bus (refer to Section 5.3). The TBI Base Address register set is not initialized to a fixed value upon reset.

Channel 0 Local Index Address—58h and 59h (byte 0 and 1)
Channel 1 Local Index Address—78h and 79h (byte 0 and 1)

Byte 1	Byte 0

NOTE:
The TBI current registers contain real-time status and may change at anytime. If a stable value is needed while reading a set of these registers, the channel should be temporarily suspended by setting the CFGSU bit in the Channel Configuration register to a 1 before these registers are read.

9.0 DETAILED PIN DESCRIPTION

9.1 EISA Interface Signals

Pin Name	Description
START#	I/O TRI-STATED (EISA CYCLE START STROBE) The START# signal provides timing control at the start of a cycle. During EISA master mode, the BMIC drives this signal low after LA<31:2> and M/IO become valid and negates START# on the rising edge of BCLK after one BCLK cycle time. During EISA slave mode, the BMIC uses this signal to indicate the start of a slave bus cycle. It is sampled on the rising edge of BCLK. Upon reset, this pin is tri-stated and placed in input mode.
CMD#	INPUT (EISA COMMAND STROBE) The CMD# provides timing control within the cycle. The 82358 Bus Controller asserts CMD# on the rising edge of BCLK, simultaneously with the negation of START#. CMD# is held low until the end of the cycle. The BMIC uses CMD# in EISA slave mode for timing control during internal Shared register read/write accesses.
M/IO	I/O TRI-STATED (EISA MEMORY/IO CYCLE STATUS PIN) M/IO is used to indicate that the type of cycle in progress is a memory cycle (high) or I/O cycle (low). M/IO is pipelined from one cycle to the next and must be latched by the addressed memory slave if needed for the whole cycle. During EISA master mode, the BMIC drives this signal. The BMIC will drive this pin high during burst and non-burst (two BCLK) cycles. The value of M/IO in a Peek/Poke or locked exchange cycle depends on the programmed value of bit 4 in the Peek/Poke Control register. During EISA slave mode, the M/IO pin is an input. As a slave, the BMIC will only respond as an I/O device. Upon reset, this pin is tri-stated and placed in input mode.
W/R	I/O TRI-STATED (EISA WRITE/READ CYCLE STATUS PIN) The W/R status signal identifies the cycle as a write (high) or read (low). W/R is pipelined from one cycle to the next and must be latched by the addressed memory slave if needed for the whole cycle. During EISA master mode, the BMIC drives this signal. During EISA slave mode, this pin is an input. Upon reset, W/R is tri-stated and placed in input mode.
EXRDY	I/O OPEN COLLECTOR (EISA READY SIGNAL) EXRDY is used by EISA I/O and memory slaves to request wait states during a cycle. Each wait state is one BCLK period. During EISA master mode, the BMIC first samples this signal on the falling edge of BCLK after CMD# is asserted. If it is low, the BMIC will insert a wiat state, and continue inserting wait states as long as EXRDY is low at each successive falling edge of BCLK. During EISA slave mode, the BMIC drives EXRDY inactive until it is ready to complete cycles addressed to it. The EXRDY pin is an open collector output.
EX32#	I/O OPEN COLLECTOR (EISA 32-BIT SLAVE RESPONSE PIN) EX32# is an open collector and is used by memory or I/O slaves to indicate their support of 32-bit transfers. During EISA master mode, the BMIC samples EX32# on the same rising edge of BCLK that START# is deasserted. The BMIC uses this pin to determine if the addressed memory slave is capable of 32-bit transfers. During peek/poke and non-burst EISA data transfers, the BMIC is a 32-bit master only and will allow the 82358 Bus controller to do all necessary bus conversions. During EISA slave mode, the BMIC drives EX32# low if it has 32-bit data to send to the EISA bus, otherwise this signal is inactive.

9.1 EISA Interface Signals (Continued)

Pin Name	Description
MASTER16#	OUTPUT OPEN COLLECTOR (EISA 16-BIT MASTER CONTROL) In master mode, the BMIC will assert MASTER16# (at the same time as START#) for one BCLK period when it is capable of downshifting from a 32-bit master to a 16-bit master. The BMIC will downshift if necessary during memory burst transfers only. The BMIC will automatically downshift from a 32- to 16-bit master if the EX32# signal is sampled inactive and the SLBURST# signal is sampled active. MASTER16# has no function in slave mode.
AEN	INPUT (EISA ADDRESS ENABLE SIGNAL) The BMIC uses AEN when in EISA slave mode to qualify I/O addresses. When negated (low), the BMIC uses AEN to decode possible accesses to its general and slot specific I/O space. When asserted (high), the address on the EISA bus will be ignored by the BMIC. AEN is sampled on the falling edge of CMD#. This signal is not used in master mode.
MSBURST#	OUTPUT TRI-STATED (EISA MASTER BURST SIGNAL) The BMIC asserts MSBURST# to indicate to the addressed memory slave that the BMIC will provide burst cycles. If the BMIC samples SLBURST# active on the rising edge of BCLK after START# is asserted, the BMIC will activate MSBURST# on the next BCLK falling edge and will proceed with burst cycles. If the BMIC samples SLBURST negated, MSBURST# will not be activated and the BMIC will proceed with either non-burst (two BCLK) or mismatched cycles, depending on the size of the slave device addressed. This signal is not used in slave mode. Upon reset, this pin is tri-stated.
SLBURST#	INPUT (EISA SLAVE BURST SIGNAL) The BMIC uses this signal in master mode to determine if the addresed slave memory is capable of supporting burst transfers. If the BMIC samples SLBURST# active on the rising edge of BCLK after START# is asserted, the BMIC will proceed with burst cycles. If the BMIC samples SLBURST negated, either non-burst (two BCLK) or mismatched cycles will be generated.
LOCK#	OUTPUT TRI-STATED (EISA RESOURCE LOCK SIGNAL) The BMIC asserts this signal to guarantee exclusive memory and I/O access during locked peek/poke exchange. Upon reset, this pin is tri-stated.
MREQ#	OUTPUT (EISA MASTER BUS REQUEST SIGNAL) MREQ# is asserted by the BMIC to request EISA bus access. The BMIC will begin driving the bus with the address and control signals on the falling edge of BCLK, two BCLKs after MAK# is sampled active. During an EISA write transfer, MREQ# will not be asserted until the FIFO on the selected channel is full. During an EISA read transfer, MREQ# will be asserted immediately after receiving a transfer request, assuming that a slave cycle is not currently in progress. Upon reset, this pin is driven inactive high.
MAK#	INPUT (EISA MASTER BUS ACKNOWLEDGE SIGNAL) The MAK# signal is asserted by the 82357 (ISP) to grant EISA bus access to the BMIC. The BMIC samples MAK# on the falling edge of BCLK and will begin driving the bus with the address and control signals on the falling edge of BCLK, two BCLKs after MAK# is sampled active. The MAK# signal may be negated by the ISP to indicate to the BMIC that another device requires EISA bus access. The BMIC will negate MREQ# to release the bus within 64 BCLKs (8 μs) of sampling MAK# negated.
EINT	OUTPUT OPEN COLLECTOR (EISA INTERRUPT REQUEST SIGNAL) The EINT line is used by the BMIC to interrupt the system CPU or EISA bus master to request service. EINT can be programmed for either edge or level-triggered operations and is an open collector output in level-triggered mode. Upon reset, EINT is placed in level-triggered mode and floating.
BCLK	INPUT (EISA BUS CLOCK) This clock signal is used by the BMIC to synchronize the EISA control signals and data transfers to the system clock. BCLK typically runs at a frequency of 8.33 MHz with a normal duty cycle of 50%. The BCLK period is sometimes extended by the 82358 (EBC) by up to one BCLK period for synchronization purposes.

9.1 EISA Interface Signals (Continued)

Pin Name	Description
RESET	INPUT (EISA RESET SIGNAL) This signal is used by the BMIC to initialize all of its internal registers and state machines to a known state. This signal is asynchronous with respect to BCLK. To reset the BMIC properly, the RESET signal must be active for eight BCLK periods.
IDAT<31:0>	I/O TRI-STATED (EISA DATA LINES/UPPER 22 ADDRESS LINES) These data signals interface to the EISA bus through external, 74F245 bi-directional TTL buffers. The upper 22 data lines are also multiplexed to function as the upper 22 EISA address lines. The 22 upper address signals are latched into external 74F573 TTL latches during transfers as necessary by the BMIC. Both the external data buffers and the address latches are controlled by the BMIC during all slave and master mode data transfers. Upon reset, these pins are tri-stated.
IADS<11:10>	(INPUT) (EISA ADDRESS INPUT LINES) These two address lines are input only and are only used during slave mode. They are used along with IADS<9:2> and EISA byte enables IBE<3:0># for I/O address decoding. The corresponding EISA output address lines LA<11:10> are part of the upper 22 address lines that are multiplexed and sent out through the upper 22 data lines.
IADS<9:2>	I/O TRI-STATED (EISA LOWER ADDRESS LINES) These eight address lines are part of the lower EISA address lines and are connected directly to the EISA bus. When the BMIC is a master, it drives these lines directly to the EISA bus. The upper 22 addresses are latched from the data bus. IADS<9:2> are pipelined from one cycle to the next and should be latched by the addressed slave if required for the whole cycle. When the BMIC is a slave, it monitors these lines along with EISA address lines IADS<11:10> and EISA byte enables IBE<3:0># for I/O address decoding. Upon reset, these pins are tri-stated and placed in input mode. The following address lines are used during I/O decoding as shown: Slot specific I/O address decoding (expansion board)—IADS<11:2> Slot specific I/O address decoding (shared registers)—IADS<11:2>/IBE<3:0># General I/O address decoding (expansion board)—IADS<9:2>
IBE<3:0>#	I/O TRI-STATED (EISA BYTE ENABLES) IBE#<3:0> are the byte enables of the EISA bus and identify the specific bytes that are active during the current EISA bus cycle. During EISA master mode, the BMIC drives these signals. IBE#<3:0> are pipelined from one cycle to the next and should be latched by the addressed slave if required for the whole cycle. During EISA slave mode, the byte enables are inputs and are used along with EISA address lines IAD<11:2> for internal shared register decoding. Upon reset, these pins are tri-stated and placed in input mode.

9.2 EISA Buffer Control Signals

Pin Name	Description
UALOE#	OUTPUT (EISA UPPER ADDRESS LATCH STROBE AND OUTPUT ENABLE) The UALOE# signal is used by the BMIC to control the external latching of the upper 22 address lines LA<31:10>. UALOE# is designed to be connected to the latch enables and output enables of the 74F573 external address latches. The BMIC updates the external address latches at the beginning of all master mode transfers. The desired address value is placed on the IDAT<31:10> lines and latched by the external latches on the falling edge of UALOE# at the beginning of the transfer. During EISA master mode to enable the EISA address lines <31:10>, the BMIC drives UALOE# low on the rising edge of BCLK, one BCLK prior to the falling edge of START#. UALOE# will remain active until the end of the cycle. During slave mode, the BMIC holds UALOE# high to disable the latches. For additional information with regards to the timing for this signal, refer to the A.C. timing and Basic Function timing sections. Upon reset, this pin is driven inactive high.
IDDIR	OUTPUT (EISA DATA DIRECTION SIGNAL) The IDDIR signal is used by the BMIC to control the direction of the external 74F245 data buffers. During data transfers from the BMIC to the EISA bus, this signal will be driven low. During data transfers from the EISA bus to the BMIC, this signal will be driven high. For additional information regarding the timing for this signal, refer to the A.C. timing and Basic Function timing sections (master and slave). Upon reset, this pin is driven high.
IDOE23# IDOE1# IDOE0#	OUTPUT (EISA DATA BYTE LANE BUFFER ENABLES) The IDOE# signals are used by the BMIC to control the output enables on the external 74F245 data buffers. The IDOE# signals will be driven so that the data buffers are enabled at the appropriate times during master and slave transfers. For additional information with regards to the timing for these signals, refer to the A.C. timing and Basic Function timing sections. Upon reset, these signals are driven inactive high.

9.3 Address Decode Signals

Pin Name	Description
IOSEL# <1:0>	OUTPUT (ADDRESS RANGE DECODE OUTPUTS) The IOSEL# signals are used by the BMIC to enable external logic on the expansion board during slot specific and general purpose I/O decode. These pins become active when the LA<11:2> address lines on the EISA bus contain a value mapped into one of the two possible I/O address decode ranges provided by the BMIC (refer to Section 7.3). Upon reset, these pins are driven inactive high.

9.4 Transfer Buffer Interface Signals

Pin Name	Description
TRQ#	OUTPUT (LOCAL DATA TRANSFER REQUEST SIGNAL) When a data transfer is desired over the Transfer Buffer interface, TRQ# is driven low, indicating to the transfer buffer logic that a transfer is following. TRQ# will remain active until the data transfer is completed or a transfer interruption occurs. Upon reset, this pin is driven inactive high.
TACK#	INPUT (LOCAL DATA TRANSFER ACKNOWLEDGE SIGNAL) External logic uses this signal to acknowlege the transfer of a data item (16-bit word) over the Transfer Buffer interface.
TLD#	OUTPUT (LOCAL ADDRESS COUNTER LOAD SIGNAL) This signal when asserted (low) is used to load the transfer start address and the transfer real-time address into an external address counter as required for data transfers (refer to Section 5.3). TLD# is asserted at the beginning of all new channel accesses to the transfer buffer logic and will remain asserted until acknowledged by TACK#. Upon reset, this pin is driven inactive high.
TDIR	OUTPUT (DATA TRANSFER DIRECTION SIGNAL) This signal is used to inform the transfer buffer logic as to the direction of the current data transfer. When driven (high), data will be transferred from the EISA bus to the expansion board. When driven (low), data will be transferred from the expansion board to the EISA bus. TDIR will be held valid whenever TLD# and TRQ# are active. TDIR will not change states when TRQ# is active. Upon reset, this pin is driven high.
TCHAN	OUTPUT (DATA CHANNEL TRANSFER CHANNEL SELECT SIGNAL) This signal is used by the BMIC to inform the transfer buffer logic as to which channel will be active during the transfer. When driven (low), transfer channel 0 is active and when driven (high), transfer channel 1 is active. TCHAN has the same timings as TDIR and will not change states when TLD# or TRQ# are active. Upon reset, this pin is driven low.
TDAT<15:0>	I/O TRI-STATED (TRANSFER DATA LINES) This bidirectional bus is the BMIC's Transfer Buffer interface data bus. It is used during data transfers between the external transfer buffer logic and the BMIC. The data transferred across the TDAT bus is word aligned. The data lines are also used to transport the transfer address to the transfer buffer logic on the expansion board (refer to Section 5.3). The TDAT bus can be unconditionally disabled by driving the TDOE# signal high. **NOTE:** During EISA write data transfers, the TDAT lines are inputs and operate independent of the value of TDOE#. Upon reset, the TDAT bus is tri-stated.
TDOE#	INPUT (TRANSFER INTERFACE DATA OUTPUT ENABLE) When driven high, this pin can be used by external logic to unconditional disable the BMIC from driving the TDAT<15:0> lines. This feature eliminates the need for the BMIC to gain prior permission to drive the TDAT bus and also allows external logic the ability to time-share the TDAT bus.
TEOP#	OUTPUT OPEN COLLECTOR (TRANSFER END-OF-PROCESS SIGNAL) This signal is an open collector signal that indicates the end of a transfer to external transfer buffer logic. TEOP# is driven low by the BMIC to indicate the end of transfer. The TEOP# pin requires an external 2.5K to 3.2K pullup resistor for proper operation.
TCLK	INPUT (TRANSFER CLOCK) All transfer control signals are synchronous to this clock. The frequency should be in the range of 16 MHz to 20 MHz to maintain a 33 Mbyte/sec burst transfer rate over the EISA bus. This clock may be completely asynchronous to the EISA BCLK signal.

9.5 Local Processor Interface Signals

Pin Name	Description
LDAT<7:0>	I/O TRI-STATED (LOCAL PROCESSOR INTERFACE DATA BUS) This bidirectional bus is used to transfer commands and status between the BMIC and the local processor on the expansion board. If a local Processor is not present, this bus will need to be connected to the ISA bus (refer to Section 7.2). Upon reset these pins are tri-stated.
LRD#	INPUT (LOCAL PROCESSOR INTERFACE READ STROBE) The local processor asserts LRD# to indicate to the BMIC that it should drive its data onto the LDAT bus. LRD# is asserted for register access to the BMIC's Local Processor interface. The LADS lines and the LCS# signal must be valid 10 ns before the falling edge of LRD# and remain valid until LRD# is deasserted.
LWR#	INPUT (LOCAL PROCESSOR INTERFACE STROBE) The local processor asserts LWR# to indicate to the BMIC that it may latch data from the LDAT bus. LWR# is asserted for write accesses to the BMIC's Local Processor interface. The LADS lines and the LCS signal must be valid 10 ns before the falling edge of LWR# and remain valid until LWR# is deasserted.
LCS#	INPUT (LOCAL PROCESSOR INTERFACE CHIP) A (low) on this pin enables LWR# and LRD# communication between the BMIC and the local processor on the expansion board. The LRD# and LWR# signals are ignored unless the LCS# signal is active. LCS# must be asserted 10 ns before LRD# and LWR# and remain active until the inactive edge of LRD# and LWR#.
LADS<1:0>	INPUT (LOCAL PROCESSOR ADDRESS SELECT) These address lines are used by the local processor to select the Local Data, Local Index, and Local Status/Control registers. The BMIC uses these registers as part of an indexing scheme to access all of its internal registers (refer to Section 3.2.6.1). **LADS1 LADSD0** 0 0 = Local Data register 0 1 = Local Index register 1 0 = Local Status/Control register 1 1 = Reserved
LINT	OUTPUT (LOCAL PROCESSOR INTERRUPT SIGNAL) This signal informs the local processor that an event has occurred which requires the local processor's attention. This pin can be programmed for either active high or active low level operations. After being asserted, LINT will not return to an inactive state until the interrupt has been serviced. The LINT signal is not an open collector output during active low operations and will require external logic if interrupts need to be tied together on the local side. Upon reset, this pin is driven high and placed in active low level mode.
LRDY	I/O (LOCAL PROCESSOR READY) This signal is the acknowledgement from the BMIC to the local processor that it is finished with the current Shared register access cycle. The LRDY pin is also used by external logic to indicate to the BMIC that a local processor is not present. If a local processor is not present, the LRDY signal must be driven low during reset (refer to Section 7.2). If a local processor is present, a weak pullup resistor must be connected to the LRDY output to insure that LRDY is high during the time reset is active.

9.6 Power Supplies

V_{CC} — 11 Power pins

V_{SS} — 13 Ground pins

Total number of power supply pins: 24

10.0 BASIC FUNCTION TIMING DIAGRAMS

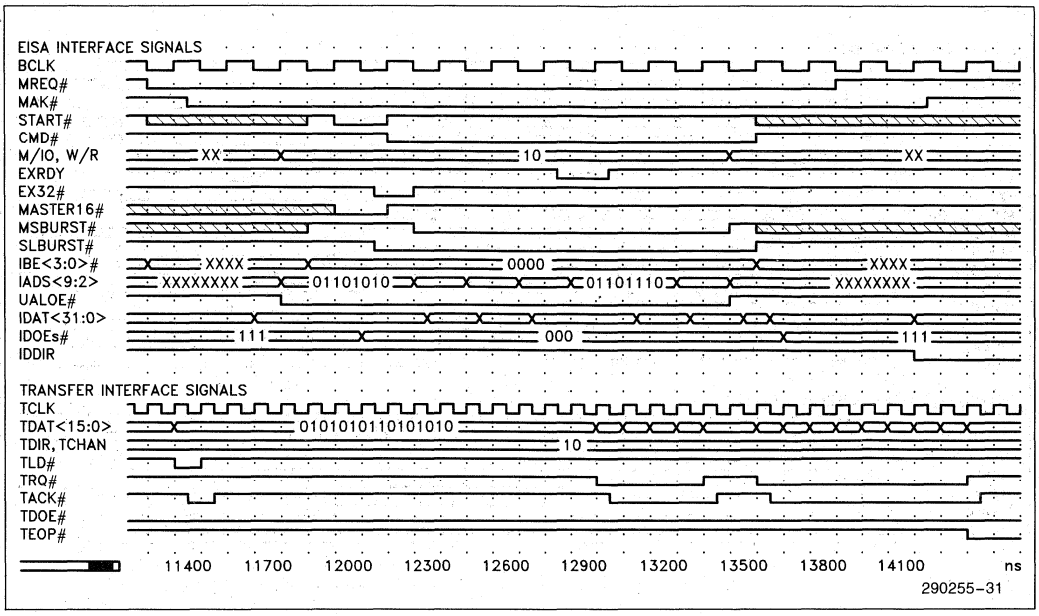


Figure 10-1. 32-Bit Burst Cycle (EISA Read)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

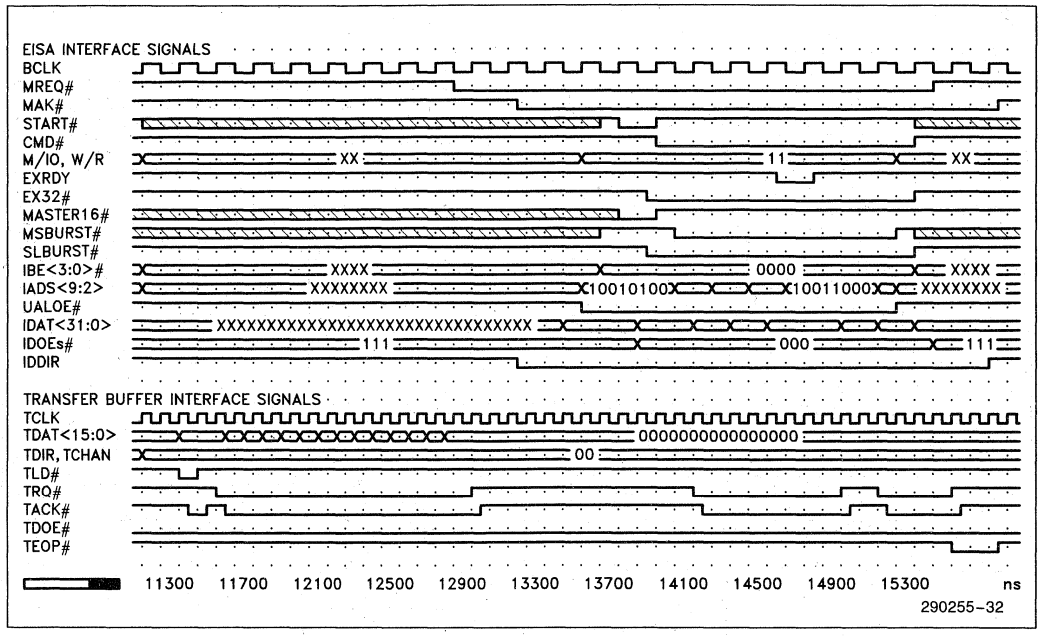


Figure 10-2. 32-Bit Burst Cycle (EISA Write)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

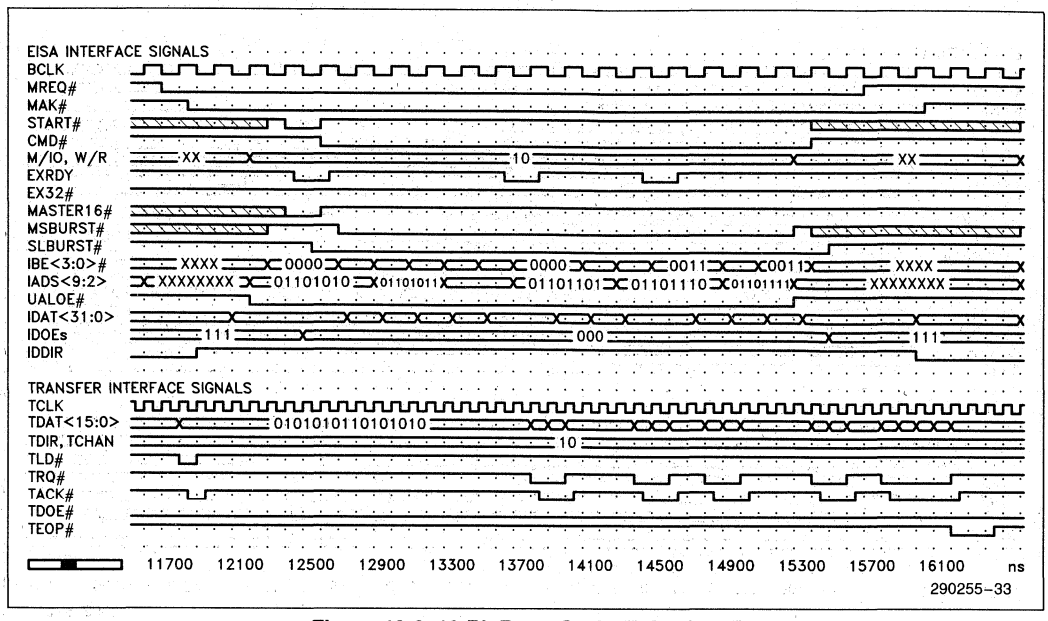


Figure 10-3. 16-Bit Burst Cycle (EISA Read)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

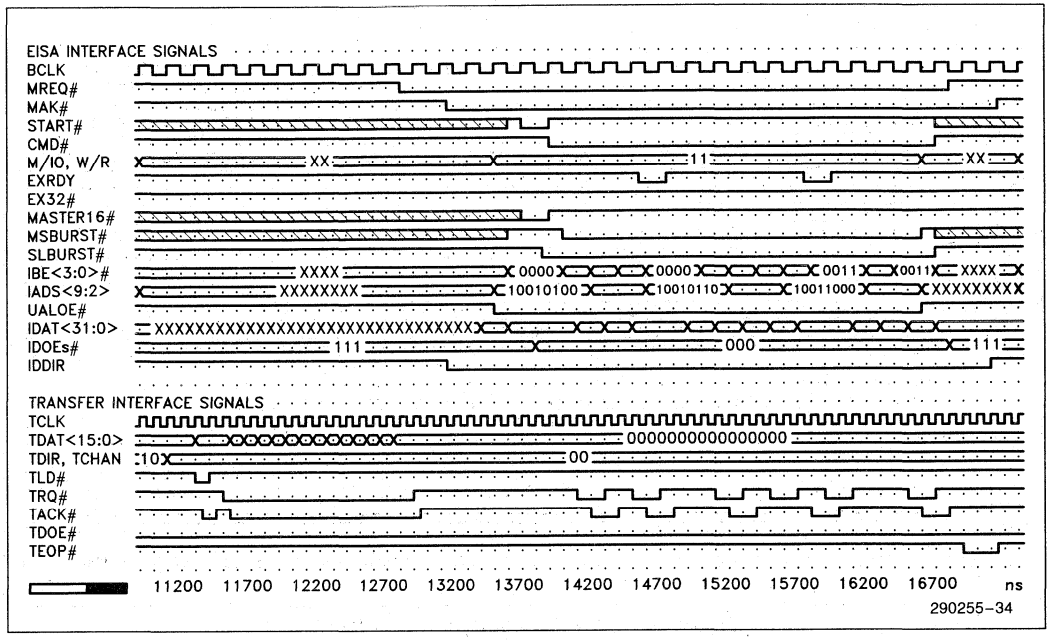


Figure 10-4. 16-Bit Burst Cycle (EISA Write)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

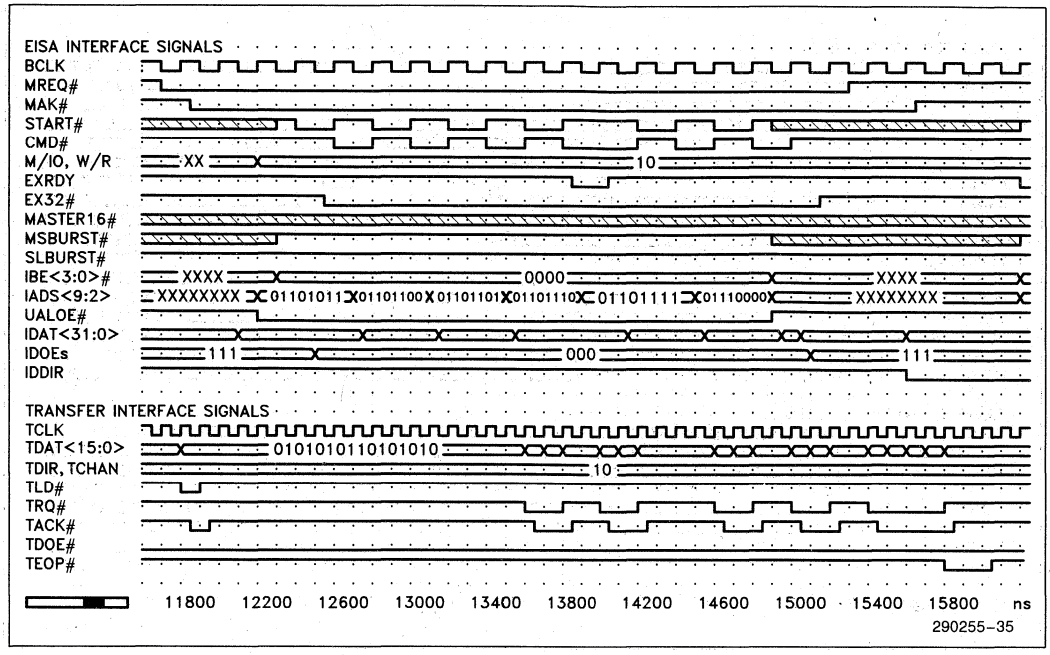


Figure 10-5. 32-Bit Non-Burst Cycle (EISA Read)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

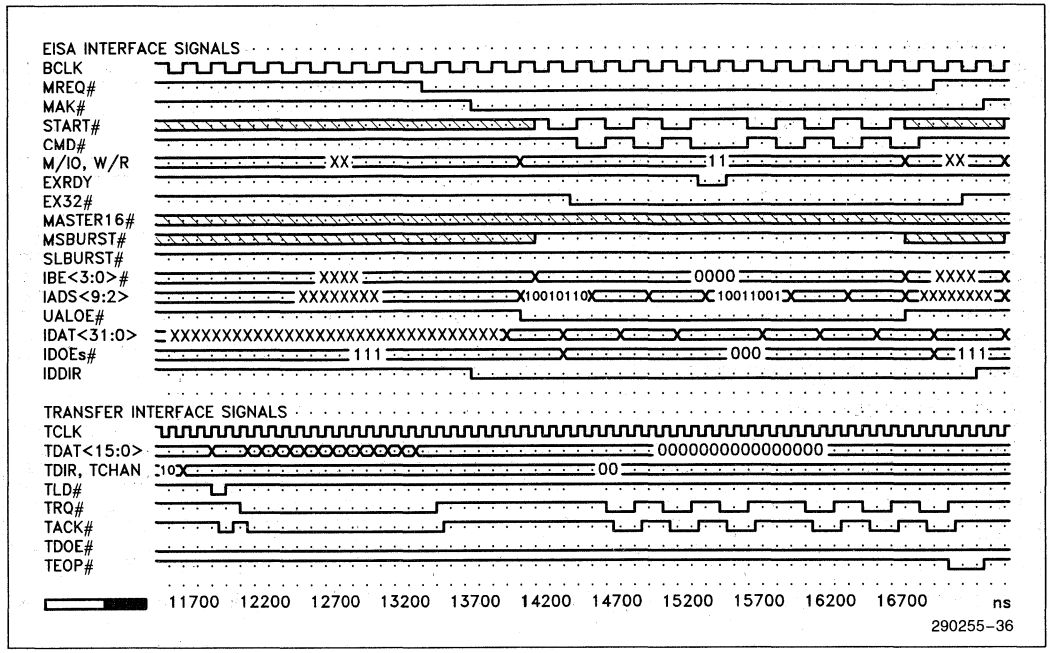


Figure 10-6. 32-Bit Non-Burst Cycle (EISA Write)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

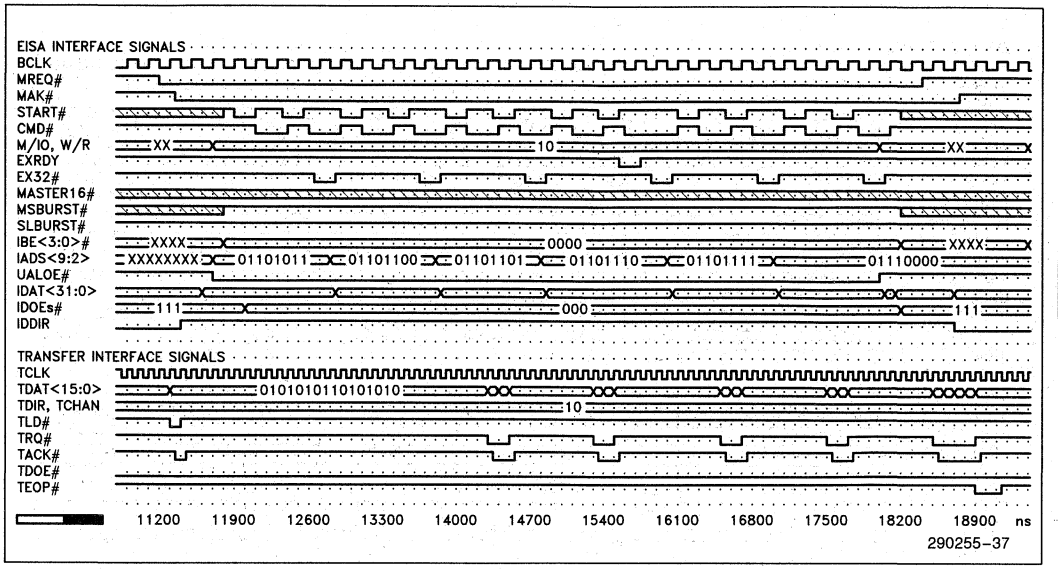


Figure 10-7. Misaligned Cycle (EISA Read)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

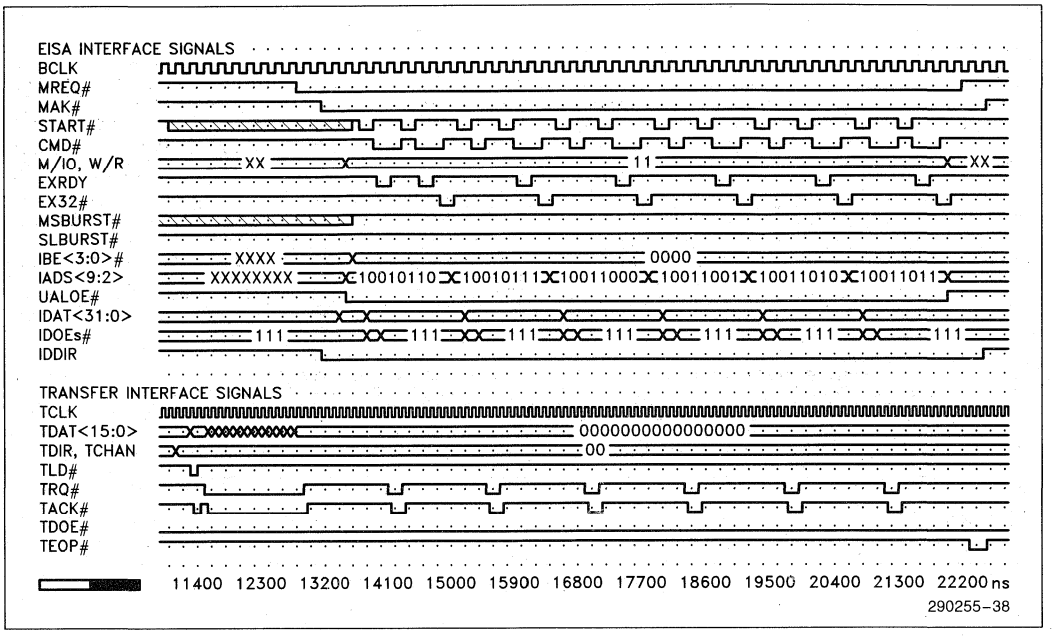


Figure 10-8. Misaligned Cycle (EISA Write)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

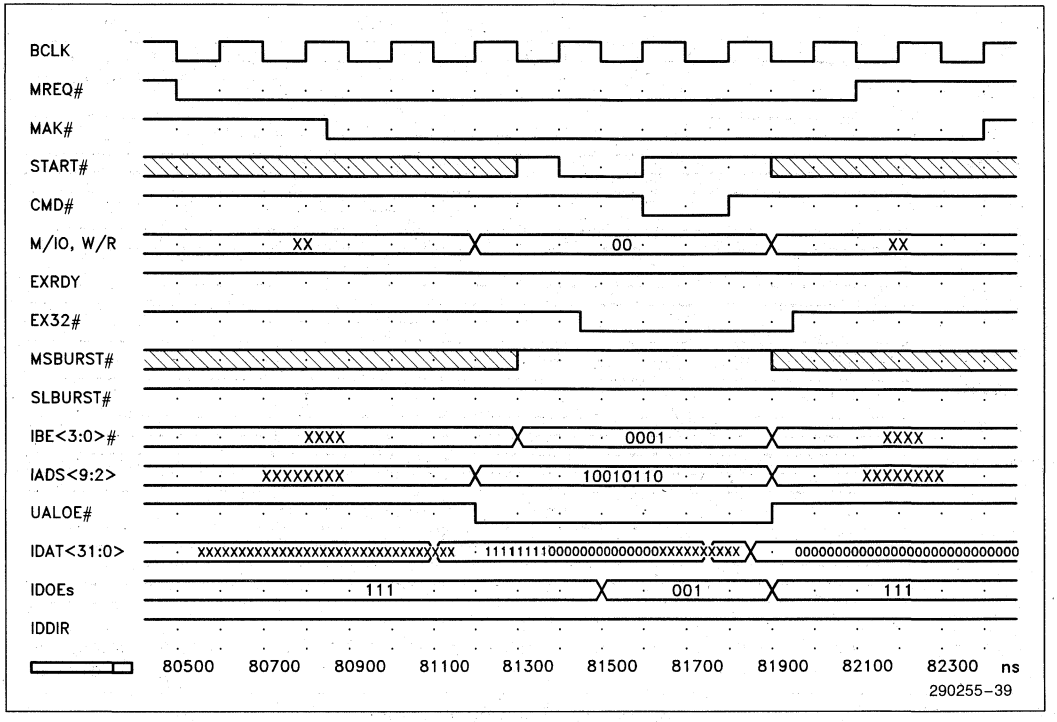


Figure 10-9. I/O Peek Cycle

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

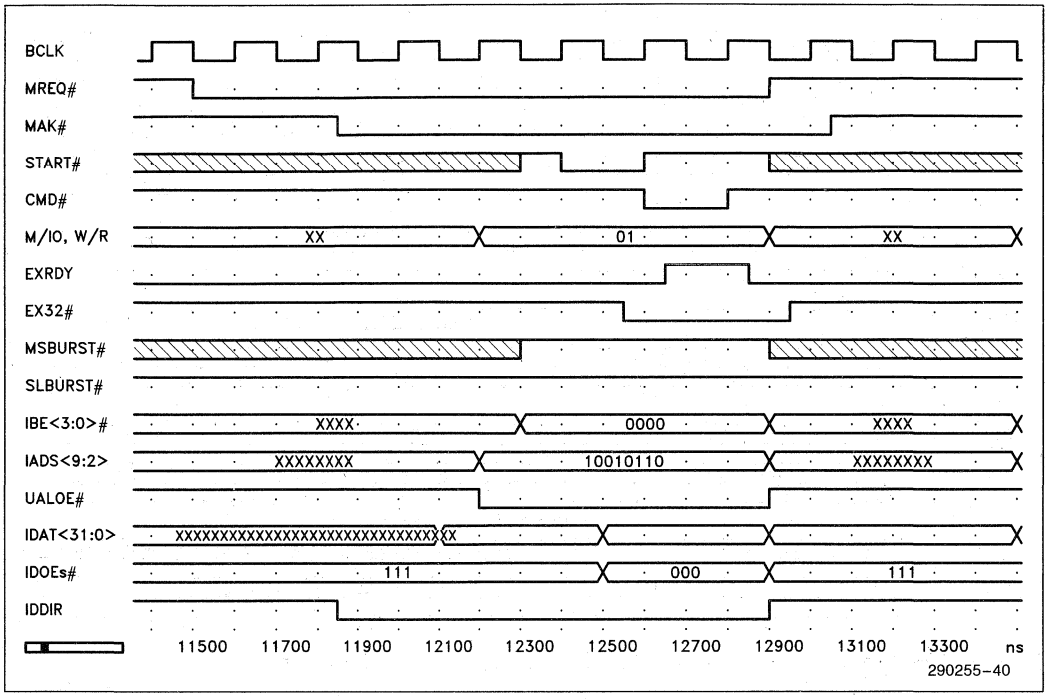


Figure 10-10. I/O Poke Cycle

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

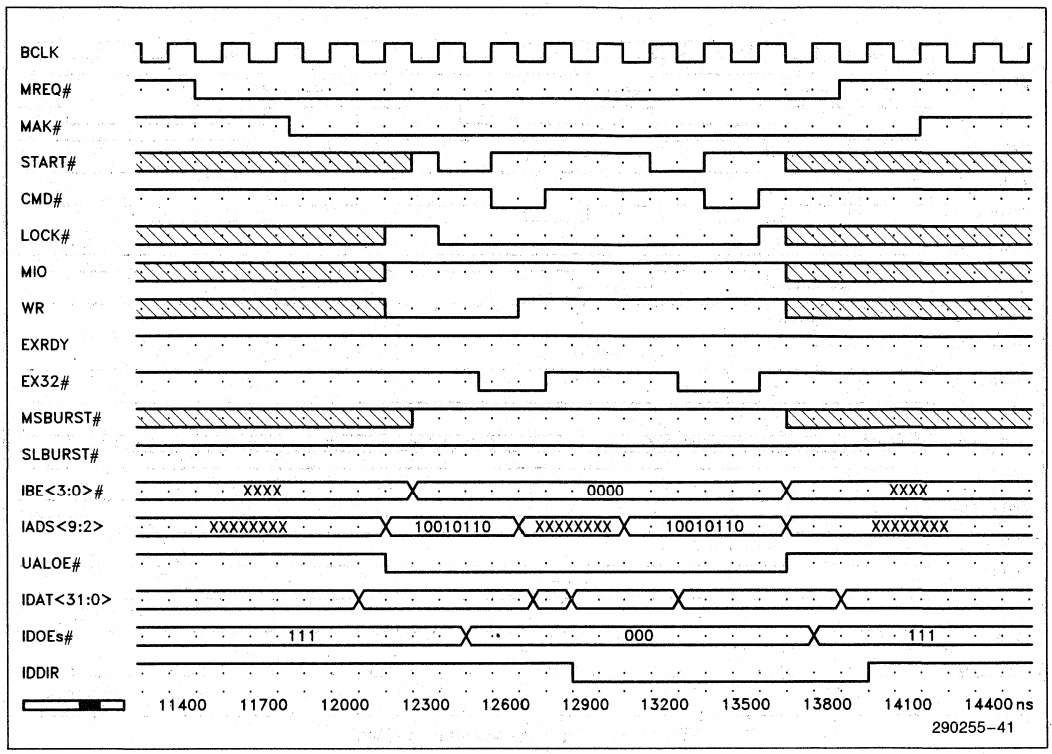


Figure 10-11. Locked Exchange Cycle

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

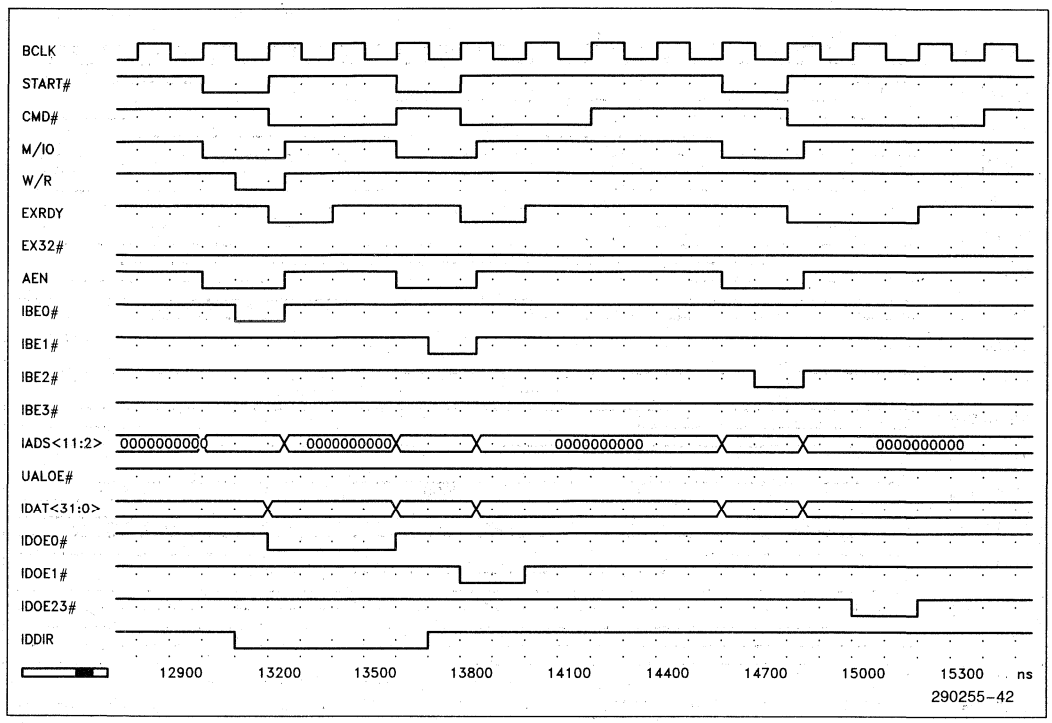


Figure 10-12. Slave Access to BMIC

11.0 D.C. SPECIFICATIONS

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

11.1 Maximum Ratings*

Case Temperature under Bias . . . −65°C to +110°C
Storage Temperature . . . . . −65°C to +150°C
Supply Voltages with
Respect to Ground . . . . . −0.5V to +6.5V
Voltage on Any Pin . . . . . −0.5V to V_{CC} + 0.5V

WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

1

11.2 D.C. Characteristics Table

T_{CASE} = 0°C to 70°C, V_{CC} = 5V ±5%, $T_{AMBIENT}$ = 0°C to 55°C

Symbol	Parameter	Limits		Units	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V	
V_{ILC}	CLOCK Input Low	−0.5	0.8	V	
V_{IHC}	CLOCK Input High	2.0	V_{CC}+0.5	V	
V_{OL1}	Output Low Voltage		0.45	V	I_{OL} = 2.5 mA
V_{OH1}	Output High Voltage	2.4		V	I_{OH} = −2.5 mA
V_{OL2}	Output Low Voltage		0.45	V	I_{OL} = 6 mA
V_{OH2}	Output High Voltage	2.4		V	I_{OH} = −4 mA
V_{OL3}	Output Low Voltage		0.45	V	I_{OL} = 24 mA
V_{OH3}	Output High Voltage	V_{CC}−0.4		V	I_{OH} = −100 μA
ILI	Input Leakage		±10	μA	
ILO	Output Leakage		±10	μA	
C_{IN}	Capacitance Input		10	pF	@ 1 MHz(2)
C_{OUT}	Capacitance Output or I/O		20	pF	@ 1 MHz(2)
C_{CLK}	BCLK or TCLK		20	pF	@ 1 MHz(2)
I_{CC}	V_{CC} Supply Current		190	mA	(3)

NOTES:

1. V_{OL1} = UALOE#, IDDIR, IDOE23#, IDOE1#, IDOE0#, LRDY, LDAT<7:0>, IDAT<31:0>, TEOP#, TDIR, TCHAN, IOSEL0#, IOSEL1#, TRQ#, TLD#, and TDAT<15:0>
   V_{OL2} = MREQ#, EINT, and LINT
   V_{OL3} = IADS<9:2>, START#, M/IO, W/R, EXRDY, MASTER16#, EX32, IBE#<3:0>, MSBURST#, and LOCK#
   V_{OH1} = UALOE#, IDDIR, IDOE23#, IDOE1#, IDOE0#, LRDY, LDAT<7:0>, IDAT<31:0>, TDIR, TCHAN, TRQ#, TLD#, IOSEL0, IOSEL1#, TDAT<15:0>, MREQ#, EINT, and LINT
   V_{OH2} = IADS<9:2>, START#, M/IO, W/R, IBE#<3:0>, MSBURST#, and LOCK#
   V_{OH3} = UALOE#, IDDIR, IDOE23#, IDOE1#, IDOE0#, IADS<9:2>, LRDY, LDAT<7:0>, IDAT<31:0>, TDIR, TCHAN, EINT, IOSEL0#, IOSEL1#, TRQ#, TLD#, TDAT<15:0>, MREQ#, LINT, IADS<9:2>, START#, M/IO, W/R, IBE#<3:0>, MSBURST#, and LOCK#
   The following outputs are open collector: EXRDY, EX32#, MASTER16#, and TEOP#; EINT is an open collector output when programmed for active low operation.
2. Sampled only
3. Tested at V_{CC} = 5.30V and Frequency = BCLK (8.33 MHz) and TCLK (20 MHz)

12.0 A.C. SPECIFICATIONS

12.1 A.C. Characteristics Tables

The A.C. specifications given in the following tables consist of output delays/float times and input setup and hold times.

T_{CASE} = 0°C to 70°C, V_{CC} = 5V ±5%, $T_{AMBIENT}$ = 0°C to 55°C

BCLK Timing

Symbol	Parameter	Min	Max	Units	Notes
t1	Period	120	250	ns	Typical = 125 ns
t2	High Time	50		ns	Measured @ 2.0V
t3	Low Time	50		ns	Measured @ 0.8V
t4	Rise Time		10	ns	(13)
t5	Fall Time		10	ns	(13)

Reset Timing

Symbol	Parameter	Min	Max	Units	Notes
t6	Pulse Width	8 (t1)		ns	

Master Timing

Symbol	Parameter	Min	Max	Units	Notes
	MREQ#				
t7	Delay ACT/Inact		33	ns	From BCLK Falling
	MAK#				
t8	Setup Time	10		ns	To BCLK Falling
t9	Hold Time	25		ns	From BCLK Falling
	IADS<9:2>, M/IO, W/R				
t10	Delay Valid	2	45	ns	From BCLK Falling(17)
	IADS<9:2>, M/IO, W/R		75	ns	From BCLK Rising(18)
t10a	Delay Valid				
t11	Delay Float		40	ns	From BCLK Falling(7)
	BE#<3:0>				
t12	Delay Valid	2	45	ns	From BCLK Falling
t13	Delay Float		40	ns	From BCLK Falling(7, 8)
	START#				
t14	Delay Act/Inact		25	ns	From BCLK Rising
t15	Delay Float		40	ns	From BCLK Falling(7, 8)
	EX32#				
t16	Setup Time	15		ns	To BCLK Rising(9)
t17	Hold Time	50		ns	From BCLK Rising(9)
	EXRDY				
t18	Setup Time	15		ns	To BCLK Falling
t19	Hold Time	2		ns	From BCLK Falling
	IDAT<31:0>				
t20	Delay Valid		27	ns	From BCLK Falling(1)
t21	Delay Float		25	ns	From BCLK Falling(7, 8)
t22	Setup Time	7		ns	To BCLK Rising(2)
t23	Hold Time	6		ns	From BCLK Rising(2)
	IDAT<31:10>				
t24	Delay Valid		45	ns	From BCLK Falling(10)
	LOCK#				
t25	Delay Act/Inact	2	60	ns	From BCLK Rising
t26	Delay Float		40	ns	From BCLK Falling(7)
	IDOE#				
t27	Delay Act/Inact		25	ns	From BCLK Falling
	UALOE#				
t28a	Delay Active		60	ns	From BCLK Rising
t28b	Delay Inactive		35	ns	From BCLK Falling
	IDDIR				
t29	Delay Act/Inact		40	ns	From BCLK Falling

Master Timing (Burst)

Symbol	Parameter	Min	Max	Units	Notes
	MSBURST#				
t30	Delay ACT/INACT		35	ns	From BCLK Falling
t31	Delay Float		40	ns	From BCLK Rising(8)
	START#, IBE#		40	ns	From BCLK Rising(19)
t31a	Delay Float				
	SLBURST#				
t32	Setup Time	15		ns	To BCLK Rising
t33	Hold Time	50		ns	From BCLK Rising
	IDOE#				
t34	Delay Act/Inact		25	ns	From BCLK Rising
	IDAT<31:0>				
t35	Setup Time (Read)	7		ns	To BCLK Rising(2)
t36	Hold Time (Read)	6		ns	From BCLK Rising(2)
t37	Delay Valid		27	ns	From BCLK Rising(1)
t38	Delay Invalid			ns	From BCLK Rising(1)
	MASTER16#				
t39	Delay Act		50	ns	From BCLK Rising
t40	Delay Float		40	ns	From BCLK Rising(7,8)

Slave Timing

Symbol	Parameter	Min	Max	Units	Notes
	IADS<11:12>, M/IO				
t41	Setup Time	120		ns	To CMD# Falling
t42	Hold Time	25		ns	From CMD# Falling
	EX32#				
t43	Delay Act/Float		54	ns	From IADS<11:2>, M/IO
t44	Delay Act/Float		34	ns	From AEN
	AEN				
t45	Setup Time	95		ns	To CMD# Falling
t46	Hold Time	25		ns	From CMD# Falling
	START#				
t47	Pulse Width	110		ns	
	BE#<3:0>, W/R				
t48	Setup Time	80		ns	To CMD# Falling
t49	Hold Time	25		ns	From CMD# Falling
	EXRDY				
t50	Delay Negated		125	ns	From START# Falling(3)
t51	Delay Float		40	ns	From BCLK Falling
	CMD#				
t52	Pulse Width	110		ns	
	IDAT<31:0>				
t53	Setup Time	−35		ns	To CMD# Falling(2)
t54	Hold Time	0		ns	From CMD# Rising(2)
t55	Delay Valid		100	ns	From BCLK Rising(1)
t56	Delay Invalid	0		ns	From CMD# Rising(1)
t57	Delay Float		50	ns	From CMD# Rising
	IDDIR				
t58	Delay Valid		50	ns	From W/R Valid
t59	Delay Invalid	2		ns	From CMD# Rising
	IDOE#				
t60	Delay Act (Read)		25	ns	From CMD# Falling
t61	Delay Inact (Read)		20	ns	From CMD# Rising
t62	Delay Act/Inact (Write)		45	ns	From BCLK Rising
	IOSEL#				
t63	Delay Active		60	ns	From IADS<11:2>
t64	Delay Inactive	5		ns	From CMD# Rising If Latched

Transfer Buffer Interface Timing

Symbol	Parameter	Min	Max	Units	Notes
	TCLK				
t65	Period	50	250	ns	
t66	High Time	18		ns	Measured @ 2.0V
t67	Low Time	20		ns	Measured @ 0.8V
	TRQ#				
t68	Delay Act/Inact		15	ns	From TCLK Rising
	TLD#				
t69	Delay Act/Inact		25	ns	From TCLK Rising
	TEOP#				
t70	Delay Act/Float		25	ns	From TCLK Rising (Output)
	TCHAN, TDIR				
t73	Setup Time	25		ns	To TLD# or TRQ# Active(11)
	TACK#				
t74	Setup Time	15		ns	To TCLK Rising
t75	Hold Time	1		ns	To TCLK Rising
	TDAT<15:0>				
t76	Delay Valid	4	25	ns	From TCLK Rising/TDOE# Falling
t77	Delay Float		25	ns	From TCLK/TDOE# Rising
t78	Setup Time	10		ns	TO TCLK Rising
t79	Hold Time	1		ns	From TCLK Rising
t80	Ratio of TCLK to BCLK	1.1			

Local Processor Interface Timing (Read Cycle)

Symbol	Parameter	Min	Max	Units	Notes
	LADS<1:0>, LCS#				
t81	Setup Time	10		ns	To LRD# Falling
t82	Hold Time	0		ns	From LRD# Rising
	LRD#				
t83	Pulse Width	150		ns	
	LDAT<7:0>				
t84	Delay Valid		130	ns	From LRD# Falling(4)
t85	Max Delay Valid	2.5 (t1) + 120		ns	From LRD# Falling(5)
t86	Delay Float		40	ns	From LRD# Rising
	LRD# (Inact) to LRD# (Act) or LWR# (Act)				
t87	Recovery Time	60		ns	

Local Processor Interface (Write Cycle)

Symbol	Parameter	Min	Max	Units	Notes
	LADS<1:0>, LCS#				
t88	Setup Time	10		ns	To LWR# Falling
t89	Hold Time	0		ns	From LWR# Rising
	LWR#				
t90	Pulse Width	100		ns	(4)
	LDAT<7:0>				
t91	Setup Time	60		ns	To LWR# Rising(4)
t92	Hold Time	10		ns	From LWR# Rising
t93	Data Valid		70	ns	From LWR# Falling(5)
	LWR# (Inact) to LWR# (Act) or LRD# (Act)				
t94	Recovery Time	60		ns	

Local Processor Ready Timing

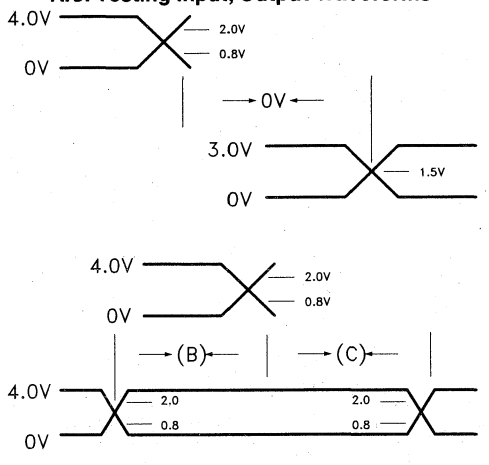
Symbol	Parameter	Min	Max	Units	Notes
	LRDY				
t95	Delay Inactive		50	ns	From LADS and LCS# Valid(5)
	Delay Active				From LRD# or LWR# Active
t96	Max Delay	3.5 (t1) + 60		ns	(5, 6)
t97	Max Delay	2.5 (t1) + 60		ns	(5, 6)
t98	Min Delay	1.5 (t1)		ns	(5, 6)
	LDAT<7:0>				
t99	Delay Valid		0	ns	From LRDY Rising(5, 12)

NOTES FOR A.C TIMINGS:

1. Specification does not include allowance for 13 ns max. and 2 ns min. into 240 pF for external buffer delay to EISA bus.
2. Specification does not include allowance for 8 ns max. and 1 ns min. into 25 pF for external input delay from EISA bus.
3. Delay includes 40 ns for pull-up rise time (300Ω into 240 pF, 2V rise).
4. Applies to all non-shared registers excluding the Peek/Poke Data registers. LRDY will remain active.
5. Applies to the Peek/Poke Data and Shared Registers. LRDY will be taken inactive as soon as LA<1:0> and LCS# are valid, and remain inactive until valid data is available, or has been written.
6. The maximum LRDY delay, 3.5 (t1) + 60 ns from LRD# or LWR#, only occurs if the local processor access loses the internal register access arbitration to an EISA access and if the following BCLK cycle is stretched. Without BCLK stretching, the maximum delay is 2.5 (t1) +60 ns. The minimum LRDY delay is 1.5 (t1). **NOTE:** The maximum BCLK stretch that will be seen by the BMIC is one BCLK period; this is assuming that the bus controller is the 82358 (EBC). If the 82358 is not used as the bus controller, the LRDY and data delay max. specs (t96/t85) will not necessarily be valid.
7. Exiting master mode, the address lines <31:2>, M-IO, LOCK# START#, IBE#<3:0>, MSBURST#, IDAT<31:0>, and W/R will float no later than the falling edge of BCLK after CMD# is deasserted.
8. During a mismatched cycle START#, IBE#<3:0>, and IDAT<31:0> will float from the first falling edge of BCLK after START# is negated.
9. Includes mismatched cycles.
10. Refers to the upper 22 EISA address lines which are multiplexed into the upper 22 data lines IDAT<31:10>. The address will be available for latching into the external address latches 45 ns from the falling edge of BCLK.
11. The TDIR and TCHAN signals are referenced to the falling edge of TRQ# during the cycles that TLD# is not requested.
12. LRDY going active will always be delayed from data valid. The maximum delay seen will be no greater than one (t1) period.
13. Characterized, not tested.
14. Under non-preempt, MREQ# will deassert a minimum of 0.5 BCLKs after the negating edge of the last CMD# of the transfer, depending on the cycle type (refer to the Basic Function Timings, Section 10.0).
15. During an EISA read transfer, the BMIC will assert TEOP# typically eight TCLKs (max 20 TCLKs) after CMD# is deasserted from the last EISA cycle, indicating end of transfer (refer to the Basic Function Timings, Section 10.0).
16. During an EISA write transfer, the BMIC will assert TEOP# two TCLKs after CMD# is deasserted, indicating end of transfer (refer to the Basic Function Timings, Section 10.0).
17. For address changes while CMD# is active.
18. During an upper address load cycle, at the beginning of a transfer sequence, CMD# is inactive.
19. For "Downshifting Cases" where the transfer is misaligned.

12.2 A.C. Characteristics Waveforms

A.C. Testing Input, Output Waveforms



290255-12

NOTE:
The input waveforms have tr < 2.0 ns from 0.8V to 2.0V

A. Output delay specification referenced from one of the following signals: BCLK, TCLK, CMD#, START#, AEN, IADS<11:2>, W/R, TDOE#, LRD#, LWR#, LADS<1:0>, LCS#, LRD#, or LWR#.

B. Minimum input setup specification referenced to one of the following signals: BCLK, TCLK, CMD#, LWR#, LRD#, TLD#, or TRQ#.

C. Minimum input hold specification referenced to one of the following signals: BCLK, TCLK, CMD#, LWR#, LRD#, TLD#, or TRQ#.

A.C. Testing: All inputs are driven at 4V for a logic "1" and 0V for a logic "0". A.C. Timings are measured from the 0.8V and 2.0V levels on the source signal to either the 0.8V and 2V or 1.5V level on the signal under test; except as noted by the following:

1. BCLK and TCLK high time measurements are made at 2.0V
2. BCLK and TCLK low time measurements are made at 0.8V
3. START#, CMD#, LRD#, and LWR# pulse width measurements are made at 0.8V

A.C. TEST LOADS

CL = 25 pF on IDAT<31:0>, IDOE#, IOSEL#<1:0>, TRQ#, TLD#, TEOP#, TDAT<15:0>, TCHAN, TDIR, LRDY, and LDAT<7:0>

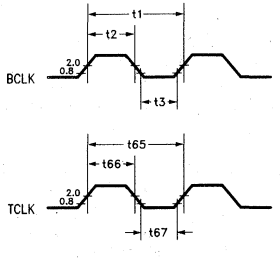
CL = 35 pF on IDDIR

CL = 50 pF on UALOE# and LINT

CL = 120 pF on MREQ# and EINT

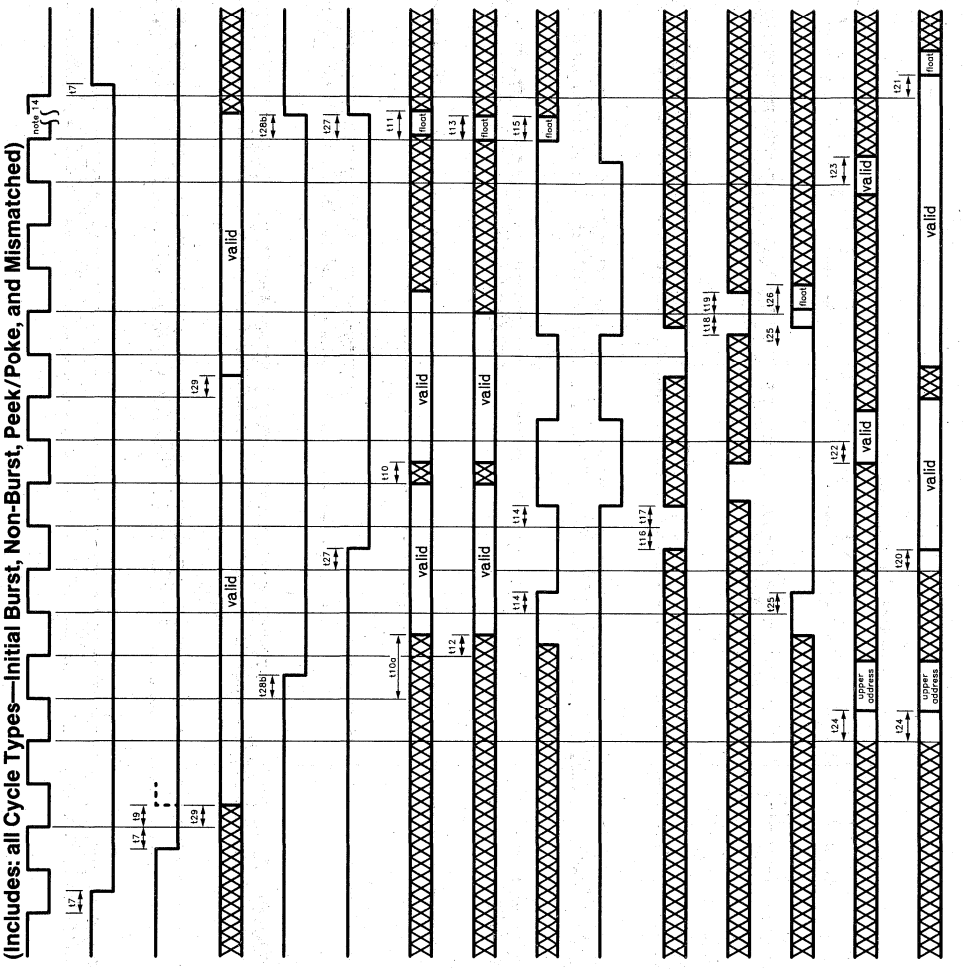
CL = 240 pF on IADS<9:2>, BE#<3:0>, W/R, START#, EX32#, LOCK, MSBURST#, MASTER16#, EXRDY, and M/IO

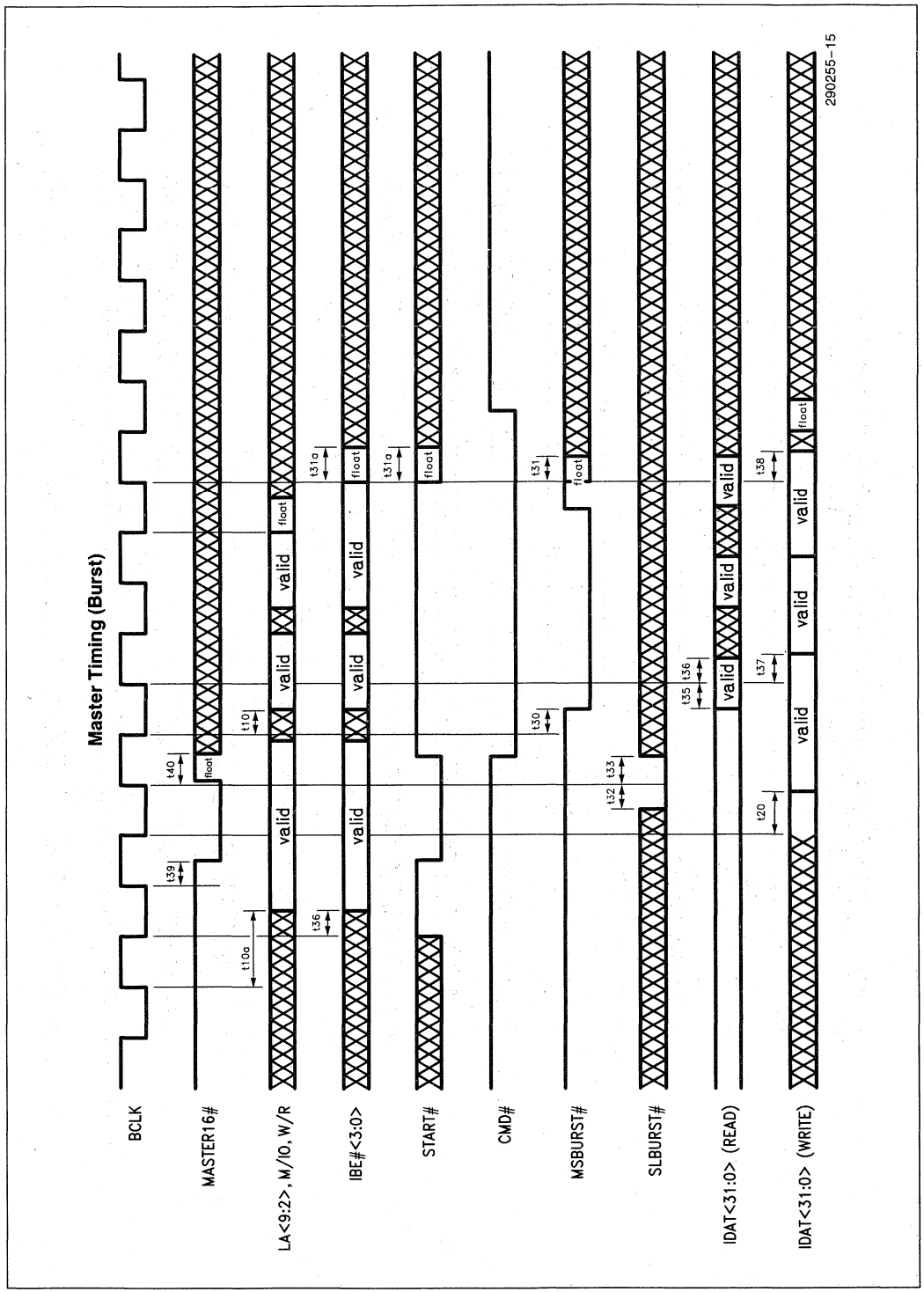
BCLK, TCLK Timing

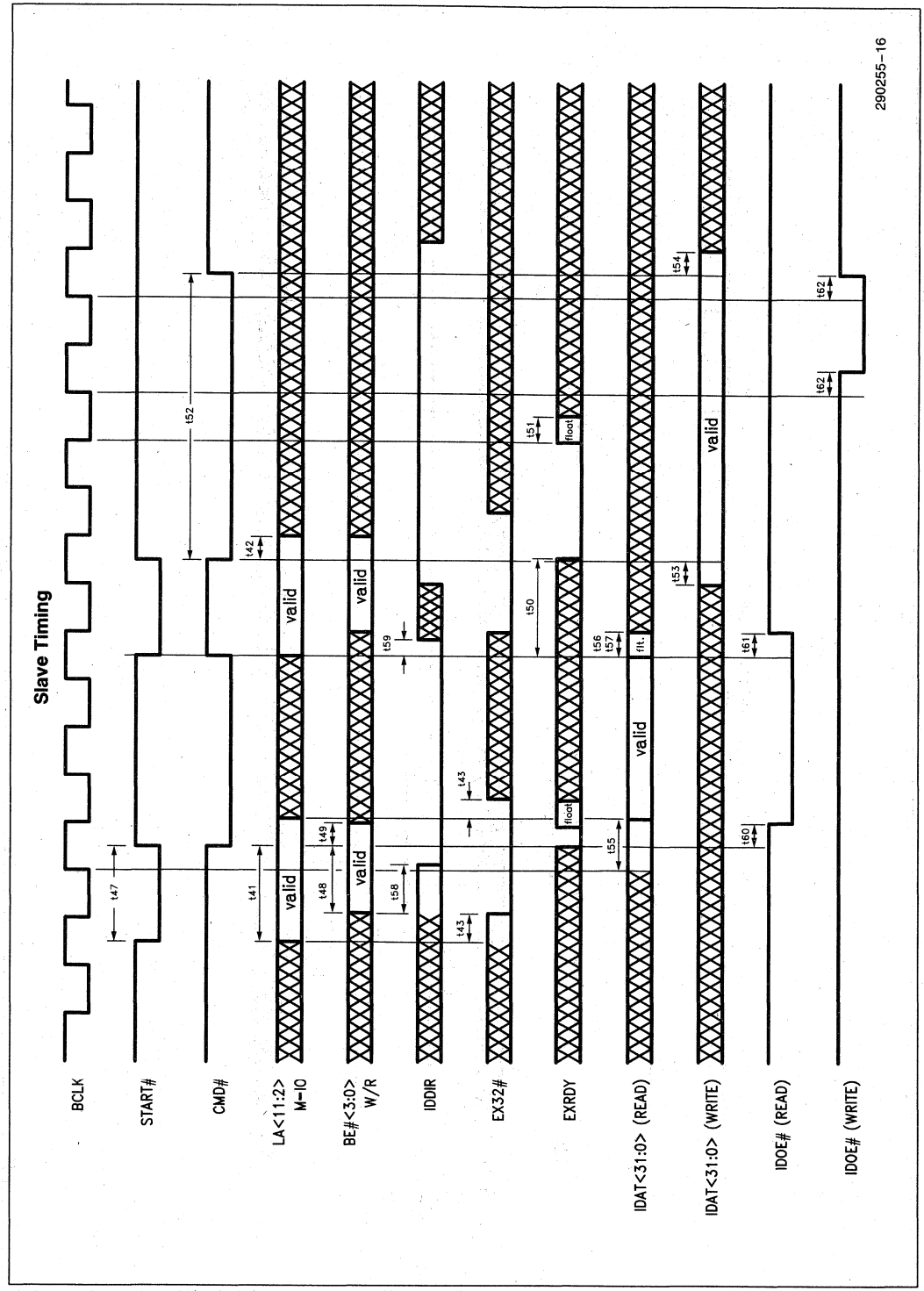


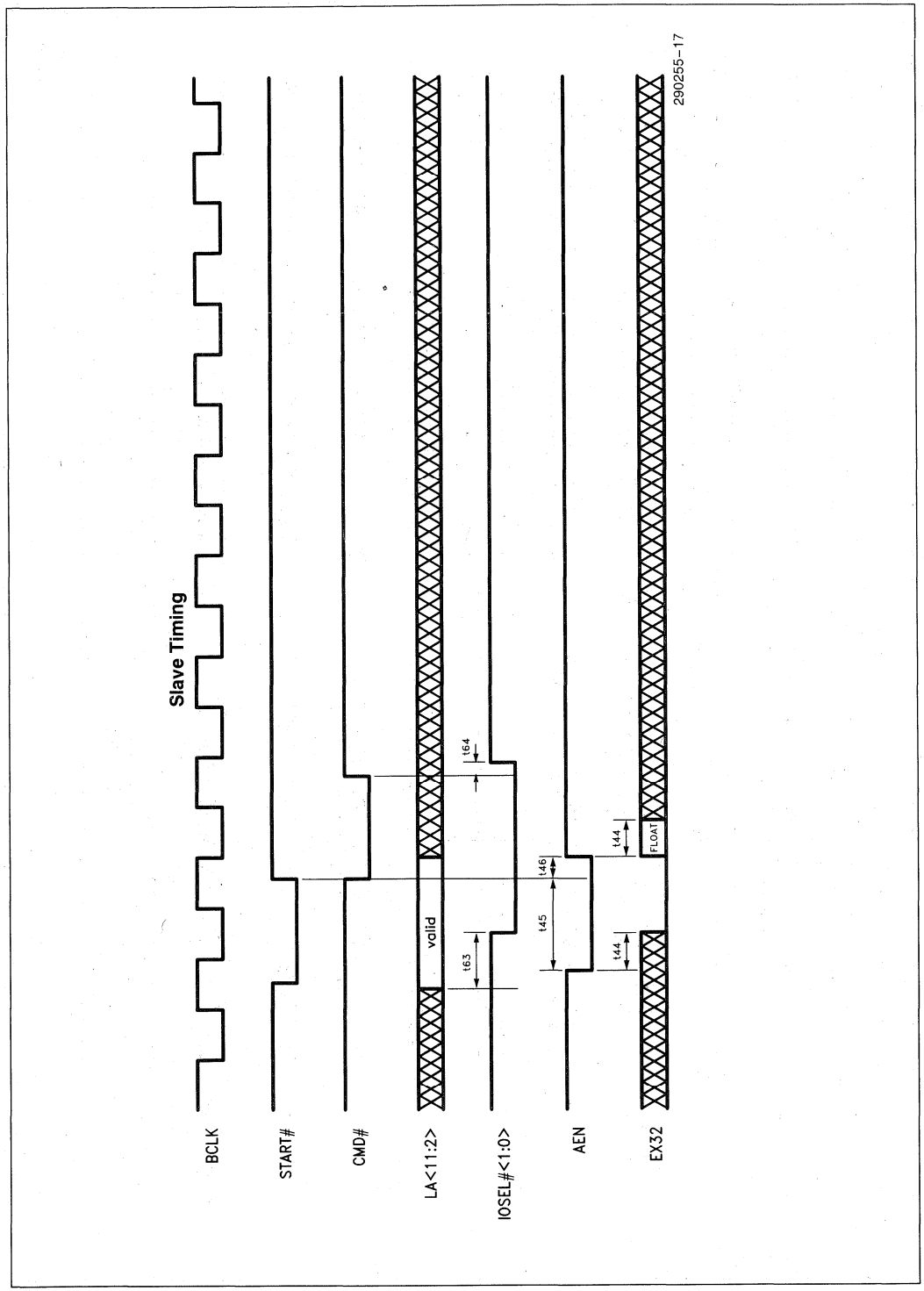
290255-13

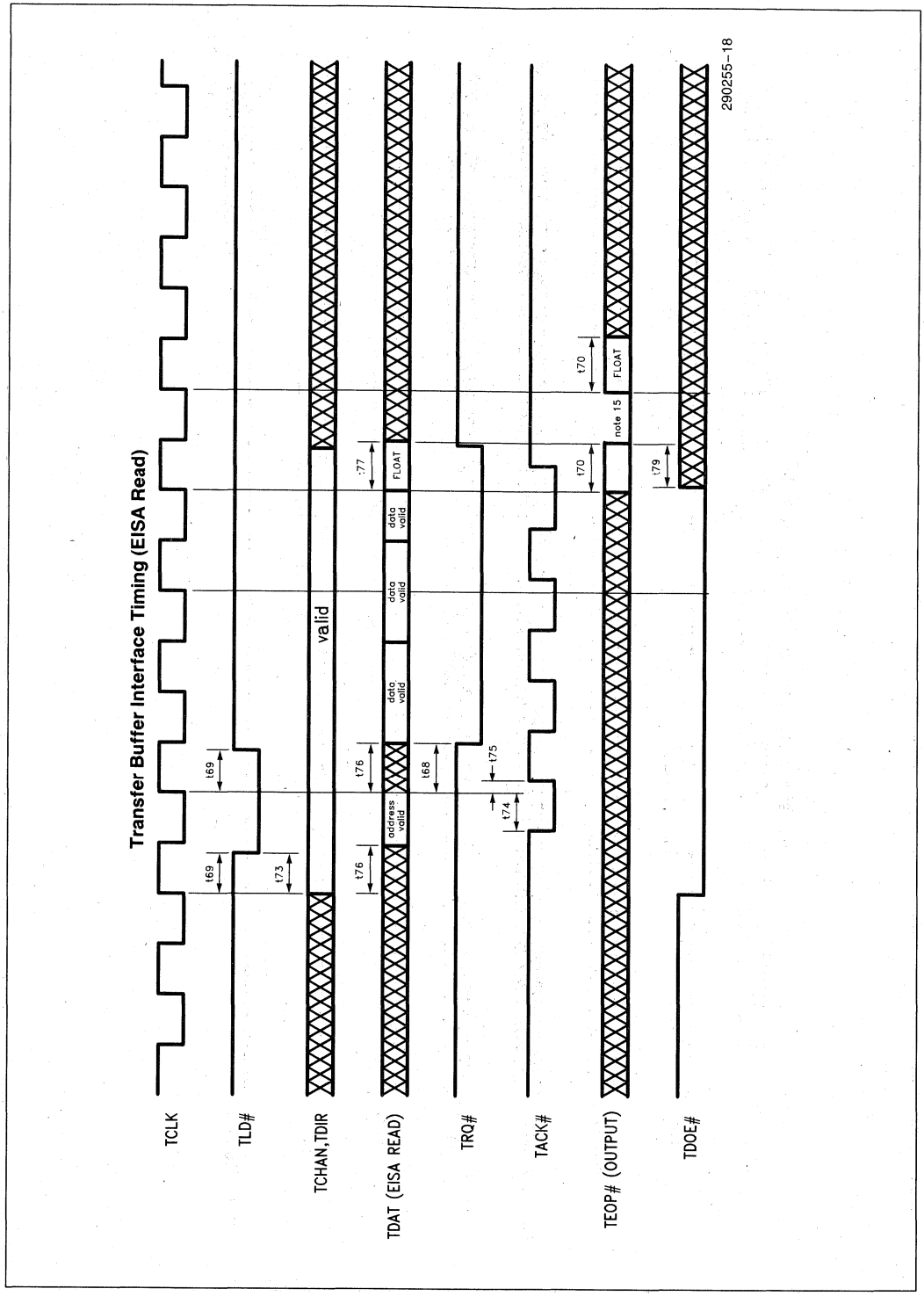
Master Timing
(Includes: all Cycle Types—Initial Burst, Non-Burst, Peek/Poke, and Mismatched)

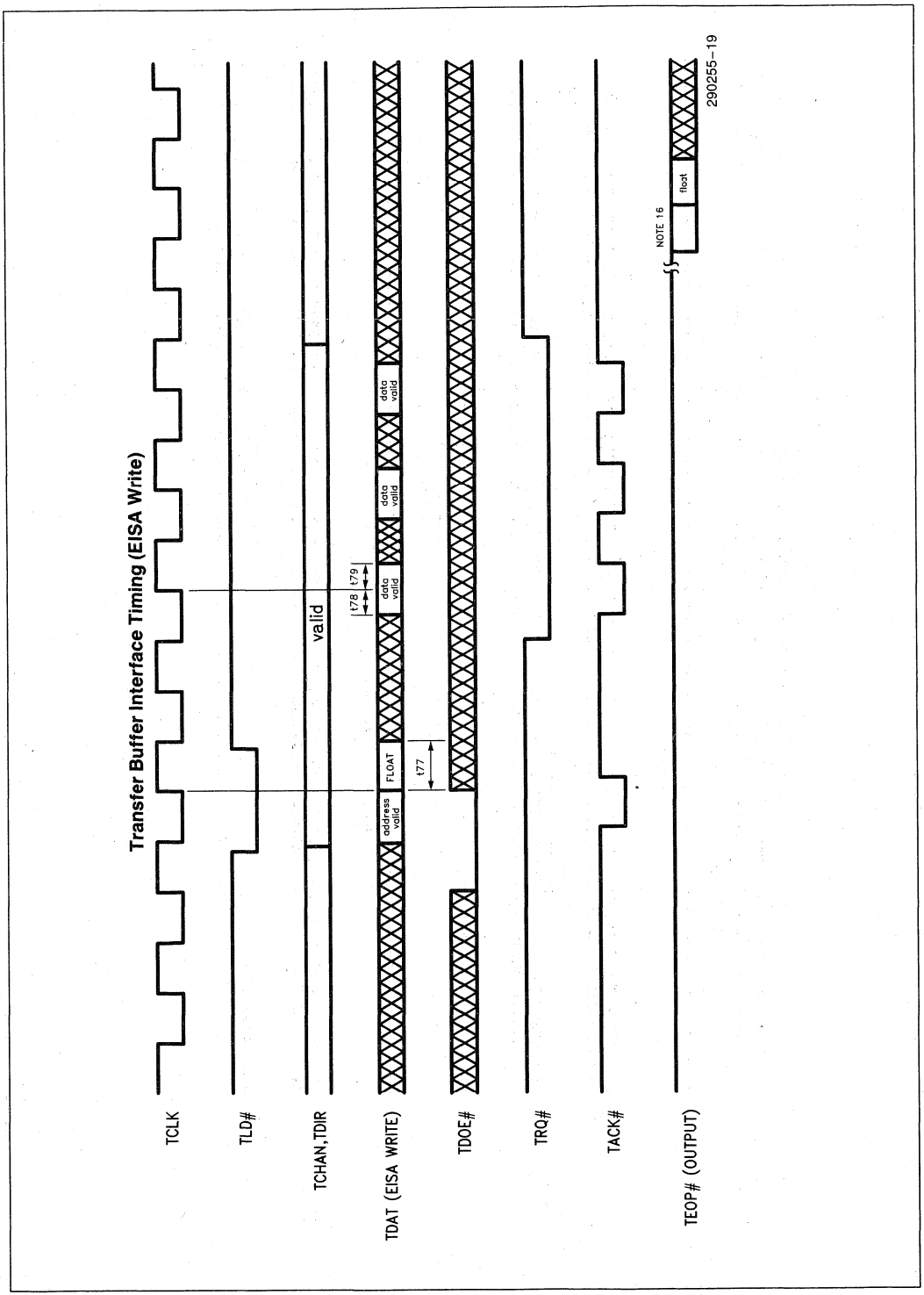


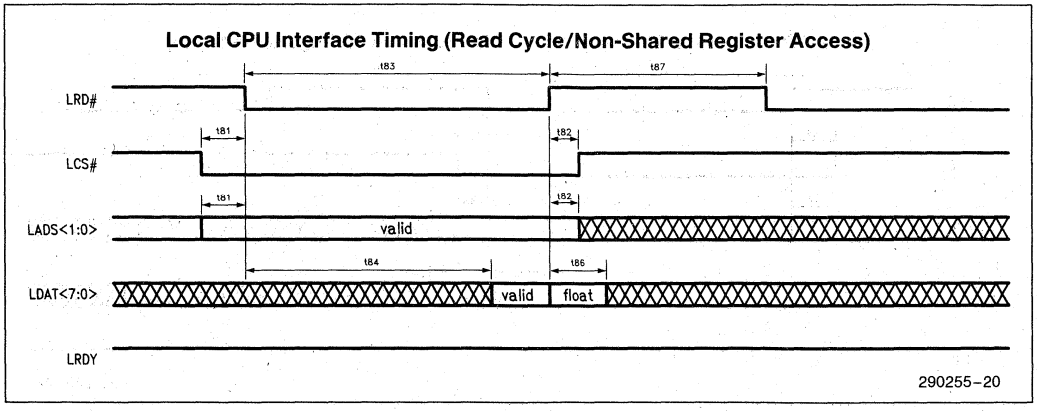

Master Timing (Burst)
BCLK
MASTER16#
LA<9:2>, M/IO, W/R
IBE#<3:0>
START#
CMD#
MSBURST#
SLBURST#
IDAT<31:0> (READ)
IDAT<31:0> (WRITE)
valid
float
t10a
t10
t39
t40
t36
t31a
t31
t30
t32
t33
t35
t37
t20
t38
290255-15

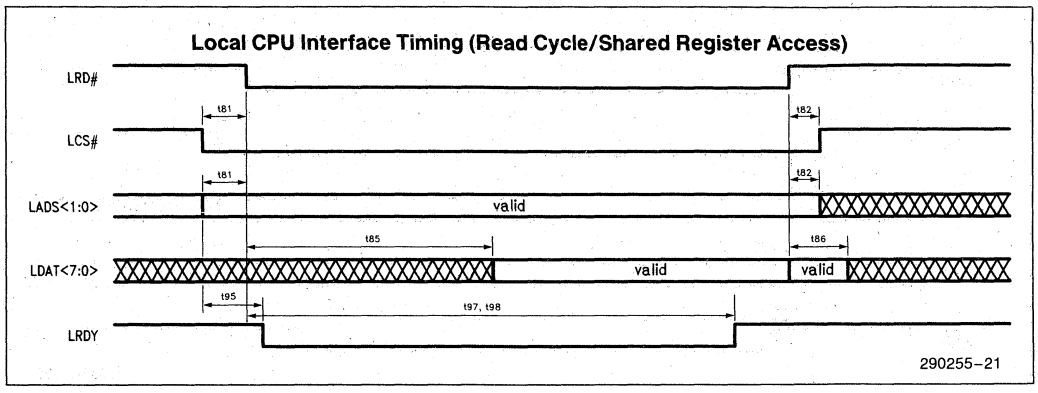

Slave Timing
BCLK
START#
CMD#
LA<11:2>
M-IO
BE#<3:0>
W/R
IODIR
EX32#
EXRDY
IDAT<31:0> (READ)
IDAT<31:0> (WRITE)
IDOE# (READ)
IDOE# (WRITE)
valid
float
filt.
t41
t42
t43
t45
t47
t48
t49
t50
t51
t52
t53
t54
t55
t56
t57
t58
t59
t60
t61
t62
290255-16


Slave Timing
BCLK
START#
CMD#
LA<11:2>
IOSEL#<1:0>
AEN
EX32
valid
FLOAT
t63
t64
t45
t46
t44
t44
290255-17

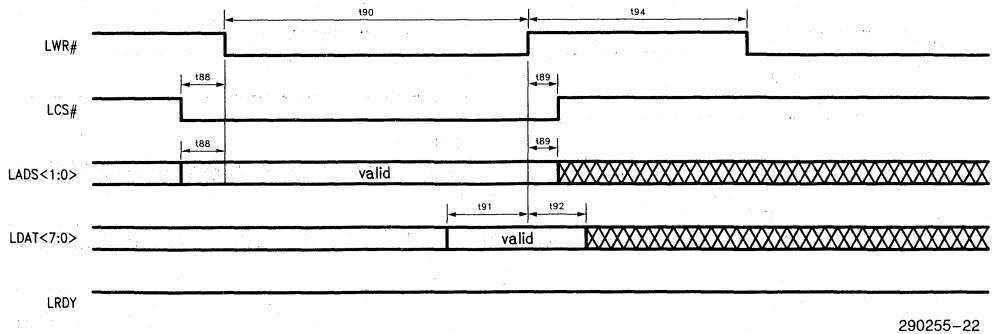

Transfer Buffer Interface Timing (EISA Read)
TCLK
TLD#
TCHAN,TDIR
TDAT (EISA READ)
TRQ#
TACK#
TEOP# (OUTPUT)
TDOE#
valid
address valid
data valid
FLOAT
note 15
t68
t69
t70
t73
t74
t75
t76
t77
t79
290255-18


Transfer Buffer Interface Timing (EISA Write)
TCLK
TLD#
TCHAN,TDIR
valid
TDAT (EISA WRITE)
address valid
FLOAT
data valid
data valid
data valid
t77
t78
t79
TDOE#
TRQ#
TACK#
TEOP# (OUTPUT)
NOTE 16
float
290255-19


Local CPU Interface Timing (Read Cycle/Non-Shared Register Access)
LRD#
LCS#
LADS<1:0>
LDAT<7:0>
LRDY
t81
t82
t83
t84
t86
t87
valid
valid
float
290255-20

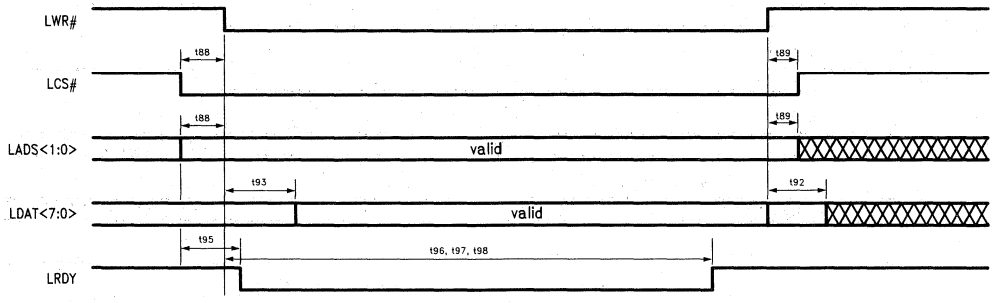

Local CPU Interface Timing (Read Cycle/Shared Register Access)
LRD#
LCS#
LADS<1:0>
LDAT<7:0>
LRDY
t81
t82
t85
t86
t95
t97, t98
valid
valid
valid
290255-21

Local CPU Interface Timing (Write Cycle/Non-Shared Register Access)



290255-22

Local CPU Interface Timing (Write Cycle/Shared Register Access)



290255-23

13.0 BMIC PN AND PACKAGE INFORMATION

13.1 Signal Overview

Name = Pin Name, Type = I—Input, O—Output, OC—Open Collector, B—Both Input and Output, BC—Both and Open Collector, Pin = Pin Location

Name	Type	Pin	Description
EISA BUS INTERFACE SIGNALS			
START#	B	84	EISA Start of Cycle
CMD#	I	102	EISA Command Strobe
M/IO	B	81	EISA Memory/IO Cycle Status Signal
W/R	B	80	EISA Write/Read Status Signal
EXRDY	I, OC	103	EISA Ready Signal
EX32#	I, OC	102	EISA 32-Bit Slave Response Signal
MASTER16#	OC	82	EISA 16-Bit Master Control Signal
IBE#<3:0>	B	64, 61, 60 59	EISA Byte Enable Lines
AEN	I	107	EISA Address Enable Signal
MSBURST#	O	96	EISA Master Burst Signal
SLBURST#	I	97	EISA Slave Burst Signal
LOCK#	O	98	EISA Resource Lock Signal
MREQ#	O	99	EISA Bus Master Request Signal
MAK#	I	100	EISA Master Bus Acknowledge Signal
EINT	BC	109	EISA Interrupt Request Signal
BCLK	I	101	EISA Bus Clock
RESET	I	125	EISA Reset Signal
IDAT<31:0>	B	Section	EISA Data Lines
IADS<11:10>	I	105, 106	EISA Address Input Lines
IADS<9:2>	B	57–55, 53, 44, 40–38	EISA Lower Address Lines
EISA BUFFER CONTROL SIGNALS			
UALOE#	O	78	EISA Upper Address Latch and Output Enable
IDDIR	O	79	EISA Data Buffer Direction Signal
IDOE23#	O	75	EISA Data Byte Line Buffer Enable (Bytes 3, 2)
IDOE#<1:0>	O	76, 77	EISA Data Byte Line Buffer Enables (Bytes 1, 0)
TRANSFER BUFFER INTERFACE SIGNALS			
TCLK	I	32	Transfer Clock
TRQ#	O	7	Transfer Data Request Signal
TACK#	I	6	Transfer Data Acknowledge Signal
TDIR	O	3	Transfer Data Direction Signal
TCHAN	O	4	Transfer Data Channel Select Signal
TLD#	O	5	Transfer Address Counter Load Signal
TDOE#	I	2	Transfer Data Bus Output Enable
TEOP#	I, OC	1	Transfer End-of-Process
TDAT<15:0>	B	Section	Transfer Data Bus Lines

13.1 Signal Overview (Continued)

Name = Pin Name, Type = I—Input, O—Output, OC—Open Collector, B—Both Input and Output, BC—Both and Open Collector, Pin = Pin Location

Name	Type	Pin	Description
LOCAL PROCESSOR INTERFACE SIGNALS			
LRD#	I	130	Local Read Signal
LWR#	I	129	Local Write Signal
LCS#	I	128	Local Chip Select Signal
LDAT<7:0>	B	121–118 115–112	Local Data Bus Lines
LADS<1:0>	B	127, 126	Local Address Register Select Signals
LRDY#	B	122	Local Ready Signal
LINT#	O	123	Local Processor Interrupt Signal
MISCELLANEOUS SIGNALS			
IOSEL#<1:0>	O	111, 110	Expansion Board Address Range Decode Signals
POWER PINS			
V_{CC}		108, 124	Power Pins for the Internal Logic
V_{SS}		42, 58	Ground Pins for the Internal Logic
V_{CCB}		12, 23, 41, 63, 74, 83, 94, 117, 132	Power Pins for the Output Buffers
V_{SSB}		13, 22, 33, 43, 54, 62, 73, 85, 95, 116, 131	Ground Pins for the Output Buffers

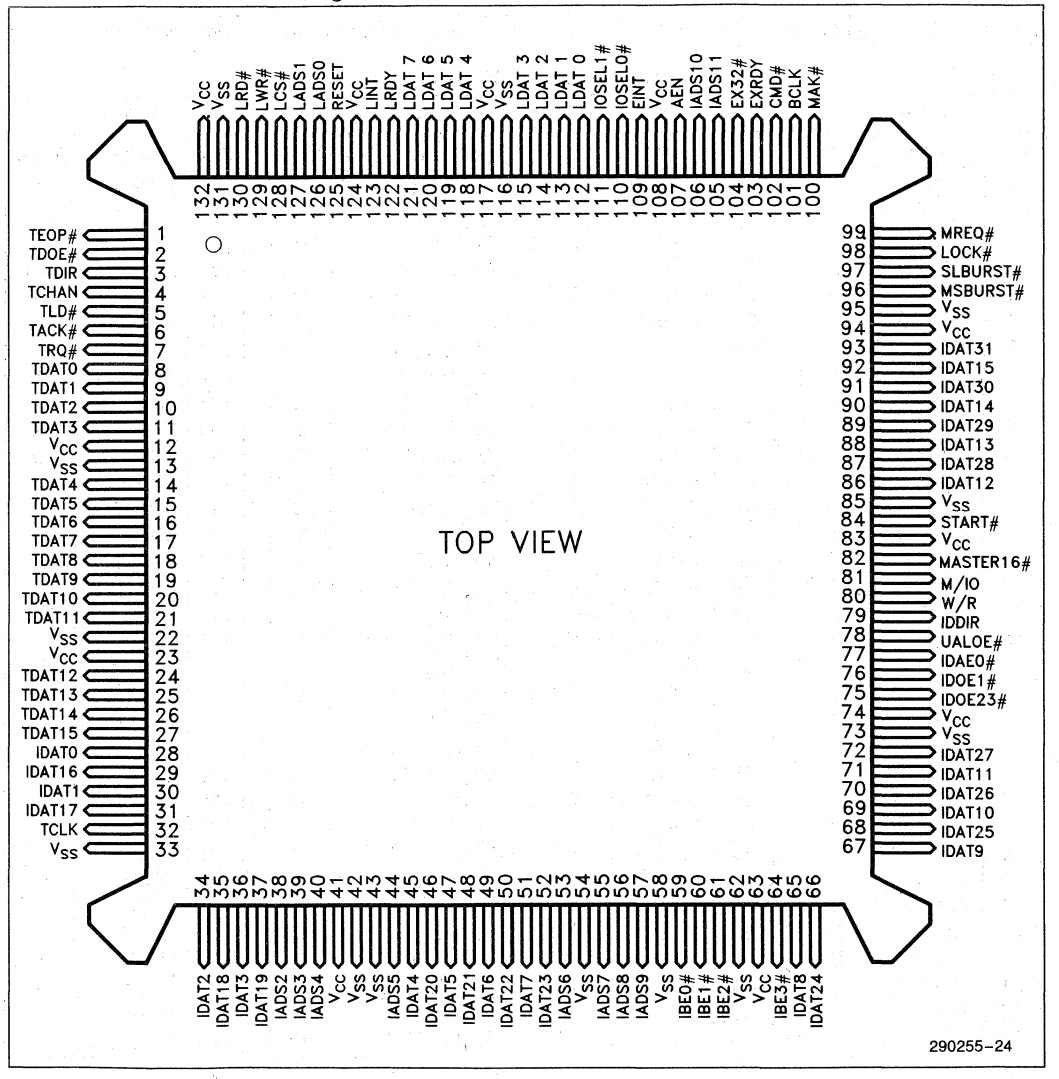
13.2 Device Pinout

I = Input, O = Output, OC = Open Collector, B = Both Input and Output, BC = Both and Open Collector

Device Pinout—132 Lead PQFP

A Row			B Row			C Row			D Row		
Pin	Label	Type	Pin	Label	Type	Pin	Label	Type	Pin	Label	Type
1	TEOP#	I, OC	34	IDAT2	B	67	IDAT9	B	100	MAK#	I
2	TDOE#	I	35	IDAT18	B	68	IDAT25	B	101	BCLK	I
3	TDIR	O	36	IDAT3	B	69	IDAT10	B	102	CMD#	I
4	TCHAN	O	37	IDAT19	B	70	IDAT26	B	103	EXRDY	I, OC
5	TLD#	O	38	IADS2	B	71	IDAT11	B	104	EX32#	I, OC
6	TACK#	I	39	IADS3	B	72	IDAT27	B	105	IADS11	I
7	TRQ#	O	40	IADS4	B	73	V_{SSB}		106	IADS10	I
8	TDAT0	B	41	V_{CCB}		74	V_{CCB}		107	AEN	I
9	TDAT1	B	42	V_{SS}		75	IDOE23#	O	108	V_{CC}	
10	TDAT2	B	43	V_{SSB}		76	IDOE1#	O	109	EINT	BC
11	TDAT3	B	44	IADS5	B	77	IDOE0#	O	110	IOSEL0#	O
12	V_{CCB}		45	IDAT4	B	78	UALOE#	O	111	IOSEL1#	O
13	V_{SSB}		46	IDAT20	B	79	IDDIR	O	112	LDAT0	B
14	TDAT4	B	47	IDAT5	B	80	W/R	B	113	LDAT1	B
15	TDAT5	B	48	IDAT21	B	81	M/IO	B	114	LDAT2	B
16	TDAT6	B	49	IDAT6	B	82	MASTER16#	OC	115	LDAT3	B
17	TDAT7	B	50	IDAT22	B	83	V_{CCB}		116	V_{SSB}	
18	TDAT8	B	51	IDAT7	B	84	START#	B	117	V_{CCB}	
19	TDAT9	B	52	IDAT23	B	85	V_{SSB}		118	LDAT4	B
20	TDAT10	B	53	IADS6	B	86	IDAT12	B	119	LDAT5	B
21	TDAT11	B	54	V_{SSB}		87	IDAT28	B	120	LDAT6	B
22	V_{SSB}		55	IADS7	B	88	IDAT13	B	121	LDAT7	B
23	V_{CCB}		56	IADS8	B	89	IDAT29	B	122	LRDY	B
24	TDAT12	B	57	IADS9	B	90	IDAT14	B	123	LINT	O
25	TDAT13	B	58	V_{SS}		91	IDAT30	B	124	V_{CC}	
26	TDAT14	B	59	IBE0#	B	92	IDAT15	B	125	RESET	I
27	TDAT15	B	60	IBE1#	B	93	IDAT31	B	126	LADS0	I
28	IDAT0	B	61	IBE2#	B	94	V_{CCB}		127	LADS1	I
29	IDAT16	B	62	V_{SSB}		95	V_{SSB}		128	LCS#	I
30	IDAT1	B	63	V_{CCB}		96	MSBURST#	O	129	LWR#	I
31	IDAT17	B	64	IBE3#	B	97	SLBURST#	I	130	LRD#	I
32	TCLK	I	65	IDAT8	B	98	LOCK#	O	131	V_{SSB}	
33	V_{SSB}		66	IDAT24	B	99	MREQ#	O	132	V_{CCB}	

13.3 132-Pin PQFP Package Pinout



290255-24

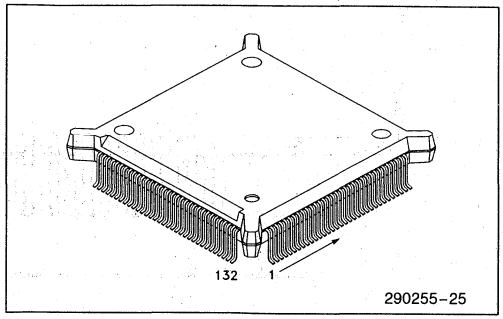
PACKAGING INFORMATION

(See Packaging Specification Order # 231369)

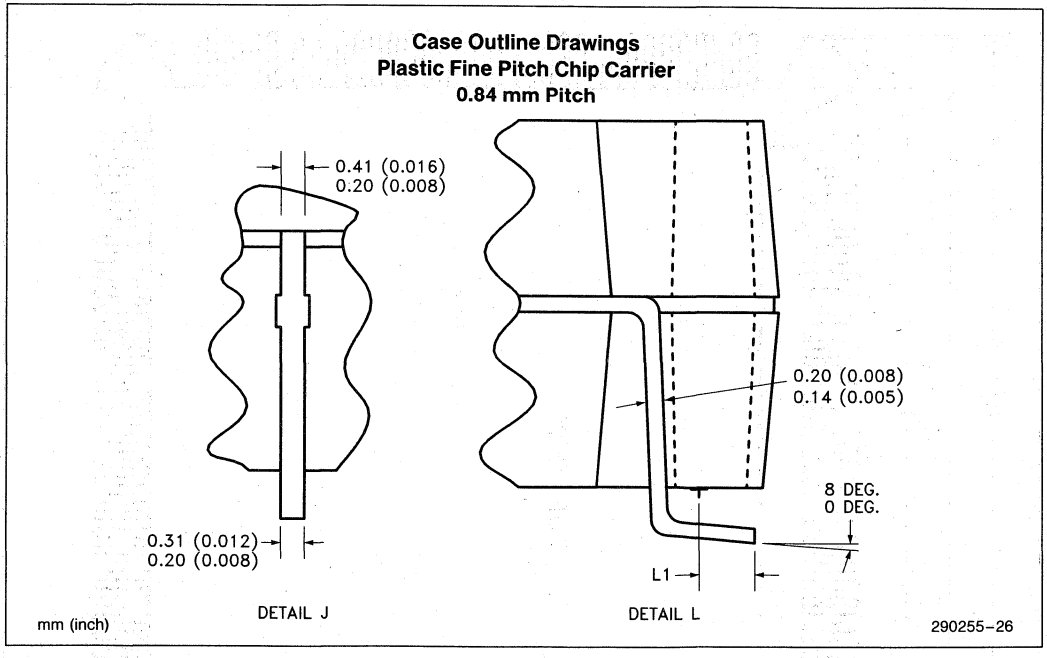
Introduction

The individual components of Intel's EISA Chip Set come in JEDEC standard Gull Wing packages (25 MIL pitch), with "bumpers" on the corners for ease of handling. Please refer to the accompanying table for the package associated with each device, and to the individual component specifications for pinouts. (Note that the individual pinouts are numbered consistently with the numbering scheme depicted in the accompanying figures.)

PLASTIC QUAD FLAT PACK (PQFP)

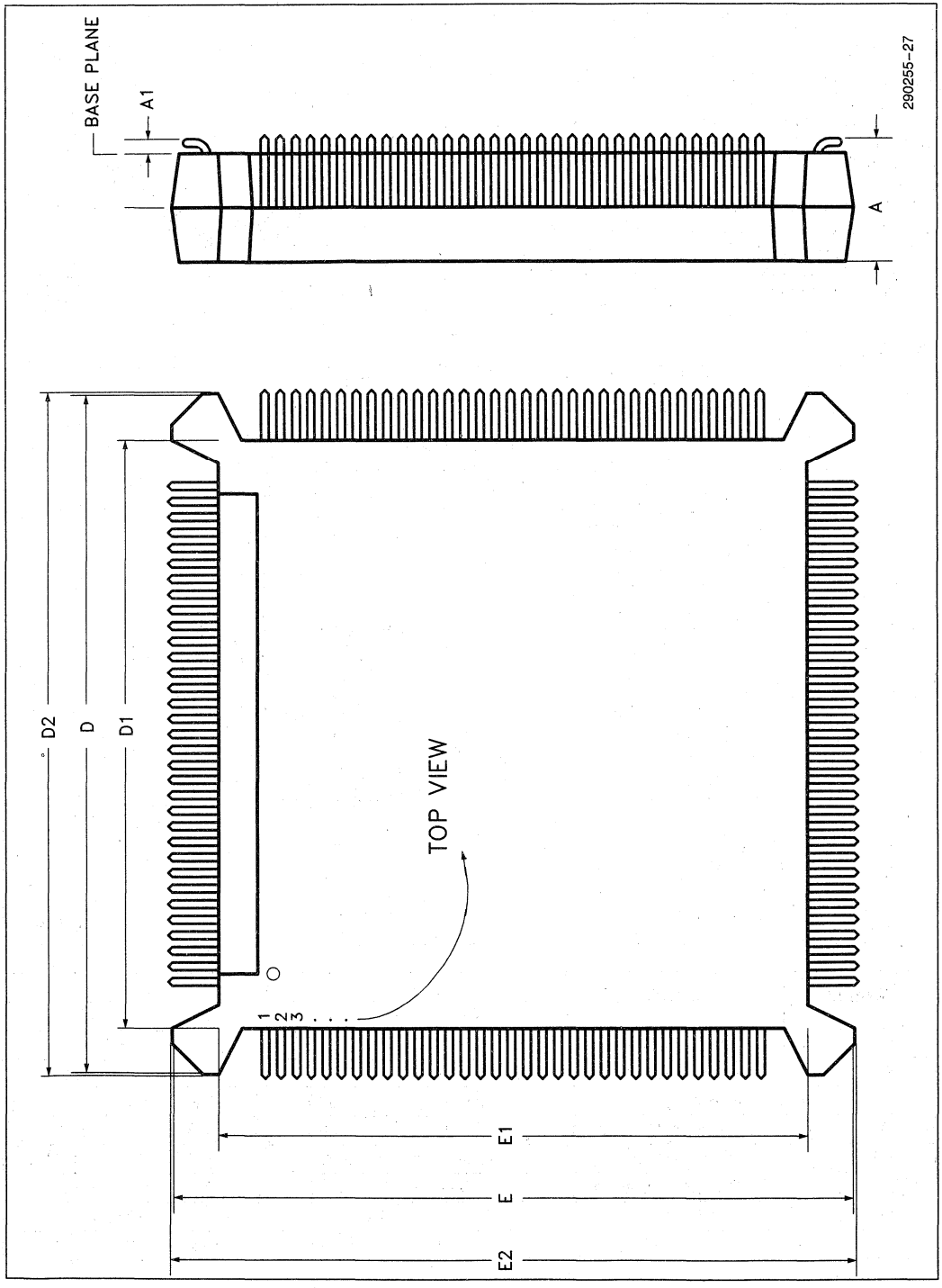


TYPICAL LEAD

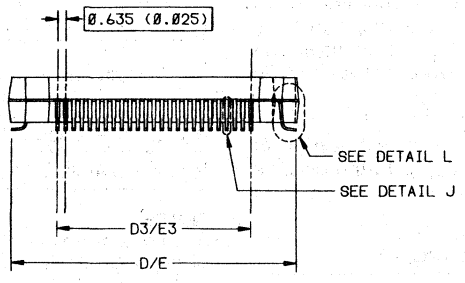


Symbol	Description	Inch		mm	
		Min	Max	Min	Max
N	Lead Count	132		132	
A	Package Height	0.160	0.170	4.06	4.32
A1	Standoff	0.020	0.030	0.51	0.76
D, E	Terminal Dimension	1.075	1.085	27.31	27.56
D1, E1	Package Body	0.947	0.953	24.05	24.21
D2, E2	Bumper Distance	1.097	1.103	27.86	28.02
D3, E3	Lead Dimension	0.800 Ref		20.32 Ref	
L1	Foot Length	0.020	0.030	0.51	0.76

13.4 PRINCIPAL DIMENSIONS & DATUMS



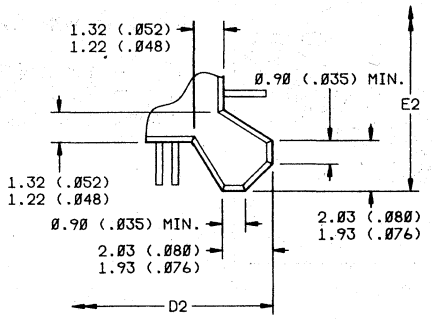
TERMINAL DETAILS



mm (inch)

290255-28

BUMPER DETAIL



mm (inch)

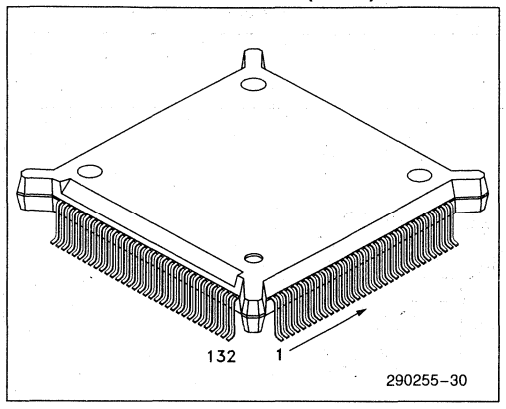
290255-29

13.5 Package Thermal Specification

The 82355 (BMIC) is specified for operation when the case temperature is within the range of 0°C–85°C. The case temperature may be measured in any environment, to determine whether the device is within the specified operating range.

The PQFP case temperature should be measured at the center of the top surface opposite the pins, as shown in the figure below.

PLASTIC QUAD FLAT PACK (PQFP)



82355 PQFP Package Thermal Characteristics

Thermal Resistance—°C/W							
Parameter	Air Flow Rate (ft/min)						
	0	50	100	200	400	600	800
θ Junction—Case	7	7	7	7	7	7	7
θ Case to Ambient	22	21	19.5	17.5	14.5	12	10

NOTES:
1. Table 2 applies to 82358 PQFP plugged into socket or soldered directly into board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$.

Process Name:

1.2μ CHMOS III P-well

I_{CC} at Hot with no Resistive Loads:

150 mA max at 85°C
Measure PQFP case temperature
at center of top surface

14.0 BMIC REGISTER ADDRESS MAP

14.1 Index Register Set

The following registers are mapped directly into the local processor interface:

Local Address	Type	Register Description
0	R/W	Local Data Register
1	R/W	Local Index Register
2	R/W	Local Status/Control Register
3	—	Reserved

14.2 Shared Register Set

EISA Address	Type	Index Address	Type	Register Description
XC80	R	00	R/W	ID Byte 0
XC81	R	01	R/W	ID Byte 1
XC82	R	02	R/W	ID Byte 2
XC83	R	03	R/W	ID Byte 3
XC84	—	04	—	Non BMIC Register (For Expansion Board Use)
XC85	—	05	—	Non BMIC Register (For Expansion Board Use)
XC86	—	06	—	Non BMIC Register (For Expansion Board Use)
XC87	—	07	—	Non BMIC Register (For Expansion Board Use)
XC88	R	08	R/W	Global Configuration Register
XC89	R/W	09	R	System Interrupt Enable/Control Register
XC8A	R/W	0A	R/W	Semaphore Port 0
XC8B	R/W	0B	R/W	Semaphore Port 1
XC8C	R	0C	R/W	Local Doorbell Enable Register
XC8D	R/W	0D	R/W	Local Doorbell Interrupt/Status Register
XC8E	R/W	0E	R	EISA System Doorbell Enable Register
XC8F	R/W	0F	R/W	EISA System Doorbell Interrupt/Status Register
XC90	R/W	10	R/W	Mailbox Register (1)
XC91	R/W	11	R/W	Mailbox Register (2)
XC92	R/W	12	R/W	Mailbox Register (3)
XC93	R/W	13	R/W	Mailbox Register (4)
XC94	R/W	14	R/W	Mailbox Register (5)
XC95	R/W	15	R/W	Mailbox Register (6)
XC96	R/W	16	R/W	Mailbox Register (7)
XC97	R/W	17	R/W	Mailbox Register (8)
XC98	R/W	18	R/W	Mailbox Register (9)
XC99	R/W	19	R/W	Mailbox Register (10)
XC9A	R/W	1A	R/W	Mailbox Register (11)
XC9B	R/W	1B	R/W	Mailbox Register (12)
XC9C	R/W	1C	R/W	Mailbox Register (13)
XC9D	R/W	1D	R/W	Mailbox Register (14)
XC9E	R/W	1E	R/W	Mailbox Register (15)
XC9F	R/W	1F	R/W	Mailbox Register (16)
XCA0–XCAF	R/W	20–2F	—	Reserved

14.3 Processor Only Register Set

Index Address	Type	Register Description
30	R/W	Peek/Poke Data Register Byte 0
31	R/W	Peek/Poke Data Register Byte 1
32	R/W	Peek/Poke Data Register Byte 2
33	R/W	Peek/Poke Data Register Byte 3
34	R/W	Peek/Poke Address Register Byte 0
35	R/W	Peek/Poke Address Register Byte 1
36	R/W	Peek/Poke Address Register Byte 2
37	R/W	Peek/Poke Address Register Byte 3
38	R/W	Peek/Poke Control Register
39	R/W	I/O Decode Range 0 Base Address
3A	R/W	I/O Decode Range 0 Control Address
3B	R/W	I/O Decode Range 1 Base Address
3C	R/W	I/O Decode Range 1 Control Address
3D	—	Reserved
3E	—	Reserved
3F	—	Reserved
40	R/W	Channel 0 Base Count Register Byte 0
41	R/W	Channel 0 Base Count Register Byte 1
42	R/W	Channel 0 Base Count Register Byte 2
43	R/W	Channel 0 Base Address Register Byte 0
44	R/W	Channel 0 Base Address Register Byte 1
45	R/W	Channel 0 Base Address Register Byte 2
46	R/W	Channel 0 Base Address Register Byte 3
47		Reserved
48	R/W	Channel 0 Configuration Register
49	W	Channel 0 Transfer Strobe Register
4A	R	Channel 0 Status Register
4B	R/W	Channel 0 TBI Base Register Byte 0
4C	R/W	Channel 0 TBI Base Register Byte 1
4D	—	Reserved
4E	—	Reserved
4F	—	Reserved
50	R	Channel 0 Current Count Register Byte 0
51	R	Channel 0 Current Count Register Byte 1
52	R	Channel 0 Current Count Register Byte 2
53	R	Channel 0 Current Address Register Byte 0
54	R	Channel 0 Current Address Register Byte 1
55	R	Channel 0 Current Address Register Byte 2
56	R	Channel 0 Current Address Register Byte 3
57	—	Reserved
58	R	Channel 0 TBI Current Register Byte 0
59	R	Channel 0 TBI Current Register Byte 1
5A	—	Reserved
5B	—	Reserved
5C	—	Reserved
5D	—	Reserved
5E	—	Reserved
5F	—	Reserved

Index Address	Type	Register Description
60	R/W	Channel 1 Base Count Register Byte 0
61	R/W	Channel 1 Base Count Register Byte 1
62	R/W	Channel 1 Base Count Register Byte 2
63	R/W	Channel 1 Base Address Register Byte 0
64	R/W	Channel 1 Base Address Register Byte 1
65	R/W	Channel 1 Base Address Register Byte 2
66	R/W	Channel 1 Base Address Register Byte 3
67	—	Reserved
68	R/W	Channel 1 Configuration Register
69	W	Channel 1 Transfer Strobe Register
6A	R	Channel 1 Status Register
6B	R/W	Channel 1 TBI Register Byte 0
6C	R/W	Channel 1 TBI Register Byte 1
6D	—	Reserved
6E	—	Reserved
6F	—	Reserved
70	R	Channel 1 Current Count Register Byte 0
71	R	Channel 1 Current Count Register Byte 1
72	R	Channel 1 Current Count Register Byte 2
73	R	Channel 1 Current Address Register Byte 0
74	R	Channel 1 Current Address Register Byte 1
75	R	Channel 1 Current Address Register Byte 2
76	R	Channel 1 Current Address Register Byte 3
77	—	Reserved
78	R	Channel 1 TBI Current Register Byte 0
79	R	Channel 1 TBI Current Register Byte 1
7A	—	Reserved
7B	—	Reserved
7C	—	Reserved
7D	—	Reserved
7E	—	Reserved
7F	—	Reserved

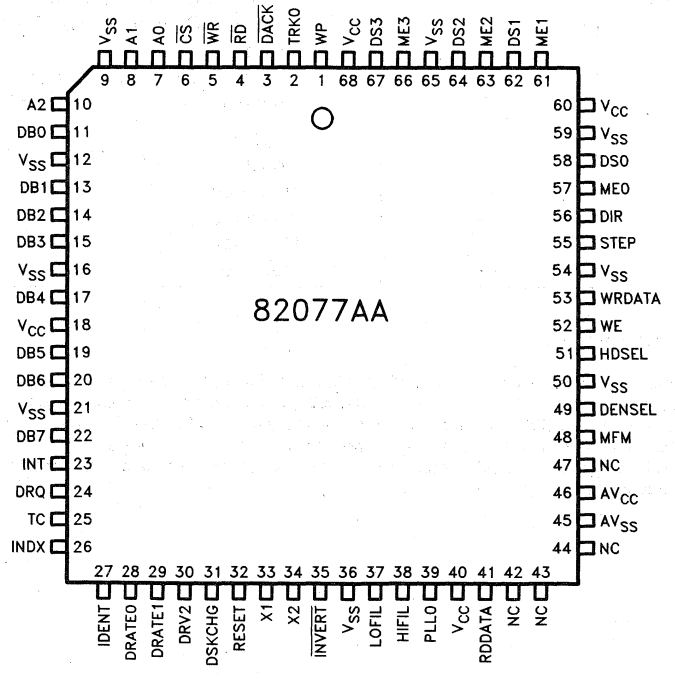
NOTES:
1. TBI = Transfer Buffer Interface
2. X = Slot number
3. All the reserved locations when read, will return a value of no practical use to the user.
4. The "non BMIC" register locations (XC84h–XC87h & 04h–07h) are locations to be used by registers implemented externally on the expansion board. The BMIC will not respond to these locations (XC84h–CC87h) when accessed from the EISA side. However, the BMIC can be programmed to support the decode of the EISA addresses (XC84h–XC87h) through its I/O decode register set (refer to Section 4.8). All "non BMIC" register locations (04h–07h) when read from the local side, will return a value of no practical use to the user.

82077AA
CHMOS SINGLE-CHIP FLOPPY DISK CONTROLLER

- **Single-Chip Floppy Disk Solution**
  - **— 100% PC AT* Compatible**
  - **— 100% PS/2* Compatible**
  - **— 100% PS/2 Model 30 Compatible**
  - **— Integrated Drive and Data Bus Buffers**
- **Integrated Analog Data Separator**
  - **— 250 Kbits/sec**
  - **— 300 Kbits/sec**
  - **— 500 Kbits/sec**
  - **— 1 Mbits/sec**
- **High Speed Processor Interface**
- **Perpendicular Recording Support**
- **Integrated Tape Drive Support**
- **12 mA Host Interface Drivers, 40 mA Disk Drivers**
- **Four Fully Decoded Drive Select and Motor Signals**
- **Programmable Write Precompensation Delays**
- **Addresses 256 Tracks Directly, Supports Unlimited Tracks**
- **16 Byte FIFO**
- **68-Pin PLCC**

The 82077AA floppy disk controller has completely integrated all of the logic required for floppy disk control. The 82077AA, a 24 MHz crystal, a resistor package and a device chip select implements a PC AT or PS/2 solution. All programmable options default to compatible values. The dual PLL data separator has better performance than most board level/discrete PLL implementations. The FIFO allows better system performance in multi-master systems (e.g. PS/2, EISA).

The 82077AA is fabricated with Intel's CHMOS III technology and is available in a 68-lead PLCC (plastic) package.



290166-1

Figure 1. 82077AA Pinout

*PS/2 and PC AT are trademarks of IBM.

82311
Micro Channel* Compatible Peripheral Family

*Micro Channel, PS/2, AT are trademarks of IBM.

October 1990
Order Number: 290167-005

82311 Micro Channel COMPATIBLE PERIPHERAL FAMILY

CONTENTS

PAGE

MICRO CHANNEL COMPATIBLE PERIPHERALS FAMILY OVERVIEW 1-511

82311 MICRO CHANNEL COMPATIBLE CHIP SET 1-512

MICRO CHANNEL INTERFACE AND SPECIFICATIONS 1-516

PACKAGING INFORMATION 1-523

82303 LOCAL I/O SUPPORT CHIP (LIO) 1-526

82304 LOCAL I/O SUPPORT CHIP (LIO) 1-536

82307 DMA/Micro Channel ARBITRATION CONTROLLER 1-551

82308 Micro Channel BUS CONTROLLER (BC) 1-563

82309 ADDRESS BUS CONTROLLER (ABC) 1-594

82077AA FLOPPY DISK CONTROLLER 1-624

Micro Channel COMPATIBLE PERIPHERALS FAMILY

High Performance/High Integration/100% Compatibility

- **Total Solution . . . High Integration VLSI Components Implement Complete Micro Channel Compatible Motherboards**
- **Single Architectural Solution for 386™ DX and 386 SX Systems**
- **High Performance**
  - **— 386 Systems to 25 MHz**
  - **— Up to 16 MB of Zero Wait State Page-Interleaved DRAM**
  - **— Interface to Industry Standard 82385 Cache Controller for Maximum Performance Memory Design**
- **100% Compatible at All Levels**
  - **— Architecture Compatible**
  - **— Register Level Compatible**
  - **— Compatible with All Micro Channel Bus Timing and Drive Characteristics**
- **82311 Chip Set Includes:**
  - **— 82303 and 82304 Local I/O Channel Support Chips**
  - **— 82307 DMA Controller/Central Arbiter**
  - **— 82308 Micro Channel Bus Controller**
  - **— 82309 Address Bus Controller**
  - **— 82077AA Floppy Disk Controller**

Intel's Micro Channel Peripheral Family consists of a complete chip set which can be used to build high performance, 100% Micro Channel compatible motherboard. The 82311 chip set features a highly integrated peripheral bus, and includes the 82077AA Single Chip Floppy Disk Controller. (The 82311 chip set does not support the 8272A or the 82072.) The chip set does support 386 systems at 16, 20, and 25 MHz, as well as 386 SX systems at 16 MHz.

The following pages describe Intel's Micro Channel Peripheral Family. The first section presents an overview of the 82311 and discusses system issues such as clock requirements and Micro Channel interface logic. Following this are the individual component descriptions and specifications.

82311
HIGH INTEGRATION Micro Channel COMPATIBLE PERIPHERAL CHIP SET

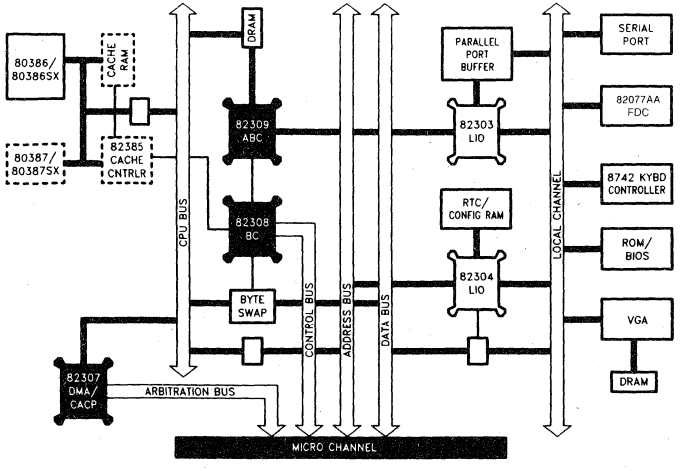
- **High Integration VLSI Components to Implement Micro Channel Compatible Motherboard**
- **Single Architectural Solution for 386 DX 16 MHz, 20 MHz and 25 MHz Systems and 386 SX 16 MHz Systems**
- **Full Compatibility with IBM Micro Channel Architecture**
- **Zero-Wait State Performance**
- **Cache Interface (82385) for Highest Performance Compatible System Implementation with 386 DX**
- **Supports up to 16 MB of Memory on Motherboard**
  — Extended Memory for OS/2 Support
- **100% IBM Compatible VGA Graphics**
- **Flexible Memory Architecture Support**
  — Up to 4 Banks of Interleaved Page Memory
  — 256K, 1M, 4M DRAM Support
- **Supports the 82077AA Single Chip Floppy Disk Controller, Which Supports 3½″ and 5¼″ Disk Drives**
- **Keyboard and BIOS Support from 3rd Party**
- **Math Coprocessor(s) Interface (387 DX, 387™ SX)**
- **Surface Mount Packaging for Small Footprint Design (0.025″ Pitch)**
- **Low Power CHMOS Technology**
- **Available in 100 & 132-Pin Plastic Quad Flat Pack Packages.**

(See Packaging Spec. # 231369)

Intel's peripheral chip family is designed to support the new generation of Micro Channel compatible systems. Intel's Micro Channel compatible peripheral solution consists of highly integrated VLSI components designed to support 386 DX systems up to 25 MHz, as well as 16 MHz 386 SX systems.

The Intel solution is based on the high performance IBM* Model 80 register model but it is highly integrated to provide full compatibility across all models. The specifications for 82311 VLSI components conform to architectural specifications defined for the Micro Channel Bus Architecture. The VLSI components are implemented in 1.5 micron CHMOS technology and packaged in space saving surface mount JEDEC flat pack packages.

*IBM is a registered trademark of IBM.



290167-82

INTRODUCTION

The new generation of Personal Computer systems from IBM offers significant technological advantages over the PC/AT and XT systems. The most significant advancement is in the *Architectural* definition of the bus—Micro Channel Bus. Unlike the AT bus, the Micro Channel is well defined in terms of bus protocol timings. To create a compatible Micro Channel system requires adherence to the Micro Channel timings and electrical drive characteristics.

All IBM Micro Channel models have increased system functionality included on the motherboard. In the older PC/AT architecture, such functionality required the addition of peripheral cards. Specific features added to the motherboard include the Serial Port, Bi-directional Parallel Port and Video Graphics Control.

Micro Channel ARCHITECTURE

The Micro Channel Bus is defined to support an open architecture providing Multi-Master capability, Multi-Device arbitration with fairness, arbitration capability and easy configurability of the total system (Programmable Option Select-POS). Providing full details about the Micro Channel Bus Architecture is beyond the scope of this document. Please refer to IBM Technical Reference Manuals on Micro Channel systems.

To provide Multi-Master capability as defined in the Micro Channel Architecture, each Master device is responsible for driving the Address, Data, arbitration and control signals. For operation reliability and compatibility there are significant constraints in terms of timing and drive levels. These constraints are well documented in IBM's Technical Reference Manual for Micro Channel systems. Intel's chip set is designed to meet the Micro Channel timings.

The Micro Channel has four modes of Memory and I/O Bus cycles. These are Default cycle, Synchronous Extended cycle, Asynchronous Extended cycle and Matched Memory cycle. Each of these bus cycles is supported by the Intel Peripheral chip set.

COMPATIBILITY METRICS

The Intel chip set provides full compatibility with the IBM Micro Channel solution. All Bus cycles comply with the Micro Channel timings. Selection of buffers for drive level with minimum delays to meet Micro Channel timings are specified in the Intel *82311 Micro Channel Compatible Peripheral Chip Set Designers Guide.*

MEMORY PERFORMANCE

With the Intel chip set, Micro Channel compatible motherboards can be designed to provide zero-wait performance. Performance is predicated on memory design and DRAM speed selection. The Intel chip set offers flexible memory design support to meet various cost/performance goals.

SYSTEM CONSIDERATIONS

System Components

82303	Local I/O Support Chip
82304	Local I/O Support Chip
82307	DMA/CACP Controller
82308	Micro Channel Bus Controller
82309	Address Bus Controller
82077AA	Floppy Disk Controller

Note that the above names/numbers are frequency independent; i.e., they refer to a generic functional VLSI device. To actually implement for example, a 20 MHz system, however, requires an 82311-20 Chip Set as opposed to an 82311-16 Chip Set. The 25 MHz version of the 82308 (dubbed the 82308HS-25) cannot be used at 16 MHz or 20 MHz.

To implement a minimum configuration Micro Channel compatible motherboard, each of the seven system components listed above are required in addition to the following components:

- 386 DX or 386 SX Microprocessor
- TTL Buffers for Various Buses in the System
- 8742 Keyboard Controller with Firmware for 101 and 102 Keyboard Interface
- Battery-Backed Real Time Clock with CMOS RAM
- Serial Port
- Memory

— ROM BIOS
— DRAMs for Main Memory
— DRAMs for VGA

- System Clock Sources
- Mechanical Connectors/Components

The Intel solution is supported by a fully compatible BIOS firmware from a third-party vendor.

82311 CHIP SET SYSTEM CLOCK REQUIREMENTS

- **Introduction**
- **Clock Definitions**
- **Clock Requirements**

INTRODUCTION

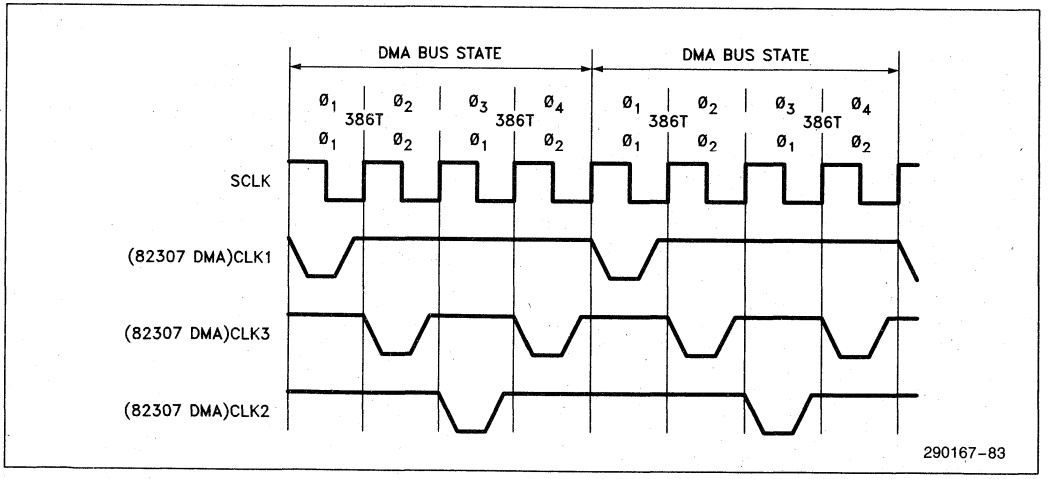
This section describes the basic clocking scheme of the host CPU (386 DX or 386 SX), LIO (82304), DMA (82307), BC (82308) and ABC (82309). Although each component spec individually describes its own clock requirements, this section describes the synchronous relationship that exists between them. (Note that several other clocks exist in a Micro Channel system. However, this section describes only those clocks that are synchronously related to the CPU clock.)

The clocking scheme essentially divides the DMA bus state into four phases as depicted in the figure. Note that there is a direct 2-to-1 mapping of 386 state to DMA state. The DMA (82307) comprehends phases by inputting distinct, active low, non-overlapping clock phases. The Address Bus Controller, Bus Controller and LIO device learn the system phase by synchronously sampling the falling edge of RESET, as described in the component specifications.

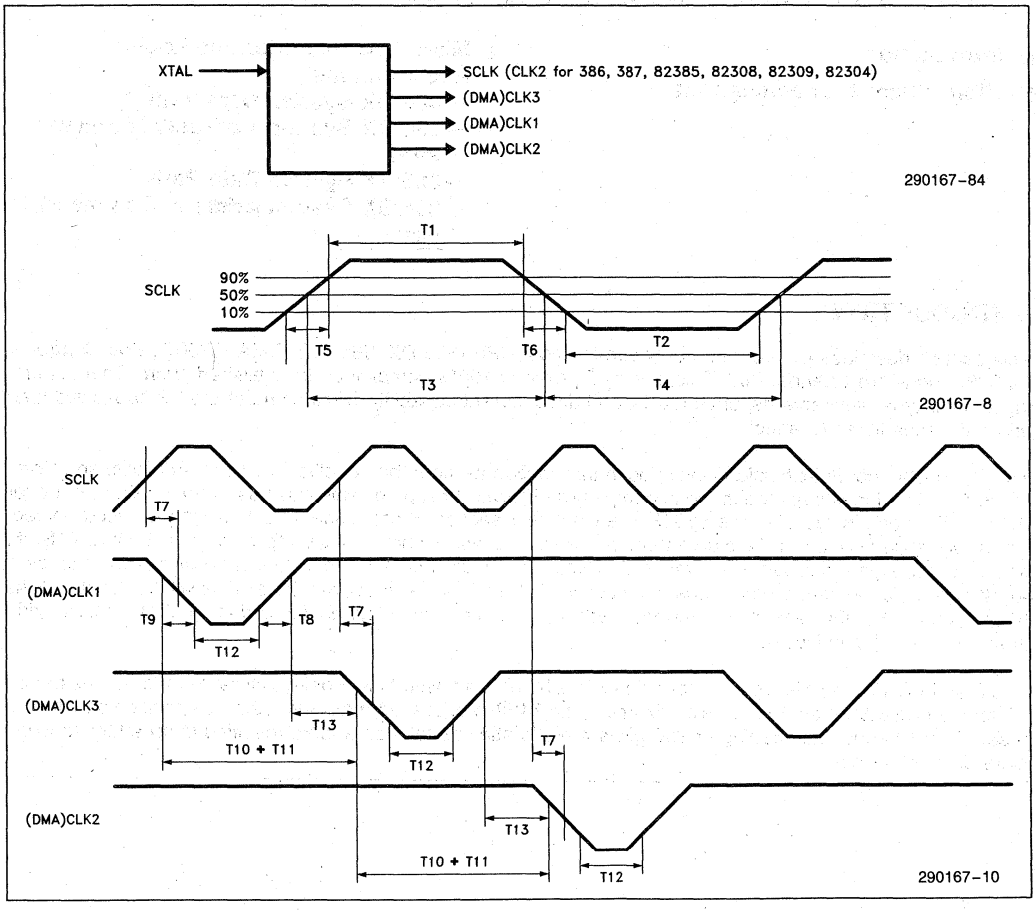
SYSTEM CLOCK CONSIDERATIONS

- SCLK load should be evenly divided between SCLKA, SCLKB, SCLKC, and SCLKD.
- The calculated characteristic impedance of all clock lines should be as near as possible to 100Ω.
- All clock lines should be kept as short as possible with no stubs.
- Clock lines should be driven at one end, with optional parallel termination at the other end. Series termination should be as close as possible to the driver.
- Guidelines in the *386™ Hardware Reference Manual,* Chapter 11, Sections 2 and 3, for clock design should be followed.
- A pull-down 100Ω resistor is needed on signal CLK3 (82309 pin 13) to terminate the signal properly.

BASIC FOUR-PHASE CLOCKING REQUIREMENT



CLOCK CIRCUIT DEFINITION



SYSTEM CLOCK REQUIREMENTS

Symbol	Parameter	Kit 16 MHz		Kit 20 MHz		Kit 25 MHz		Notes
		Min	Max	Min	Max	Min	Max	
T1	SCLK High Time (90%)	8		6.5		5.5		
T2	SCLK Low Time (10%)	8		6.5		5.5		
T3	SCLK High Time (50%)	12		10		9		1
T4	SCLK Low Time (50%)	12		10		9		1
T5	SCLK Rise Time		3.5		3.5		3.5	
T6	SCLK Fall Time		3.5		3.5		3.5	
T7	SCLK-To-DMACLK(N) Skew	−2	3	−2	3	−2	3	2
T8	DMACLK(N) Rise Time		2		2		2	
T9	DMACLK(N) Fall Time		2		2		2	
T10	SCLK Period							
T11	DMACLK-To-DMACLK Skew	−2	2	−2	2	−2	2	2
T12	DMACLK Low Time	15		15		12		
T13	DMACLK Non-Overlap Time	4		4		2		

NOTES:
1. Needed to enforce a duty cycle between 40% and 60% (45% and 55% at 25 MHz).
2. Limiting skew to this level is recommended.

Micro Channel INTERFACE AND SPECIFICATIONS

- **Introduction**
- **Micro Channel Specifications**
- **Micro Channel Interface Logic Requirements**
  - **— 386 DX System Data Path**
  - **— 386 DX System Address/Command Path**
  - **— 386 SX System Data Path**
  - **— 386 SX System Address Command Path**

INTRODUCTION

This section describes the interface between the host CPU (386 DX, 386 SX), DMA (82307), Bus Controller (82308) and Micro Channel Bus. This interface provides 100% compliance to published Micro Channel timings, driver type requirements, drive levels, and drive current capability. Timings meet the full capacitive load allowed on the Micro Channel.

The Micro Channel Specifications included in this section assume the specific TTL Data, Address, and Command Path interfaces depicted in the accompanying figures. Timing analysis was based on the Bus Controller AC specifications included in the 82308 Bus Controller section. Worst case TTL analysis was used, except when two related signals share a path through the same physical chip. (For example, since MMCCMD#, S0#, and S1# propagate through the same 74F241 package in an 386 system, one signal will not experience a worst case delay while the other sees a best case. Rather, it is assumed that the signals will track within 2 ns of each other.) For this reason, it is important to follow the recommendations detailed at the end of this section in the Interface Logic Notes.

The F and AS TTL logic is typically specified into a 50 pF load, worst case delays were derated at 1 ns per 50 pF for loads greater than 50 pF. As an example, the 74F241 published maximum delay is specified as 7 ns. To meet Micro Channel bus loading of 250 pF, a 4 ns derating factor was added, resulting in an effective worst case delay of 11 ns.

DEFAULT CYCLE SPECIFICATIONS — ALL KITS

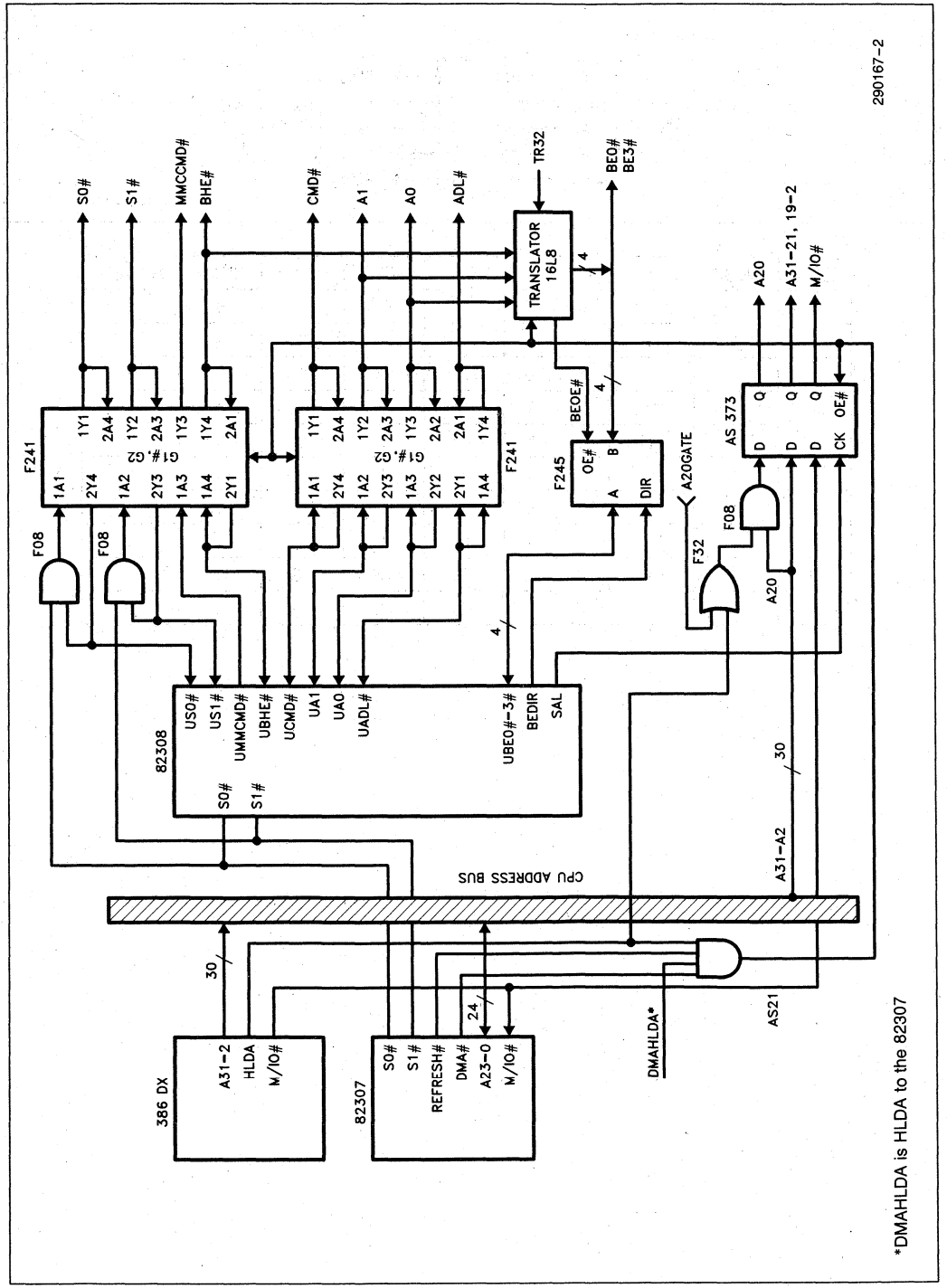
Symbol	Parameter	Min	Max
T1	Status active from ADDR,M/IO#,REFRESH#	10	
T2	CMD# active from Status active	55	
T3	ADL# active from ADDR,M/IO#,REFRESH#	45	
T4	ADL# active to CMD# active	40	
T5	ADL# active from Status active	12	
T6	ADL# pulse width	40	
T7	Status hold from ADL# inactive	25	
T8	ADDR,M/IO#,REFRESH#,SBHE# hold frm ADL# INACTIVE	25	
T9	ADDR,M/IO#,REFRESH#,SBHE# hold frm CMD# ACTIVE	30	
T10	Status hold from CMD# active	30	
T11	SBHE# setup to ADL# inactive	40	
T12	SBHE# setup to CMD# active	40	
T13	CDDS16/32 active from ADDR,M/IO#,REFRESH#		55
T14	CDSFDBK# active from ADDR,M/IO#,REFRESH#		60
T15	CMD# active from ADDRESS valid	85	
T16	CMD# pulse width	90	
T17	Write data setup to CMD# active	0	
T18	Write data hold from CMD# inactive	30	
T19	Status to Read Data valid (Access Time)		125
T20	Read Data valid from CMD# active		60
T21	Read Data hold from CMD# inactive	0	
T22	Read Data bus tri-state from CMD# INACTIVE		40
T23	CMD# active to next CMD# active	190	
T23A	CMD# inactive to next CMD# active	80	
T23B	CMD# inactive to next ADL# active	40	
T24	Next Status active from Status Inactive	30	
T25	Next Status active to CMD# Inactive		20
T26	CHRDY INACTIVE FROM ADDR VALID		60
T27	CHRDY INACTIVE FROM STATUS ACTIVE		30
T28	CHRDY RELEASE FROM CMD# ACTIVE		30
T28D	READ DATA VALID FROM CMD# ACTIVE		160
T29S	READ DATA VALID FROM CHRDY RELEASE		60
T31	BE#(0–3) from Addr valid (32-Bit Masters Only)		40
T32	BE#(0–3) active from SBHE#,A0,A1 active		30
T33	BE#(0–3) active to CMD# active	10	

MATCHED MEMORY CYCLE SPECIFICATIONS

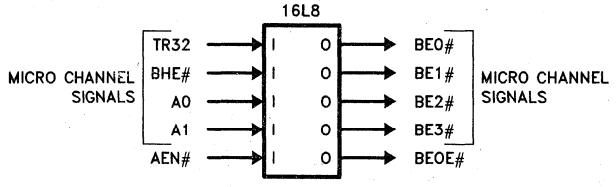
ALL KITS

Symboi	Parameter	Min	Max
T1	ADDR VALID TO STATUS ACTIVE	10	
T2	Status valid to MMCCMD# active	82	
T3	ADDR hold from MMCCMD# active	20	
T4	Status hold from MMCCMD# active	25	
T5	CDDS16/32 active from ADDR valid		55
T6	MMCR# active from ADDR valid		55
T7	CDSFDBK# active from ADDR valid		60
T8	ADDR valid to MMCCMD# active	100	
T9	MMCCMD# pulse width	85	
T10	Write Data valid to MMCCMD# active	0	
T11	Write Data hold from MMCCMD# inactive	30	
T12A	Read Data valid from Status active		145
T12B	for non-aligned xfers (16b < = = > 32b)		145
T13A	Read Data valid from MMCCMD# active		60
T13B	for non-aligned xfers (16b < = = > 32b)		60
T14	Read Data hold from MMCCMD# inactive	0	
T15	Read Data off dly from MMCCMD# inactive		40
T16	MMCCMD# active to next MMCCMD# active	180	
T17	CDCHRDY valid from ADDR valid		70
T18	CDCHRDY valid from Status active		30
T23	Status inactive pulse width	30	
T24	MMCCMD# inactive to Status active	5	
T25	MMCCMD# inactive pulse width	85	
T26	MMCCMD# ACTIVE TO NEXT STATUS ACTIVE	90	

386 DX SYSTEM ADDRESS/COMMAND PATH



386 DX SYSTEM BYTE ENABLE TRANSLATOR PAL

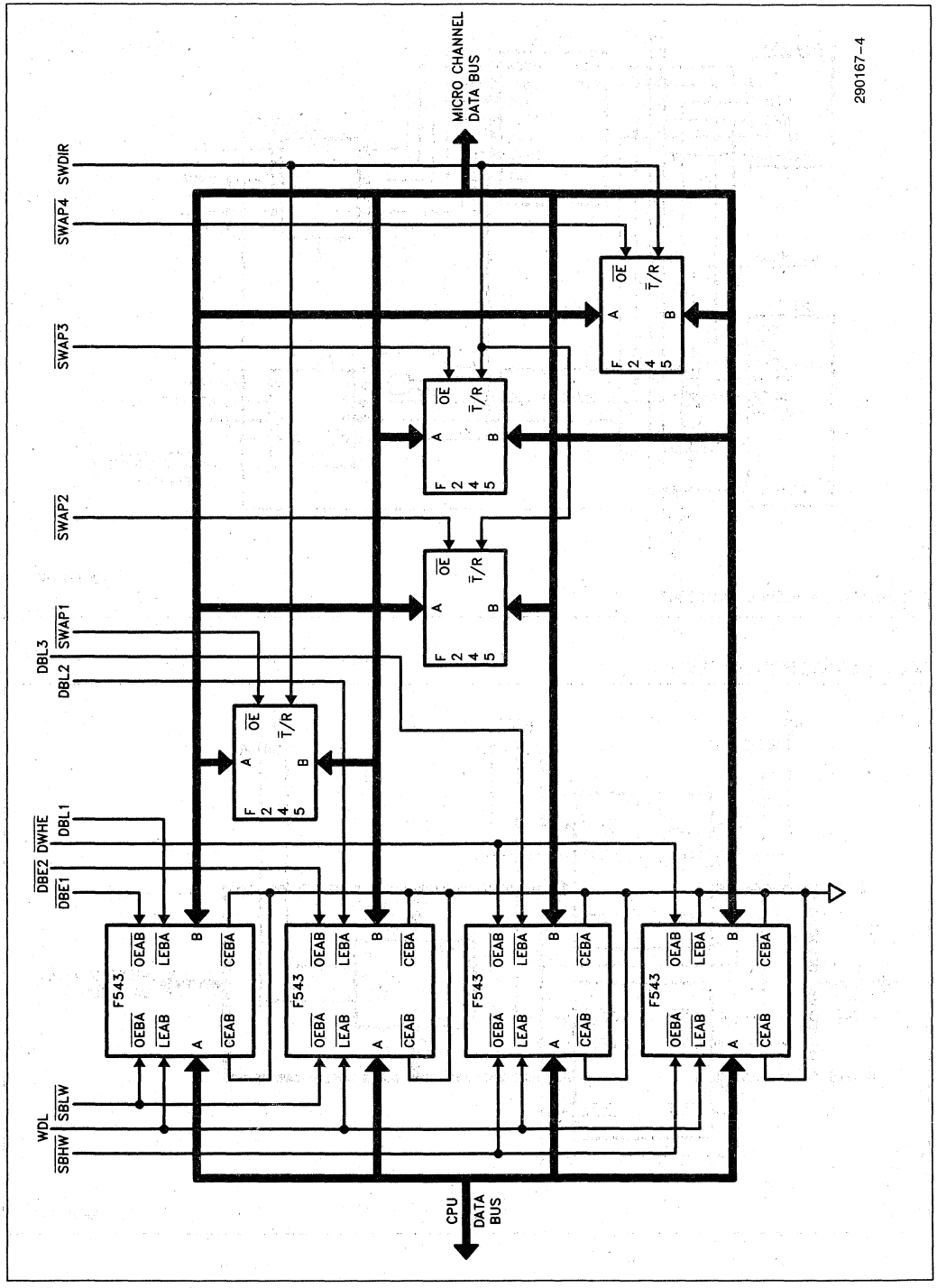


290167–3

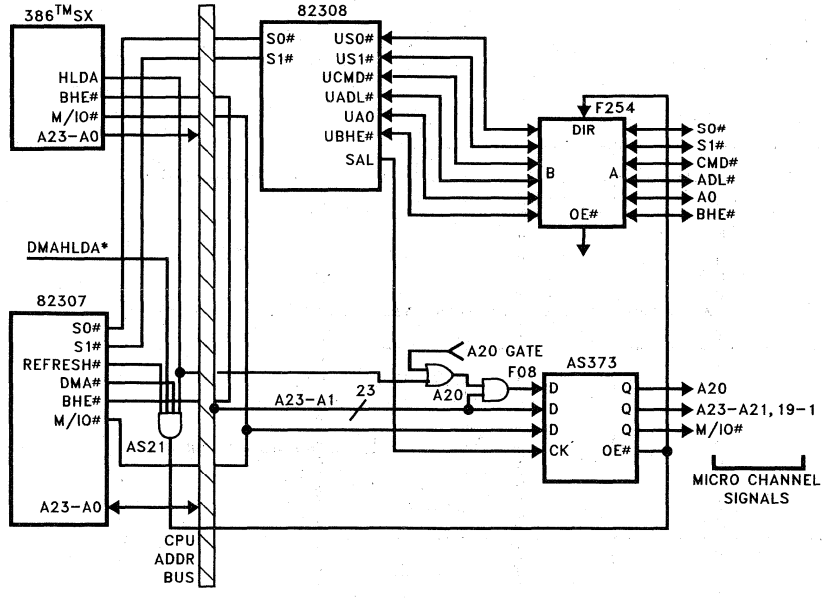
TR32	AEN#	BHE#	A1	A0	BE3#	BE2#	BE1#	BE0#	BEOE#
1	1	0	0	0	1	1	0	0	1
1	1	0	0	1	1	1	0	1	1
1	1	0	1	0	0	0	1	1	1
1	1	0	1	1	0	1	1	1	1
1	1	1	0	0	1	1	1	0	1
1	1	1	0	1	1	1	0	1	1
1	1	1	1	0	1	0	1	1	1
1	1	1	1	1	0	1	1	1	1
0	X	X	X	X	TS	TS	TS	TS	0
X	0	X	X	X	TS	TS	TS	TS	0

TS = TRISTATE

386 DX SYSTEM DATA PATH

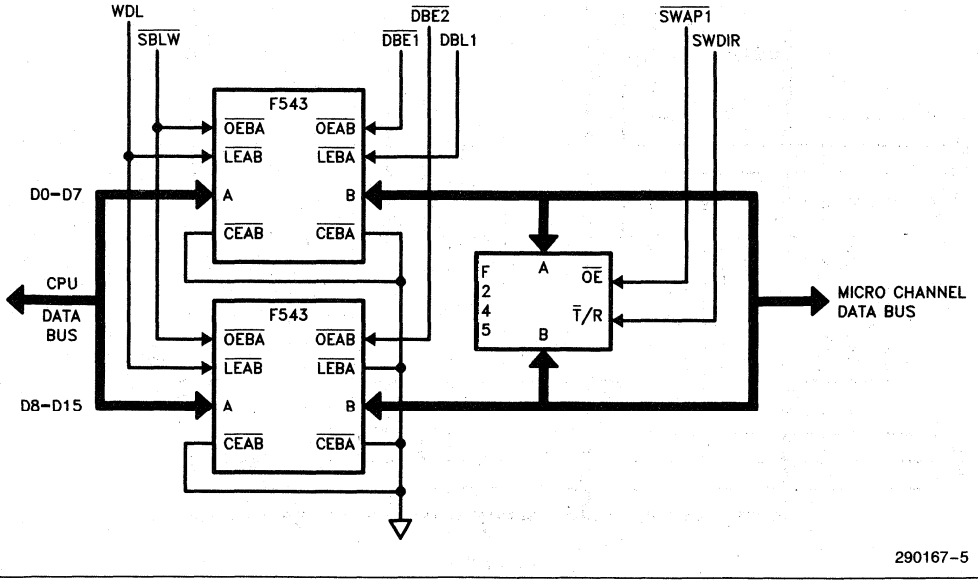


386 SX SYSTEM ADDRESS/COMMAND PATH



*DMAHLDA is HLDA to the 82307

386 SX SYSTEM DATA PATH



Micro Channel Interface Logic Notes

386 DX SYSTEM

1. The F08 gates in the S0#, S1# path are required at 20 MHz and at 25 MHz. They should not be used at 16 MHz.
2. In an 82385 system, the A20 gate logic is on the 386 DX local bus, and is thus not required on the 82385 local bus as shown in the diagram.
3. The F08 gates in the S0#, S1# path and the F08 in the A20 path should all be from the same TTL package.
4. It is important that S0#, S1#, BHE#, and MMCCMD# go through the same F241 package, and that CMD#, ADL#, A0, and A1 go through the same package.

386 SX SYSTEM

1. The F08 gates in the S0#, S1# path and in the A20 path should all be from the same package.

PLASTIC PACKAGING INFORMATION

(See Packaging Spec. Order # 231369)

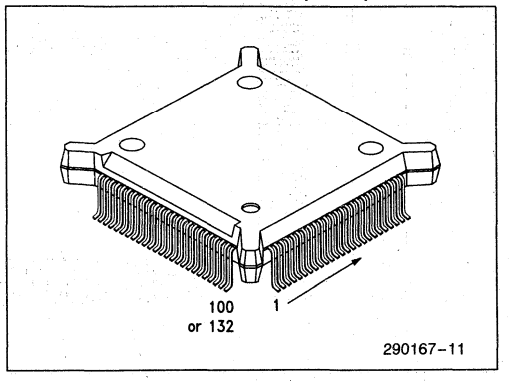
Introduction

The individual components of Intel's Micro Channel Compatible Peripheral Chip Sets come in JEDEC standard Gull Wing packages (25 MIL pitch), with "bumpers" on the corners for ease of handling. Please refer to the accompanying table for the package associated with each device, and to the individual component specifications for pinouts. (Note that the individual pinouts are numbered consistently with the numbering scheme depicted in the accompanying figures.)

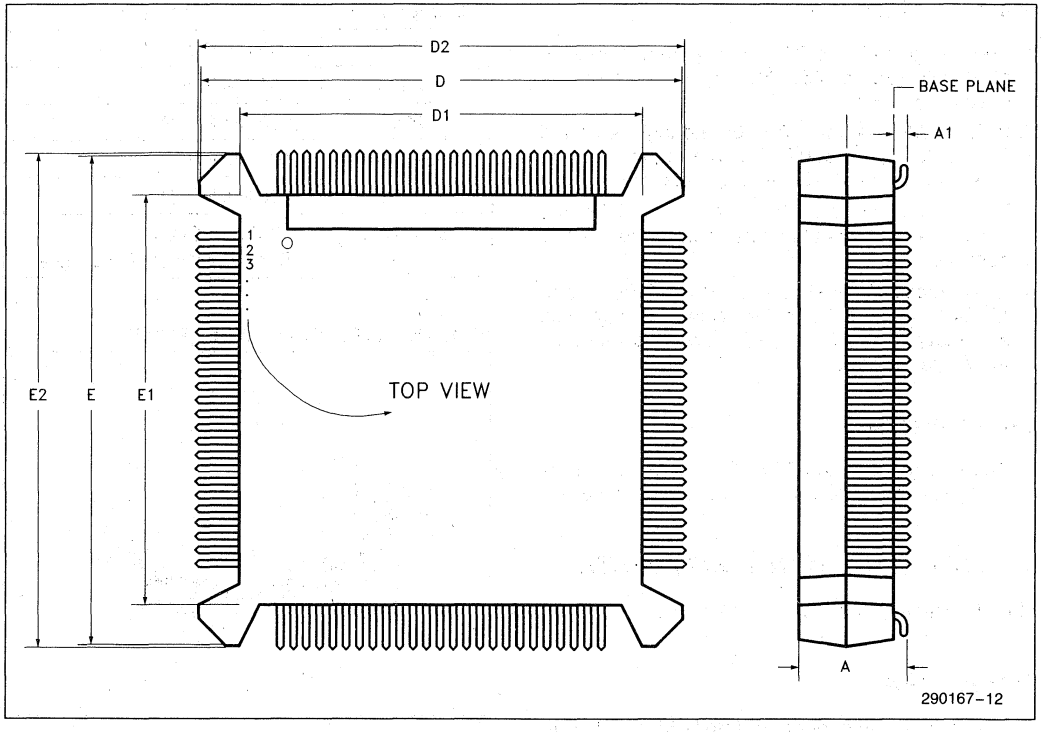
MICRO CHANNEL COMPATIBLE PERIPHERAL FAMILY COMPONENT PACKAGES

Component	Package
82303	100 Pin PQFP
82304	132 Pin PQFP
82307	132 Pin PQFP
82308	100 Pin PQFP
82309	100 Pin PQFP
82077AA	68-Pin PLCC, See Component Data Sheet

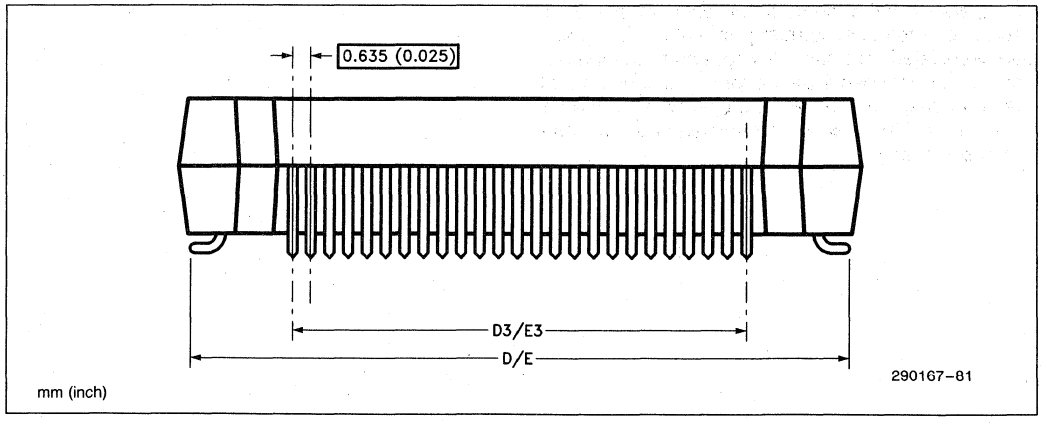
PLASTIC QUAD FLAT PACK (PQFP)



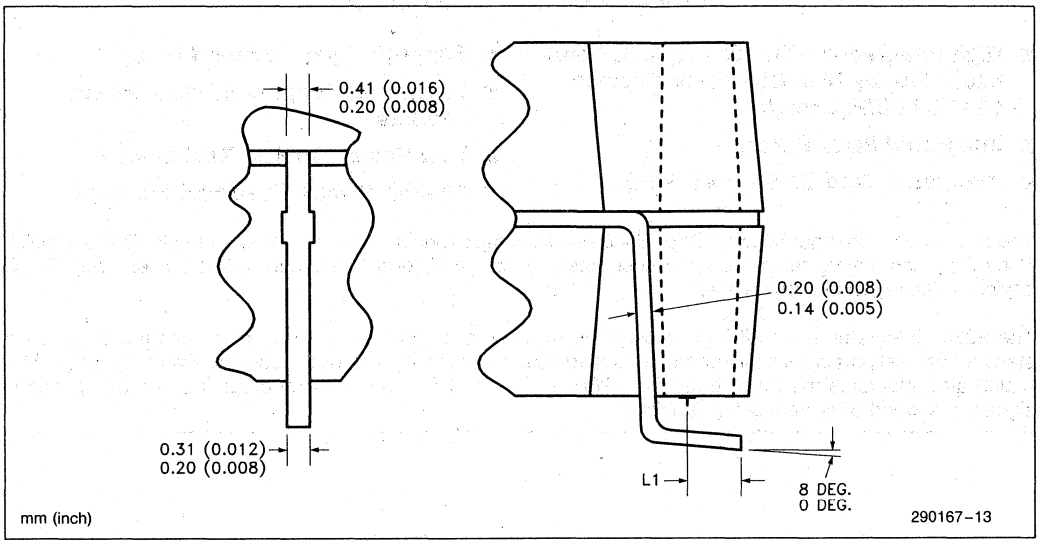
PRINCIPAL DIMENSIONS & DATUMS



TERMINAL DETALS



TYPICAL LEAD



Case Outline Drawings
Plastic Fine Pitch Chip Carrier

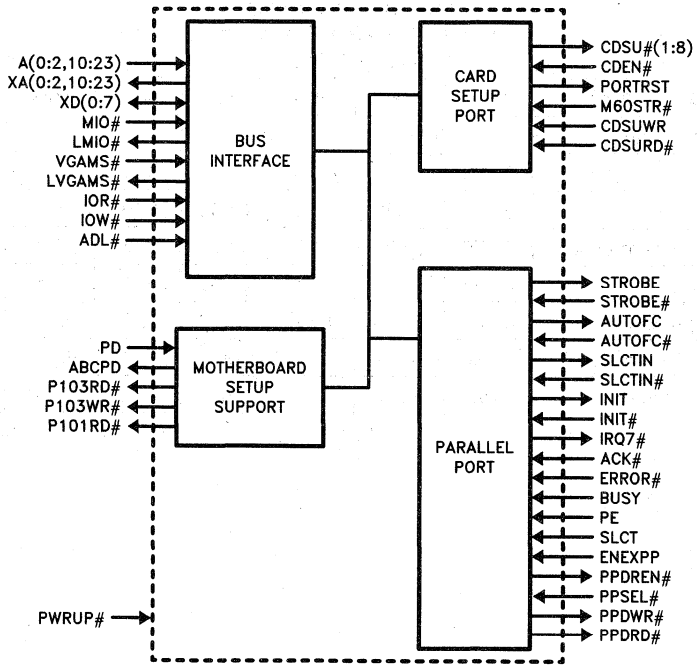
		0.025 inch Pitch				0.84 mm Pitch			
Symbol	**Description**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
N	Lead Count	100		132		100		132	
A	Package Height	0.160	0.170	0.160	0.170	4.06	4.32	4.06	4.32
A1	Standoff	0.020	0.030	0.020	0.030	0.51	0.76	0.51	0.76
D, E	Terminal Dimension	0.875	0.885	1.075	1.085	22.23	22.48	27.31	27.56
D1, E1	Package Body	0.747	0.753	0.947	0.953	18.97	19.13	24.05	24.21
D2, E2	Bumper Distance	0.897	0.903	1.097	1.103	22.78	22.94	27.86	28.02
D3, E3	Lead Dimension	0.600 Ref		0.800 Ref		15.24 Ref		20.32 Ref	
L1	Foot Length	0.020	0.030	0.020	0.030	0.51	0.76	0.51	0.76
		Inch				mm			

82303
LOCAL I/O SUPPORT CHIP

- **High Integration—The 82304, 82303 and 82077 Floppy Disk Controller Replace 50 IC's in IBM Design**
- **Integrated Parallel Port**
- **Integrated Card Setup Port (96H)**
- **Supports System Board Setup**
- **Integrated Peripheral Bus Address Latches**
- **Low Power CHMOS Technology**
- **100-Pin Plastic Quad Flat Package**

The 82303 Local Channel Support Chip, along with its companion chip (the 82304) and the 82077 Floppy Disk Controller, significantly reduce system cost, design effort, and form factor constraints by replacing 50 IC devices in an equivalent IBM system.

The 82303 integrates most all logic required to implement a parallel port. This port operates either as a standard parallel port or as a Microchannel architecture compatible "extended mode" (bi-directional) port. The 82303 also integrates the Card Setup Port (96H) and several peripheral bus address latches, and provides signals in support of system setup functions.



290184-1

November 1988
Order Number: 290184-001

Introduction

The 82303 is a high integration device intended for Microchannel compatible system designs. It integrates the Microchannel Card Setup Port, a parallel port, several peripheral bus address latches, and a variety of system board setup functions. The 82303, in conjunction with its sister chip the 82304 and the 82077 Floppy Disk Controller, replaces approximately 50 IC devices in an equivalent IBM system. Included as an appendix to this data sheet is a functional logic diagram of the 82303 that will facilitate understanding of the part. Note that the 82304 and 82303 integrate a variety of system ports. For programming and register level details, please refer to the IBM Technical Reference Manual.

Bus Interface

The Bus Interface unit interfaces the 82303 to the Microchannel and peripheral busses. It inputs the unlatched Microchannel address, latches it for internal use, and makes the latched version available externally for other peripheral bus resources. It also provides additional latches for decodes generated from the Microchannel address.

Parallel Port

The 82303 integrates most all logic required to implement a standard or "extended-mode" parallel port. The only logic not integrated is that which directly drives the physical parallel port connector, specifically one '05 (open collector) inverter package and one '652 data buffer. (This allows the system design to stay clear of directly exposing a VLSI component to an external connector.) The parallel port can serve as LPT1, LPT2, or LPT3, as dictated by the decode received via the input parallel port decode PPSEL#.

Card Setup Port

The 82303 integrates the Card Setup Port (96H), which generates the card setup lines to the individual Microchannel connectors. This port also features a software generated reset capability that resets the Microchannel, serial port, and parallel port independently of the rest of the system.

Motherboard Setup Support

The 82303 generates decoded read/write strobes for system board setup port 103H, and a read strobe for setup port 101H. It also generates a version of the system board POS decode (ABCPD) that is then forwarded to the 82309 Address Bus Controller. Note that other system board setup ports can be easily implemented externally using the same PD (POS Decode) that the 82303 uses.

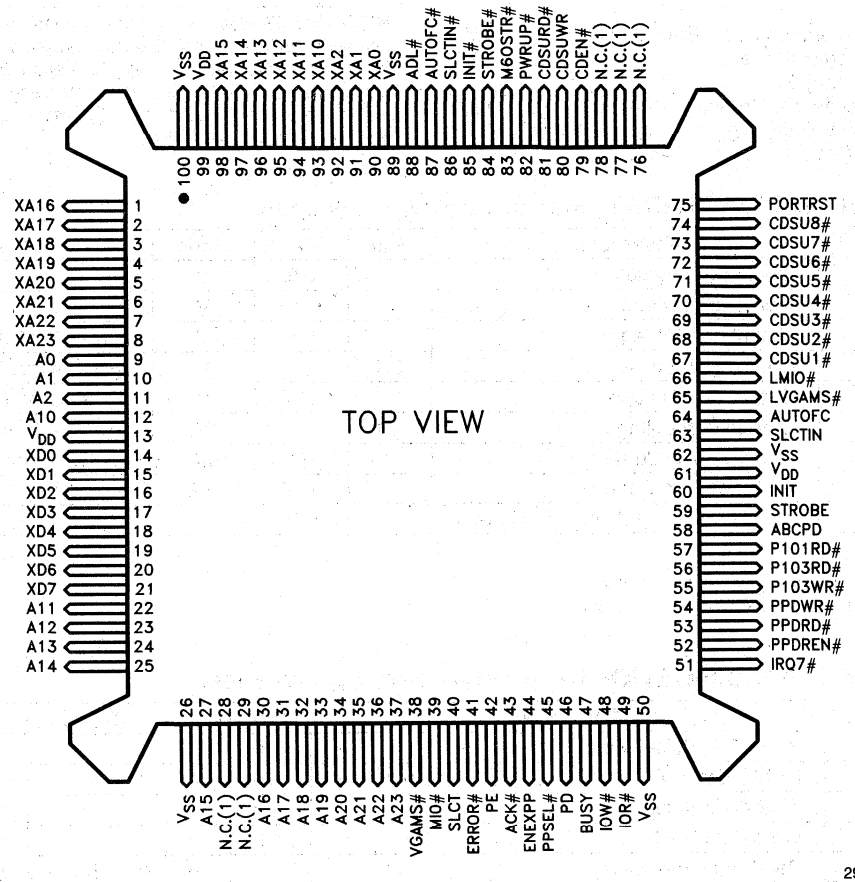
82303 Local Channel Support Chip Pin Definitions

Signal Name	Pin Number	I/O	Description
PWRUP#	82	I	Power-up reset input. Brings 82303 to initial known state.
A[0:2, 10:23]	9–12, 22–25, 27, 30–37	I	Microchannel address inputs. These signals are internally latched. (Note that in systems in which a full 24-bit peripheral address is not required, the upper significant address latches may be used as general purpose decode latches.)
XA[0:2, 10:23]	90–98, 1–8	O	Peripheral Bus Address. These outputs are latched versions of the Microchannel address inputs.
XD[0:7]	14–21	I/O	Bi-directional peripheral data bus.
MIO#	39	I	Microchannel MIO# indicator.
LMIO#	66	O	Latched Microchannel MIO# indicator. The MIO#/LMIO# pin combination may be used as a general purpose latch if LMIO# is not required.
VGAMS#	38	I	VGA memory buffer decode.

82303 Local Channel Support Chip Pin Definitions (Continued)

Signal Name	Pin Number	I/O	Description
LVGAMS#	65	O	Latched VGA memory buffer decode. The VGAMS#/LVGAMS# pin combination may be used as a general purpose latch if LVGAMS# is not required.
IOR#, IOW#	49, 48	I	82303 read/write strobes.
ADL#	88	I	Microchannel ADL# input.
PD	46	I	POS decode. Decode driven in response to accesses to system board setup Ports 100, 101, 103–107H.
ABCPD	58	O	Address Bus Controller (82309) POS decode. This is simply the PD input gated by an active IOW# signal, and insures that the 82309 does not see a decoding glitch.
P101RD#, P103RD#, P103WR#	57, 56, 55	O	Various system board setup port read/write strobes.
CDSU#[1:8]	67–74	O	Card setup signals to the Microchannel slots.
CDEN#	79	I	Port 100-107H decode used as qualifier for card setup signals.
PORTRST	75	O	Microchannel reset signal. The "OR" of the power-up reset and the reset function built into Port 96H.
M60STR#	83	I	Model 60 strap. When low, the 82303 will drive Port 96H data in either a Port 96 or 97H read. (This is in keeping with the Model 50/60 definition.) When high, the 82303 will remain tri-stated during a Port 97H read.
CDSUWR	80	I	Port 96–97H write strobe.
CDSURD#	81	I	Port 96–97H read strobe.
STROBE, AUTOFC, SLCTIN, INIT	59, 64, 63, 60	O	Parallel port control outputs. These signals are externally buffered with open collector inverters before driving the parallel port connector.
STROBE#, AUTOFC#, SLCTIN#, INIT#	84, 87, 86, 85	I	Parallel port control inputs.
IRQ7#	51	O	Parallel port interrupt request.
ACK#, ERROR#, BUSY, PE, SLCT	43, 41, 47, 42, 40	I	Parallel port status inputs.
ENEXPP	44	I	Enable parallel port extended mode. Allows parallel port to operate bi-directionally.
PPSEL#	45	I	Parallel port chip select.
PPDREN#	52	O	Enables the external '652 parallel port data buffer to be used bi-directionally. This signal is a function of the Control port direction bit (bit 5) and the ENEXPP input.
PPDWR#, PPRD#	54, 53	O	Parallel port data buffer write/read strobes.
V_{DD}	13, 61, 99		Power.
V_{SS}	26, 50, 62, 89, 100		Ground.
N.C.	28, 29, 76, 77, 78		No Connect.

82303 100-Pin PQFP Pinout



290184-2

NOTE:
1. N.C. pins must be left not connected.

82303 PARAMETRICS ABSOLUTE MAXIMUM RATINGS*

Case Temperature Under Bias .... −40°C to +85°C
Storage Temperature ..... −65°C to +150°C
Voltage to any Pin
with Respect to Ground −0.3V to +(V_{CC} +0.3)V
DC Supply Voltage (V_{CC}) ..... −0.3V to +7.0V
DC Input Current ..... ±10 mA

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_C = 0°C to +70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage		0.8	V	
V_{IH}	Input High Voltage	2.0		V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 4 mA (Note 1)
V_{OH}	Output High Voltage	2.4		V	I_{OH} = 4 mA (Note 1)
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2 mA (Note 2)
V_{OH}	Output High Voltage	2.4		V	I_{OH} = 2 mA (Note 2)
I_{CC}	Power Supply Current		180	mA	No DC Loads
I_{LI}	Input Leakage Current		±10	μA	$V_{SS} < V_{IN} < V_{CC}$
I_{OZ}	TRI-STATE Output Leakage Current		±10	μA	$V_{SS} < V_{OUT} < V_{CC}$

NOTES:
1. CDSU# [1:8], XA [0:2, 10:23], XD[0:7].
2. All outputs other than those listed in Note 1.

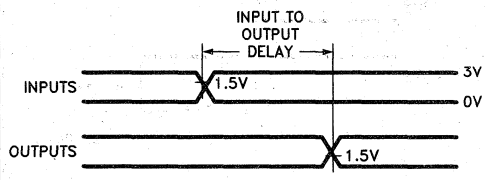
82303 A.C. SPECIFICATIONS T_C = 0°C to +70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	C_L (pF)	Notes
T_1	PWRUP#, Pulse Width	500			
T_2	IOR#, IOW#, CDSURD#, CDSUWR Pulse Width	170			
T_3	Write Data Setup	25			(Note 1)
T_4	Write Data Hold	10			(Note 2)
T_5	Read Data Valid Delay		60	100	(Note 3)
T_6	Read Data Float Delay		40	100	(Note 5)
T_7	Status Inputs to XD[0:7]		35	100	(Note 6)
T_8	Write Strobe Delays		35	50	(Note 7)
T_9	Read Strobe Delays		40	50	(Note 8)
T_{11}	PPSEL#, PD Setup to IOR#, IOW# ↓	20			
T_{12}	PPSEL#, PD Hold from IOR#, IOW# ↑	5			
T_{13}	XA[0:2, 10:23] DLY from ADL# ↓		35	100	
T_{14}	A[0:2, 10:23], VGAMS#, MIO# Setup to ADL# ↑	30			
T_{17}	CDSU# [1:8] Delay from CDEN#		28	75	
T_{18}	IOR# ↑ to ADL# ↓	30			
T_{19}	LVGAMS#, LMIO# Delay from ADL# ↓		35	50	

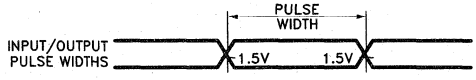
NOTES:
1. To IOW# or CDSUWR active, whichever is appropriate.
2. From IOW# or CDSUWR inactive, whichever is appropriate.
3. From IOR# or CDSURD# active, whichever is appropriate.
5. From IOR# or CDSURD# inactive, whichever is appropriate.
6. Parallel port status inputs include SLCT, PE, BUSY, ERROR#, and ACK#.
7. Write strobes include P103WR# and PPDWR#.
8. Read strobes include P103RD#, P101RD#, and PPDRD#.

82303 Drive Levels/Measurement Points for A.C. Specifications

Input/Output Pulse Widths

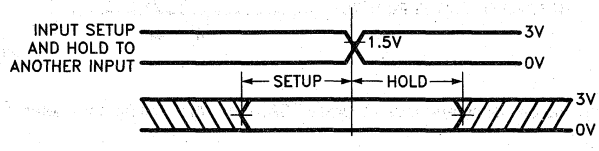


290184-3



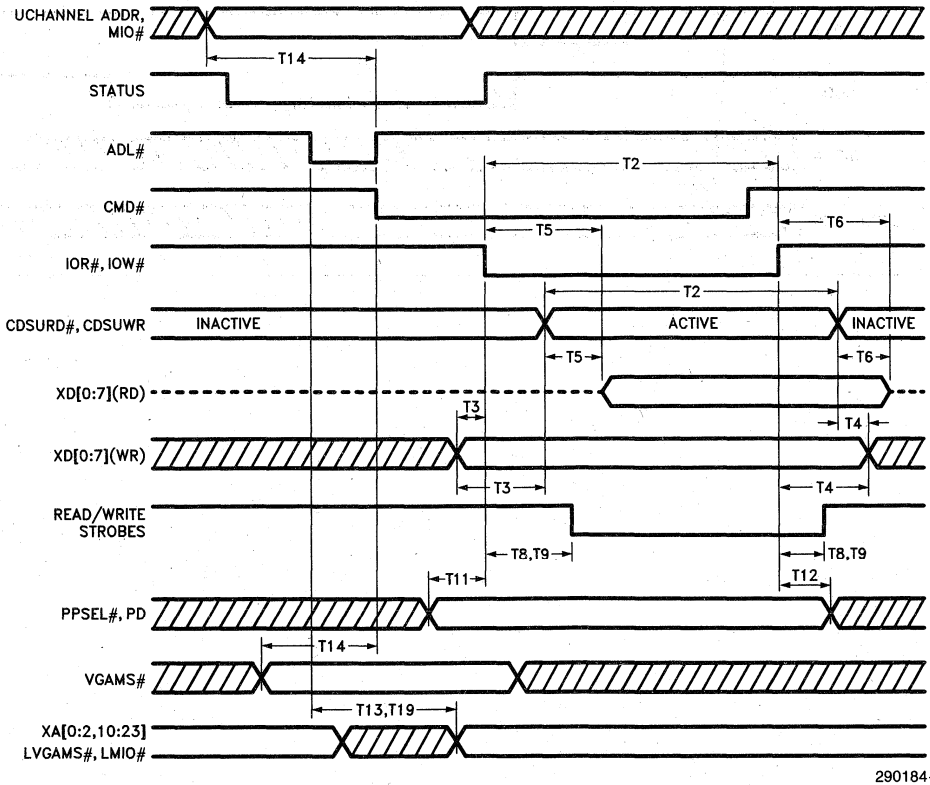
290184-4

Input Setup and Hold to another Input

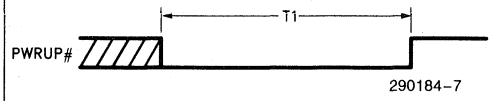


290184-5

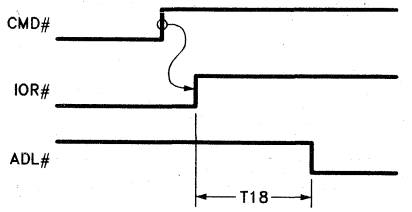
NOTE:
Input Waveforms have $T_R \leq 2.0$ ns from 0.8V to 2.0V.



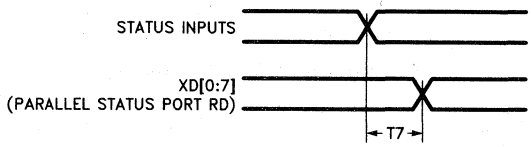
290184-6



290184-7

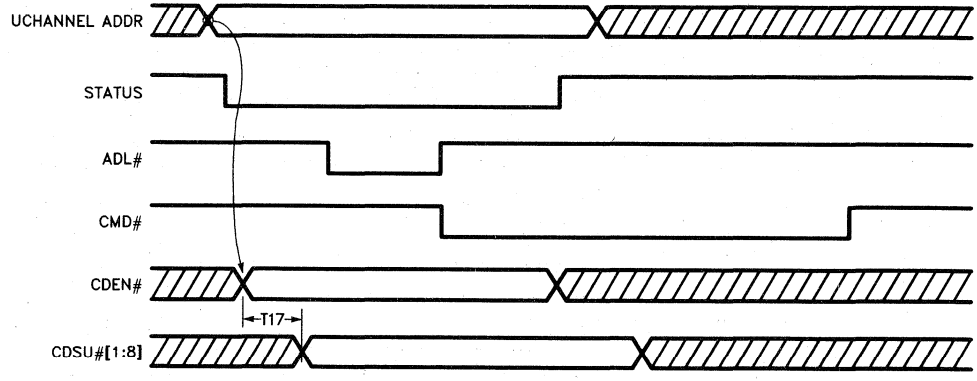


290184-8



290184-9

NOTE:
T7 is the flow through propagation delay, and thus assumes T5A, is not the limiting parameter.



290184-10

APPENDIX
82303 INTERNAL LOGIC DIAGRAMS

These logic diagrams are provided to aid in understanding the basic functionality of the 82303, and should not be used to estimate signal loading, propagation delays, or any other timing behavior.

The clocked latches in the diagrams are functionally equivalent to 7474 type TTL latches. The transparent latches are equivalent to 74373 type TTL latches except that the gate input is active low rather than active high.

The truth table for the combinatorial PAL is as follows:

RD

PSEL#	IOR#	XA0	XA1	PPDRD#	PPCRD#	PPSRD#
0	0	0	0	0	1	1
0	0	0	1	1	0	1
0	0	1	0	1	1	0

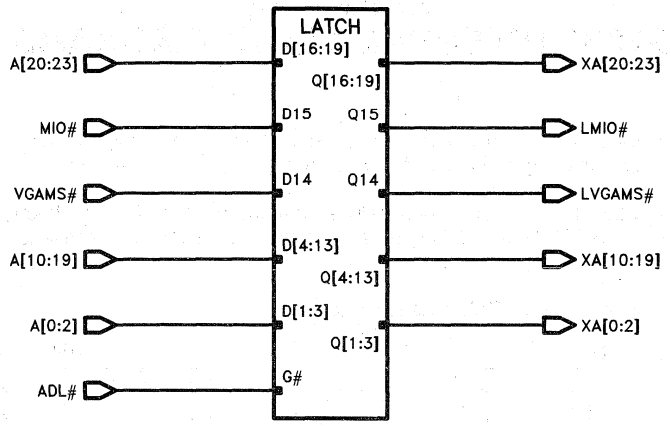
SELPOS	IOR#	XA2	XA1	XA0	POS1RD#	POS1RD#
1	0	0	0	1	0	1
1	0	0	1	1	1	0

WR

PSEL#	IOW#	XA0	XA1	PPDWR#	PPCWR#
0	0	0	0	0	1
0	0	0	1	1	0

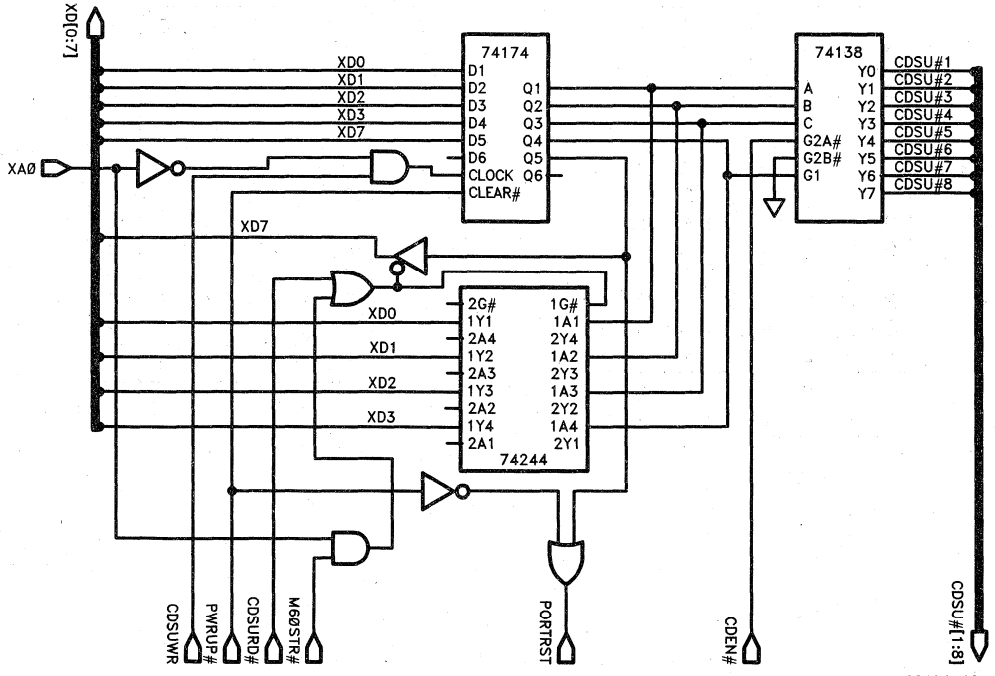
SELPOS	IOW#	XA2	XA1	XA0	POS103WR#
1	0	0	1	1	0

Address Latch
Transparent Latch × 19

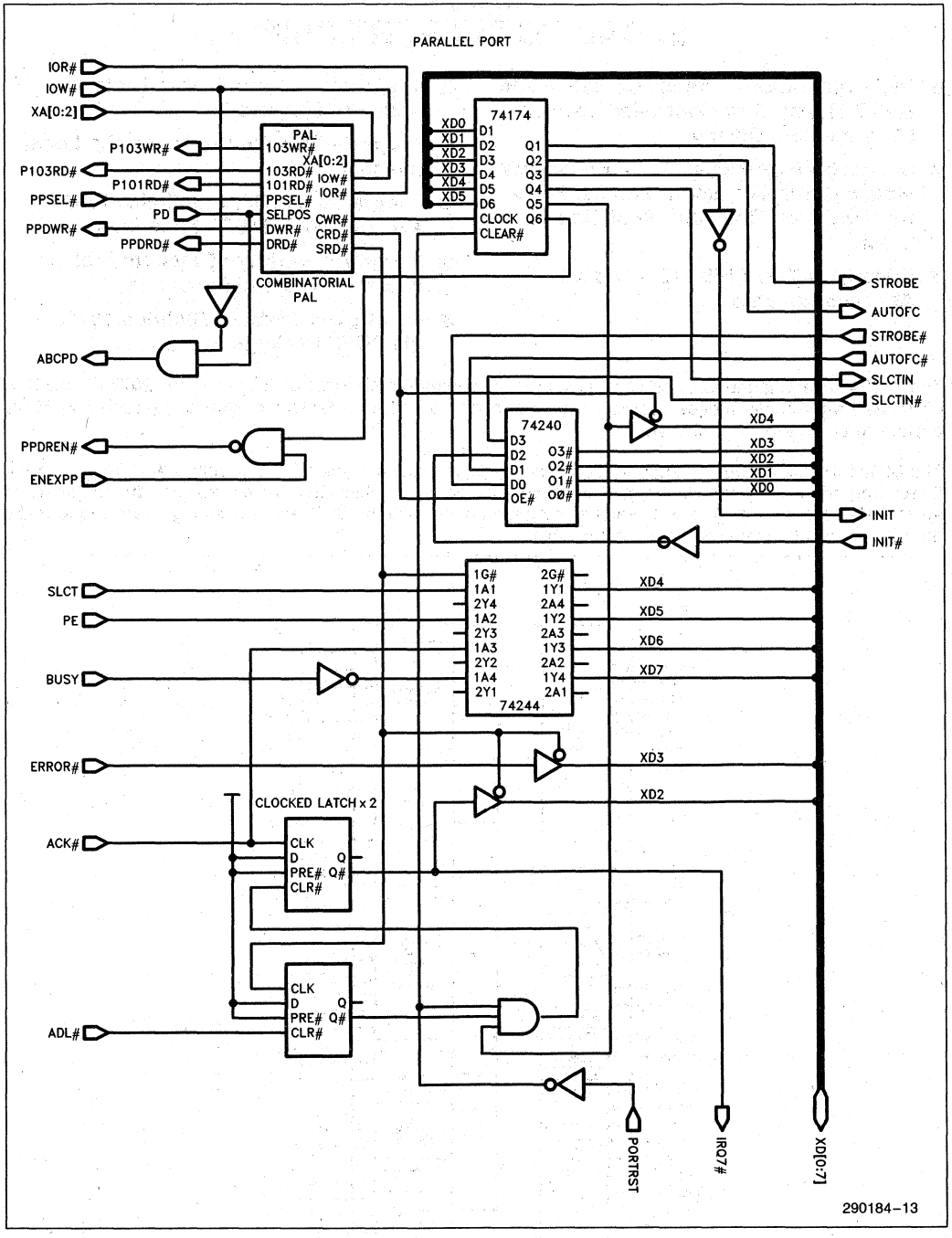


290184-11

Card Setup Port



290184-12

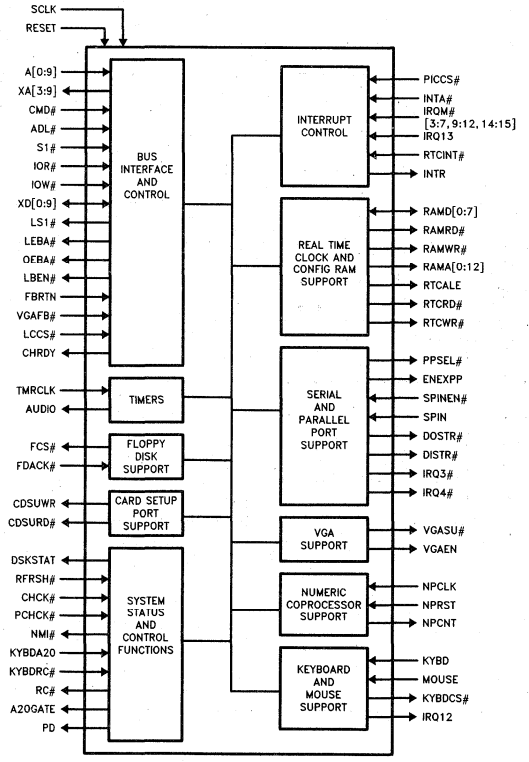

PARALLEL PORT
IOR#
IOW#
XA[0:2]
P103WR#
P103RD#
P101RD#
PPSEL#
PD
PPDWR#
PPDRD#
PAL
103WR#
XA[0:2]
103RD#
101RD#
PPSEL#
SELPOS
DWR#
DRD#
IOW#
IOR#
CWR#
CRD#
SRD#
COMBINATORIAL PAL
ABCPD
PPDREN#
ENEXPP
74174
XD0
XD1
XD2
XD3
XD4
XD5
D1
D2
D3
D4
D5
D6
CLOCK
CLEAR#
Q1
Q2
Q3
Q4
Q5
Q6
74240
D3
D2
D1
D0
OE#
O3#
O2#
O1#
O0#
1G#
1A1
2Y4
1A2
2Y3
1A3
2Y2
1A4
2Y1
2G#
1Y1
2A4
1Y2
2A3
1Y3
2A2
1Y4
2A1
74244
SLCT
PE
BUSY
ERROR#
ACK#
ADL#
CLOCKED LATCH x 2
CLK
D
PRE#
CLR#
Q
Q#
STROBE
AUTOFC
STROBE#
AUTOFC#
SLCTIN
SLCTIN#
INIT
INIT#
PORTRST
IRQ7#
XD[0:7]
290184-13

82304
LOCAL I/O SUPPORT CHIP

- **High-Integration—The 82304, 82303 and 82077 Floppy Disk Controller Replace 50 IC's in IBM Design**
- **Supports I/O Peripherals . . . Keyboard/ Mouse Controller, Serial/Parallel Ports, Configuration RAM, and Real Time Clock**
- **Integrates Two 8259 PIC's and All Associated Logic**
- **Integrates Programmable Timer Counters 0, 2 and 3**
- **Supports VGA Controller on the Local Channel**
- **Integrates the OS/2 Optimized HOT A20 and HOT RESET Functions**
- **Integrates Variety of System Status/ Control Ports and Functions**
- **Low Power CHMOS Technology/132-Pin PQFP Package**

The 82304 Local Channel Support Chip, along with its companion chip (the 82303) and the 82077 Floppy Disk Controller, significantly reduce system cost, design effort, and form factor constraints by replacing 50 IC devices in an equivalent IBM system.

The 82304 integrates logic to support local bus I/O peripherals and the VGA Controller. Also integrated are three programmable timer/counters, two "8259-like" programmable interrupt controllers, and a variety of system status/control ports and functions. Integrated along with the 8259 PIC's is all logic required to make the PIC's Microchannel architecture compatible.



290185-1

September 1989
Order Number: 290185-002

INTRODUCTION

The 82304 is a high integration device intended for Microchannel compatible system designs. It essentially integrates the 82306 Local Channel Support chip, two 8259 Programmable Interrupt Controllers, and a wide assortment of TTL circuitry. The 82304, in conjunction with its sister chip the 82303 and the 82077 Floppy Disk Controller, replaces approximately 50 IC devices in an equivalent IBM system. Included as an appendix to this data sheet is a functional logic diagram of the 82304 that should facilitate understanding of the part. Note that the 82304, 82303 and 82077 integrate a variety of system ports. For programming and register level details, please refer to the IBM Technical Reference Manual, 82077 data sheet, and 8259A data sheet.

BUS INTERFACE AND CONTROL

The Bus Interface and Control unit interfaces the 82304 to the Microchannel and peripheral busses. It inputs the unlatched Microchannel address, latches it for internal use, and makes the latched version available externally for other peripheral bus resources. It also provides signals to control an external 74F543 latching data transceiver that sits between the Microchannel and peripheral data busses. The bus interface unit also provides functions such as cycle extension on behalf of slower peripherals, and support of the Microchannel architecture's system feedback function.

SYSTEM TIMERS

The 82304 integrates the timers required for multitask time slice interrupt (timer 0), audio tone generation (timer 2), and "watch-dog" function (timer 3). These timers are accessed via ports 40, 42, 43, 44, and 47H.

FLOPPY DISK SUPPORT

The 82304 provides the decode signal required by the 82077 Floppy Disk Controller. The decode addresses ports 3F0–3F7H. The 82304 also inputs the 82077's DMA acknowledge in support of the system feedback function.

VGA SUPPORT

The 82304 supports the VGA setup and enable/disable functions. Specifically, bit 5 of integrated port 94H is used to put the VGA into setup mode, and this mode is reflected to the VGA on the 82304's VGASU# pin. Also, the 82304 integrates bit 0 of port 3C3H, which is used to enable/disable the system board VGA subsystem as indicated by the 82304's VGAEN output.

CARD SETUP PORT SUPPORT

The 82304 provides decoded read/write strobes for ports 96–97H. Port 96H is the card setup port, which is integrated on the 82303 chip. Port 97H is currently reserved by IBM.

INTERRUPT CONTROL

The 82304 integrates two 8259 Programmable Interrupt Controllers, and all additional logic required to make these interrupt controllers Microchannel architecture compatible. Specifically, the Microchannel definition requires that interrupts be active low and level sensitive. This allows a wire-OR system implementation. Integrated logic includes inverters for incoming interrupts, as the 8259 treats level sensitive interrupts as active high. Additionally, logic to inhibit the 8259's from being programmed in edge-triggered mode is integrated.

REAL TIME CLOCK AND CONFIGURATION RAM SUPPORT

The 82304 integrates all logic required to support an external battery backed up real time clock chip and static RAM. Note that while the IBM implementation supports a 2K RAM, the 82304 makes provision to support either a 2K or 8K RAM. The real time clock is accessed via ports 70–71H, while the RAM is accessed via ports 74–76H. (RAM data is accessed via port 76H, while ports 74H and 75H serve as an indirect address latch for the RAM.) The 82304 also integrates the logic required to enforce the Microchannel architecture's password security function. Specifically, writes to port 70H are monitored. If a write to 70H is attempting to access offsets 38–3FH in the real time clock chip's onboard RAM, and if the security bit in 92H indicates that these offsets are off limits, then no address latch signal is generated to the real time clock chip.

SERIAL AND PARALLEL PORT SUPPORT

The 82304 provides various functions in support of an external serial port chip (the 16550A), and a par-

allel port (integrated on the 82303 chip). The 82304 provides decoded read/writes strobes for the serial port chip, as well as converting a serial port interrupt into either IRQ3# or IRQ4#, depending on whether the serial port is configured as COMM1 or COMM2. When configured as COMM1, the serial port is decoded at ports 3F8–3FFH. As COMM2, the port is decoded from 2F8–2FFH. Configuration is done via the integrated system setup port 102H.

The 82304 generates a parallel port chip select that maps to LPT1, LPT2, or LPT3, depending on how system setup port 102H is programmed. As LPT1, the parallel port is decoded at ports 3BC, 3BD, 3BE, and 3BFH. LPT2 maps to ports 378, 379, 37A, and 37BH. LPT3 maps to 278, 279, 27A, and 27BH. The 82304 also generates a signal that indicates whether the parallel port is to operate in its normal output-only mode, or its "extended" bi-directional mode. The mode is selected via system setup port 102H.

NUMERIC COPROCESSOR SUPPORT

The 82307 DMA Controller, in response to a software command, issues a pulse to reset the 80387 or 80387SX Numeric Coprocessor. The 82304 inputs this pulse and effectively stretches it out to insure that the 80387 reset input pulse is long enough to meet its internal reset requirements. (Note that the 80387 reset pulse out of the 82304 must be externally synchronized to the 80387 clock so as to convey the system phase to the 80387.) The 82304 also manipulates CHRDY to extend the bus cycle that initiates the reset so as to tie up the CPU until the 80387's reset and initialization requirements are met.

KEYBOARD AND MOUSE SUPPORT

The 82304 provides a chip select for the 8742 Keyboard Controller. This decode maps to ports 60 and 64H. The 82304 also integrates the logic required to both latch and subsequently clear keyboard and mouse interrupts.

SYSTEM STATUS AND CONTROL FUNCTIONS

The 82304 integrates a variety of system status and control functions and ports. Integrated ports include:

Port 61H System Control Port B
Port 92H System Control Port A
Port 91H Card Selected Feedback Register
Port 94H System Board Setup Port
Port 102H System Board POS Port
Port 70H NMI Enable (Write Only)

The functions and register level details of these ports are documented in the IBM Technical Reference.

82304 LOCAL CHANNEL SUPPORT CHIP PIN DEFINITIONS

Symbol	Pin No.	Type	Description
SCLK	50	I	**INPUT CLOCK:** Tied to same clock as host CPU.
RESET	64	I	**SYNCHRONIZED POWER-UP RESET:** Resets 82304 and synchronizes internal clock to system phase.
A[0:9]	85–81, 78–74	I	**MICROCHANNEL ADDRESS:** Address Lines are internally latched.
XA[3:9]	73–67	O	**PERIPHERAL BUS ADDRESS:** These are latched versions of the Microchannel address.
CMD#	97	I	**MICROCHANNEL CMD# INPUT**
ADL#	100	I	**MICROCHANNEL ADL# INPUT**
S1#	1	I	**MICROCHANNEL S1# INPUT**
IOR#, IOW#	96, 95	I	**82304 READ AND WRITE STROBES**
XD[0:7]	94–90, 88–86	I/O	**BI-DIRECTIONAL DATA BUS**
LS1#	131	O	**LATCHED VERSION OF MICROCHANNEL S1# INPUT**
LEBA#, OEBA#	130, 129	O	**EXTERNAL 74543 DATA BUFFER CONTROL SIGNALS:** These signals control data timing on the peripheral data bus.
LBEN#	52	O	**LOCAL (PERIPHERAL) BUS ENABLE:** The 82304 generates this in response to address decodes of peripheral bus ports. It is typically "OR"ed with other system qualifiers to enable the peripheral bus data buffer.

82304 LOCAL CHANNEL SUPPORT CHIP PIN DEFINITIONS (Continued)

Signal Name	Pin No.	I/O	Description
FBRTN	41	I	**SYSTEM FEEDBACK:** This input receives the OR of the system feedback signals of the Microchannel slots. It is internally latched and "OR"ed with other feedback sources, and the result made available via a Port 91H read (Bit 0).
VGAFB#	42	I	**VGA FEEDBACK.**
LCCS#	37	I	**LOCAL CHANNEL CHIP SELECT:** Activated for the I/O address range 0–3FFH (CPU or DMA master) or 100–3FFH (Microchannel Master). LCCS# is internally latched.
CHRDY	5	O	**CHANNEL READY:** The 82304 deasserts CHRDY to extend accesses to certain peripheral bus resources, specifically the keyboard controller, real time clock and serial port. Also, CHRDY is used to tie up the CPU during numeric coprocessor resets.
TMRCLK	45	I	**1.193 MHz CLOCK INPUT:** Drives clock inputs of system timers 0 and 2.
AUDIO	128	O	**OUTPUT OF SYSTEM TIMER 2 GATED BY BIT 1 OF PORT 61H:** It drives the Microchannel audio sum node.
FCS#	46	O	**FLOPPY DISK CONTROLLER (82077) CHIP SELECT:** Responds to I/O range 3F0–3F7H.
FDACK#	49	I	**FLOPPY DISK CONTROL DMA ACKNOWLEDGE:** Internally latched and "OR"ed with other system feedback sources.
VGASU#	44	O	**VGA SETUP:** Puts the VGA into setup mode when active, according to Bit 5 of Port 94H.
VGAEN	43	O	**VGA ENABLE:** Enables/Disables motherboard VGA according to Bit 0 of Port 3C3H.
CDSUWR	126	O	**CARD SETUP WRITE STROBE:** Active High command generated during writes to Port 96–97H.
CDSURD#	125	O	**CARD SETUP READ STROBE:** Active low command generated during reads from Port 96–97H.
DSKSTAT	127	O	**FIXED DISK STATUS:** Controls the fixed disk activity light. It is active when either Bit 6 or Bit 7 of Port 92H is set.
RFRSH#	51	I	**REFRESH CYCLE INDICATOR:** Diagnostics can monitor refresh activity via Bit 4 of Port 61H.
CHCK#	54	I	**MICROCHANNEL CHECK INDICATOR:** Used to report adapter errors.
PCHCK#	55	I	**DRAM PARITY ERROR:** Driven in response to motherboard memory parity errors.
NMI#	53	O	**NON-MASKABLE INTERRUPT REQUEST TO CPU:** This acts as an open drain output that allows for an external wire "OR" with other NMI sources.
KYBDA20	60	I	**A20 GATE SIGNAL OUT OF THE KEYBOARD CONTROLLER:** Internally "OR"ed with the alternate A20 switch incorporated in Bit 1 of Port 92H.
KYBDRC#	58	I	**CPU RESET SIGNAL OUT OF THE KEYBOARD CONTROLLER:** It is internally "OR"ed with the alternate reset function of Bit 0 of Port 92H.
RC#	59	O	**RESET CPU:** Resets CPU via Port 92H (Bit 0) or the KYBDRC# input.
A20GATE	57	O	**A20 GATE SIGNAL:** The "OR" of Bit 1 of Port 92H and the KYBDA20 input.

82304 LOCAL CHANNEL SUPPORT CHIP PIN DEFINITIONS (Continued)

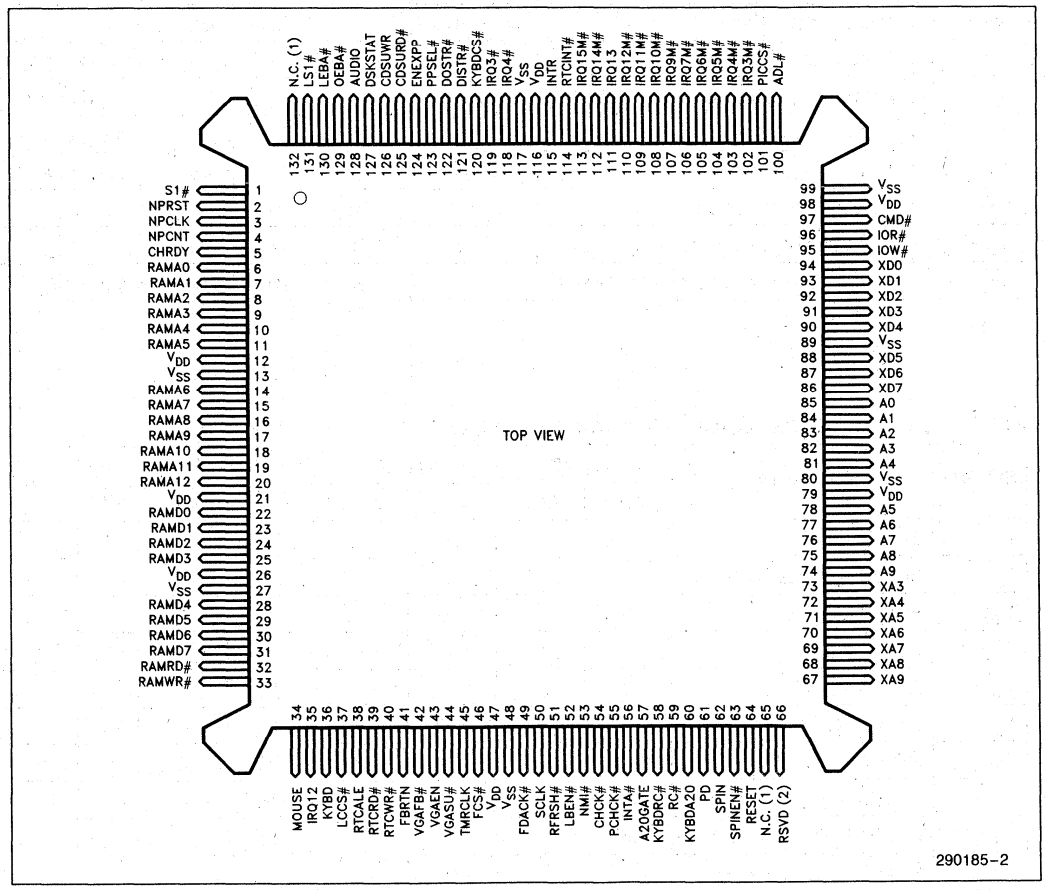
Signal Name	Pin No.	I/O	Description
PD	61	O	**POS DECODE:** An active high decode of system board setup ports 100, 101, 103–107H. (Port 102H is integrated on the 82304.)
PICCS#	101	I	**CHIP SELECT FOR THE INTEGRATED 8259 PROGRAMMABLE INTERRUPT CONTROLLERS (PIC)**
INTA#	56	I	**INTERRUPT ACKNOWLEDGE:** Generated by the bus controller during interrupt acknowledge cycles.
IRQM# [3:7, 9:12, 14:15]	102–110, 112–113	I	**MICROCHANNEL INTERRUPT INPUTS.**
IRQ13	111	I	**INTERRUPT INPUT USED TO REPORT NUMERIC COPROCESSOR ERRORS.**
RTCINT#	114	I	**INTERRUPT INPUT FROM REAL TIME CLOCK.**
INTR	115	O	**MASKABLE INTERRUPT REQUEST TO CPU.**
RAMD[0:7]	22–25, 28–31	I/O	**REAL TIME CLOCK AND CONFIGURATION RAM DATA BUS.**
RAMRD#, RAMWR#	32, 33	O	**READ/WRITE STROBES TO CONFIGURATION RAM:** Generated during accesses to Port 76H.
RAMA[0:12]	6–11, 14–20	O	**CONFIGURATION RAM ADDRESS BUS:** Internal RAM address latches are written to via Ports 74–75H.
RTCALE	38	O	**REAL TIME CLOCK ADDRESS LATCH ENABLE.**
RTCRD#, RTCWR#	39, 40	O	**REAL TIME CLOCK READ/WRITE STROBES.**
PPSEL#	123	O	**PARALLEL PORT CHIP SELECT:** Maps to LPT1, LPT2, or LPT3 as controlled by Bits 5 and 6 of system board setup Port 102H.
ENEXPP	124	O	**PARALLEL PORT EXTENDED MODE ENABLE:** This mode is controlled via bit 7 of system board setup Port 102H.
SPINEN#	63	I	**SERIAL PORT INTERRUPT ENABLE.**
SPIN	62	I	**SERIAL PORT INTERRUPT.**
IRQ3#, IRQ4#	119, 118	O	**SERIAL PORT INTERRUPT:** Configured to either COMM1 (IRQ4#) or COMM2 (IRQ3#). Selection is done via Bit 3 of system board setup Port 102H.
DOSTR#, DISTR#	122, 121	O	**WRITE/READ STROBES FOR SERIAL PORT.**
NPCLK	3	I	**CLOCK FOR NUMERIC PROCESSOR RESET PULSE STRETCHER.**
NPRST	2	I	**NUMERIC PROCESSOR RESET REQUEST INPUT**
NPCNT	4	O	**NUMERIC PROCESSOR COUNT:** Numeric processor reset signal typically synchronized externally and fed to 80387 or 80387SX.
KYBD	36	I	**INTERRUPT REQUEST INPUT FROM KEYBOARD CONTROLLER:** It is internally latched, and then subsequently cleared by a keyboard controller read.

82304 LOCAL CHANNEL SUPPORT CHIP PIN DEFINITIONS (Continued)

Signal Name	Pin No.	I/O	Description
MOUSE	34	I	**INTERRUPT REQUEST INPUT FROM KEYBOARD CONTROLLER'S MOUSE PORT:** It is internally latched and subsequently cleared by a keyboard controller read.
KYBDCS#	120	O	**KEYBOARD CONTROLLER CHIP SELECT.**
IRQ12	35	O	**LATCHED VERSION OF MOUSE INPUT INTERRUPT REQUEST.**
V_{DD}	12, 21, 26, 47, 79, 98, 116		**POWER.**
V_{SS}	13, 27, 48, 80, 89, 99, 117		**GROUND.**
NC	65, 132		**NO CONNECT.**
RSVD	66		**RESERVED.**

1

82304 132-Pin PQFP Pinout



NOTES:
1. N.C. pins must be left not connected.
2. This pin is reserved . . . must be tied to ground in system.

82304 Parametrics

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

Absolute Maximum Ratings*

Case Temperature Under Bias .... −40°C to +85°C
Storage Temperature ..... −65°C to +150°C
Voltage to any Pin with Respect to Ground .... −0.3V to +(V_{CC} +0.3)V
DC Supply Voltage (V_{CC}) ..... −0.3V to +7.0V
DC Input Current ..... ±10 mA

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. Electrical Characteristics T_C = 0°C to +70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage		0.8	V	
V_{IH}	Input High Voltage	2.0		V	
V_{IL}	Input Low Voltage		0.8	V	SCLK
V_{IH}	Input High Voltage	V_{CC}−0.8		V	SCLK
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 4 mA (Note 1)
V_{OH}	Output High Voltage	2.4		V	I_{OH} = 4 mA (Note 1)
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2 mA (Note 2)
V_{OH}	Output High Voltage	2.4		V	I_{OH} = 2 mA (Note 2)
I_{CC}	Power Supply Current		180	mA	No DC Loads
I_{LI}	Input Leakage Current		±10	μA	$V_{SS} < V_{IN} < V_{CC}$
I_{OZ}	TRI-STATE Output Leakage Current		±10	μA	$V_{SS} < V_{OUT} < V_{CC}$

NOTES:
1. DSKSTAT, XA[3:9], XD[0:7].
2. All outputs other than those listed in Note 1.

82304 A.C. Electrical Specifications T_C = 0°C to +70°C, V_{CC} = 5V ±10%

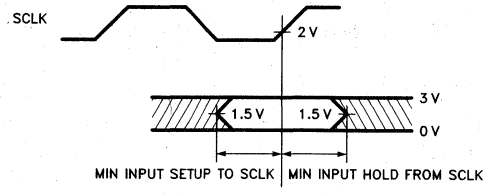
Symbol	Parameter	KIT-16		KIT-20		KIT-25		C_L (pF)	Notes
		Min	Max	Min	Max	Min	Max		
T_1	SCLK Period	31.25		25		20			
T_{2A}	SCLK High/Low Time	12		10		8			
T_{2B}	SCLK High/Low Time	8		6.5		6			
T_3	Reset Setup	10		10		10			
T_4	Reset Hold	3		3		3			
T_5	Reset Pulse Width	500		500		500			
T_6	RC# Pulse Width	75	150	75	150	75	150	50	
T_7	TMRCLK High/Low Time	300		300		300			
T_8	PICCS#, FDACK# Setup	30		30		30			
T_9	PICCS#, FDACK# Hold	0		0		0			
T_{10}	IOR#, IOW#, INTA# Pulse Width	200		170		170			

82304 A.C. Electrical Specifications T_C = 0°C to +70°C, V_{CC} = 5V ±10% (Continued)

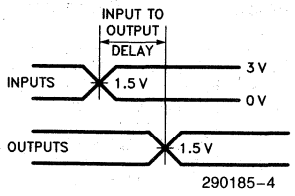
Symbol	Parameter	KIT-16		KIT-20		KIT-25		C_L (pF)	Notes
		Min	Max	Min	Max	Min	Max		
T_{11}	Write Data Setup	25		25		25			
T_{12}	Write Data Hold	20		20		20			
T_{13}	Read Data Valid Delay	0	90	0	90	0	90	100	
T_{14}	Read Data Float Delay	0		0		0		100	
T_{15}	RAMD[0:7] to XD[0:7] Delay		36		36		36	100	
T_{17}	CHRDY Delay	0	80	0	80	0	80	25	
T_{18}	CHRDY Inactive Pulse Width	280		280		230		25	5
T_{19}	Address Decode Delays from ADL# ↓	0	62	0	62	0	62	50	1
T_{20A}	Write Strobe Delays from IOW# ↓	0	40	0	40	0	40	50	2
T_{20B}	CDSUWR, DOSTR# Delays from IOW# ↑	0	35	0	35	0	35	50	
T_{20C}	RTCWR#, RAMWR# Delay from CMD# ↑	0	33	0	33	0	33	50	6
$T_{21A,B}$	Read Strobe Delays	0	40	0	40	0	40	50	3
T_{22}	RTCALE Min Pulse Width	120		120		110		50	3
T_{23}	XA[3:9] Delay from ADL# ↓		35		35		35	100	5
T_{24}	S1#, LCCS#, A[0:9] Setup to ADL# ↑	30		30		30			
T_{25}	FBRTN Setup to CMD# ↓	15		15		15			
T_{26}	VGAFB# Setup to CMD# ↑	15		15		15			
T_{27}	LEBA# Delay from CMD#		26		26		26	25	
T_{28}	OEBA# Delay from CMD# ↓		30		30		30	25	
T_{30}	CHRDY ↓ Delay		38		38		38	25	4
T_{31}	NPCNT ↑ Delay	128 NPCLKS		128 NPCLKS		128 NPCLKS		50	5
T_{32}	CHRDY ↑ Delay	192 NPCLKS		192 NPCLKS		192 NPCLKS		25	5
T_{33}	NPCLK High/Low Time	12		12		12			
T_{34}	LS1# Delay from ADL# ↓		35		35		35	50	

NOTES:
1. Address decodes include FCS#, PPSEL#, KYBDCS#, PD and LBEN#.
2. Write strobes include RTCWR#, CDSUWR#, DOSTR#, and RAMWR#.
3. Read strobes include RTCRD#, CDSURD#, DISTR# and RAMRD#.
4. From later or NPRST ↑ or CMD# ↓.
5. Functional Specification . . . Not tested.
6. CMD# ↑ causes RTCWR# ↑ and RAMWR# ↑, while IOW# ↑ causes RAMD[0:7] to float. The 82304 insures that CMD#-to-RAMWR#/RTCWR# is at least 5 ns faster than IOW# to RAMD[0:7] float, assuming loading on RAMD[0:7] is greater than or equal to loading on RAMWR# or RTCWR#. This provides a minimum of 5 ns data hold time for the real time clock and SRAM, assuming CMD# and IOW# reach the 82304 at the same instant. Typically, more than 5 ns is provided, since IOW# is generated from, and thus delayed from CMD#.
7. Specification applies to software reset generated via port 92H.

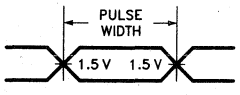
82304 DRIVE LEVELS/MEASUREMENT POINTS FOR A.C. SPECIFICATIONS



290185-3

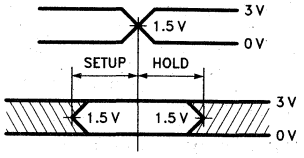


290185-4



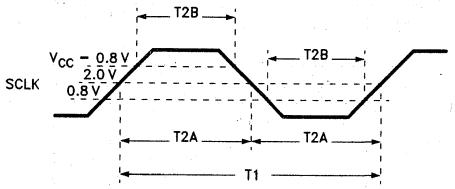
290185-5

Input/Output Pulse Widths,
Input Clock (Other than SCLK)
HIGH/LOW Times

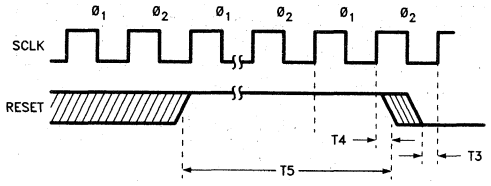


290185-6

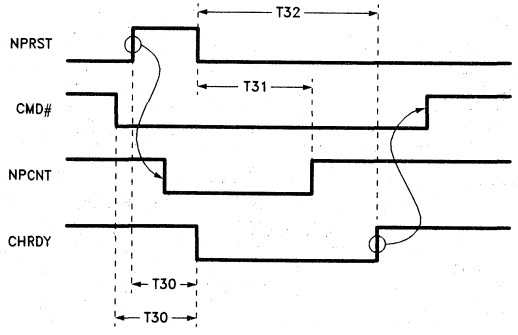
Input Setup and Hold to
another Input (Other than SCLK)



290185-7

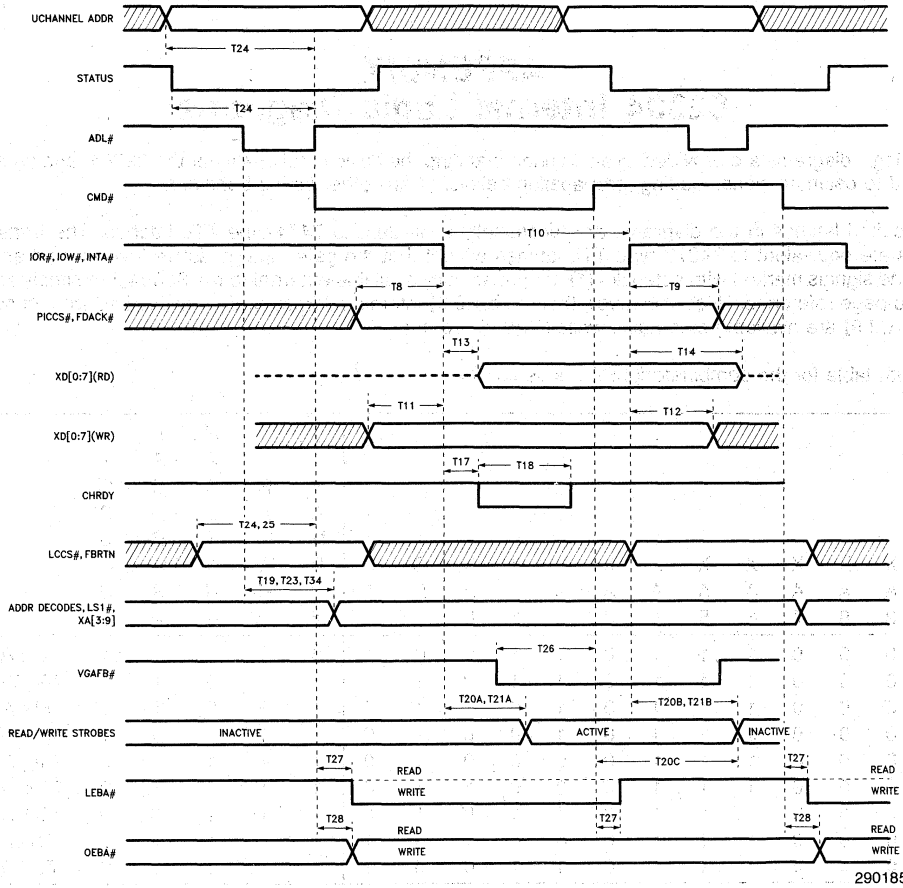


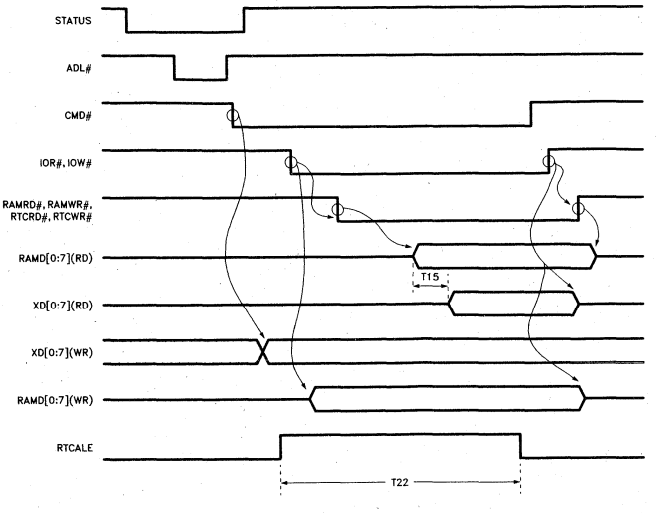
290185-8



290185-9

NOTE:
Input Waveforms have $T_R \leq 2.0$ ns from 0.8V to 2.0V.


UCHANNEL ADDR
T24
STATUS
T24
ADL#
CMD#
T10
IOR#, IOW#, INTA#
T8
T9
PICCS#, FDACK#
T13
T14
XD[0:7](RD)
T11
T12
XD[0:7](WR)
T17
T18
CHRDY
T24, 25
LCCS#, FBRTN
T19, T23, T34
ADDR DECODES, LS1#, XA[3:9]
T26
VGAFB#
T20A, T21A
T20B, T21B
READ/WRITE STROBES
INACTIVE
ACTIVE
INACTIVE
T27
T20C
T27
READ
WRITE
LEBA#
T28
T27
T28
OEBA#
290185-10


STATUS
ADL#
CMD#
IOR#, IOW#
RAMRD#, RAMWR#, RTCRD#, RTCWR#
RAMD[0:7](RD)
T15
XD[0:7](RD)
XD[0:7](WR)
RAMD[0:7](WR)
RTCALE
T22
290185-11

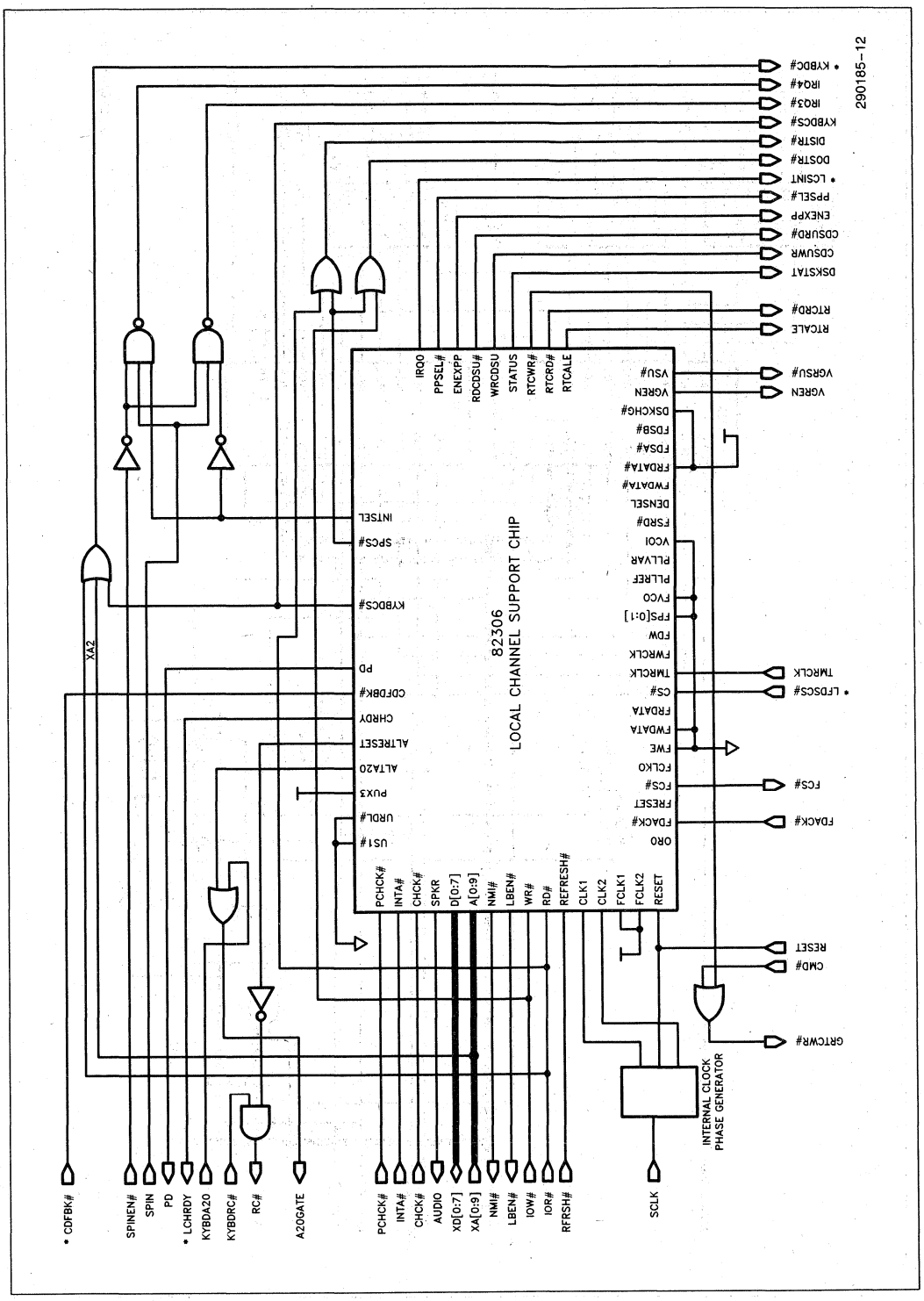
APPENDIX
82304 Internal Logic Diagrams

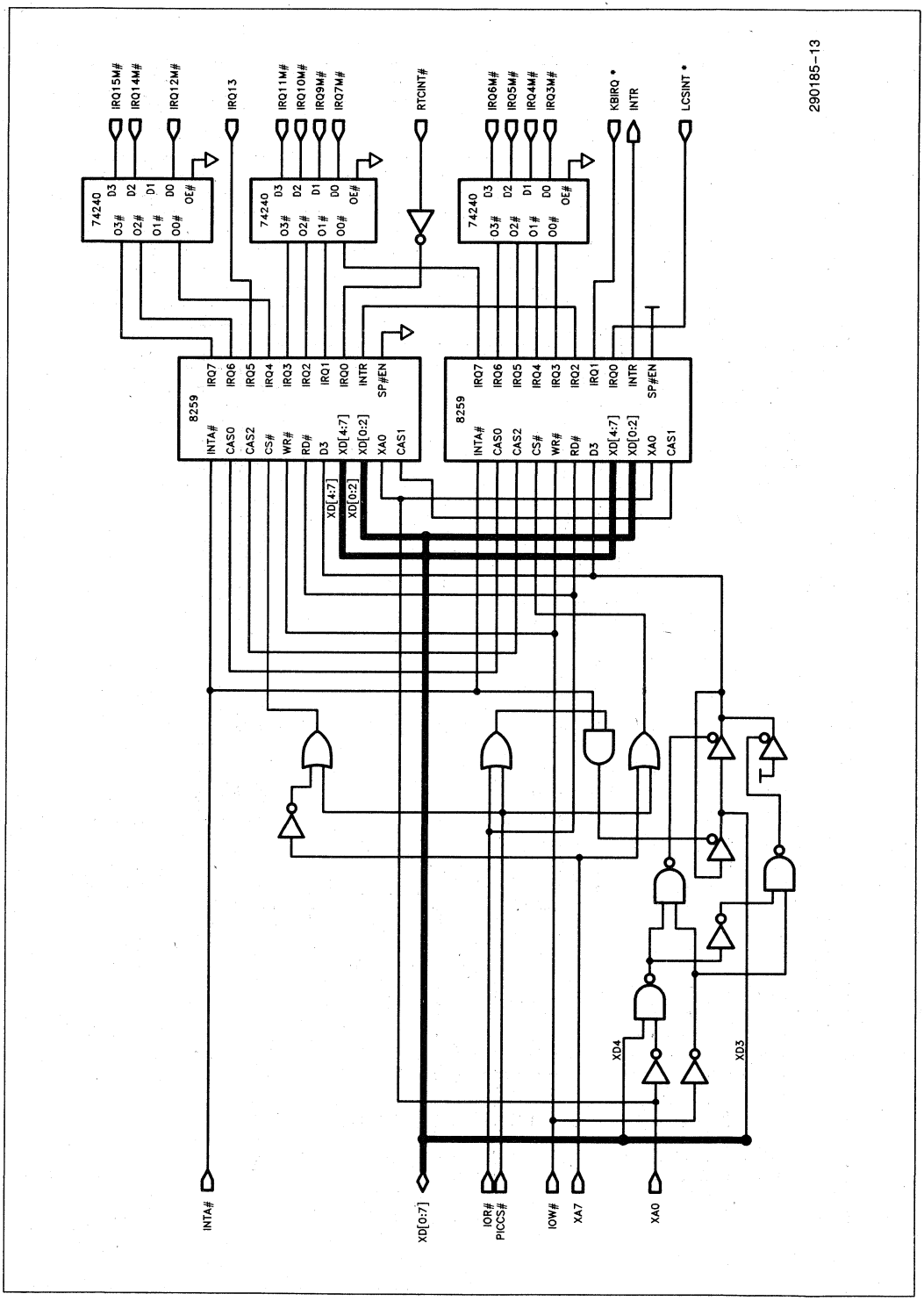
These logic diagrams are provided to aid in understanding the basic functionality of the 82304, and should not be used to estimate signal loading, propagation delays, or any other timing behavior.

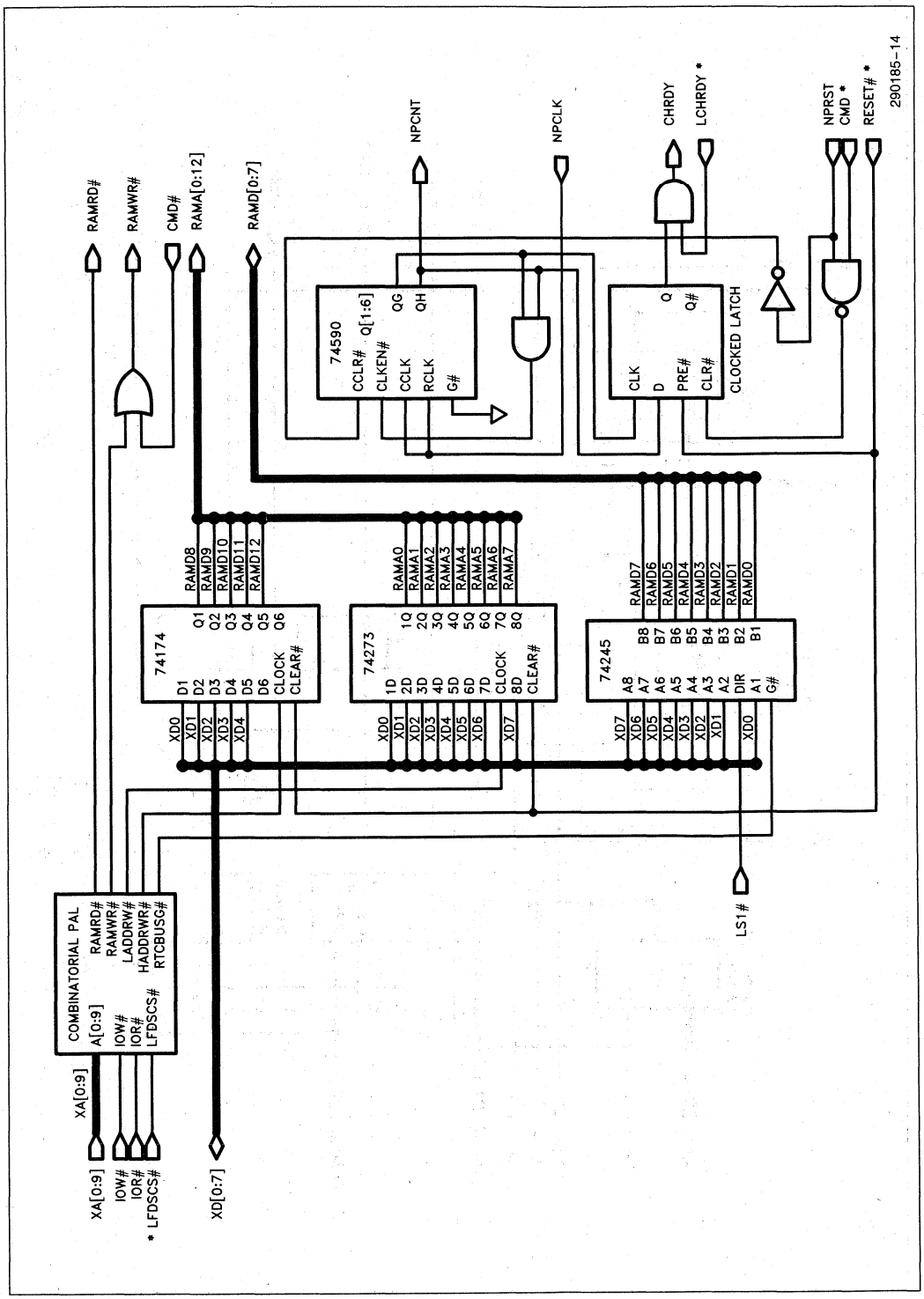
The clocked latches in the diagrams are functionally equivalent to 7474 type TTL latches. The transparent latches are equivalent to 74373 type TTL latches except that the gate input is active low rather than active high. The signals marked with asterisks (*) are not actually available external to the 82304, but simply serve as page-to-page references. Note however, that the XA[0:9] internal address bus is not marked with an asterisk. Only XA[3:9] are available externally, while XA[0:2] are not.

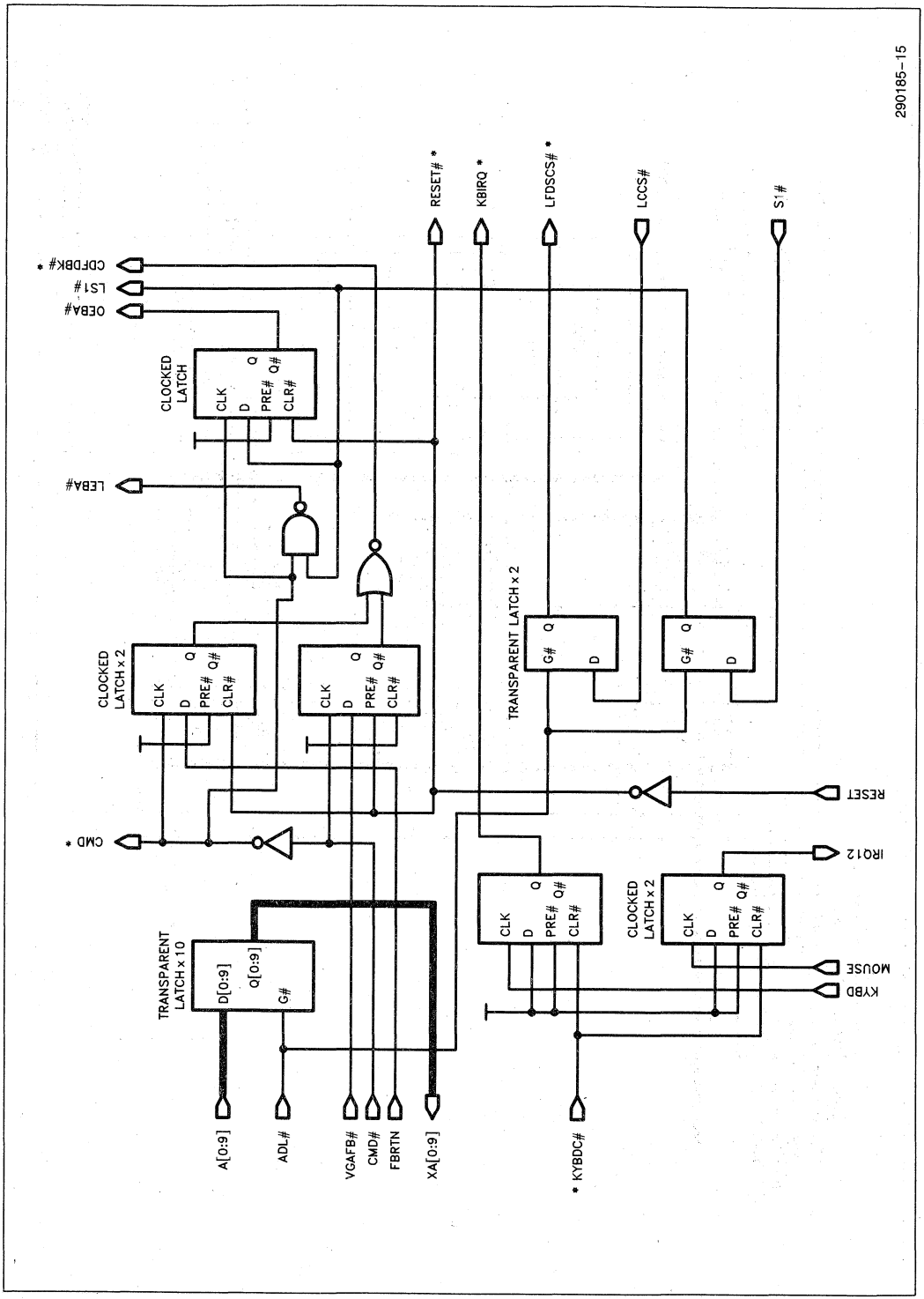
The truth table for the combinatorial PAL is as follows:

LFDSCS#	XA9	XA8	XA7	XA6	XA5	XA4	XA3	XA2	XA1	XA0	IORC#	IOWC#	RAMRD#	RAMWR#	HADRWR#	LADRWR#	RTCBUSG#	
0	0	0	0	1	1	1	0	1	1	0	0	1	0	1	1	1	1	76H READ
0	0	0	0	1	1	1	0	1	1	0	1	0	1	0	1	1	1	76H WRITE
0	0	0	0	1	1	1	0	1	0	1	1	0	1	1	0	1	1	75H WRITE
0	0	0	0	1	1	1	0	1	0	0	1	0	1	1	1	0	1	74H WRITE
0	0	0	0	1	1	1	0	0	0	0	1	0	1	1	1	1	0	70H WRITE
0	0	0	0	1	1	1	0	0	0	1	1	0	1	1	1	1	0	71H WRITE
0	0	0	0	1	1	1	0	1	1	0	1	0	1	1	1	1	0	76H WRITE
0	0	0	0	1	1	1	0	0	0	1	0	1	1	1	1	1	0	71H READ
0	0	0	0	1	1	1	0	1	1	0	0	1	1	1	1	1	0	76H READ


290185-12
82306
LOCAL CHANNEL SUPPORT CHIP
INTERNAL CLOCK PHASE GENERATOR
XA2
* KYBDC#
IRQ4#
IRQ3#
KYBDCS#
DISTR#
DOSTR#
* LCSINT
PPSEL#
ENEXPP
CDSURD#
CDSUWR
DSKSTAT
RTCRD#
RTCALE
VGRSU#
VGREN
TMRCLK
* LFDCS#
FCS#
FDACK#
RESET
CMD#
GRTCWR#
IRQO
PPSEL#
ENEXPP
RDCDSU#
WRCDSU
STATUS
RTCWR#
RTCRD#
RTCALE
VSU#
VGREN
DSKCHG#
FDSB#
FDSA#
FRDATA#
FWDATA#
DENSEL
FSRD#
VCOI
PLLVAR
PLLREF
FVCO
FPS[0:1]
FDW
FWRCLK
TMRCLK
CS#
FRDATA
FWDATA
FWE
FCLKO
FCS#
FRESET
FDACK#
ORO
INTSEL
SPCS#
KYBDCS#
PD
CDFDBK#
CHRDY
ALTRESET
ALTA20
PUX3
URDL#
US1#
PCHCK#
INTA#
CHCK#
SPKR
D[0:7]
A[0:9]
NMI#
LBEN#
WR#
RD#
REFRESH#
CLK1
CLK2
FCLK1
FCLK2
RESET
* CDFBK#
SPINEN#
SPIN
PD
* LCHRDY
KYBDA20
KYBDRC#
RC#
A20GATE
PCHCK#
INTA#
CHCK#
AUDIO
XD[0:7]
XA[0:9]
NMI#
LBEN#
IOW#
IOR#
RFRSH#
SCLK


290185-13
IRQ15M#
IRQ14M#
IRQ12M#
IRQ13
IRQ11M#
IRQ10M#
IRQ9M#
IRQ7M#
RTCINT#
IRQ6M#
IRQ5M#
IRQ4M#
IRQ3M#
KBIRQ *
INTR
LCSINT *
74240
D3
D2
D1
D0
OE#
O3#
O2#
O1#
O0#
8259
IRQ7
IRQ6
IRQ5
IRQ4
IRQ3
IRQ2
IRQ1
IRQ0
INTR
SP#EN
INTA#
CAS0
CAS2
CS#
WR#
RD#
D3
XD[4:7]
XD[0:2]
XA0
CAS1
XD4
XD3
INTA#
XD[0:7]
IOR#
PICCS#
IOW#
XA7
XA0


290185-14
RAMRD#
RAMWR#
CMD#
RAMA[0:12]
RAMD[0:7]
NPCNT
NPCLK
CHRDY
LCHRDY *
NPRST
CMD *
RESET# *
74590
CCLR#
CLKEN#
CCLK
RCLK
G#
Q[1:6]
QG
QH
CLOCKED LATCH
CLK
D
PRE#
CLR#
Q
Q#
74174
74273
74245
RAMD8
RAMD9
RAMD10
RAMD11
RAMD12
RAMA0
RAMA1
RAMA2
RAMA3
RAMA4
RAMA5
RAMA6
RAMA7
RAMD7
RAMD6
RAMD5
RAMD4
RAMD3
RAMD2
RAMD1
RAMD0
XD0
XD1
XD2
XD3
XD4
XD5
XD6
XD7
LS1 #
COMBINATORIAL PAL
A[0:9]
IOW#
IOR#
LFDSCS#
RAMRD#
RAMWR#
LADDRW#
HADDRWR#
RTCBUSG#
XA[0:9]
XD[0:7]
* LFDSCS#


290185-15
RESET# *
KBIRQ *
LFDSCS# *
LCCS#
S1#
CDFDBK# *
LS1#
OEBA#
CLOCKED LATCH
CLK
D
Q
PRE#
Q#
CLR#
LEBA#
CLOCKED LATCH x 2
TRANSPARENT LATCH x 2
G#
RESET
CMD *
IRQ12
MOUSE
KYBD
TRANSPARENT LATCH x 10
D[0:9]
Q[0:9]
A[0:9]
ADL#
VGAFB#
CMD#
FBRTN
XA[0:9]
* KYBDC#

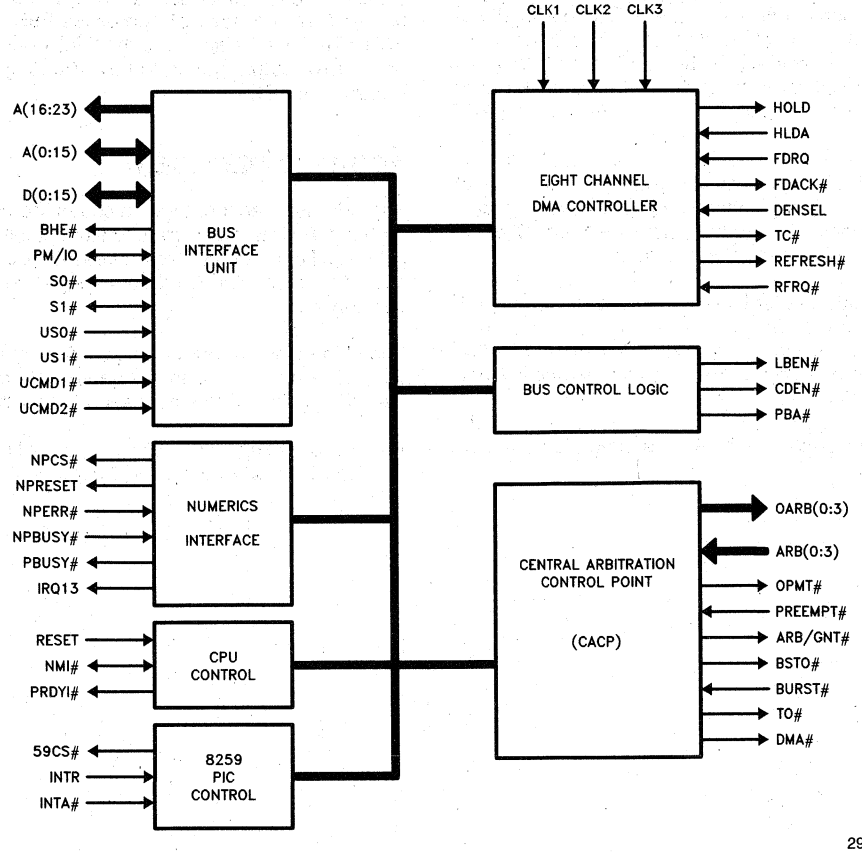
82307
DMA/Micro Channel ARBITRATION CONTROLLER

- **8 Channel DMA Controller (8/16-Bit)**
- **Integrated Central Arbitration Control Point**
- **Refresh Address Generation/Cycling**
- **Numerics Co-processor Interface**
- **Address Decoding**
  - **— Numeric Coprocessor**
  - **— Interrupt Controller**
  - **— POS Address Space for Expansion Slots**
- **Low Power CHMOS Technology**
- **132-Pin Plastic Quad Flat Pack Packaging**
  (See Packaging Spec., Order # 231369)

1

The 82307 DMA/Micro Channel Arbitration Controller is a register level implementation of the equivalent VLSI device in IBM Micro Channel systems. The Central Arbitration Control Point (CACP) as defined in the Micro Channel Architecture for bus arbitration is integrated in this VLSI device.

The 82307 also integrates the Address decoder logic for generating decodes for numeric coprocessor, Interrupt controllers and POS address space.



290186-1

August 1990
Order Number: 290186-002

DMA FUNCTION

The 82307 features eight 8/16-bit channels, 24-bit addressing capability, and operates in two-cycle transfer mode as defined in the Micro Channel architecture. The DMA controller owns the bus for both halves of the transfer cycle.

The DMA function in the 82307 also supports the motherboard Floppy Disk Controller. Upon receiving the DMA request from the FDC, the DMA controller arbitrates for the Micro Channel bus on behalf of the FDC. The FDC is acknowledged once the bus is granted to initiate the data transfer.

Micro Channel ARBITRATION

The other major function of the 82307 DMA controller is to provide Micro Channel Arbitration. It provides full Micro Channel bus arbitration capability according to the 18-level priority scheme. During normal operation priority 0–15 are assigned with 15 being the lowest priority for the CPU. Priority level −1 which has the higher priority than 0 is also assigned to CPU (switched from 15 to −1) during NMI error recovery. The highest priority −2 is used for refresh.

The bus arbitration priority is asserted via the ARB0–ARB3 signals by the requesting masters to gain control. Bus granting is assigned by the priority level. During an arbitration cycle no Micro Channel master is allowed to drive the bus.

The bus can be preempted by the 82307 when arbitrating on behalf of DMA, or when it is requested to run a refresh cycle, or to respond to NMI error recovery. The preempting of the bus can also be initiated by Micro Channel masters.

The CACP Control Port 090H is integrated on chip.

Micro Channel REFRESH ADDRESS GENERATION/CYCLING

The actual refresh request is generated by the 82309 Address Bus Controller. Upon receiving this request the 82307 DMA Controller gains bus control and executes the refresh cycle. Address generation for the Micro Channel refresh cycle is generated by the 82307 DMA controller.

NUMERICS COPROCESSOR INTERFACE

The 82307 DMA controller supports the numerics coprocessor interface. It provides software transparency required to interface an 80387 to 80386 processor. Ports F0H and F1H are integrated on the 82307.

The numerics coprocessor interface support includes the chip select decode for coprocessor internal register accesses of addresses F8H, FAH and FCH. The 82307 also alerts the CPU of any coprocessor error output generated by asserting the interrupt request IRQ13.

ADDRESS DECODER

The Address Decoder logic decodes the chip select for the 8259 Interrupt Controllers and generates P0S Address Space output for addresses 100H through 107H to support the Card set-up signals for the expansion slots.

The chip select output is for both 8259s in the system, so it must be externally gated with local channel address bit A7 to select each actual device.

For programming and register level details, please refer to IBM PS/2 Technical Reference Manual.

82307 DMA/Micro Channel Arbitration Controller Pin Definitions

Signal Name	Pin Number	I/O	Description
A<0:23>	13-2, 130-124, 122-118	A0-A15 B A16-A23 O	Processor local address bus. A16-A23 are output only and are driven when the DMA controller is bus master. A0-A15 are bi-directional. They are inputs when the CPU is master, allowing the CPU access to the chip's internal ports. They are outputs when the DMA is master.
D<0:15>	18-21, 23-31, 35-37	B	Processor local data bus. When the DMA is master, it drives this bus in a write cycle, and samples it during a read cycle. When the CPU is master, the bus is used to access DMA's internal registers.
S0#, S1#	87, 85	B	CPU or DMA cycle status indicators. The DMA drives these signals when it is bus master. When a slave, the DMA inputs these signals to track CPU cycles.
BHE#	84	O	Byte high enable. It is driven when the DMA owns the bus and tristated otherwise. This signal ties directly to the CPU BHE# output in an 80386SX machine.
PM/IO#	81	B	CPU memory / I/O indicator. The DMA drives PM/IO# when bus master, and inputs it when it is slave.
US0#, US1#	43, 44	I	Micro Channel status pins. Generated by the Bus controller when the CPU or DMA is master. When a slot-resident master owns the bus, it generates US0M# and US1M#, and the DMA inputs these so as to recognize when the slot-resident master relinquishes the bus. (The slot-resident master end-of-transfer is recognized when US0M#, US1M#, the channel CMD# signal, and the channel BURST# signal are all negated.)
OARB0-OARB3	58-55	O	DMA/CACP arbitration bus outputs. These signals are driven by the DMA/CACP to arbitrate on behalf of a floppy disk service at priority level 2.
PBA#	107	O	Processor Bus Access. The signal indicates a CPU bus access to the numeric coprocessor or to one of the DMA/CACP registers.
ARB3-ARB0	59-62	I	DMA/CACP arbitration bus inputs. These signals tie directly to the Micro Channel. All competing masters including the DMA/CACP drive these during an arbitration cycle, and the master with the highest priority takes control of the Micro Channel after the arbitration cycle is complete.
OPMT#	92	O	Preempt Bus Master. DMA/CACP drives this output whenever it wishes to preempt the current bus master. This can occur when arbitrating on behalf of a DMA channel service or when arbitrating on behalf of a refresh request, or when arbitrating on behalf of the CPU so as to let it respond to a NMI (non-maskable interrupt) request.
PREEMPT#	45	I	Wired "OR" of the PREEMPT# signals from all Micro Channel masters, including the DMA/CACP (PMTO#). It signifies that a master wishes to force an arbitration cycle.
ARB/GNT#	63	O	Arbitration Cycle indicator. The DMA/CACP drives this Micro Channel signal high to signify an arbitration cycle. During the arbitration cycle, all competing masters drive their priorities onto the arbitration bus (ARB03-ARB00). The falling edge of ARBGNT# signifies the end of the arbitration cycle, at which time the master with the highest priority takes control of the bus. If no master competes for the bus, the pullups on ARB03-ARB00 will read binary 1111 by default, which is the normal operating priority of the CPU.

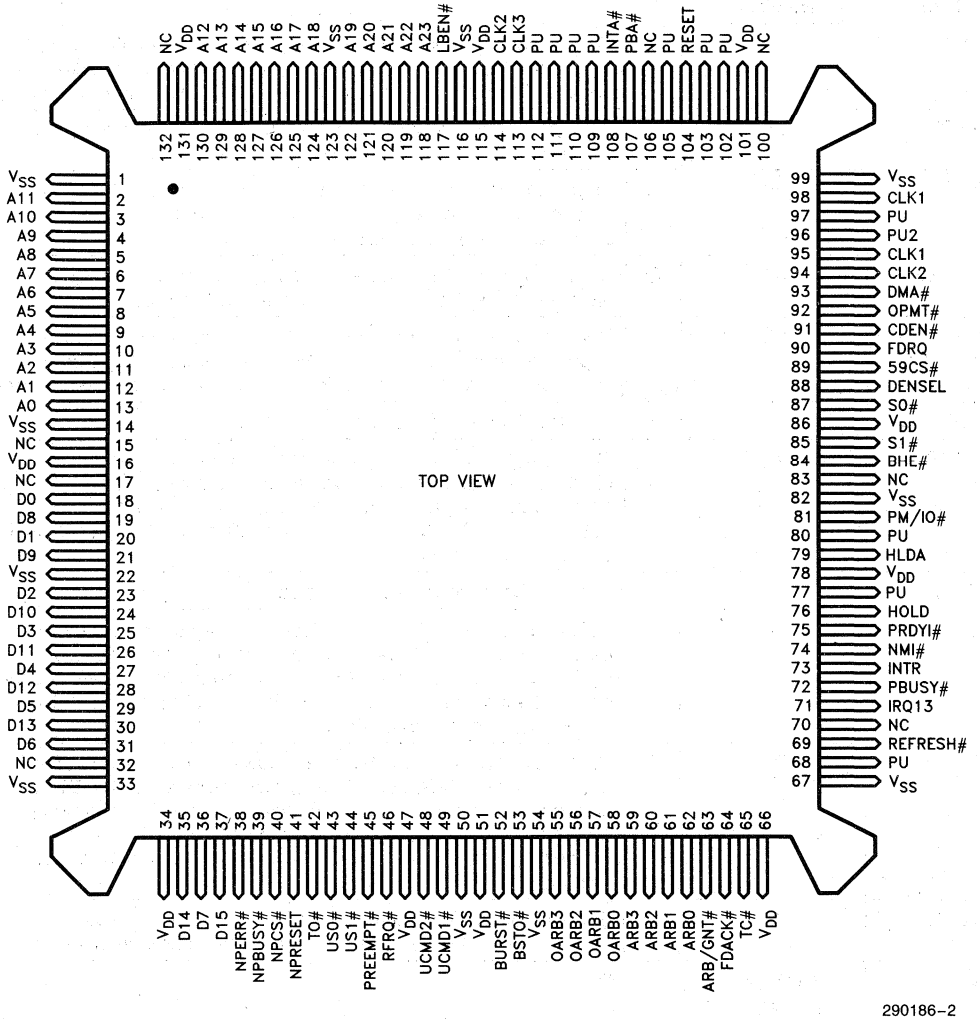
82307 DMA/Micro Channel Arbitration Controller Pin Definitions (Continued)

Signal Name	Pin Number	I/O	Description
BSTO#	53	O	Burst Output. The DMA/CACP drives this output in order to own the Micro Channel for multiple cycles. Specifically, since all PS/2 DMA cycles are two-cycle, BSTO# is driven to allow the DMA controller to own the bus for both halves of a two-cycle transfer.
BURST#	52	I	Burst Request Input. It is an input to the CACP from the current master wishing to own the bus for multiple cycles. It is derived by "OR"ing the Micro Channel BURST# signal with the DMA/CACP BSTO# signal.
HOLD, HLDA	76, 79	(HOLD=O) (HLDA=I)	Hold/Hold Acknowledge to the CPU.
FDRQ, FDACK#	90, 64	(FDRQ=I) (FDACK#=O)	Floppy DMA Request, Acknowledge signals. The motherboard FDC requests DMA service via FDRQ. In response, the DMA/CACP arbitrates for the Micro Channel on behalf of the floppy disk system. Once the DMA/CACP has gained control of the bus, it acknowledges the FDC via FDACK#.
TC#	65	O	DMA Transfer Complete.
UCMD1#, UCMD2#	49, 48	I	Micro Channel Command Inputs. These inputs are driven directly by the Micro Channel CMD# signal.
RFRQ#	46	I	Refresh Cycle Request from the Address Bus Controller.
REFRESH#	69	O	Refresh Cycle Signal. The DMA/CACP drives this output active during refresh cycles. This signal is buffered to become the Micro Channel REFRESH# signal.
59CS#	89	O	Interrupt Controller Chip Select (8259s). Note that this output is activated if either interrupt controller is selected. It is then externally gated with the local I/O channel address bit A7 to distinguish between controller 1 and controller 2.
DENSEL	88	I	Density Selected for the motherboard FDC
CDEN#	91	O	Card Setup Enable. During system setup, a bit pattern is written to port 96H to select a particular slot for configuration. CDEN# enables the decode of these bits to send an active CDSETUP# signal to the selected slot. CDEN# is simply a combinatorial (non-clocked) decode of ports 100H–107H.
CLK1, CLK2, CLK3	95, 98, 94, 114, 113	I	Clock Inputs
RESET	104	I	Power-up System Reset
INTR, INTA#	73, 108	I	Interrupt Request/Acknowledge. The PS/2's 8259 based interrupt system generates interrupt requests to the CPU via INTR. In response, the CPU fetches the appropriate interrupt vector from the interrupt controller in an interrupt acknowledge cycle. The Bus controller decodes the CPU status outputs, and drives INTA# to identify a CPU interrupt acknowledge cycle. The DMA/CACP monitors this activity via its INTR and INTA# inputs. In response to INTA#, the DMA/CACP drives LBEN# so as to enable the 8259 vector onto the Micro Channel. The CACP uses INTR to ensure that the CPU has an opportunity to service an interrupt within one "fairness" cycle; i.e., it prevents the CPU from being totally locked out by higher priority arbiters.

82307 DMA/Micro Channel Arbitration Controller Pin Definitions (Continued)

Signal Name	Pin Number	I/O	Description
NMI#	74	B	Non-Maskable Interrupt to force arbitration cycle to allow CPU bus ownership. As an output, this appears to the system as an open drain, which allows for an external wire "OR" with other NMI sources.
PRDYI#	75	I	Processor Ready Input. The Bus controller generates this signal to terminate CPU and DMA cycles.
NPCS#	40	O	Chip select for the Numeric Coprocessor. It is an unlatched decode that acts as a chip select for CPU accesses to the numeric coprocessor's internal registers.
NPRESET	41	O	Numeric Coprocessor Reset. It resets the numeric coprocessor either upon a system reset or under software control.
NPERR#	38	I	Numeric Coprocessor Error Input. It is an input from the numeric coprocessor error output. The DMA/CACP uses it to generate an interrupt request (IRQ13) to inform the CPU of a coprocessor error.
NPBUSY#	39	I	Numeric Coprocessor Busy
PBUSY#	72	O	Processor Busy Output. It drives the CPU numeric coprocessor busy input. It is activated normally when the coprocessor is busy executing an instruction, but is also activated when a coprocessor error is detected. The CPU will not attempt to utilize the coprocessor as long as PBUSY# is active.
IRQ13	71	O	Numeric Coprocessor Error Interrupt
LBEN#	117	O	Local Bus Enable. This signal is used to enable the data buffers between the Micro Channel and local I/O bus. It is activated for decoded accesses to the 8259 interrupt controllers, as well as for interrupt acknowledge cycles. It is also driven during the DMA acknowledge cycle to the FDC.
DMA#	93	O	DMA/CACP as the Bus Master. It is driven low at the end of an arbitration cycle (ARB/GNT# falling) to indicate that the DMA controller has gain control of the Micro Channel. It is negated during arbitration cycles, and is negated when either the CPU or slot-resident master owns the bus. It is also negated during refresh cycles.
TO#	42	O	Bus Timeout signal. The DMA/CACP also issues an NMI to the CPU in response to the Bus timeout, and forces an arbitration cycle.
V_{DD}	16, 34, 47, 51, 66, 78, 86, 101, 115, 131		Power
V_{SS}	1, 14, 22, 33, 50, 54, 67, 82, 99, 116, 123		Ground
NC	15, 17, 32, 70, 83, 100, 106, 132		No Connect
PU	68, 77, 80, 97, 102, 103, 105, 109–112	I	Pull Up
PU2	96	I	Pull Up. This input must have its own pullup.

82307 DMA/Micro Channel Arbitration Controller



290186-2

NOTES:
Important!! No other node allowed to share pull-up with PU2.
NC = No Connect
PU = Pull-Up
—Pull-Up Resistor Value = 2K to 10K
—No more than three nodes to a single pull-up resistor.

82307 PARAMETRICS

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

ABSOLUTE MAXIMUM RATINGS*

Case Temperature under Bias .... −40°C to +85°C
Storage Temperature ..... −65°C to +150°C
Voltage to Any Pin with Respect to Ground ..... −0.3V to (V_{CC}+0.3)V
DC Supply Voltage (V_{CC}) ..... −0.3V to +7.0V
DC Input Current ..... ±10 mA

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

1

D.C. CHARACTERISTICS

T_C = 0°C to +70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Conditions
V_{IL}	Input Low Voltage		0.8	V	
V_{IH}	Input High Voltage	2.0		V	
V_{IL}	Input Low Voltage		0.8	V	CLK1, CLK2, CLK3
V_{IH}	Input High Voltage	V_{CC} − 0.8		V	CLK1, CLK2, CLK3
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = 2 mA
I_{CC}	Power Supply Current		180	mA	No DC Loads
I_{LI}	Input Leakage Current		±10	μA	$V_{SS} < V_{IN} < V_{CC}$
I_{OZ}	Tri-State Output Leakage Current		±10	μA	$V_{SS} < V_{OUT} < V_{CC}$

82307 DMA CONTROLLER A.C. SPECS

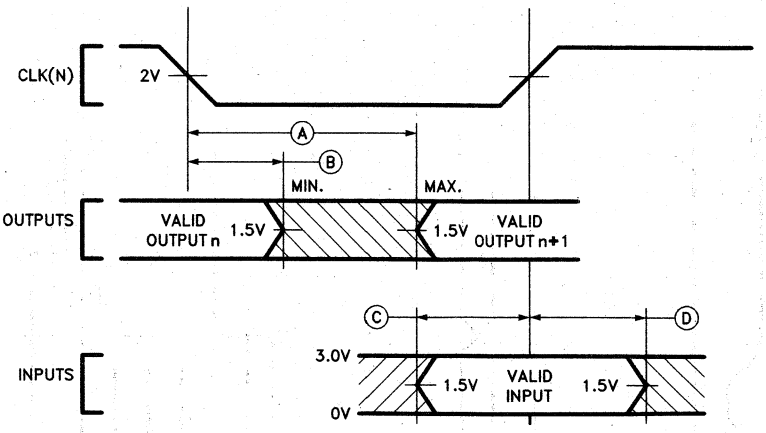
T_C = 0°C to +70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Kit 16 MHz		Kit 20 MHz		Kit 25 MHz		C_L (pF)	Notes
		Min	Max	Min	Max	Min	Max		
T1	CLK1, CLK2, CLK3 LOW TIME	15		15		14			
T2	CLK(N) NON-OVERLAP TIME	4		4		0			
T3	RESET(IN), NPRESET(OUT) PULSE WIDTH	500		500		500		50	
T4	TO# PULSE WIDTH	60		60		60		50	
T5	A23–A0, PM/IO#, BHE#, REFRESH# DELAY	4	35	4	35	4	32	75	
T6	A23–A0, PM/IO#, BHE#, STATUS FLOAT DELAY	4	40	4	40	4	40	75	
T7	WRITE DATA VALID DELAY	2	35	2	35	2	28	75	
T8	WRITE DATA FLOAT DELAY	2	40	2	35	2	35	75	
T9	READ DATA SETUP TIME	15		13		11			
T10	READ DATA HOLD TIME	4		4		4			
T11A	STATUS VALID DELAY (TPHL)	2	25	2	25	2	20	75	
T11B	STATUS VALID DELAY (TPLH)	2	35	2	35	2	30	75	
T12	ADDR-TO-STATUS SETUP	35		27		22		75	
T13A	PRDYI# SETUP TIME	25		20		20			2, 3
T13B	PRDYI# SETUP TIME	8		8		8			2, 3
T14	PRDYI# HOLD TIME	5		5		5			2
T15	A15–A0, PM/IO# SETUP TIME	35		35		35			
T16	A15–A0, PM/IO# HOLD TIME	10		10		10			
T17	STATUS SETUP TIME	26		26		24			
T18	STATUS HOLD TIME	10		10		10			
T19	WRITE DATA SETUP TIME	25		25		25			1
T20	WRITE DATA HOLD TIME	25		25		25			1
T21	READ DATA VALID DELAY	2	100	2	100	2	100	75	
T22	READ DATA FLOAT DEALY	8	40	8	35	8	35	75	
T23	HOLD DELAY	2	35	2	32	2	32	50	
T24	ARB/GNT# DELAY FROM EOT	30		30		30		50	
T25	ARB/GNT# PULSE WIDTH	6 × CLK3		6 × CLK3		6 × CLK3		50	4
T26	OARB3–OARB0 DELAY		32		32		32	50	
T27	OPMT# INACTIVE DELAY		35		35		35	50	
T28	BSTO# DELAY	2	40	2	40	2	40	50	
T29	TC# DELAY	2	36	2	33	2	33	50	
T30	FDACK#, DMA# DELAY	2	40	2	40	2	40	50	
T32	LBEN# VALID DELAY	0	35	0	35	0	35	50	
T33	CDEN# VALID DELAY	2	35	2	35	2	35	25	
T34	NPCS# DELAY	2	65	2	65	2	50	50	
T35	59CS# VALID DELAY	6	42	6	42	6	42	50	
T36	PBA# DELAY	2	45	2	45	2	45	50	

NOTES:

1. Write data is sampled on different clock edges by different DMA internal registers. T19 is speced relative to the earliest sampling edge, while T20 is relative to the latest edge.
2. PRDYI# must be inactive and stable according to these specs at all DMA state boundaries except at the end of the TC boundary at which the cycle is to be terminated.
3. T13A must be met to insure the 82307 properly recognizes PRDYI# at the end of the cycle. T13B must be met to insure that status (S0#, S1#) activation is not delayed if the 82307 has a cycle pending and will move directly into a TS state upon completion of the current cycle. If T13B is not met, the status valid delay (T11A) may no longer apply since PRDYI# rather than CLK1 may gate status activation.
4. T25 is specified as 3 times the CLK3 period. This is a typical value which is not tested. Also, this spec does not comply with Micro Channel timings at 25 MHz. To remedy this situation, external hardware, such as the PAL used in the 82311 Designer's Guide, must be implemented in order to guarantee T25's compatibility with Micro Channel specifications.

DRIVE LEVELS AND MEASUREMENT POINTS FOR A.C. SPECIFICATIONS

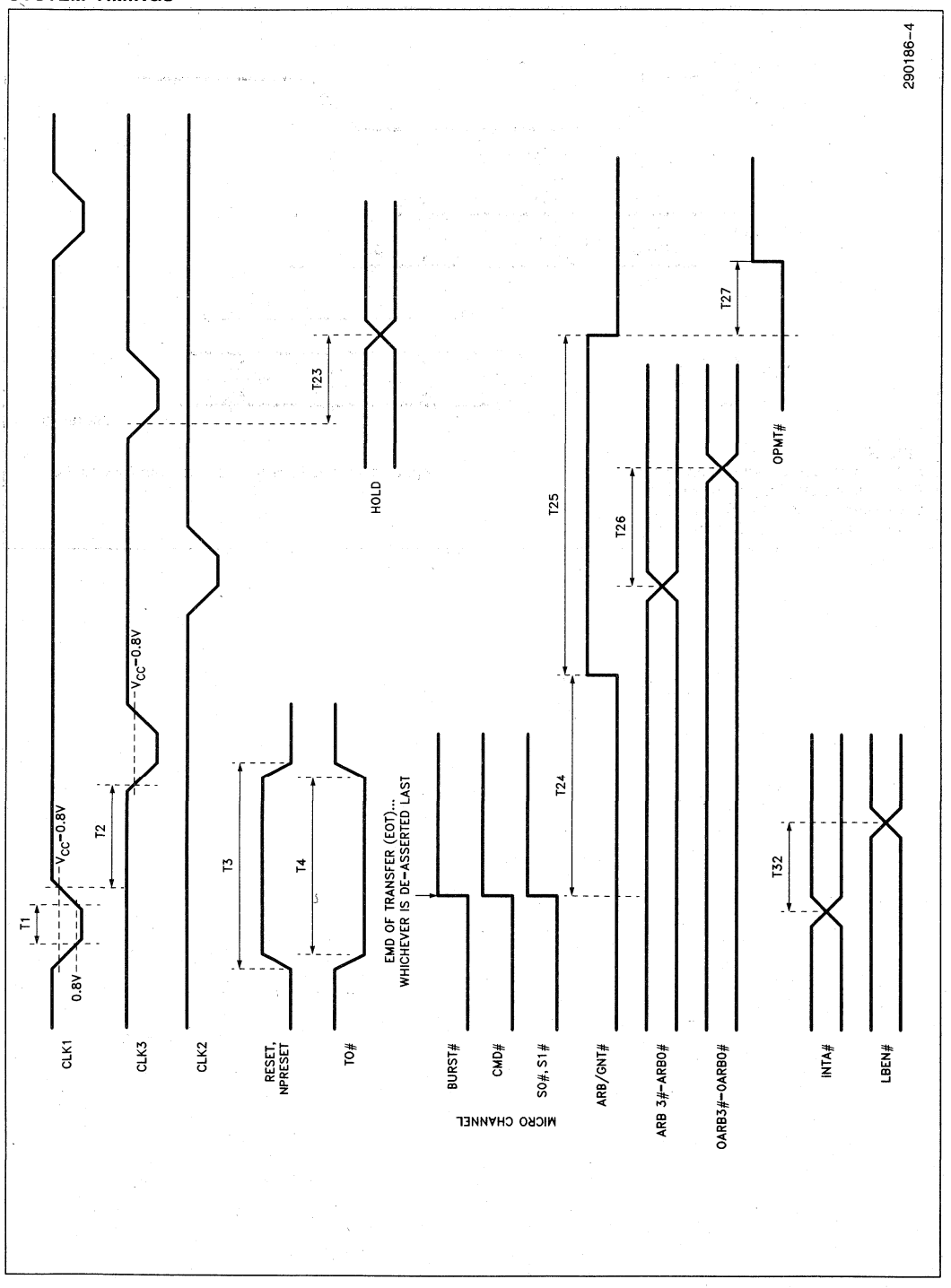


290186-3

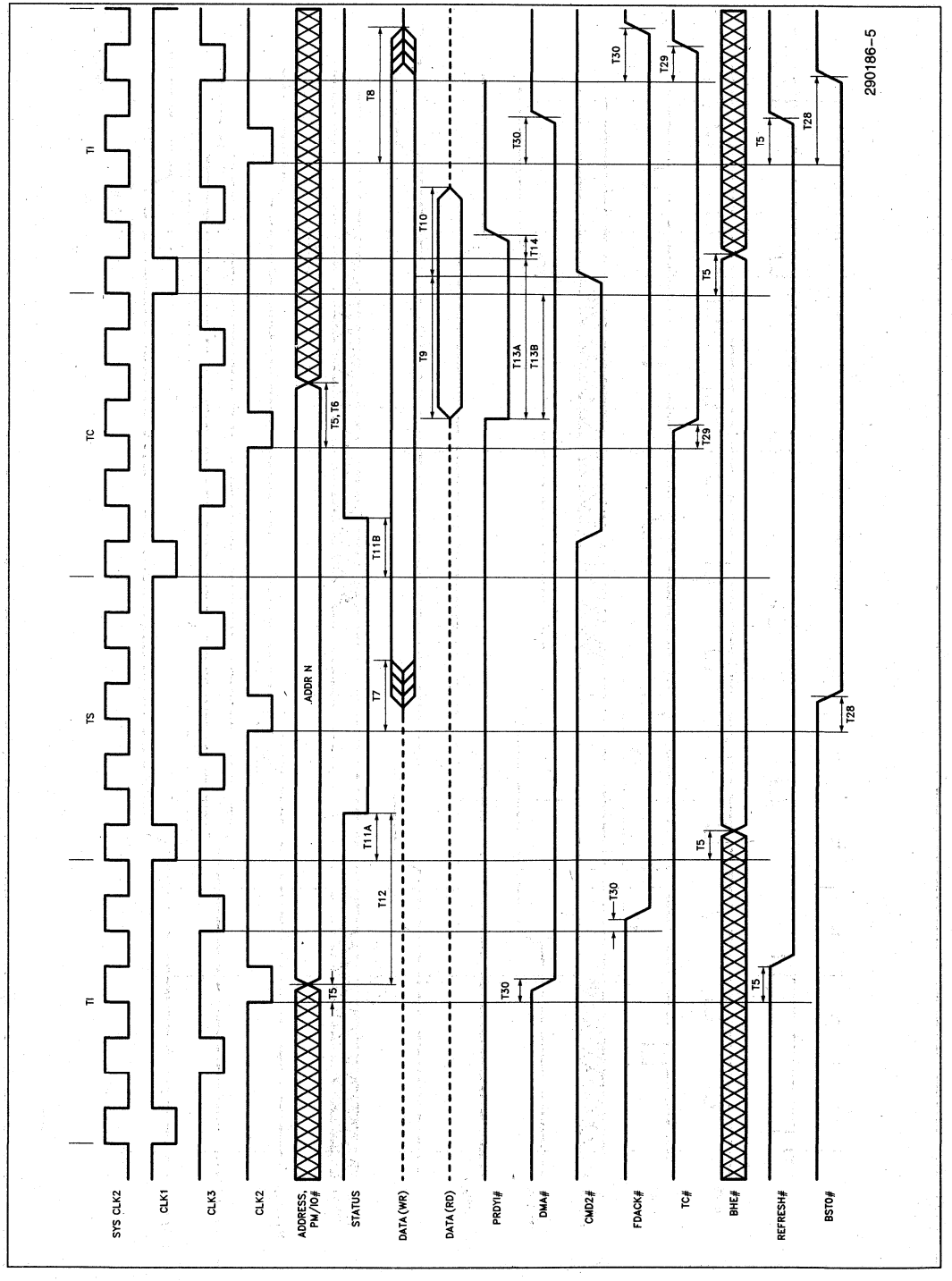
LEGEND:
A. Maximum Output Delay Specification.
B. Minimum Output Delay Specification.
C. Minimum Input Setup Specification.
D. Minimum Input Hold Specification.

Input waveforms have tr ≤ 2.0 ns from 0.8V to 2.0V.

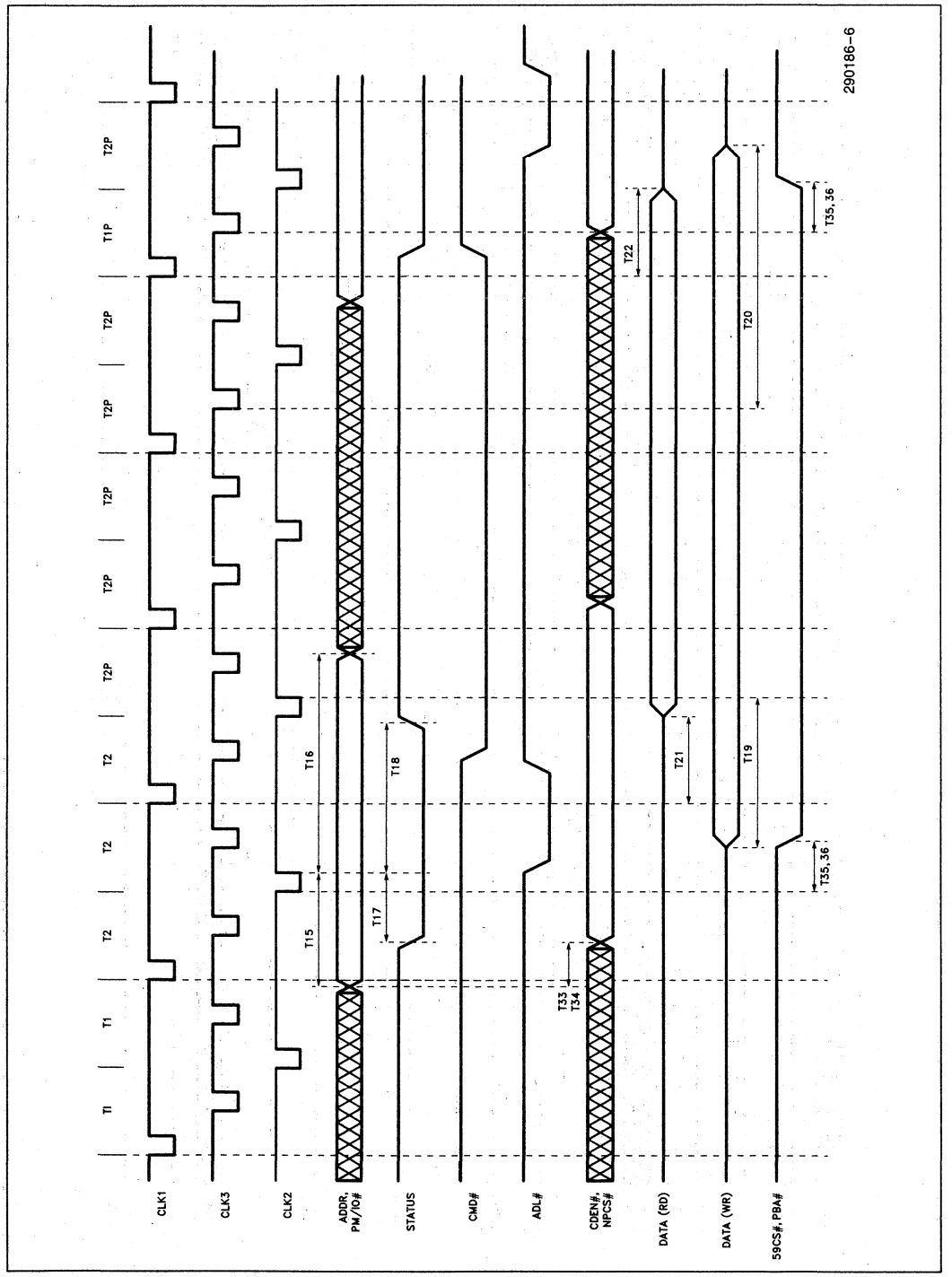
SYSTEM TIMINGS



DMA ... MASTER MODE



DMA ... SLAVE MODE



82308
Micro Channel BUS CONTROLLER

- **Micro Channel Compatible Bus Control**
- **Supports 8-, 16- or 32-Bit Data Transfers on the Micro Channel**
- **Optional Hardward Enforced I/O Recovery Mechanism**
- **Cache Controller (82385) Interface to Maximize Performance for 80386 Based Systems**
- **Low Power CHMOS Technology**
- **100-Pin Plastic Quad Flat Pack Packaging**

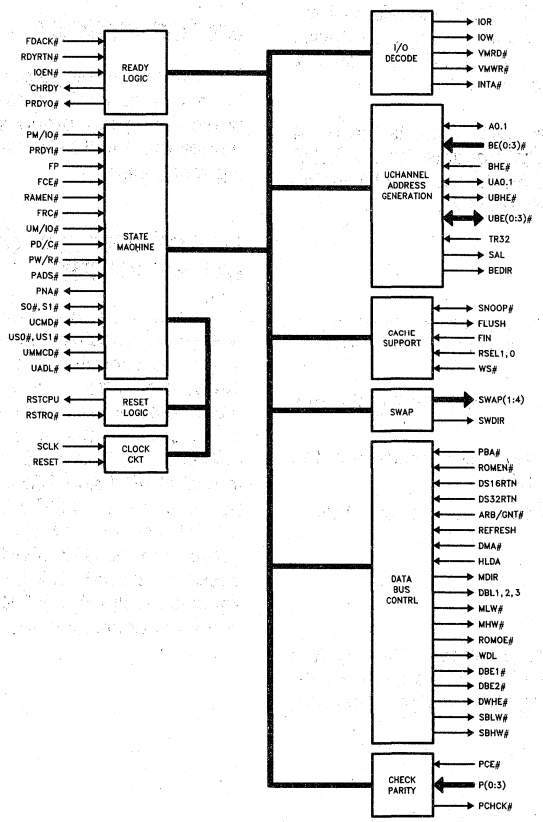
(See Packaging Spec., Order # 231369)

1

The 82308 Micro Channel Bus Controller is the complementary device to the 82309 Address Bus Controller. It is designed to facilitate data transfers between the Microprocessor, DMA, Memory and Micro Channel bus. It generates the appropriate data conversion and alignment control signals to implement an external byte swap mechanism for transferring data of equal and different widths.

The 82308 Bus Controller generates all control signals necessary to run Micro Channel Memory and I/O Bus cycles for both 80386 and 80386SX processors.

To implement the highest performance 80386 based system with the 82385 cache controller, the Bus Controller features special cache hardware interface signals.



290187-1

STATE MACHINE

The primary purpose of the state machine is to generate the Micro Channel signals for processor and DMA cycles. These Micro Channel signals are: S0#, S1#, ADL#, MMCCMD# and CMD#. The state machine also generates the PNA# signal required by the 386 CPU and generates the DMA S0# and S1# based on the 386 processor status.

DATA TRANSFER

The 82308 Bus Controller directs the transfer of data between the 32-bit 80386 bus and 32-bit, 16-bit or 8-bit devices on the Micro Channel. For 16-bit transfers initiated by the 80386SX or DMA, the Bus Controller provides the control signals to facilitate transfers to 16-bit or 8-bit devices on the Micro Channel data bus and vice versa.

In addition to providing the transceiver direction, latch, and enable signals, the Bus Controller manipulates Address signals for the DMA and Micro Channel. For example, a 32-bit access to an 8-bit device is broken into four cycles, and the BC automatically sequences A1 and A0 in each cycle.

The 82308 Bus Controller also supports the ROM BIOS by providing the output enable for the BIOS based on the decoded BIOS address signal from the Address Bus Controller (ROMEN#).

RESET DETECT

The reset detect logic generates a synchronous CPU reset signal based on any of the following events:

— An active low-pulse on the RC# input to the Bus Controller

— Processor Shutdown Condition based on the Processors' status signals

— Power-up condition as determined by the RESET input.

I/O SUPPORT

The 82308 Bus Controller generates Memory and I/O Read and Write signals for devices on the motherboard. It also generates the Interrupt Acknowledge signal.

The Bus Controller extends motherboard device accesses by de-asserting CHRDY until a read or write strobe is generated. This gives the peripheral device an opportunity to extend the cycle even further if required by driving its own CHRDY inactive after it detects the read or write strobe. The motherboard device decode is performed by the 82309 Address Bus Controller.

CACHE SUPPORT

The 82308 Bus Controller supports 82385 Cache Controller interface signals to allow maximum system performance in cache based 386 systems.

CACHE FLUSH

The Bus Controller generates a synchronous flush output signal (FLUSH) to the cache controller whenever the flush request (FIN) is generated.

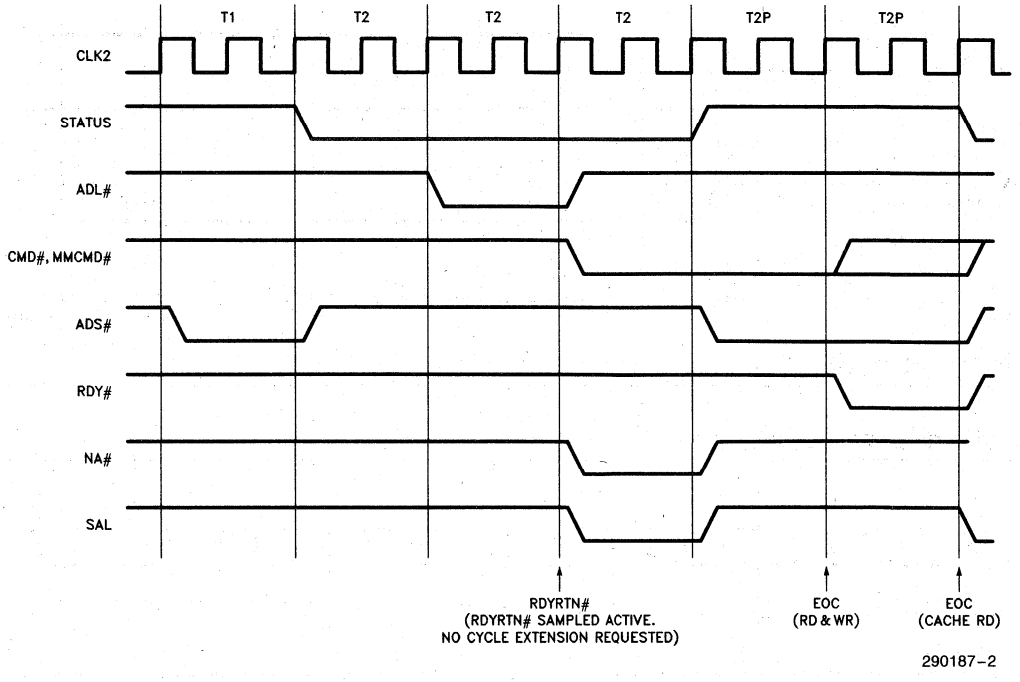
SNOOP STROBE

The Snoop Strobe output of the 82308 Bus Controller is a synchronized strobe indicating a valid address during non-processor write cycles. It is compatible with the 82385 cache controller's bus watching mechanism.

HARDWARE ENFORCED I/O RECOVERY

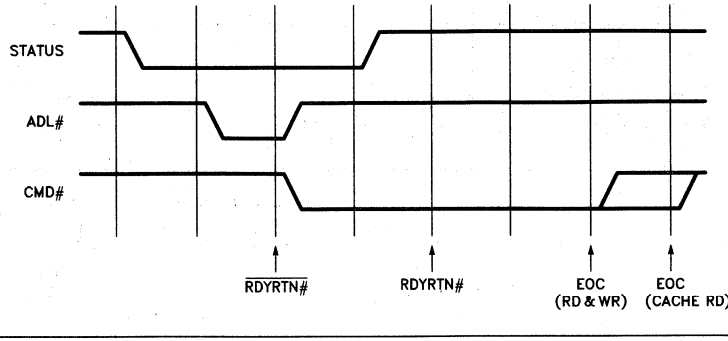
Certain I/O devices require a minimum delay between consecutive accesses. Typically, software loops are executed in the I/O routine to force the delay, but software loops cannot guarantee minimum delay times in all cases. The 82308 Bus Controller provides the option of enforcing I/O recovery in hardware. This mechanism is controlled by two inputs (RSEL1 and RSEL0), which select one of four possible minimum I/O recovery times. At the end of a CPU initiated I/O cycle, an internal timer is triggered, and the 82308 will not allow the next I/O access to proceed until the timer has timed out. The specific functioning of RSEL1 and RSEL0 is detailed in the pin definitions and A.C. Timing specifications.

16 MHz CPU DEFAULT CYCLE (FP = 0)

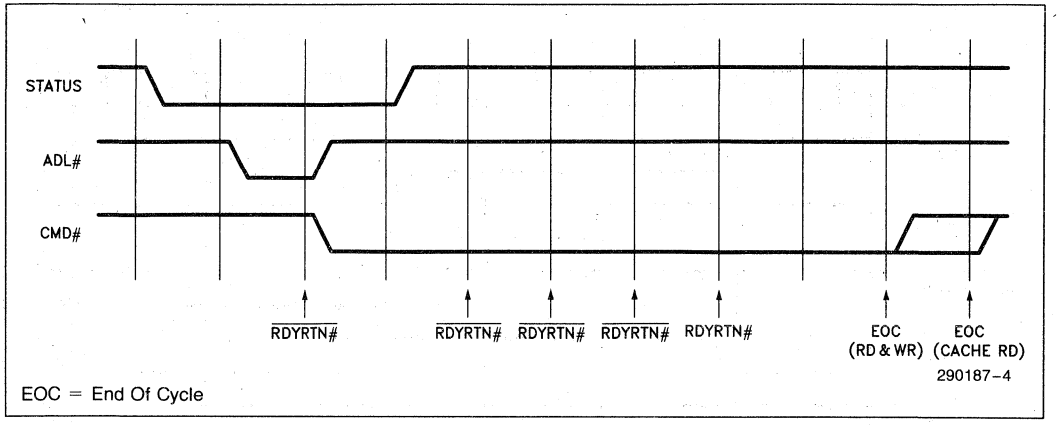


NOTE:
MMCCMD# only driven in CPU initiated cycles.
EOC = End Of Cycle.

16 MHz CPU SYNCHRONOUS EXTENDED

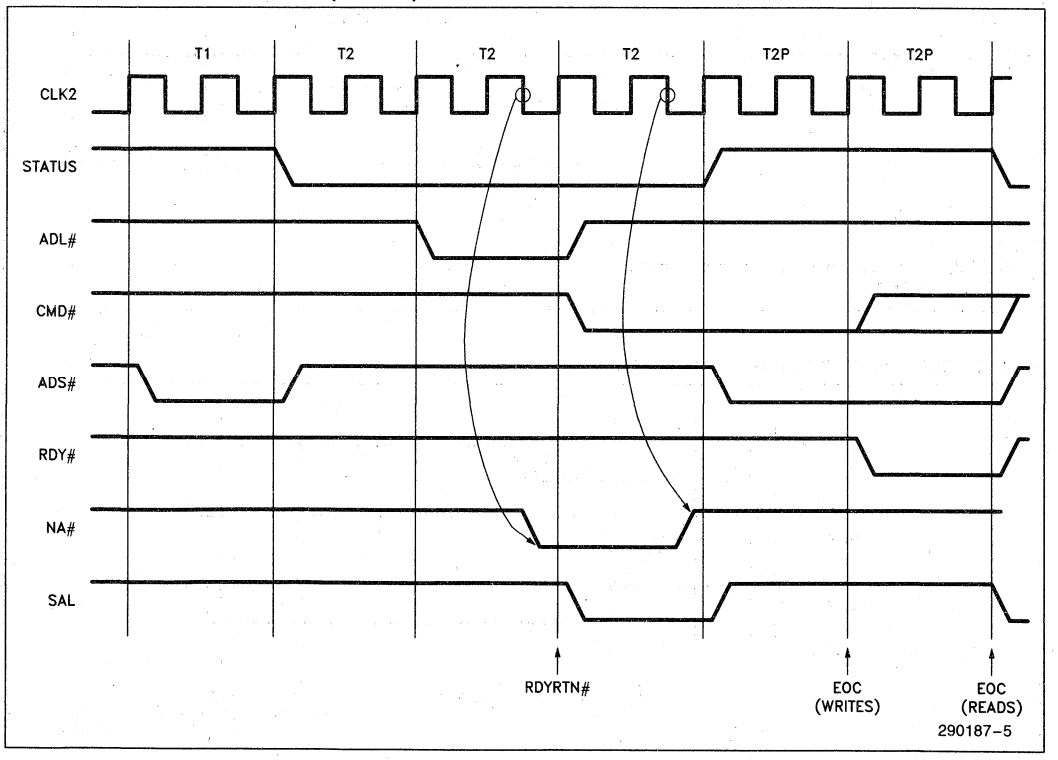


16 MHz CPU ASYNCHRONOUS EXTENDED

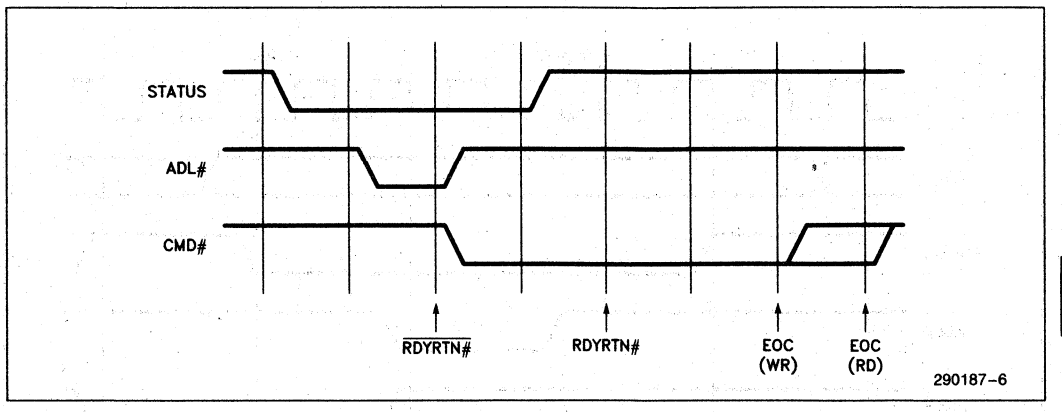


EOC = End Of Cycle

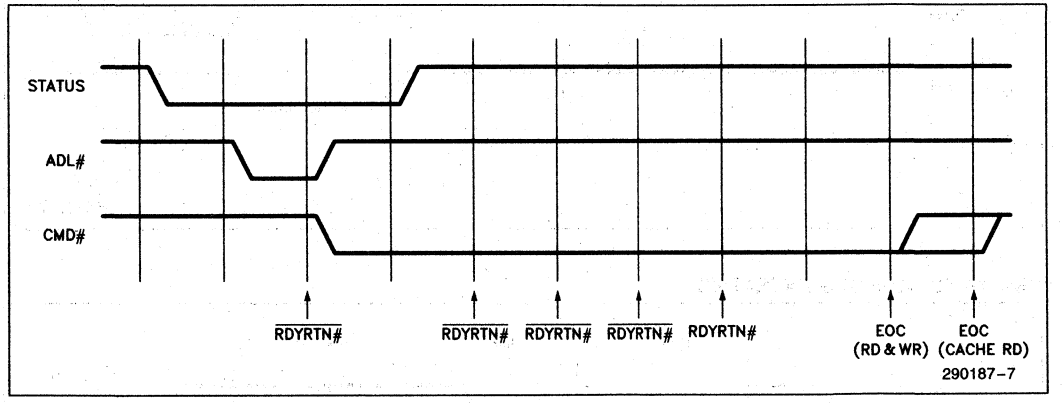
20 MHz CPU DEFAULT CYCLE (FP = 1)



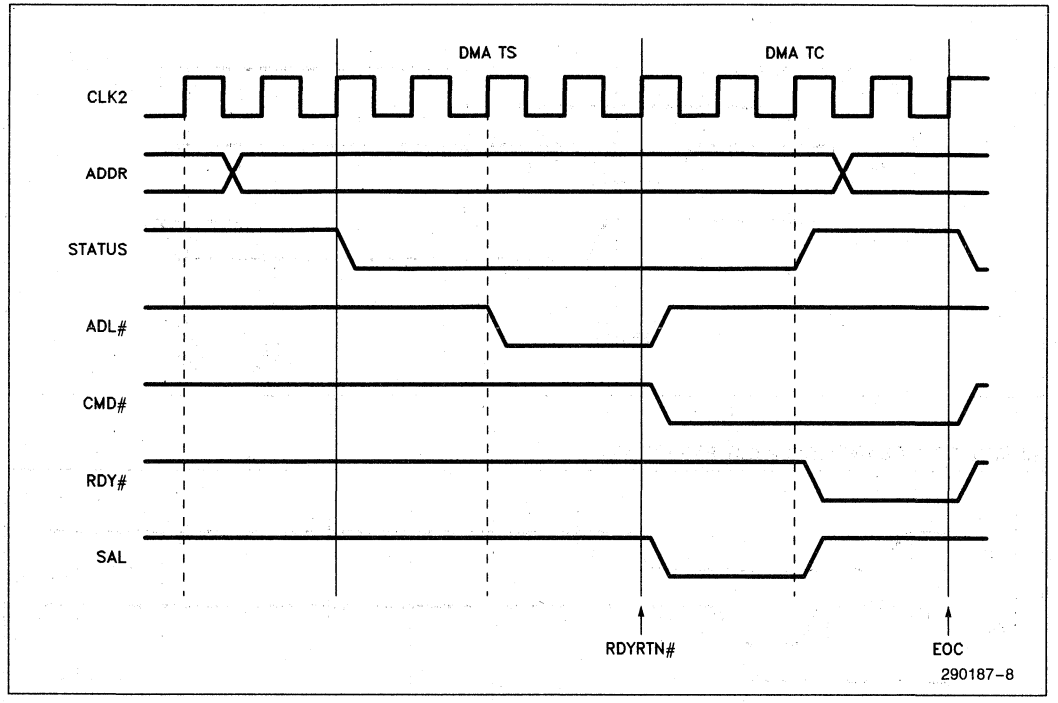
20 MHz CPU SYNCHRONOUS EXTENDED



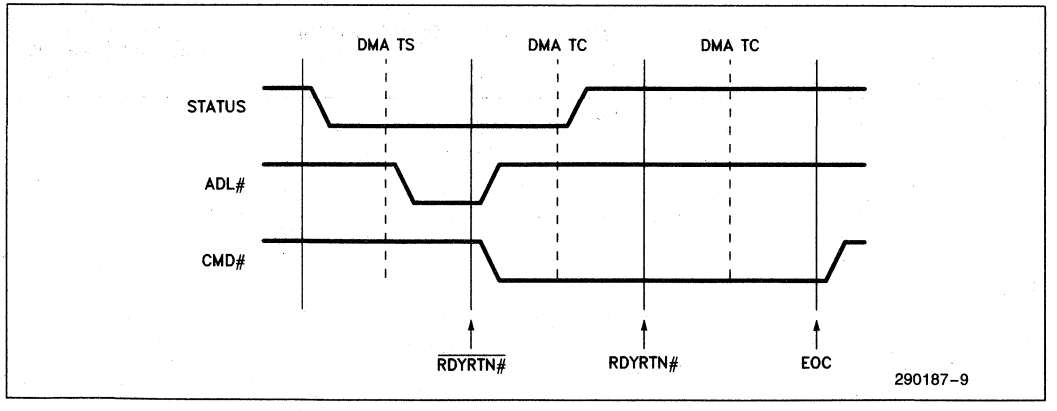
20 MHz CPU ASYNCHRONOUS EXTENDED



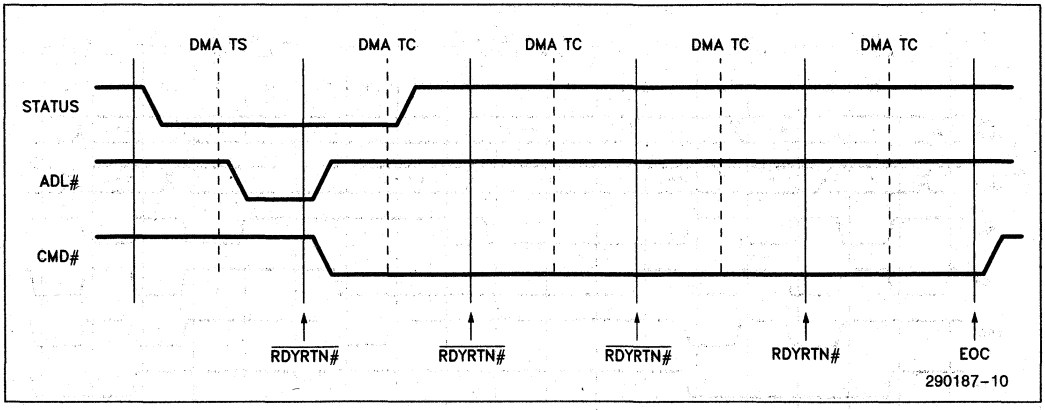
DMA DEFAULT CYCLE



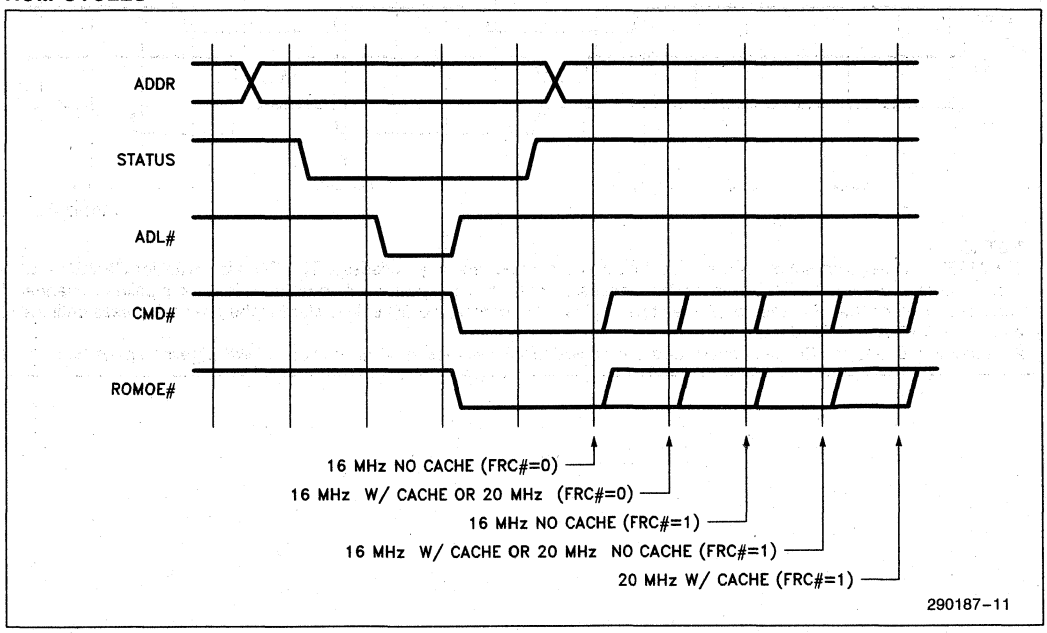
DMA SYNCHRONOUS EXTENDED



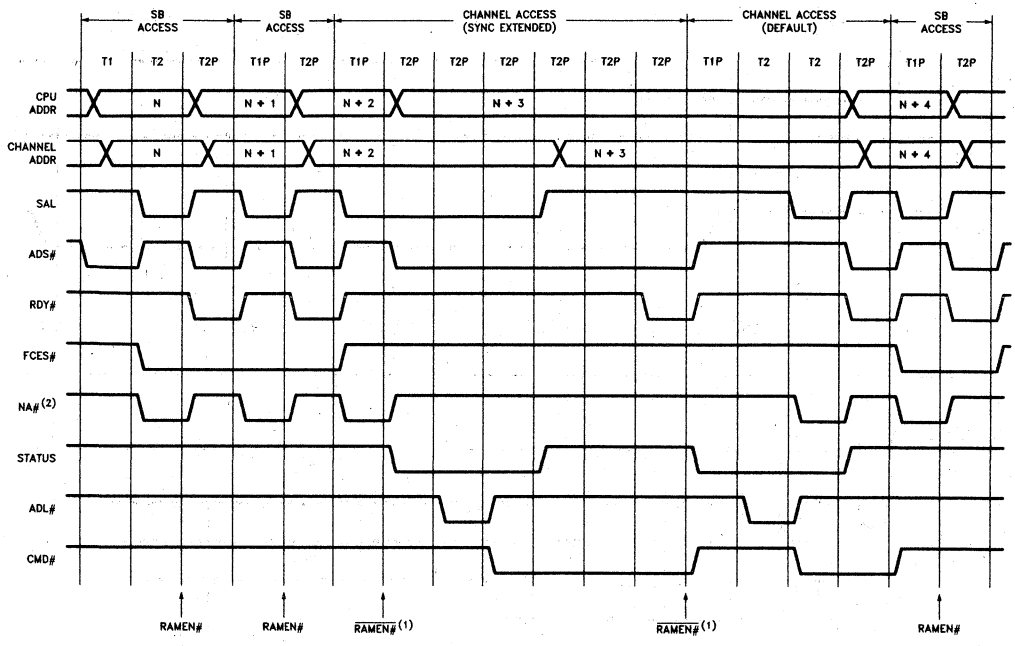
DMA ASYNCHRONOUS EXTENDED



ROM CYCLES



CPU SYSTEM BOARD MEMORY/Micro Channel MEMORY ACCESSES



290187-12

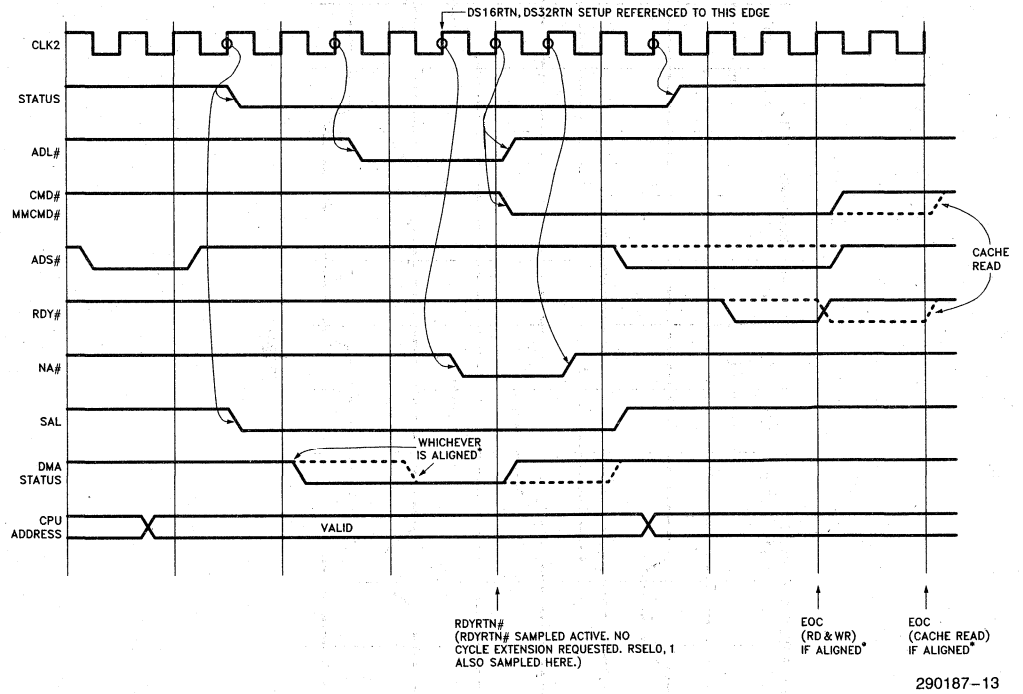
NOTES:

1. RAMEN# distinguishes between system board and channel memory accesses. The BC must wait for RAMEN# to resolve before driving STATUS in a channel access. Thus, in non-pipelined channel accesses or pipelined channel accesses in which the ABC sees only one state of pipelined address, the BC delays starting the channel access until the end of the T1P or first T2 state.
2. In memory cycles the BC must drive NA# before RAMEN# resolves in order to sustain 0WS pipelined page hits.

82308HS-25 MICRO CHANNEL BUS CONTROLLER TIMING DIAGRAMS

The 82308HS-25 provides Micro Channel Bus Control for 25 MHz 80386 systems. It is 100% function and pin compatible with the 82308-16/20 Bus Controller, so minimal system re-design is required to upgrade current 16 MHz or 20 MHz systems to 25 MHz. (Note that the 82308HS-25 FP input must be tied high.)

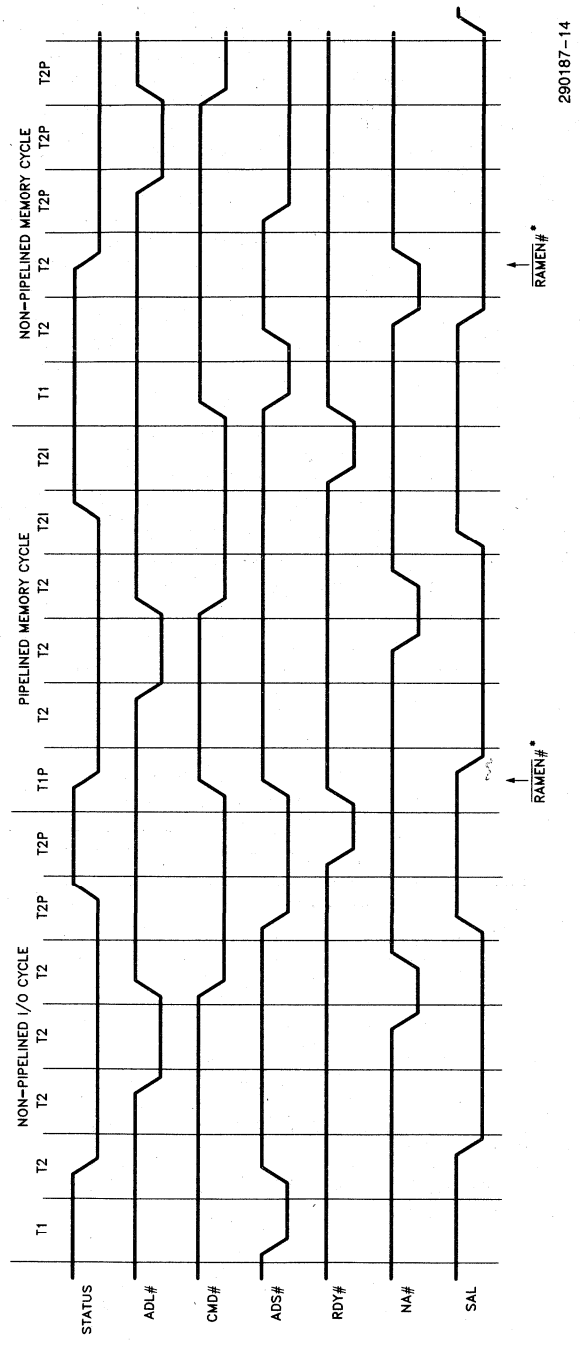
Although the 82308HS-25 is functionally identical to the 82308-16/20, its internal state machine and external timing behavior are modified to insure full compatibility with published Microchannel timings at the increased CPU frequency, and to accommodate 25 MHz system and component specifications. This addendum to the 82308-16/20 data sheet provides the basic timing diagrams for the 82308HS-25, highlighting the specific clock edges that either sample specific inputs, or else trigger specific outputs. All AC specification output delays are referenced to the "causal" clock edge, and input setup/hold times are referenced to the sampling clock edge. Any signal not specifically addressed in these diagrams behaves just as it does in the 82308-16/20. (Note in the AC specifications that notes numbered 21 or greater apply only to the 82308HS-25.) The 82311 Micro Channel Compatible Peripheral Chip Set Designer's Guide contains additional 25 MHz system design information.



290187–13

*If necessary, a wait state is added to CPU I/O cycles to align them with even DMA state boundaries. (Recall that DMA states are 4 clock phases wide while 80386 states are only 2 phases.)

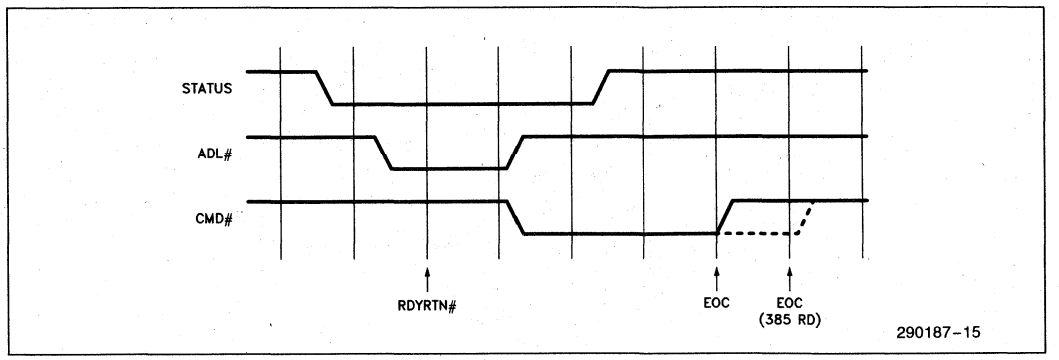
82308HS 25 MHz CPU Default I/O Cycle



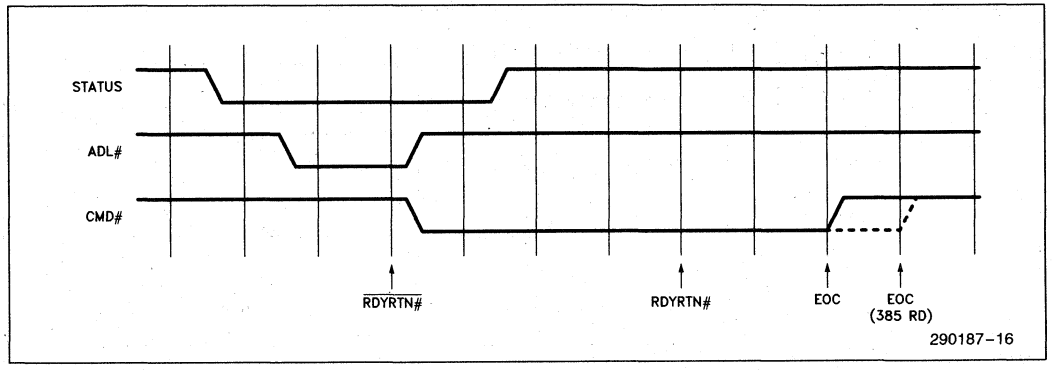
*Ramen# distinguishes between (non-broadcast) system board and channel memory accesses. (The 82308-25 samples Ramen# and Romen# on the phase 2 clock edge.) The BC must wait for Ramen# to resolve "not true" before driving status in a channel access. Thus, in a non-piplined channel memory access, the BC delays starting the channel cycle until after the first T2 state.

82308HS 25 MHz CPU Default Micro Channel Cycles

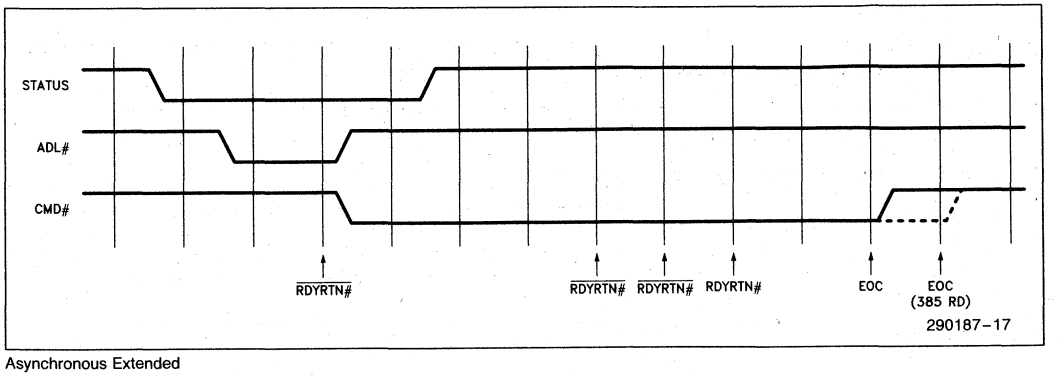
82308HS 25 MHz CPU EXTENDED CYCLES



Default

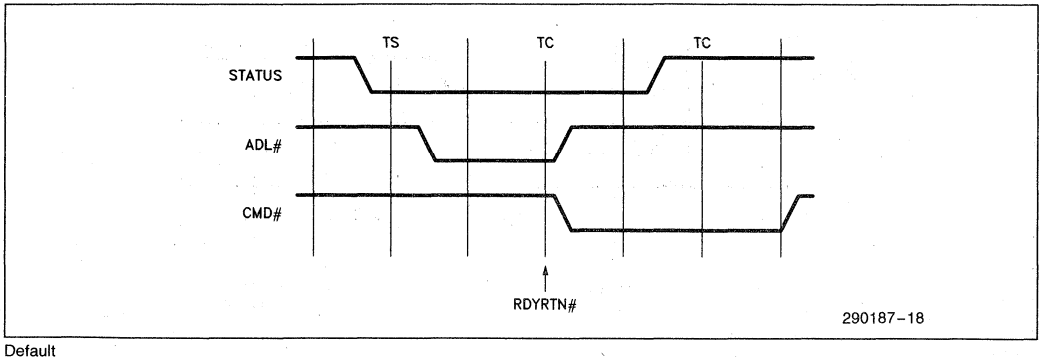


Synchronous Extended

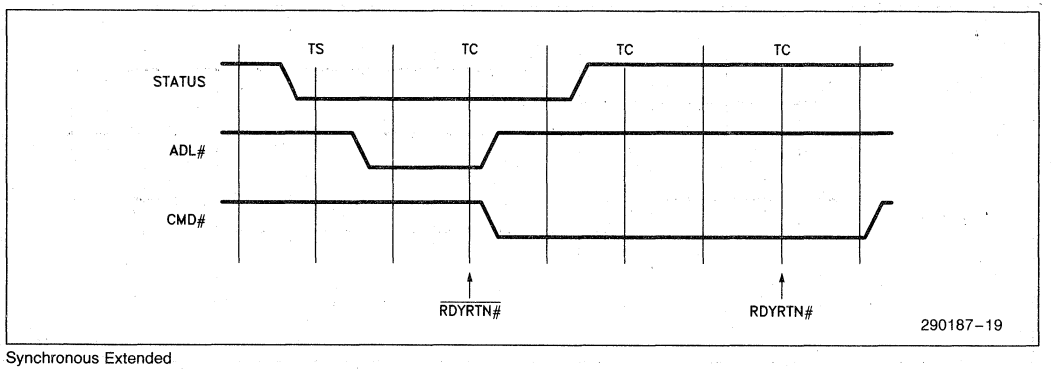


Asynchronous Extended

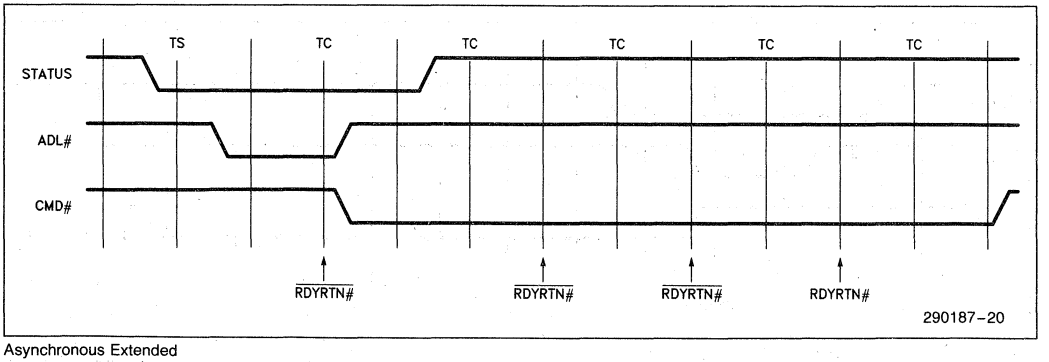
82308HS 25 MHz DMA EXTENDED CYCLES



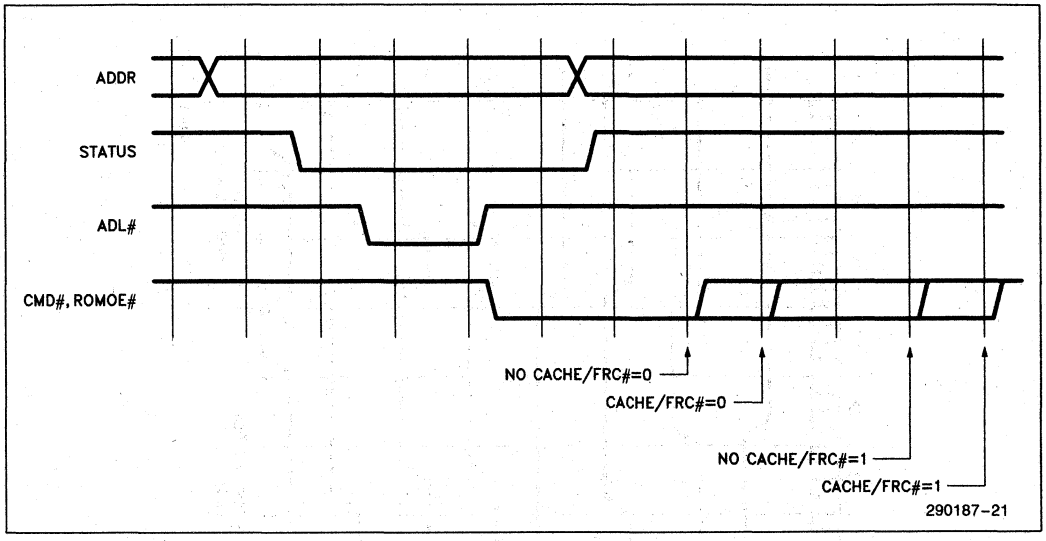
Default



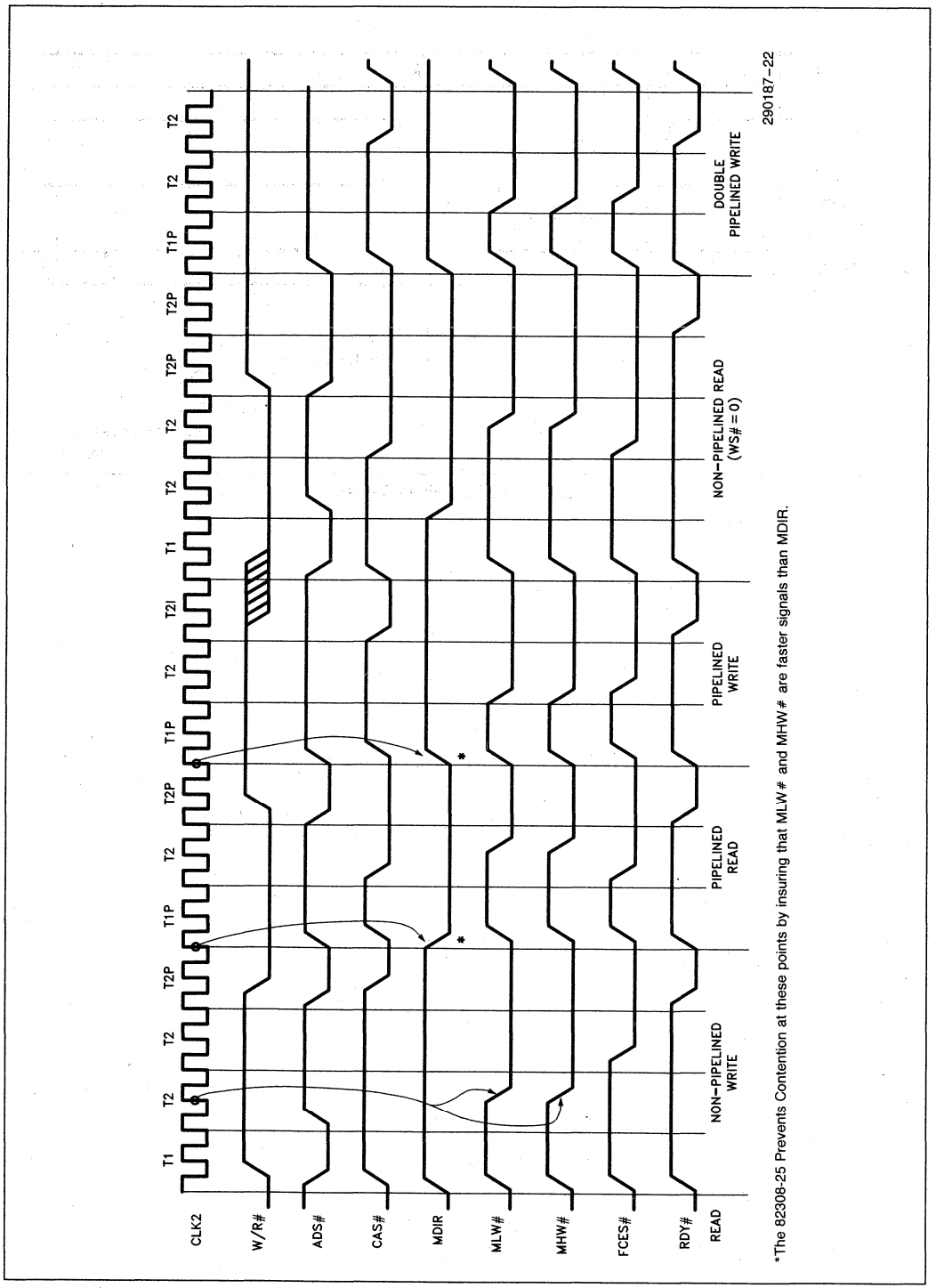
Synchronous Extended



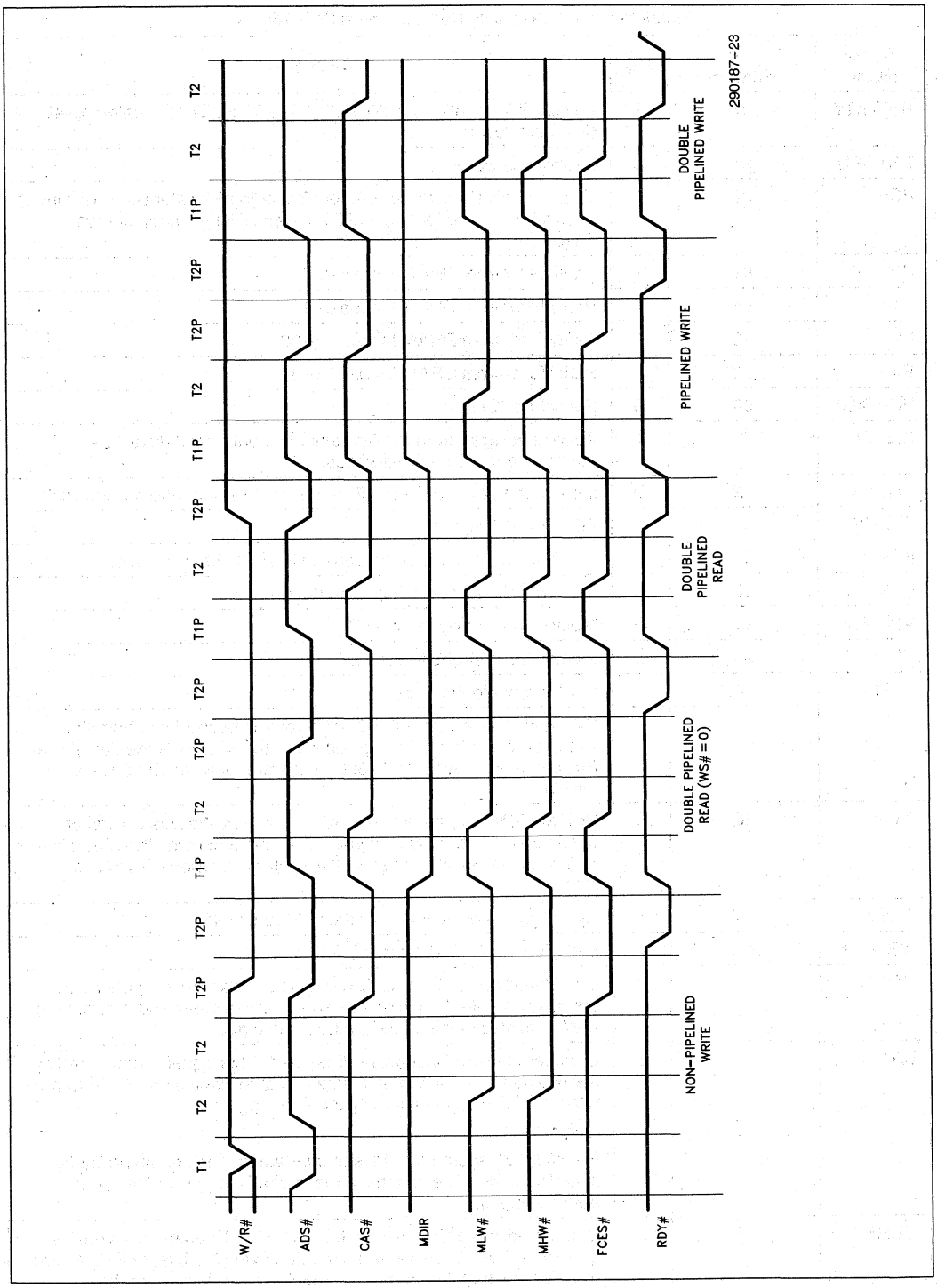
Asynchronous Extended



82308HS 25 MHz ROM Cycles



82308HS 25 MHz CPU Access to System Board Memory/Data Bus Transceiver Control



82308HS 25 MHz CPU Access to System Board Memory/Data Bus Transceiver Control

82308 Micro Channel Bus Controller Pin Definitions

Signal Name	Pin Number	I/O	Description
RSTRQ#	87	I	Logical NOR of 8042 pin 20 and LCS ALTRESET to initialize reset (Software reset).
RSTCPU	91	O	Microprocessor Reset.
PCE#	82	I	Enable Parity Check from Memory Encoding Register Bit 0. This input should be tied low for a model 60 system. (Parity check always enabled.)
PO	83	I	Parity Error from DRAM for bits 0–7.
P1	84	I	Parity Error from DRAM for bits 8–15.
P2	85	I	Parity Error from DRAM for bits 16–23.
P3	86	I	Parity Error from DRAM for bits 24–31.
PCHCK#	90	O	Parity Error Output.
IOEN#	79	I	Active low signal from the Address Bus controller indicating a motherboard I/O device address.
CHRDY	92	O	Input to the Channel Ready Return logic to extend the current cycle.
FDACK#	81	I	FDC DACK# signal.
IOR	97	O	I/O Read signal for motherboard devices (8042, 8259, etc.).
IOW	98	O	I/O Write signal for motherboard devices.
VMWR#	96	O	Memory write strobe to the VGA.
VMRD#	95	O	Memory read strobe to the VGA.
INTA#	94	O	INTA# Input to the 8259.
A0	9	B	CPU and DMA Address 0. A0 will be driven by the Bus controller based on the byte enable signals when the 386 owns the bus. It is a Bus controller input in an 80386SX system, or when the DMA is master.
A1	10	B	CPU and DMA Address 1. A1 will be driven by the Bus controller based on the byte enable signals when the 386 owns the bus. It is a Bus controller input in an 80386SX system, or when the DMA is master.
BHE#	100	I	Byte High Enable signal from the CPU and DMA.
BE0–3#	11–12, 14–15	I	Byte Enable bits 0–3 from the 80386.
UA0	3	B	Unbuffered Micro Channel Address bit 0. This signal is generated by the bus controller based on the byte enable signals and A0 from the DMA. It is also manipulated by the swap logic.
UA1	4	B	Unbuffered Micro Channel Address bit 1. This signal is generated by the bus controller based on the byte enable signals and A1 from the DMA. It is also manipulated by the swap logic. **NOTE:** For 80386SX systems, UA1 is unconnected, and should be lightly pulled up (10K). The channel A1 is latched along with the upper address lines.)
UBHE#	16	B	Unbuffered Micro Channel System Bus High Enable. This signal is generated from BE0–3# when the 386 owns the bus, or BHE# from the DMA. In 80386SX systems, UBHE# is a reflection of the 80386SX BHE# output.

82308 Micro Channel Bus Controller Pin Definitions (Continued)

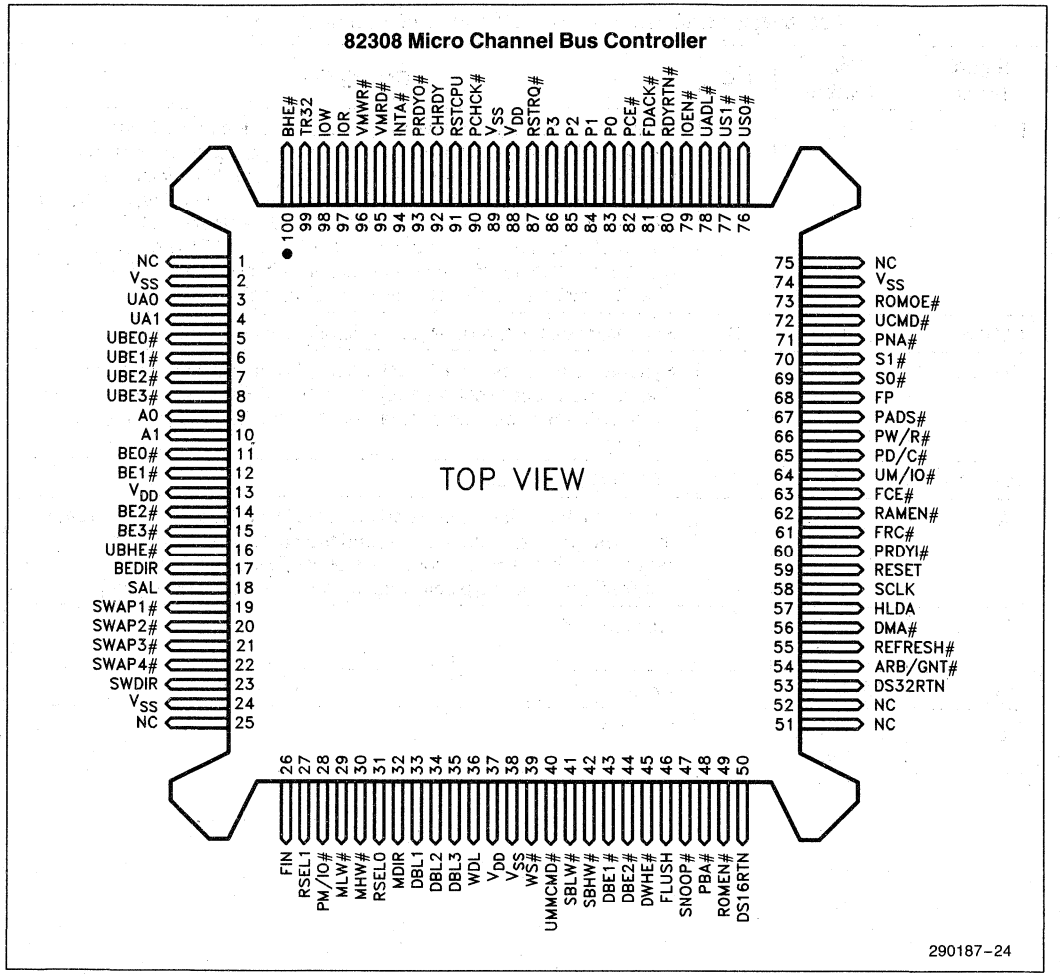
Signal Name	Pin Number	I/O	Description
UBE0–3#	5–8	B	Unbuffered Micro Channel Byte Enable bits 0–3. These byte enable signals are driven by the bus controller when the CPU or DMA is master. (An external PAL is required to generate the channel byte enables on behalf of a 16-bit channel master that requests translation.)
TR32	99	I	Translate 32 from the Micro Channel to indicate 32 bit masters driving BE0–3# (when inactive). (Tie high for 80386SX system.)
SAL	18	O	Latch enable for the system address bus. This signal controls the address latch between the CPU bus and channel.
BEDIR	17	O	Direction control for the UBE0–3# transceiver. It is high when the Bus controller is driving UBE0–3#.
PBA#	48	I	Indicates that the DMA or numeric coprocessor has been selected and is using the local data bus.
ROMEN#	49	I	Decode that inidicates that the BIOS ROM has been selected.
DS16RTN	50	I	Micro Channel Data Size 16 signal.
DS32RTN	53	I	Micro Channel Data Size 32 signal.
ARB/GNT#	54	I	Micro Channel ARB/-GNT status.
REFRESH#	55	I	Refresh Indicator.
DMA#	56	I	Indicates that the DMA owns the bus.
HLDA	57	I	CPU HLDA input. Indicates CPU controls local address and data bus if low.
WDL	36	O	Latch enable signal for latching data bus D0–31 for generating Micro Channel D0–31 signal. Insures Micro Channel write data hold time spec is met.
DBE1#	43	O	Output Enable for driving data on CPU D0–7 onto Micro Channel D0–7 during CPU or DMA writes or Micro Channel DRAM reads.
DBE2#	44	O	Output Enable for driving data on CPU D8–15 onto Micro Channel D8–15 during CPU or DMA writes or Micro Channel DRAM reads.
DWHE#	45	O	Output Enable for driving data on CPU D16–31 onto Micro Channel D16–31.
SBLW#	41	O	Output enable for driving data on Micro Channel D0–15 onto CPU D0–15.
SBHW#	42	O	Output enable for driving data on Micro Channel D16–31 onto CPU D16–31.
MDIR	32	O	Direction control for transferring data between the CPU Data Bus and the DRAM memory data bus.
MLW#	29	O	Output enable for the transceiver between the CPU data bus D0–15 and the DRAM memory data bus.
MHW#	30	O	Output enable for the transceiver between the CPU data bus D16–31 and the DRAM memory data bus.
ROMOE#	73	O	Output Enable signal for the BIOS ROMs.
SWAP1#	19	O	Transceiver enable for transferring data between Micro Channel Data Bus 0–7 and 8–15.

82308 Micro Channel Bus Controller Pin Definitions (Continued)

Signal Name	Pin Number	I/O	Description
SWAP2#	20	O	Transceiver enable for transferring data between Micro Channel Data Bus 0–7 and 16–23.
SWAP3#	21	O	Transceiver enable for transferring data between Micro Channel Data Bus 8–15 and 24–31.
SWAP4#	22	O	Transceiver enable for transferring data between Micro Channel Data Bus 0–7 and 24–31.
SWDIR	23	O	Direction control for Micro Channel Data Bus transceivers.
DBL1	33	O	Latch enable for latching Micro Channel Data Bus 0–7.
DBL2	34	O	Latch enable for latching Micro Channel Data Bus 8–15.
DBL3	35	O	Latch enable for latching Micro Channel Data Bus 16–23.
SCLK	58	I	Microprocessor Clock.
RESET	59	I	Synchronized reset input to synchronize the internal clock with the processor phase.
RDYRTN#	80	I	Channel Ready Return signal from Micro Channel (active low).
PRDYO#	93	O	Microprocessor ready signal.
FP	68	I	Processor Speed Select. (20 MHz = 1, 16 MHz = 0.)
PRDYI#	60	I	Synchronized microprocessor ready input.
FRC#	61	I	Fast ROM Cycle Select. When tied low, ROM cycles are run as Micro Channel default read cycles. When tied high, additional wait states are inserted to accommodate slower ROMs.
RAMEN#	62	I	Decode that indicates a system board DRAM access.
FCE#	63	I	Input that directs BC to terminate a CPU system board DRAM access by generating READY# in the next clock cycle.
UM/IO#	64	I	Micro Channel Memory/IO status.
PD/C#	65	I	CPU D/C# output.
PW/R#	66	I	CPU W/R# output.
PADS#	67	I	CPU ADS# output. (Indicates address valid.)
PNA#	71	O	Next Address Signal for address pipelining.
S0#, S1#	69, 70	B	DMA Status lines; input by the BC when DMA is master, and output by the BC when CPU is master.
UADL#	78	B	Micro Channel Address Latch Signal.
UMMCMD#	40	O	Micro Channel Matched Memory Command Signal.
UCMD#	72	B	Micro Channel Command Signal.
US0#, US1#	76, 77	B	Micro Channel Status.
RSEL1 RSEL0	27, 31	I I	These two signals are used for hardware enforced I/O recovery. They are sampled at the leading edge of UCMD# during CPU initiated I/O cycles, and are used to select one of four possible I/O recovery times. At the end of the I/O cycle, an internal timer is triggered, and then times out after the selected I/O recovery time. The next I/O cycle is not allowed to proceed into the active UCMD# phase until the internal timer times out. RSEL1,0 can be strapped for a particular time, or else driven from a combinatorial address decode.

82308 Micro Channel Bus Controller Pin Definitions (Continued)

Signal Name	Pin Number	I/O	Description
FIN	26	I	Asynchronous cache flush request input. A pulse on FIN causes a cache flush. Also, if FIN is left active for a long period of time, the 82385 will be kept in flush mode for as long as FIN is active. The exception to this is when the BC is directed to do a software initiated CPU reset when FIN is active. The BC will de-activate FLUSH for a period of time surrounding the falling edge of RSTCPU so as to prevent the 82385 from entering its self-test mode. If FIN is still active after the reset, then FLUSH will be re-activated.
FLUSH	46	O	Synchronous flush request to the 82385 Cache Controller.
SNOOP#	47	B	Synchronous strobe to the cache controller to indicate valid address during a non-processor memory write. SNOOP# is sampled at reset to indicate the presence of a cache. (1 = cache present, 0 = no cache.)
PM/IO#	28	I	CPU M/IO# output.
WS#	39	I	This input, if tied low, inserts an additional wait state into CPU reads from system board memory beyond the number of wait states requested via the FCE# input. It is primarily intended for cache applications, which typically require increased CPU data setup.
NC	1, 25, 51, 52, 75		No Connect
V_{DD}	13, 37, 88		Power
V_{SS}	2, 24, 38, 74, 89		Ground



290187-24

NOTE:
NC = No Connect

82308 PARAMETRICS

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

ABSOLUTE MAXIMUM RATINGS*

Case Temperature under Bias .... −40°C to +85°C
Storage Temperature ..... −65°C to +150°C
Voltage to Any Pin with Respect to Ground ..... −0.3V to (V_{CC}+0.3)V
DC Supply Voltage (V_{CC}) ..... −0.3V to +7.0V
DC Input Current ..... ±10 mA

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

1

D.C. CHARACTERISTICS

T_C = 0°C to +70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Conditions
V_{IL}	Input Low Voltage		0.8	V	
V_{IH}	Input High Voltage	2.0		V	
V_{IL}	Input Low Voltage		0.8	V	SCLK
V_{IH}	Input High Voltage	V_{CC} − 0.8		V	SCLK
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 4 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = 4 mA
I_{CC}	Power Supply Current		180	mA	No DC Loads
I_{LI}	Input Leakage Current		±10	μA	$V_{SS} < V_{IN} < V_{CC}$
I_{OZ}	Tri-State Output Leakage Current		±10	μA	$V_{SS} < V_{OUT} < V_{CC}$

82308 Micro Channel BUS CONTROLLER A.C. SPECS

T_C = 0°C to +70°C, V_{CC} = 5V ±10%

Symbol	Parameter	82308 Kit 16 MHz		82308 Kit 20 MHz		82308HS Kit 25 MHz		C_L (pF)	Notes	Conditions
		Min	Max	Min	Max	Min	Max			
T1	SCLK PERIOD	31.25		25		20				
T2A	SCLK HIGH/LOW TIME (50%)	12		10		8				
T2B	SCLK HIGH/LOW TIME (90%)	8		6.5		6				
T3	RESET SETUP	10		10		10				
T4	RESET HOLD	4		4		4				
T5A	RSTCPU DELAY	2	30	2	30	2	27	25		
T5B	RSTCPU PULSE WIDTH	1980		1580		1260		25	14, 20	
T6A	FLUSH DELAY	5	41	5	34	5	28	50		
T6B	FLUSH PULSE WIDTH	240		190		150		50	15, 20	
T6C	SNOOP# DELAY	5	41	5	34	5	28	50		
T7	FIN PULSE WIDTH	25		25		25				
T8A	COMMAND I/O RECOVERY PULSE WIDTH	600		600		600			20	RSEL1,0 = 01
T8B	COMMAND I/O RECOVERY PULSE WIDTH	2500		2500		2500			20	RSEL1,0 = 10
T8C	COMMAND I/O RECOVERY PULSE WIDTH	10000		10000		10000			20	RSEL1,0 = 00
T8D	COMMAND I/O RECOVERY PULSE WIDTH	0		0		0			20	RSEL1,0 = 11
T9	RSEL0,1 SETUP	15		15		15			4	
T10	RSEL0,1 HOLD	20		20		20			4	
T11A	PW/R#, PD/C#, PM/IO# SETUP	25		22		13				
T11B	PADS# SETUP	25		22		17				
T12	PADS#, PW/R#, PD/C#, PM/IO# HOLD	4		4		4				
T13A	UM/IO# SETUP TO UADL# ↓	20		20		20				
T13B	UM/IO# SETUP TO SCLK	20		20		20			8	
T14A	UM/IO# HOLD FROM UADL# ↑ OR UCMD# ↓	20		16		16			7	
T14B	UM/IO# HOLD FROM SCLK	20		16		16			8	
T15	PRDYO# DELAY	4	30	4	24	3	20	75		
T16	PRDYI# SETUP	18		18		15				
T17	PRDYI# HOLD	3		3		3				
T18	PNA# DELAY	0	25	0	25	3	28	25		
T19A	UADL# DELAY (TPHL)	2	24	2	24	2	22	25		
T19B	UADL# DELAY (TPLH)	2	24	2	24	2	22	25		
T20A	UCMD# DELAY (TPHL)	2	23	2	20	2	20	25	2	
T20B	UCMD# DELAY (TPLH)	2	25	2	23	2	20	25		
T21A	UMMCMD# DELAY (TPHL)	2	23	2	20	2	20	25	2, 5	
T21B	UMMCMD# DELAY (TPLH)	2	25	2	23	2	20	25		
T22A	US0#, US1# (TPHL) DELAY	2	27	2	26	2	24	25	2	
T22B	US0#, US1# (TPLH) DELAY	2	27	2	26	2	24	25		
T22C	US0#, US1# (TPHL) DELAY FROM S0#, S1#	0	27	0	30	0	30	25		
T23	S0#, S1# SETUP	20		20		15				
T24	S0#, S1# DELAY	2	35	2	30	2	22	50		
T25	A0, A1 DELAY FROM BE0-3#	2	35	2	30	2	25	25		
T26	UBE0-3#, UA0-1#, UBHE# DELAY FROM BE0-3#, A0-1, BHE#	2	30	2	28	2	28	25		
T28	SAL DELAY	2	36	2	30	2	27	100		
T30A	SBHW#, SBLW# INACTIVE DELAY FROM SCLK	2	30	2	25	2	32	50	11, 21	
T30B	SBHW#, SBLW# DELAY FROM UADL#	2	40	2	40	2	40	50	25	

82308 Micro Channel BUS CONTROLLER A.C. SPECS

T_C = 0°C to +70°C, V_{CC} = 5V ±10% (Continued)

Symbol	Parameter	82308 Kit 16 MHz		82308 Kit 20 MHz		82308HS Kit 25 MHz		C_L (pF)	Notes	Conditions
		Min	Max	Min	Max	Min	Max			
T30C	SBHW#, SBLW# DELAY FROM UCMD#	5	35	5	35	5	35	50	10, 25	
T30D	SWDIR DELAY FROM UADL#	2	40	2	40	2	40	100	12	
T30E	DWHE# DELAY FROM UADL#	2	40	2	40	2	40	50		
T30F	DWHE# DELAY FROM UCMD#	5	45	5	45	5	45	50	25	
T30G	DBE1#, DBE2# DELAY FROM UADL#	2	40	2	40	2	40	25		
T30H	DBE1#, DBE2# DELAY FROM UCMD#	5	45	5	45	5	45	25	25	
T30I	SWAP1-4# DELAY FROM UADL#	2	40	2	40	2	40	25	25	
T30J	SWAP1-4# DELAY FROM UCMD#	5	35	5	35	5	35	25	13, 25	
T30K	SBHW#, SBLW# ACTIVE DELAY FROM SCLK	2	40	2	40	2	40	50	11	
T30L	SWDIR DELAY FROM UCMD#	5	40	5	40	5	40	100	13, 25	
T30M	SWDIR DELAY FROM DS16RTN, DS32RTN	2	35	2	35	2	35	100	12, 25	
T30N	SWAP1-4# DELAY FROM UCMD#	2	35	2	35	2	35	25	11, 25	
T31	DBL1-3 SETUP TO UCMD#, UMMCMD#	−4		−4		−4		25		
T32	WDL DELAY	2	32	2	26	2	21	50		
T33	DBL1-3 DELAY ↑	2	27	2	25	2	22	25		
T34A	MHW#, MLW# ACTIVE DELAY FROM SCLK ↑	2	30	2	25	2	25	50	16	
T34B	MHW#, MLW# INACTIVE DELAY FROM SCLK ↑	2	30	2	25	2	32	50	17, 21, 24	
T34C	MDIR DELAY FROM SCLK	2	40	2	40	2	40	125	22, 24	
T34D	MHW#, MLW# DELAY FROM UADL#	2	45	2	45	2	45	50	25	
T34E	MHW#, MLW# DELAY FROM UCMD#	5	40	5	40	5	40	50	25	
T34F	MDIR DELAY FROM UADL#	2	45	2	45	2	45	125	25	
T34H	ROMOE# DELAY FROM SCLK	2	28	2	28	2	28	50		
T34I	MHW#, MLW# ACTIVE DELAY FROM SCLK ↓	2	35	2	35	2	35	50	18, 23	
T34J	MHW#, MLW# INACTIVE DELAY FROM SCLK ↑	2	35	2	35	2	35	50	19, 24	
T35	FCE# SETUP	15		15		7				
T36	FCE# HOLD	3		3		3				
T37	ROMEN#, RAMEN# SETUP	20		19		19				
T38A	DS16RTN, DS32RTN SETUP TO SCLK	30		30		25			6	
T38B	DS16RTN, DS32RTN SETUP TO UADL# ↑	15		15		15			3	
T41	RDYRTN# SETUP TO SCLK	10		10		10			9	
T42	RDYRTN# HOLD FROM SCLK	4		4		4				
T43	IOEN#, FDACK# SETUP TO UADL# ACTIVE	10		10		10			25	
T44A	P0-P3 SETUP TO SCLK	0		0		0			11	
T44B	P0-P3 SETUP TO UCMD#	3		3		3			13, 25	
T45	P0-P3 HOLD	12		12		12				
T46A	CHRDY DELAY FROM IOEN#	2	30	2	30	2	30	25		
T46B	CHRDY DELAY FROM STATUS	2	25	2	25	2	25	25	25	
T46C	CHRDY ACTIVE FROM MB COMMAND	100		90		70		25	1, 20	
T47A	MB COMMAND DELAY FROM UCMD# ACTIVE	75		75		75		100	1, 20	
T47C	MB COMMAND DELAY FROM UCMD# INACTIVE	3	40	3	40	3	40	100	1, 25	
T48	MB COMMAND PULSE WIDTH	250		225		190		100	1, 12, 20	

NOTES:
1. MB Commands include IOR, IOW, INTA#, VMRD# and VMWR#.
2. These specs and cycle edge definitions support a worst case "effective" data setup of 40 ns for a 385 system at 20 MHz. (Effective setup means setup to the "386-like" front end created by the 385.)
3. Spec applies only when master resides on Micro Channel.
4. RSEL0,1 should be tied high or low, or else driven from a combinatorial address decode.
5. UMMCMD# is only driven when the CPU is master.
6. Applies only when CPU is master.
7. T14A applies to the later of ADL# ↑ or CMD# ↓.
8. T13B, T14B apply to CPU or DMA master.
9. RDYRTN# is an asychronous input. Meeting T41 simply guarantees recognition at a particular clock edge.

NOTES:
10. Applies when DMA is master.
11. Applies when CPU is master.
12. Applies when CPU or DMA is master.
13. Applies when DMA or channel master is master.
14. Functional Spec . . . Not Tested (= 64 SCLK Periods).
15. Functional Spec . . . Not Tested (= 8 SCLK Periods).
16. READS AND PIPELINED WRITES
17. READS
18. NON-PIPELINED WRITES
19. WRITES
20. Functional Spec . . . Not Tested

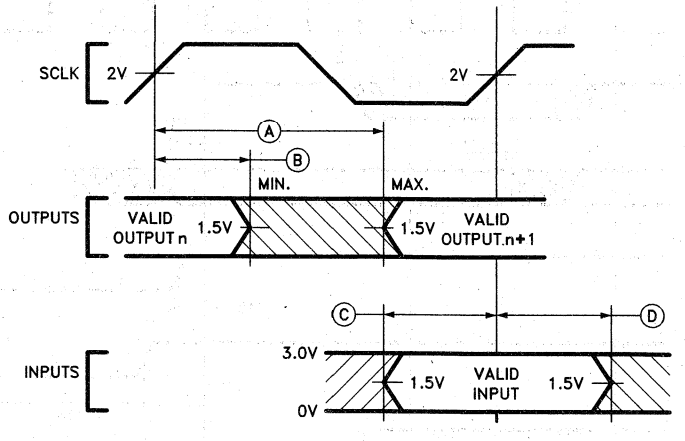
NOTES (82308HS-25 ONLY):
21. Contention will not occur when a write immediately follows a read because the 82385 insures at least one BTI state between a read followed by write sequence during which the data bus remains tri-stated.
22. MDIR is generated and speced from SCLK ↑ instead of from SCLK ↓ at it is in the 82308-16/20.
23. In non-piped writes, MHW# and MLW# are generated from the phase 2 SCLK ↑ edge rather than from SCLK ↓ as in the 82308-16/20.
24. Since MDIR toggles from the same clock edge that MHW# and MLW# are de-asserted from, contention is prevented by 82308-25 internal design such that MDIR is guaranteed to be slower than MHW# or MLW#.
25. Only tested when UADL#, UCMD#, US0#, and US1# are inputs; i.e., when Micro Channel is master.

82308SX Micro Channel Bus Controller A.C. Specs

The following spec is the only exception to the A.C. specs listed above.

Symbol	Parameter	82308SX			
		16 MHz		20 MHz	
		Min	Max	Min	Max
t_{18}	PNA# Delay	0	20	0	20

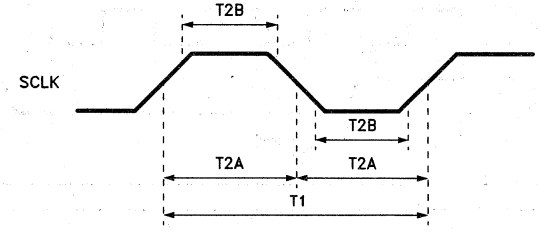
DRIVE LEVELS AND MEASUREMENT POINTS FOR A.C. SPECIFICATIONS



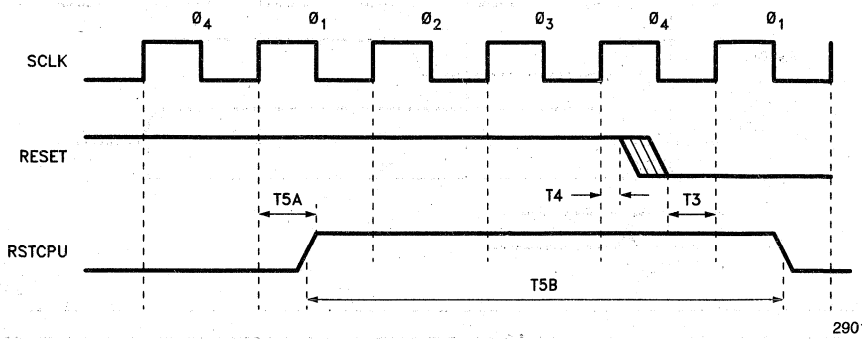
290187-25

LEGEND:
A. Maximum Output Delay Specification.
B. Minimum Output Delay Specification.
C. Minimum Input Setup Specification.
D. Minimum Input Hold Specification.

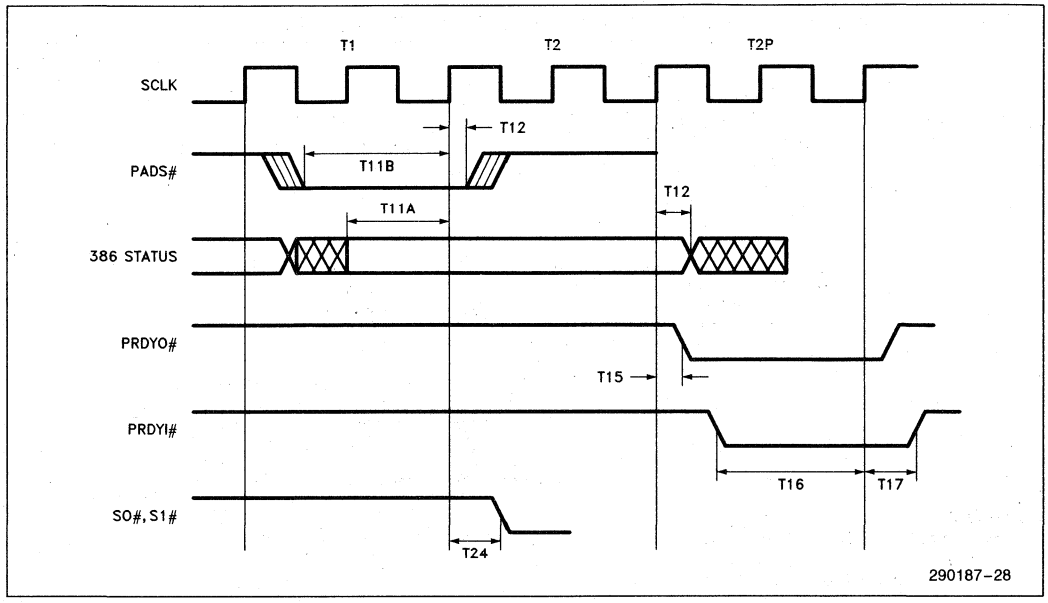
Input waveforms have tr ≤ 2.0 ns from 0.8V to 2.0V.

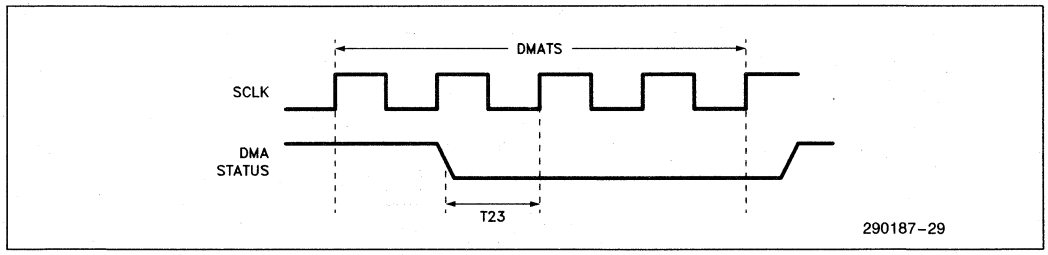


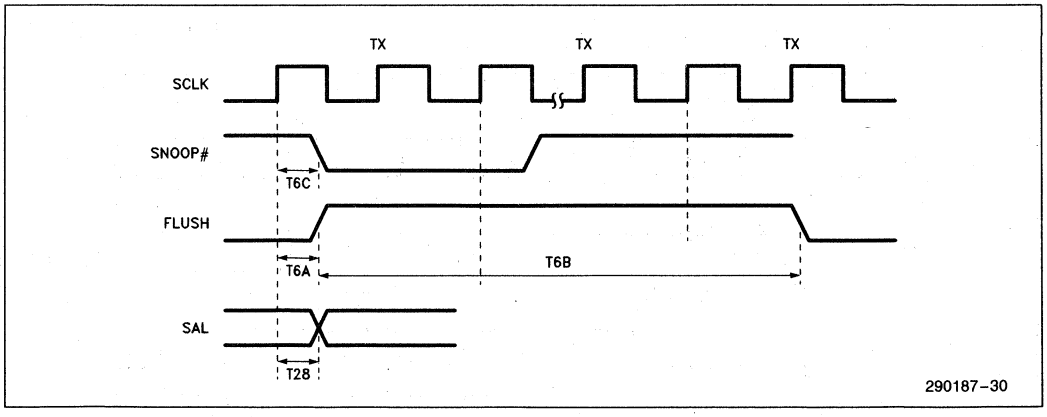
290187-26

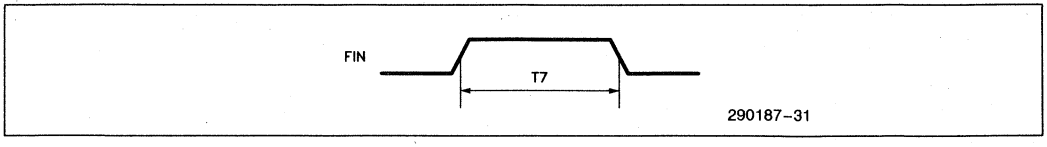


290187-27

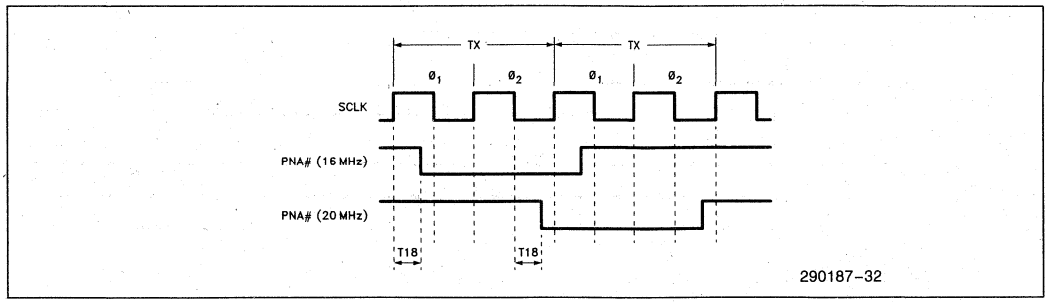

T1
T2
T2P
SCLK
T12
PADS#
T11B
T12
T11A
386 STATUS
PRDYO#
T15
PRDYI#
T16
T17
SO#, S1#
T24
290187-28

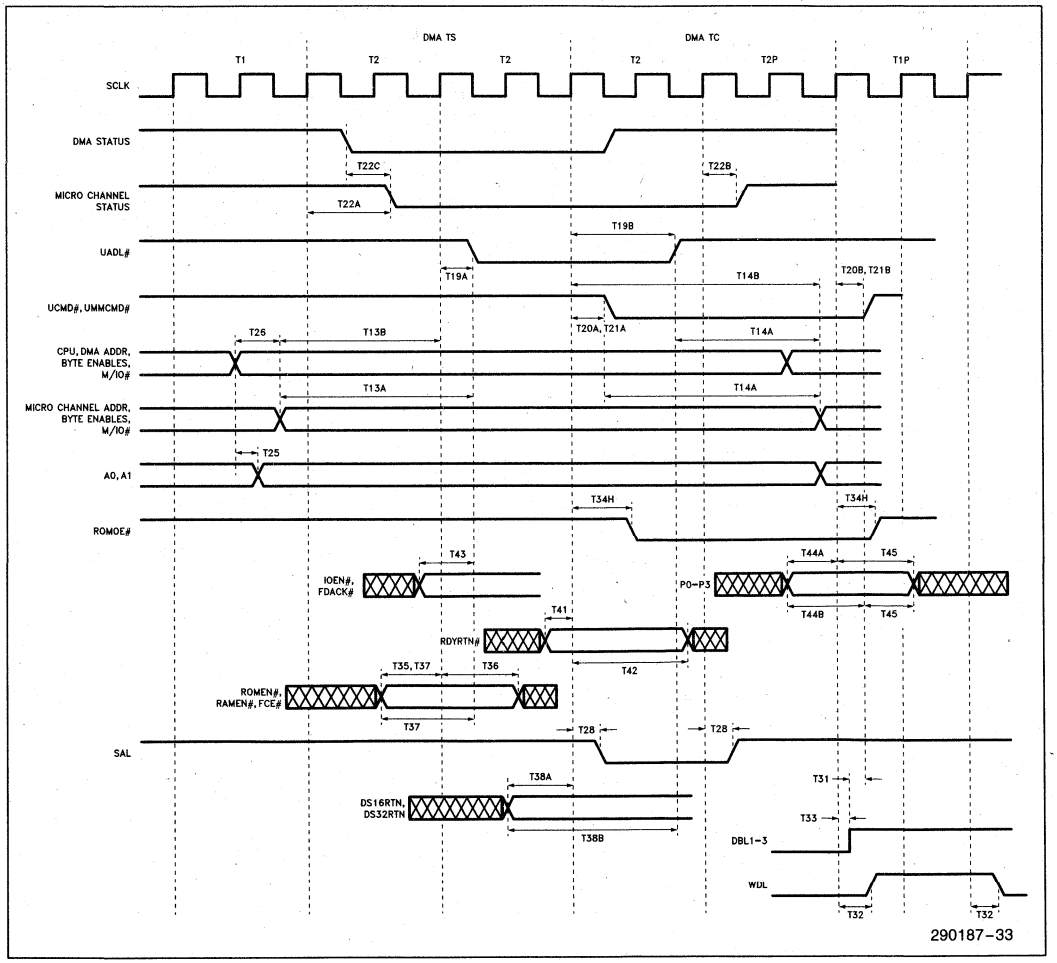

DMATS
SCLK
DMA STATUS
T23
290187-29

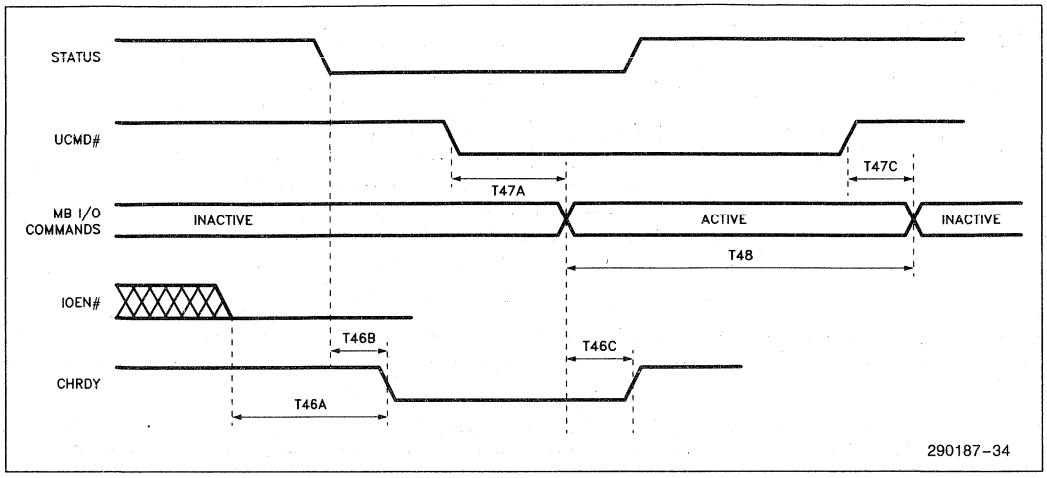

TX
TX
TX
SCLK
SNOOP#
T6C
FLUSH
T6A
T6B
SAL
T28
290187-30


FIN
T7
290187-31

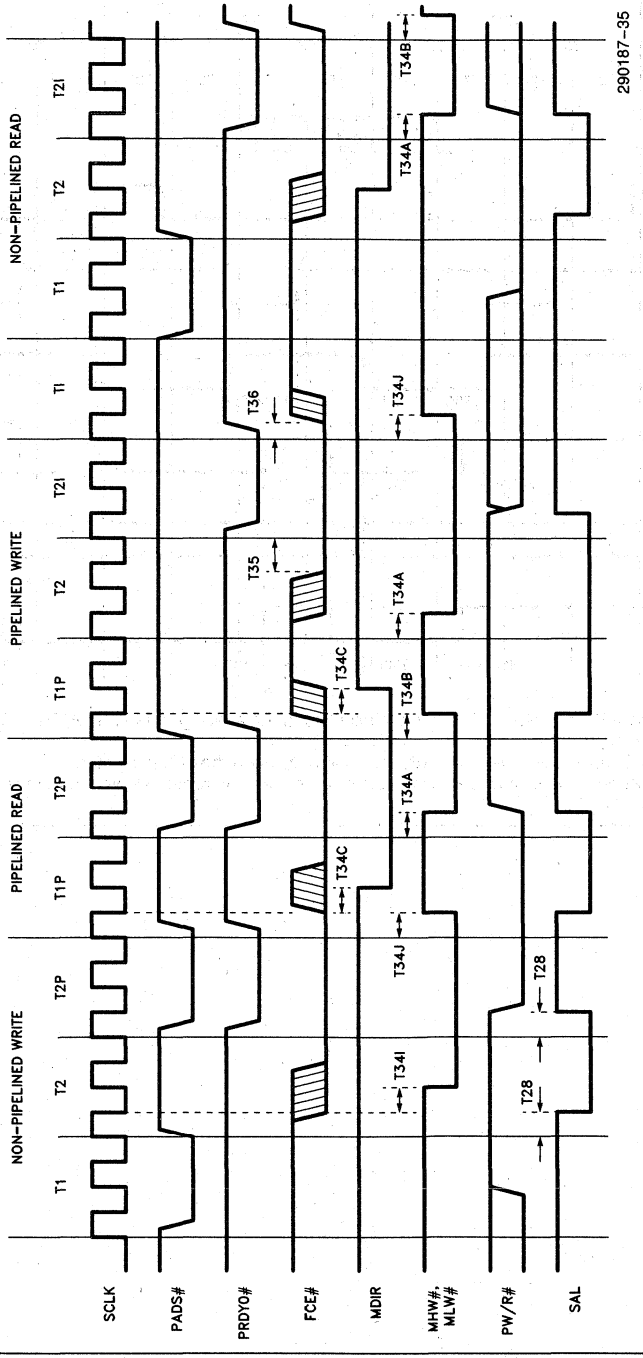
PNA# TIMING


TX
TX
Ø1
Ø2
Ø1
Ø2
SCLK
PNA# (16 MHz)
PNA# (20 MHz)
T18
T18
290187-32

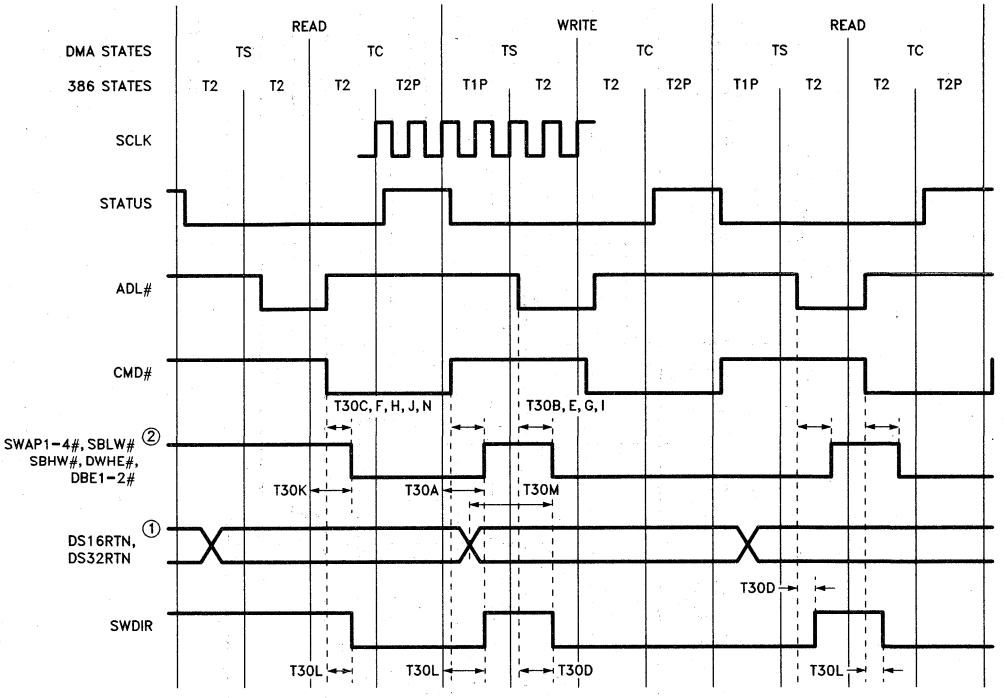

DMA TS
DMA TC
T1
T2
T2
T2
T2P
T1P
SCLK
DMA STATUS
T22C
T22B
MICRO CHANNEL STATUS
T22A
T19B
UADL#
T19A
T14B
T20B, T21B
UCMD#, UMMCMD#
T20A, T21A
T14A
T26
T13B
CPU, DMA ADDR, BYTE ENABLES, M/IO#
T13A
T14A
MICRO CHANNEL ADDR, BYTE ENABLES, M/IO#
T25
A0, A1
T34H
T34H
ROMOE#
T43
T44A
T45
IOEN#, FDACK#
P0-P3
T44B
T45
T41
RDYRTN#
T42
T35, T37
T36
ROMEN#, RAMEN#, FCE#
T37
T28
T28
SAL
T38A
T31
DS16RTN, DS32RTN
T33
DBL1-3
T38B
WDL
T32
T32
290187-33


STATUS
UCMD#
T47A
T47C
MB I/O COMMANDS
INACTIVE
ACTIVE
INACTIVE
T48
IOEN#
T46B
T46C
CHRDY
T46A
290187-34

MEMORY DATA BUFFER CONTROL
80386 ACCESSES TO MOTHERBOARD DRAM



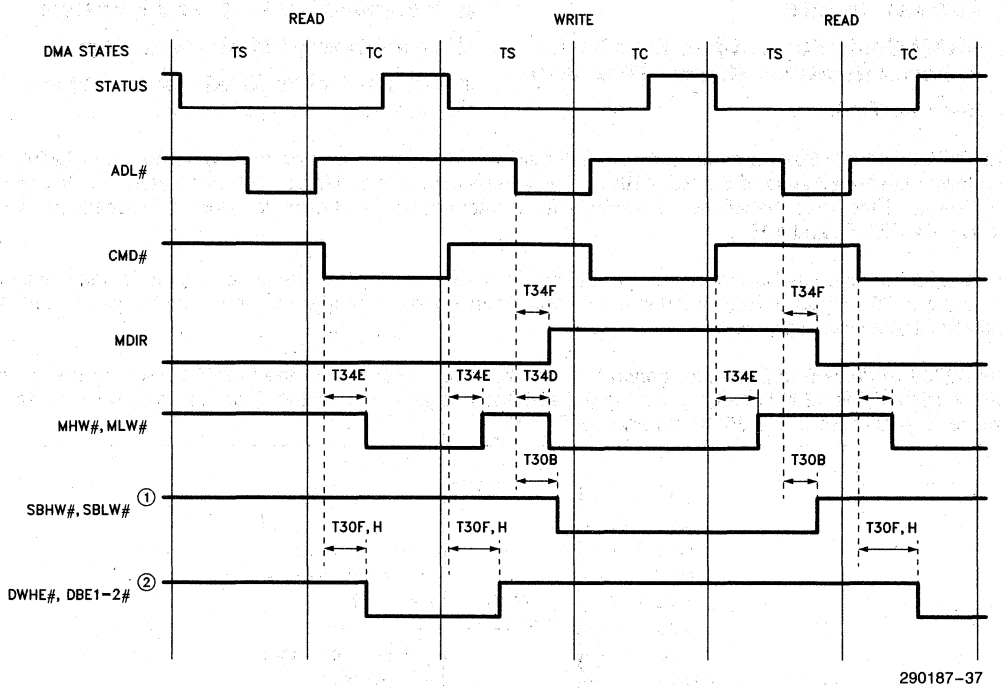
Micro Channel DATA BUFFER CONTROL 386/DMA MASTER, DATA STEERING FOR CHANNEL MASTER TO CHANNEL SLAVE



290187-36

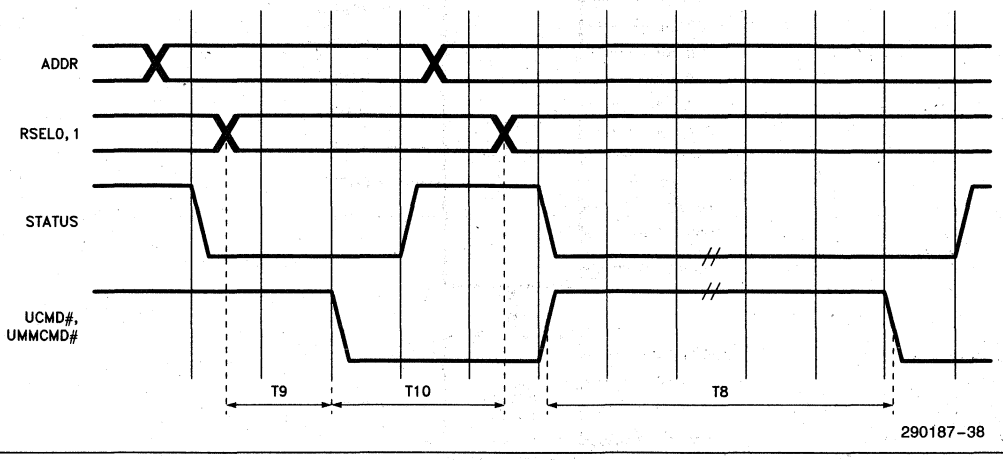
①T30D applies to SWDIR only when all involved combinatorial data size inputs (DS16RTN, DS32RTN) are available early enough not to delay SWDIR.
②T30A and T30K apply only to SBHW# and SBLW#, and only when the 386 is master.
③T30B applies here when DMA data is routed to system board memory from the lower half of the data bus (D0–D15) to the upper half (D16–D31).

MEMORY/Micro Channel DATA BUFFER CONTROL DMA/CHANNEL MASTER ACCESSES TO MB DRAM



① Channel Master writes to MB DRAM.
② Channel Master reads from MB DRAM.

(HARDWARE ENFORCED) I/O RECOVERY TIMING



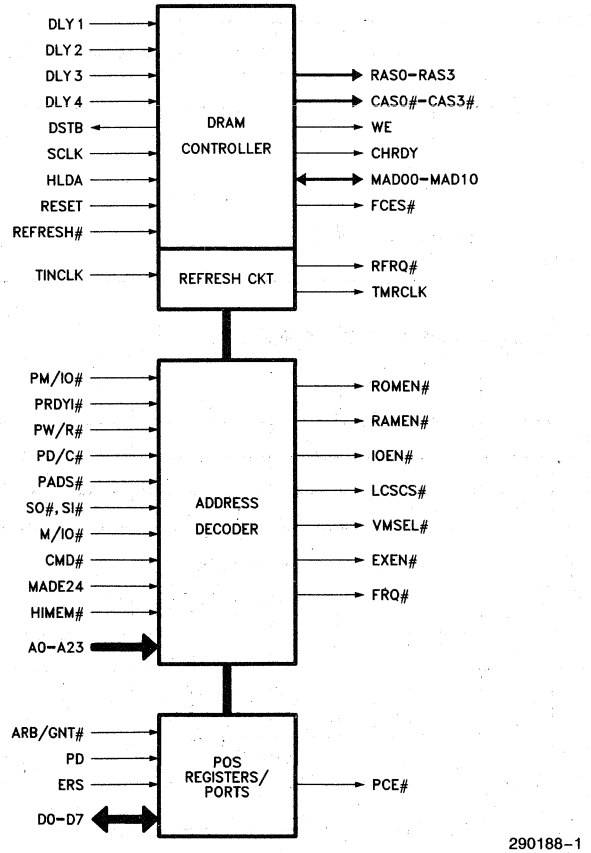
82309
ADDRESS BUS CONTROLLER

- **Address Decoder**
- **DRAM Controller . . . Up to Four Banks of Page Interleaved Memory (Max 16M)**
- **Refresh Timer**
- **Integrated I/O Ports and Registers**
- **Low Power CHMOS Technology**
- **100-Pin Plastic Quad Flat Packaging**
  (See Packaging Spec., Order # 231369)

The 82309 Address Bus Controller provides Address decoding for devices on the motherboard, including the shadowed DRAM address of the ROM BIOS. The Address Bus Controller also has integrated DRAM controller, Refresh Timer and miscellaneous registers for memory control and error recovery, specifically ports E0, E1, E3, E4, E5, E7 and 103.

The 82309 Address Bus Controller provides the designer several price/performance choices for the configuration of up to 16 MBytes of Page Interleave DRAM memory on the motherboard. Up to four banks of 256K, 1M and 4M DRAMs are supported.

The 82309 Address Bus Controller generates periodic refresh requests to the 82307 DMA controller to run refresh cycles. The 82309 does not use the Refresh Address generated by the DMA controller but provides its own refresh address to the 256K, 1M and 4M DRAMs.



290188-1

October 1990
Order Number: 290188-003

PORTS AND REGISTERS

Configuration bits SS1 and SS2 control the function of the Ports and Register Block. The Ports and Register Block, in turn, control the function of the Refresh Timer and the address mapping of the motherboard DRAMs and the BIOS EPROMs.

SS1 and SS2 essentially select one of four definitions of the memory encoding registers (E0, E1), error trace registers (E3, E4, E5, E7), and motherboard POS setup port (103). These defintions are depicted in Table 0, and go by the names System A, System B, System C and System D.

System A presents a Model 50/60 compatible definition of these ports. Specifically, Port 103 is defined as it is in the IBM PS/2 Model 50/60 Technical Reference and ports E0–E7 are non-existent. System B presents a Model 80 compatible definition of these ports, as detailed in the IBM PS/2 Model 80 Technical Reference.

System B has a limitation in that due to the definition of the card enable bits in ports E0 and E1 (described later), it is limited to 4 Mbytes of system board memory. System C overcomes this by making the card enable bits "free form"; i.e., accessible as read/write bits, but otherwise meaningless in terms of their effect on the system. System C allows a Model 80 type system to provide up to 16 Mbytes of system board memory.

System D provides a Model 50/60 compatible definition of port 103 and a Model 80 compatible definition of ports E0–E7. This system is targeted for designs that wish to present a Model 50/60 port definition, but wish to make use of features provided in the Model 80 register set, specifically the ability to copy ROM into RAM for increased performance. This system requires external logic (approx. ½ of a 16L8 PAL) that essentially makes E0–E7 disappear from a software point of view once the ROM has been copied into RAM. (Details are provided in the *82311 Micro Channel Compatible Peripheral Chip Set Designer's Guide.*)

In systems A and D, bit 0 (the Memory Enable Bit) is the only accessible bit in Port 103. This bit can be accessed via channel I/O Read and/or Write operations. When Port 103 is read only bit 0 is driven, all other data bus bits remain tristated. This bit is set to a 1 by RESET. When the Memory Enable Bit = 0 all of the motherboard DRAM is disabled (but still refreshed). In both systems A and D, the refresh timer produces a 400 ns pulse every 15.12 μs. In system D, mapping is controlled by ports E0 and E1, as described in a moment. In system A, the other functions of the ports and register block are as follows:

— The Split in the first megabyte is located at 640 Kbytes.

— If the motherboard DRAM space equals 16 Mbytes then the remaining DRAM is disabled, otherwise the remaining 384 Kbytes are remapped to the first 384 Kbytes past the end of

Table 0. Configuration Bits SS1, SS2 Definition

Config Bits		System	Description
SS1	SS2		
0	0	A	Model 50/60 Compatible Port 103(1) Registers E0–E7 Non-Accessible
1	0	B	Model 80 Compatible Port 103 Error Trace Registers E3, E4, E5 and E7 Accessible Memory Encoding Registers E0 and E1 Accessible Compatible Card Enable Bits in E0 and E1
1	1	C	Model 80 Compatible Port 103 Error Trace Registers E3, E4, E5 and E7 Accessible Memory Encoding Registers E0 and E1 Accessible Free Form Card Enable Bits In E0 and E1
0	1	D	Model 50/60 Compatible Port 103 Error Trace Registers E3, E4, E5 and E7 Accessible (But Not Typically Used) Memory Encoding Registers E0 and E1 Accessible Free Form Card Enable Bits In E0 and E1

NOTES:
1. Port 103 is a motherboard POS port; i.e., accessible only when the motherboard is in Setup Mode.

the motherboard DRAM address space. (i.e., if there are 4 Mbytes of DRAM then the split is remapped to address 00400000 → 0045FFFF.)

— The BIOS EPROMs are mapped to both 000E0000 → 000FFFFF and FFFE0000 → FFFFFFFF.

In systems B and C, port 103 is defined as follows:

— Port 103 bit 0 (the Memory Enable Bit) is not accessible.

— Port 103 bit 1 (the Refresh Rate Bit) is accessible for write operations only. If this bit is a 1 then the Refresh Timer produces an approximately 400 ns long pulse once every 15.12 μs. If this bit is a 0 then the Refresh Timer produces a continuous stream of 400 ns pulses with a period of approximately 800 ns. This bit is set to a 1 by RESET.

In systems B, C and D, ports E0, E1, E3, E4, E5 and E7 are defined as follows:

— Four of the Read only Micro Channel Error Trace Registers (Ports 00E3, 00E4, 00E5 and 00E7) are accessible. (Typically, a system D design will not utilize these registers and will thus not require any external logic to implement any error register support.) These registers sample SA<02:23>, M/IO#, D/C# and ARB/GNT# on every rising edge of the ERS input pin. The bit assignments for these registers are as follows:

Bit	00E3	00E4	00E5	00E7
7	SA23	SA15	SA07	—
6	SA22	SA14	SA06	—
5	SA21	SA13	SA05	—
4	SA20	SA12	SA04	—
3	SA19	SA11	SA03	—
2	SA18	SA10	SA02	—
1	SA17	SA09	M/IO#	—
0	SA16	SA08	ARB/GNT#	D/C#

These four registers are all set to 00 by RESET. When Register E7 is read, only data bus bit 0 is driven by the ABC, data bus bit 1–7 remain tristated.

— Registers E0 and E1 are accessible via the channel for both I/O read and I/O write operations. These two registers control the address mapping of both the motherboard DRAMs and the BIOS EPROMs. (The reset state of E0 and E1 is FF.) The two most significant bits of both of these registers are free form register bits and have no effect on the functioning of the ABC. The functioning of the two next most significant bits (bits 5 & 4, the card enable bits) of both of these registers are controlled by configuration bits SS2 and SS1 as discussed in a moment.

— The four least significant bits (bits 3, 2, 1 & 0) of register E1 are defined as follows:

Bit 3	2	1	0	
0	.	.	.	- Memory beyond split Enabled
1	.	.	.	- Memory beyond split Disabled
.	0	.	.	- Split is at 640K (000A0000)
.	1	.	.	- Split is at 512K (00080000)
.	.	0	.	- BIOS ROMs deactivated in 000E0000 to 000FFFFF BIOS ROMs active in FFFE0000 to FFFFFFFF Shadow RAM Write Protected
.	.	1	.	- BIOS ROMs active in 000E0000 to 000FFFFF BIOS ROMs active in FFFE0000 to FFFFFFFF Shadow RAM Writeable.
.	.	.	0	- Parity Checking enabled
.	.	.	1	- Parity Checking disabled

If the memory beyond the Split is enabled by bit 3 then the four least significant bits (bits 3, 2, 1 & 0) of register 00E0 define the address range in memory where the portion of the first megabyte of system RAM beyond the split will be remapped. Bits 3, 2, 1 & 0 of this register correspond to address bits 23, 22, 21 & 20 of the remap location for this memory.

Bit 2 of register E1 defines the partitioning of the first megabyte of the motherboard DRAM. Figure 0 details the effect of this bit. The S in the remap addresses represents the value of the four least significant bits of register E0.

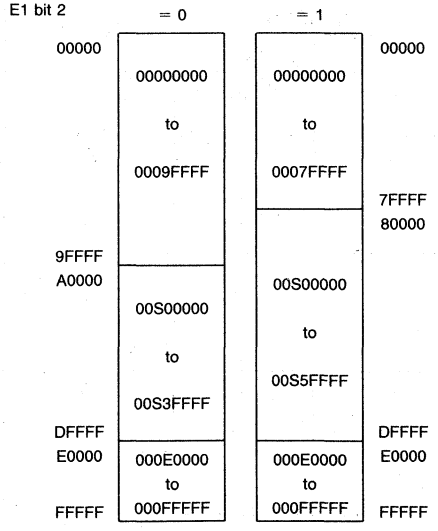


Figure 0. Partition of First Megabyte of DRAM

There is DRAM mapped in the address range 000E0000 to 000FFFFF. The function of this DRAM is controlled by bit 1 of register E1. If bit 1 = 1 then this RAM is writeable but not readable (thus the BIOS EPROMs can be Shadowed by Reading and Writing to the same address). If bit 1 = 0 then the BIOS EPROMs are disabed and this area of RAM is read enabled but write protected.

When bit 1 of register E1 is a 1 both the ROMEN# and the RAMEN# signals will respond to accesses in the range 000E0000 to 000FFFFF. In this way, the 82308 Bus Controller knows to direct reads to ROM and writes to RAM to allow shadowing. (In system D, if memory is disabled via bit 0 of port 103, then ROM is enabled in 000E0000 to 000FFFFF, regardless of the status of bit 1 in E1.)

Bit 0 of register E1 is output to the Bus Controller on the PCE# pin for use as an (active low) Parity Check Enable control bit.

— In system B, the amount of the physical motherboard DRAM that is accessable is controlled by the card enable bits (bits 5 and 4 of registers E0 and E1). These four bits act as enables (active low) for each of the first four megabytes of the physical motherboard DRAM space. Any additional DRAM controlled by the ABC will be refreshed but is otherwise disabled.

Register				Function
E0		E1		
Bit 5	Bit 4	Bit 5	Bit 4	
0	X	X	X	Megabyte #3 Enabled
1	X	X	X	Megabyte #3 Disabled
X	0	X	X	Megabyte #2 Enabled
X	1	X	X	Megabyte #2 Disabled
X	X	0	X	Megabyte #1 Enabled
X	X	1	X	Megabyte #1 Disabled
X	X	X	0	Megabyte #0 Enabled
X	X	X	1	Megabyte #0 Disabled

All megabytes that are enabled by these bits are mapped into one continuous block (with the exception of the Split from the first active megabyte) starting at address 00000000. (Thus if megabyte #0 is disabled, then the rest of the megabytes are remapped down to the range 00000000 to 002FFFFF, etc.)

In systems C and D, bits 5 and 4 of both registers E0 and E1 are free form register bits and have no effect on the functioning of the ABC.

NOTE:
During pipelined mode, an accidental write to Port E0h with E1h data can occur when the latched addresses (A2:A0) are allowed to change to the next address (pipelined) before the write strobe (UCCMD#) goes inactive. For example, this can occur when (A2:A0) change from "000" to "001" while UCCMD# is active. This condition is not seen on designs which use the Intel Cache Controller (82385), since it does not allow pipelining for I/O write cycles. In systems without an 82385, UCMMD# must be synchronized prior to being input to the 82309 CMD# pin. The workaround that follows provides a comprehensive solution independent of software sequence. UCMMD# must be "ORed" with the result of the inverted 82308 IOW signal "NORed" with DMARDY#. DMARDY# is synchronized from the 82308 PRDYO# signal. The latch that generates DMARDY# is shown on Sheet 4 in the 82311 Micro Channel Peripheral Chip Set Designer's Guide schematics.

DRAM CONTROLLER

The DRAM controller supports page interleaved memory designs in the configurations shown in Table 1. This table also details which channel address bits map to which DRAM address bits. Note that even though options D and G are two-bank options, the ABC thinks of these banks as 0 and 2, not banks 0 and 1, i.e., use RAS0, RAS2, CAS0# and CAS2#.

Table 1 describes the basic memory configurations A through N. However, a wide variety of additional options can be easily realized by building on A through N with minimal external address decode logic. These additional options include the ability to mix DRAM types (for example 256K and 1M DRAMs in the same system), and allow for a great deal of flexibility in memory upgrade paths. Examples of how to do this are included in the *82311 Micro Channel Compatible Peripheral Chip Set Designer's Guide*.

Figure 1 shows the BHE#, BE#, and A0 latch for the 82309 to DRAM interface.

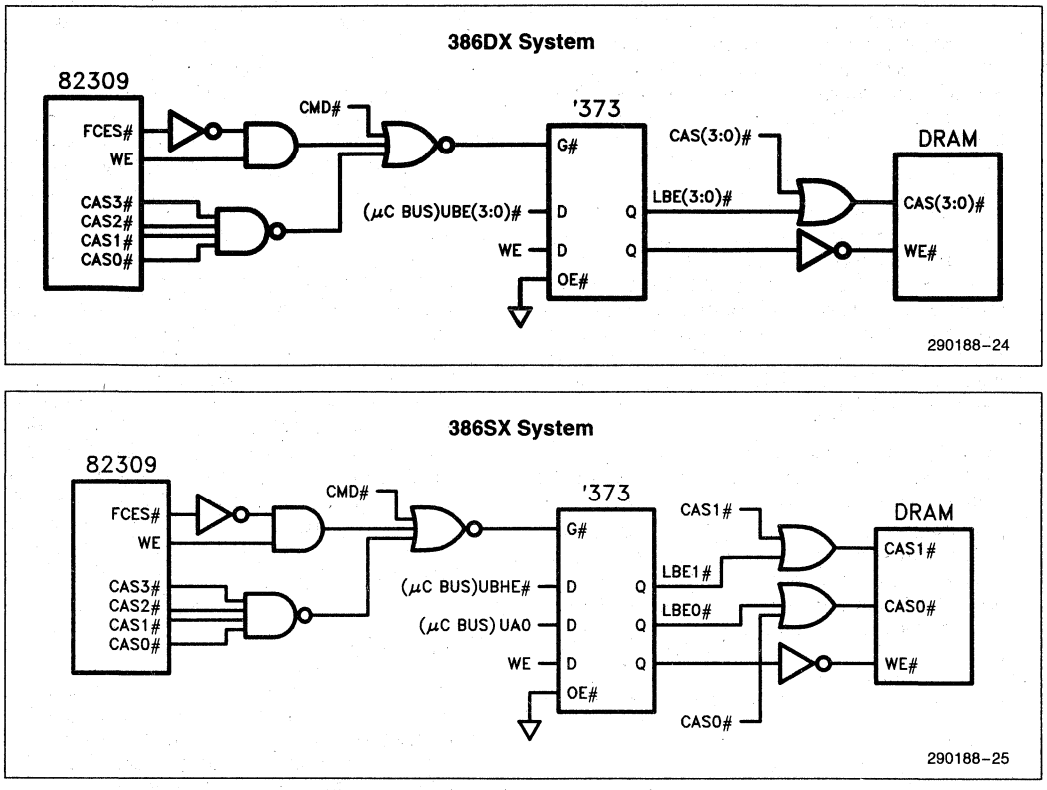


Figure 1. BHE#, BE#, A0 Latch Control

Table 1. Memory Configuration Options and Channel Address-To-DRAM Address Mapping

Opt	Size	Memory Configuration Options	Page Size
A	1M	1 Bank of 256K DRAMs (x 32) Page Mode	512
B	1M	2 Banks of 256K DRAMs (x 16) Page Mode	512
C	2M	1 Bank of 1M DRAMs (x 16) Page Mode	1024
D	2M	2 Banks of 256K DRAMs (x 32) Page Mode	512
E	2M	4 Banks of 256K DRAMs (x 16) Page Mode	512
F	4M	1 Bank of 1M DRAMs (x 32) Page Mode	1024
G	4M	2 Banks of 1M DRAMs (x 16) Page Mode	1024
H	4M	4 Banks of 256K DRAMs (x 32) Page Mode	512
I	8M	1 Bank of 4M DRAMs (x 16) Page Mode	2048
J	8M	2 Banks of 1M DRAMs (x 32) Page Mode	1024
K	8M	4 Banks of 1M DRAMs (x 16) Page Mode	1024
L	16M	1 Bank of 4M DRAMs (x 32) Page Mode	2048
M	16M	2 Banks of 4M DRAMs (x 16) Page Mode	2048
N	16M	4 Banks of 1M DRAMs (x 32) Page Mode	1024

Opt	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
A					Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws		
B					Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Bs	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	
C				Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	
D				Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Bs	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws		
E				Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Bs	Bs	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	
F			Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws		
G			Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Bs	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	
H			Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Bs	Bs	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws		
I		Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	
J		Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Bs	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws		
K		Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Bs	Bs	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	
L	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws		
M	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Bs	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	
N	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Ps	Bs	Bs	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws	Ws		

Ps ≥ Page Select Ws ≥ Word Select Bs ≥ Bank Select

NOTE:
Options A, C, F, I & L use Bank 0
Options D & G use Bank 0 & 2
Options B, J & M use Bank 0 & 1
Options E, H, K & N use all Banks

Opt	Ps											Ws										Bs	
	10	09	08	07	06	05	04	03	02	01	00	10	09	08	07	06	05	04	03	02	01	00	01	00
A	><	><	11	12	13	19	18	17	16	15	14	><	><	02	10	03	09	08	07	06	05	04	><	><
B	><	><	11	12	13	19	18	17	16	15	14	><	><	02	01	03	09	08	07	06	05	04	><	10
C	><	11	20	12	13	19	18	17	16	15	14	><	01	02	10	03	09	08	07	06	05	04	><	><
D	><	><	20	12	13	19	18	17	16	15	14	><	><	02	10	03	09	08	07	06	05	04	11	><
E	><	><	20	12	13	19	18	17	16	15	14	><	><	02	01	03	09	08	07	06	05	04	11	10
F	><	21	20	12	13	19	18	17	16	15	14	><	11	02	10	03	09	08	07	06	05	04	><	><
G	><	21	20	12	13	19	18	17	16	15	14	><	01	02	10	03	09	08	07	06	05	04	11	><
H	><	><	20	21	13	19	18	17	16	15	14	><	><	02	10	03	09	08	07	06	05	04	11	12
I	22	21	20	12	13	19	18	17	16	15	14	01	11	02	10	03	09	08	07	06	05	04	><	><
J	><	21	20	22	13	19	18	17	16	15	14	><	11	02	10	03	09	08	07	06	05	04	><	12
K	><	21	20	22	13	19	18	17	16	15	14	><	01	02	10	03	09	08	07	06	05	04	11	12
L	23	21	20	22	13	19	18	17	16	15	14	12	11	02	10	03	09	08	07	06	05	04	><	><
M	23	21	20	22	13	19	18	17	16	15	14	01	11	02	10	03	09	08	07	06	05	04	><	12
N	><	21	20	22	23	19	18	17	16	15	14	><	11	02	10	03	09	08	07	06	05	04	13	12

Typically, zero wait state pipelined page hit performance can be achieved at 16 MHz using 100 ns or 120 ns DRAMs, resulting in an aggregate of 0.5 to 0.8 wait states on average. The same DRAMs at 20 MHz will yield 1 wait state page hits.

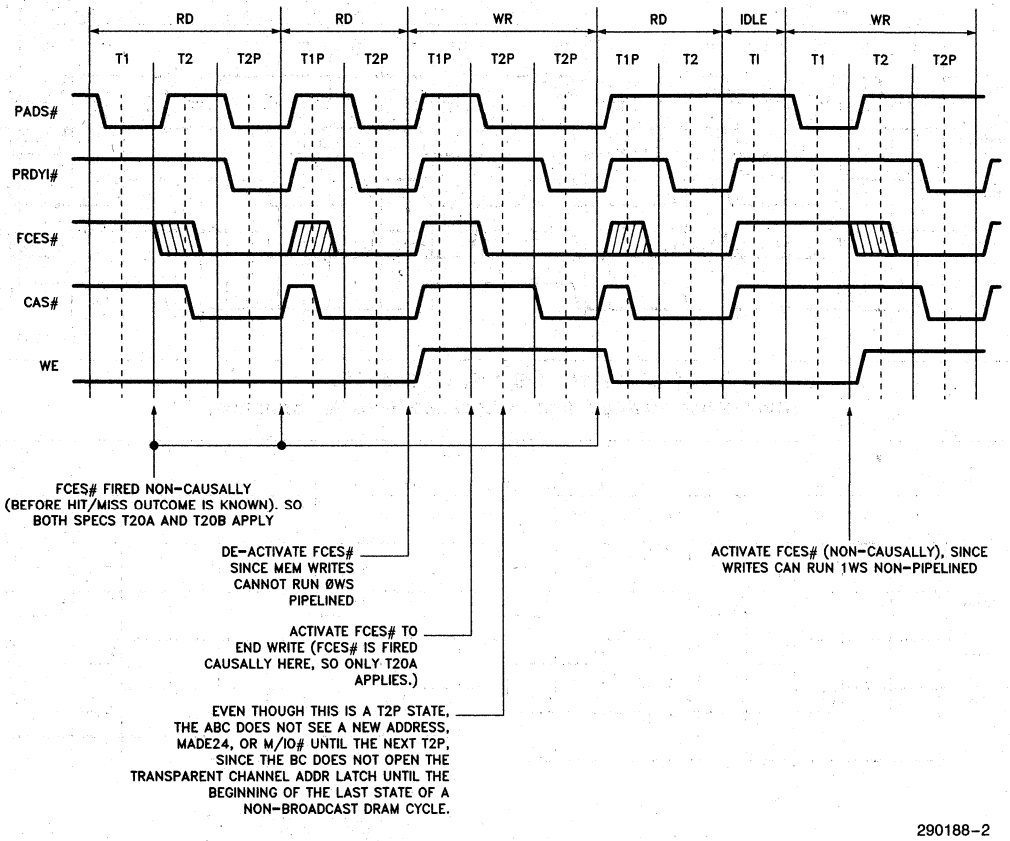
At power-up, the 82309 Address Bus Controller samples its memory address bus to determine the desired system configuration. (This operation is described in detail later in the data sheet under "MAD BUS RESET CONFIGURATION".) The three configuration switches, C0, C1 and C2 are used to select a specific performance level as measured in page hit/page miss wait states. DRAM selection involves not only selecting a DRAM, but also choosing delay line taps to control the sequence of DRAM control signals, and then choosing the performance level that can be reliably supported using a particular DRAM and set of delay line taps. The next several pages describe all the available configuration options, and following this is a DRAM/Delay Tap selection guide along with some sample calculations.

Table 2. 82309 ABC Configuration and CPU Performance(3)

	Config Inputs			Pipelined Read		Pipelined Write		Non-Pipelined Read		Non-Pipelined Write		Reference Figures
	C0	C1	C2	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
(1)	0	0	0	0	2	1(2)	2	1	3	1(2)	3	1, 4
(1)	0	0	1	0	3	1(2)	3	1	4	1(2)	4	1, 4
(1)	0	1	0	0	4	1(2)	4	1	5	1(2)	5	1, 4
	0	1	1	1	4	1	4	2	5	2	5	2, 5
	1	0	0	1	5	1	5	2	6	2	6	2, 5
	1	0	1	1	6	1	6	2	7	2	7	2, 5
	1	1	0	1	7	1	7	2	8	2	8	2, 5
	1	1	1	2	7	2	7	3	8	3	8	3, 5

NOTES:

1. These three configuration options feature 0WS pipelined page read hits. Strapping for one of these directs the ABC to determine whether a cycle is a page hit or miss, and to generate CAS# one SCLK phase earlier then the other options. Hence, these options are only supported at 16 MHz.
2. Note that both pipelined and non-pipelined write page hits run 1WS in these three configuration options.
3. The ABC completely controls the wait state counts in memory cycles according to this table via its FCES# output. The BC can however, (via its WS# strap) insert an additional wait state beyond those stated above in memory reads. (The WS# strap is intended for cache systems, which typically require additional data setup.)



NOTE:
Once the ABC fires FCES# to terminate a DRAM cycle, it remains active until the ABC detects the actual CPU end of cycle via the PRDYI# input. This allows the BC or any other external device to extend DRAM cycles if desired.

Figure 2. 0/2, 0/3, 0/4 Page Hits
(Cycles Named According to Pipelined Read Performance)

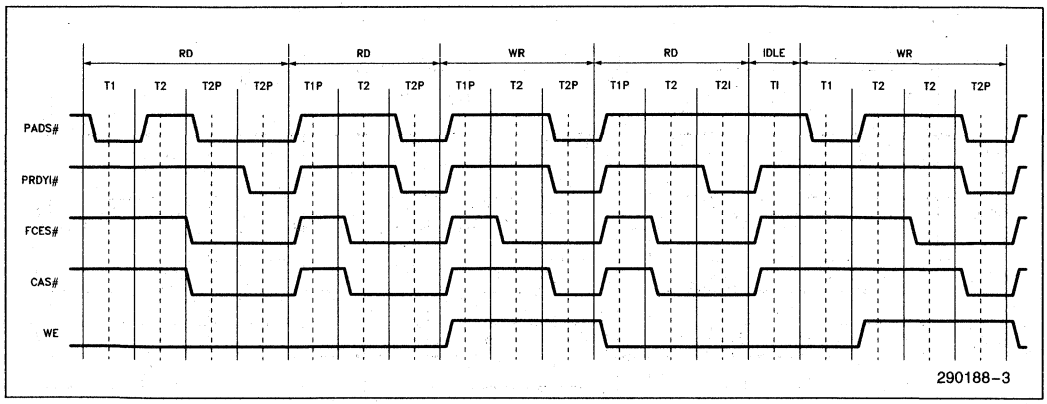


Figure 3. 1/4, 1/5, 1/6, 1/7 Page Hits (Cycles Named According to Pipelined Read Performance)

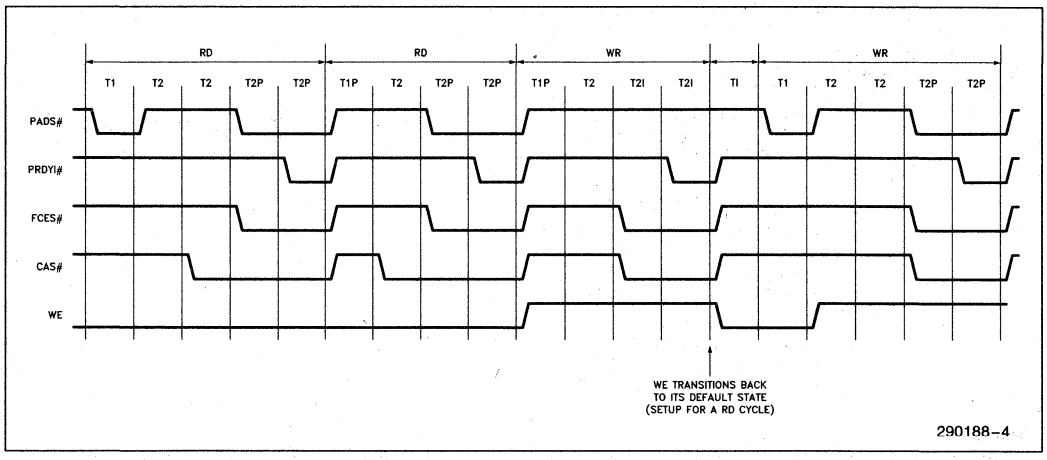


Figure 4. 2/7 Page Hit

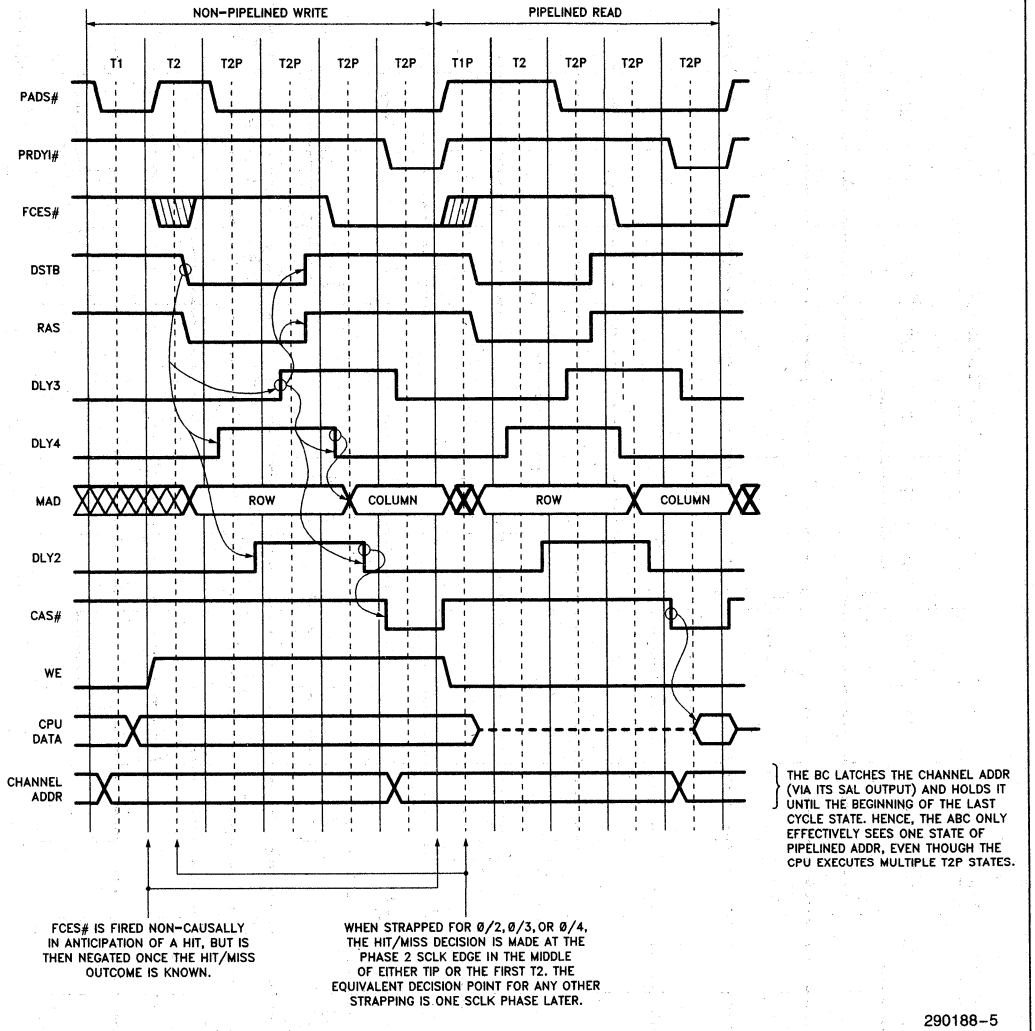


Figure 5. 0/2, 0/3, 0/4 Page Misses (Diagram Depicts 0/3 Operation. In 0/2, FCES# Fired One State Earlier. In 0/4, FCES# Fired One State Later.)

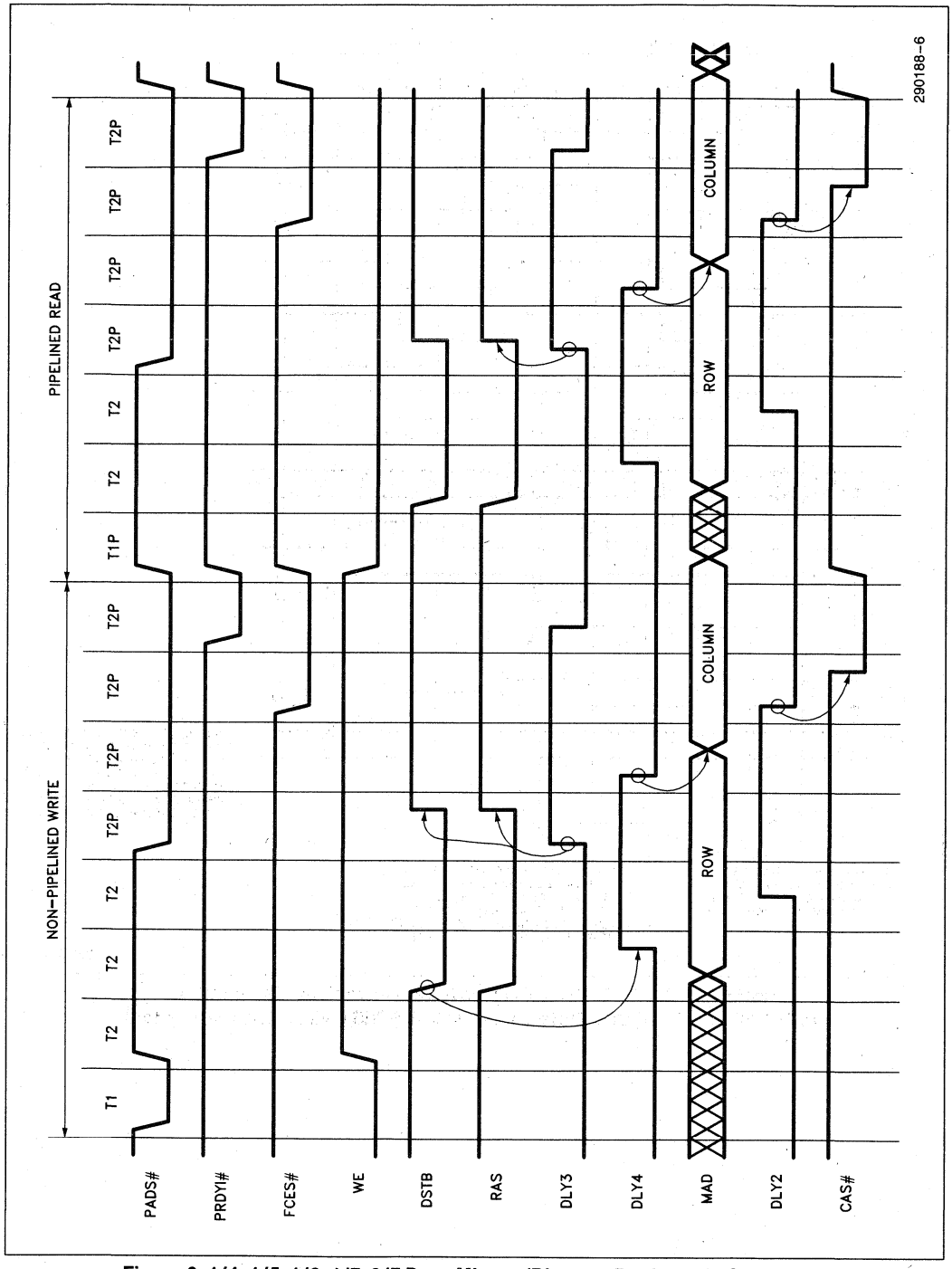
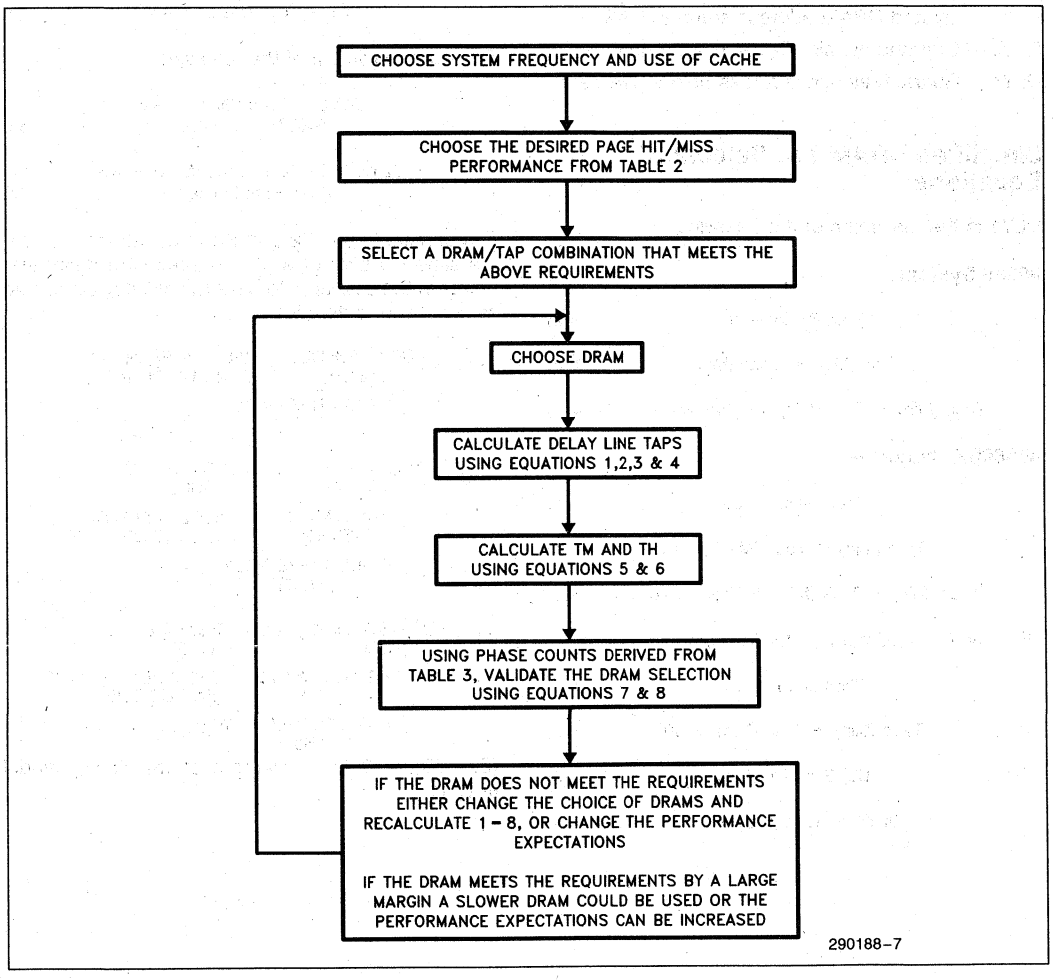


Figure 6. 1/4, 1/5, 1/6, 1/7, 2/7 Page Misses (Diagram Depicts 1/5 Operation. FCES# is Fired One State Earlier in 1/4 Operation, One State Later in 1/6 Operation, and Two States Later in Either 1/7 or 2/7 Operation.)

DRAM AND DELAY LINE TAP SELECTION

This chapter illustrates the methods that should be used to determine the delay line taps for a given DRAM, and the number of wait states a given DRAM will require.

The Flow Chart below should be used to select a DRAM/Delay line tap combination that meets the performance requirements of a system.



DELAY LINE TAP SELECTION

Function of the 4 Delay Line Taps

DLY1— Guarantees max. DRAM data from CHRDY on Micro Channel

DLY2— Guarantees the minimum RAS to CAS delay and DRAM address setup to CAS

DLY3— Guarantees min. RAS# precharge time

DLY4— Guarantees min. address hold to RAS#

Simplified DRAM Tap Selection Equations

DLY1 is the maximum of the following:

80386 System—

$$\text{Trac (Max)} - 10 \quad (1a)$$

$$\text{Trcd (Min)} + \text{Tcac (Max)} \quad (1b)$$

$$\text{Tasc (Min)} + \text{Trah (Min)} + \text{Tcac (Max)} + 20 \quad (1c)$$

80386SX System—

$$\text{Trac (Max)} - 25 \quad (1a)$$

$$\text{Trcd (Min)} + \text{Tcac (Max)} - 15 \quad (1b)$$

$$\text{Tasc (Min)} + \text{Trah (Min)} + \text{Tcac (Max)} + 5 \quad (1c)$$

DLY2 is the maximum of the following:

$$\text{Trcd (Min)} + 10 \quad (2a)$$

$$\text{Tasc (Min)} + \text{Trah (Min)} + 30 \quad (2b)$$

$$\text{DLY3} = \text{Trp (Min)} \quad (3)$$

$$\text{DLY4} = \text{Trah (Min)} + 10 \quad (4)$$

DRAM Access Time Calculations

Two access time parameters have been derived, one for hits (Th) and one for misses (Tm). These are the time from the decision to start a DRAM access to the time that data is available to the motherboard CPU.

$$\text{Th} = \text{Tcac (Max)} + \text{K1} \quad (5)$$

Tm is the maximum of the following:

$$\underset{\text{(RAS Path Limited)}}{\text{DLY3} + \text{Trac (Max)} + \text{K2}} \quad (6a)$$

$$\underset{\text{(CAS Path Limited)}}{\text{DLY3} + \text{DLY2} + \text{Tcac (Max)} + \text{K3}} \quad (6b)$$

The constants K1, K2 and K3 in equations 5 and 6 are simply a sum of all the propagation delay elements in the appropriate data access path including capacitive load derating:

$$\text{K1} = \underset{\text{(T41A)}}{\text{ABC CAS\# DLY}} + \underset{\text{(INCLUDE DERATE)}}{\text{CAS\# BUFFER DLY}} + \underset{\text{('F657)}}{\text{DATA BUFFER DLY}} = 44.5$$

$$\text{K2} = \underset{\text{(T32E)}}{\text{ABC DSTB DLY}} + \underset{\text{('AS02)}}{\text{NOR GATE DLY}} + \underset{\text{(T32G)}}{\text{ABC RAS DLY}} + \underset{\text{(INCLUDE DERATE)}}{\text{RAS BUFFER DLY}} + \underset{\text{('F657)}}{\text{DATA BUFFER DLY}} = 79$$

$$\text{K3} = \underset{\text{(T32E)}}{\text{ABC DSTB DLY}} + \underset{\text{('AS02)}}{\text{2X NOR GATE DLY}} + \underset{\text{(T34)}}{\text{ABC CAS\# DLY}} + \underset{\text{(INCLUDE DERATE)}}{\text{CAS\# BUFFER DLY}} + \underset{\text{('F657)}}{\text{DATA BUFFER DLY}} = 81.5$$

(See Figure 6 for a diagram of the timing model used.)

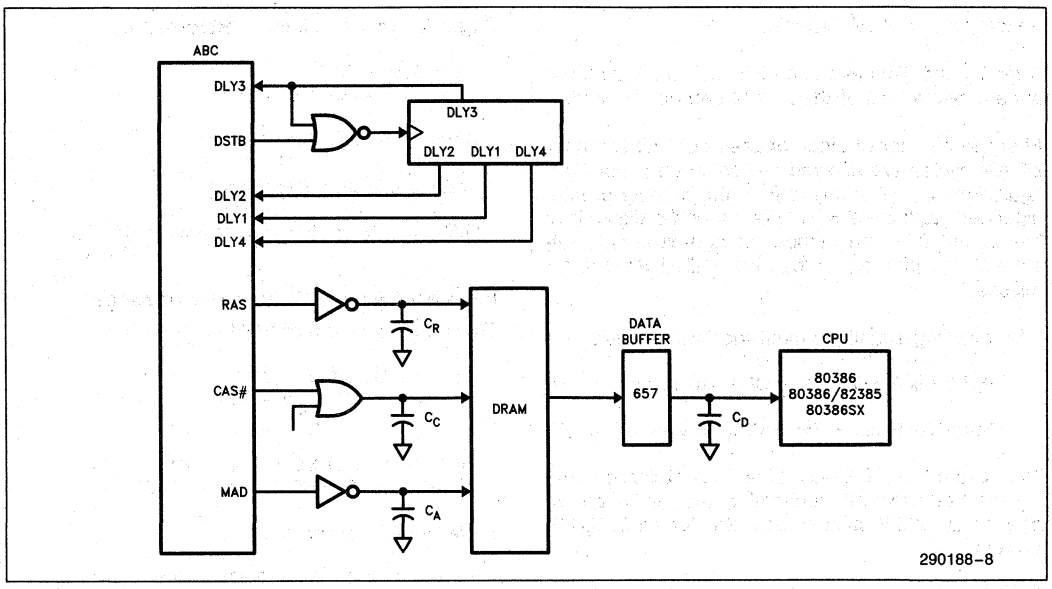


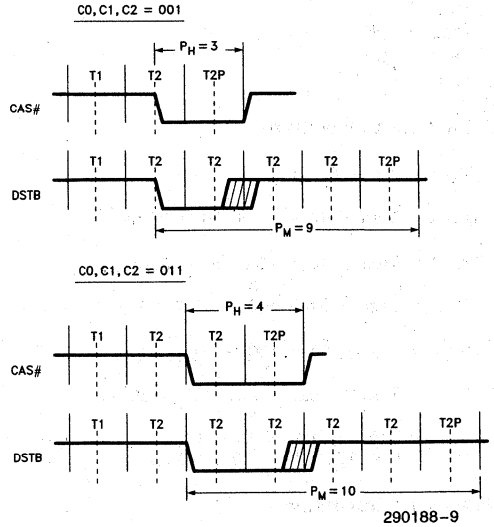
Figure 7. DRAM Timing Analysis Model

Tables 2 and 3 define the number of DRAM wait states that the motherboard CPU will see for all combinations of the configuration bits C0, C1 and C2.

Table 3. Configuration and DRAM Calculation Clock Phase Counts

Config Inputs			SCLK Phases	
C0	C1	C2	PH	PM
0	0	0	3	7
0	0	1	3	9
0	1	0	3	11
0	1	1	4	10
1	0	0	4	12
1	0	1	4	14
1	1	0	4	16
1	1	1	6	16

PH, PM Definition Examples



NOTES:
1. The phase counts are from the clock edge that either fires CAS# (Hit) or fires DSTB (Miss) to the end of cycle, as shown above.
2. Cache systems typically require additional read data setup. The BC WS# (Wait State) strap inserts an additional wait state into system board memory reads, and can be used to accommodate this increased setup if required. If WS# is tied low, then the phase counts above all increase by two.

Validation of DRAM Selection

After Th and Tm have been calculated the performance expectations of the DRAM can be checked.

First the number of clock phases both hit and miss DRAM cycles are allowed are calculated. For Configurations 0, 1 & 2 this is 2 × the number of non-pipelined waitstates + 1. For other Configurations this is just 2 × the number of non-pipelined waitstates. The phases for hits are called Ph, Pm for misses.

The following equations must then be satisfied:

$$Ph/(2*Clk\ Freq.) - Th - CPU\ Data\ setup \geq 0 \quad (7)$$

$$Pm/(2*Clk\ Freq.) - Tm - CPU\ Data\ setup \geq 0 \quad (8)$$

Two examples of Delay Line Tap Selection and DRAM Performance Verification are given below, one for an 80386 system and one for an 80386SX system.

Sample Calculation—80386 20 MHz 100 ns DRAMs 1/5 Performance

Target Dram

Key Specs (ns)

Trac	100
Trp	80
Trah	15
Trcd	25
Tasc	0
Tcac	35

Delay Line Calculations

DLY3 = Trp = 80 ns

DLY4 = Trah + 10 = 15 + 10 = 25 ns

DLY2 = Trcd + 10 = 25 + 10 = 35 ns
or
= Tasc + Trah + 30 = 0 + 15 + 30 = 45 ns

DLY1 = Trac − 10 = 100 − 10 = 90 ns
or
= Trcd + Tcac = 25 + 35 = 60 ns
or
= Tasc + Trah + Tcac + 20
= 0 + 15 + 35 + 20 = 70 ns

Delay Line Summary

DLY1	90
DLY2	45
DLY3	80
DLY4	25

Page Hit Access Time & Performance

Th = Tcac + 44.5
= 35 + 44.5 = 79.5 ns

80386 Data Setup Time = 10 ns

1 Waitstate Margin (Pipelined)

Ph/(2*CLK Freq.) − Th − 386 Data Setup = 100 − 79.5 − 10
= 10.5 ns

Page Miss Access Time & Performance

Tm is the maximum of EQN 6a and 6b.

Tm = DLY3 + Trac + 79
= 80 + 100 + 79 = 259

or

Tm = DLY3 + DLY2 + Tcac + 81.5
= 80 + 45 + 35 + 81.5 = 241.5

5 Waitstate (Pipelined) Margin

Pm/(2* CLK Freq.) − Tm − 80386 Data Setup = 300 − 259 − 10
= 31 ns

Sample Calculation—80386SX 16 MHz 100 ns DRAMs

0/3 Performance

Only DLY1 changes

DLY1 = Trac − 25 = 100 − 25 = 75 ns

Delay Line Summary

DLY1	75
DLY2	45
DLY3	80
DLY4	25

Page Hit Access Time & Performance

Th = Tcac + 44.5
= 35 + 44.5 = 79.5

80386SX Data Setup = 5 ns

0 Waitstate Margin (Pipelined)

Ph/(2* CLK Freq.) − Th − 80386SX Data Setup > = 0
93.75 − 79.5 − 5 = 9.25 ns

Page Miss Access Time & Performance

Tm = 259

(Same as for 20 MHz 386 Case)

3 Waitstate Margin (Pipelined)

Pm/(2* CLK Freq.) − Tm − 80386SX Data Setup > = 0
281.25 − 259 − 5 = 17.5 ns

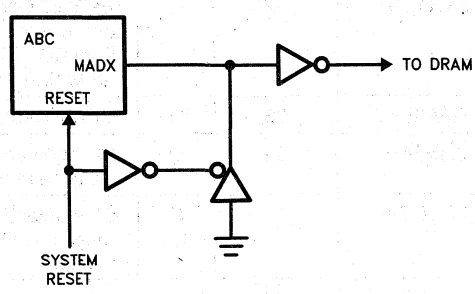
MAD BUS RESET CONFIGURATION

The ABC samples the MAD bus at the falling edge of RESET to determine system configuration as shown:

Table 4

MAD Bus Bits										Options
10	9	8	7	6	5	4	3	2	1	0	
0	0										256K DRAMs
0	1										1M DRAMs
1	1										4M DRAMs
										0	32 Bit Memory
										1	16 Bit Memory
									0		SS1 = 0
									1		SS1 = 1
							0	0			Invalid
							0	1			Single Bank
							1	0			Two Banks
							1	1			Four Banks
						0					Reserved
						1					Normal Mode
					0						C2 = 0
					1						C2 = 1
				0							C1 = 0
				1							C1 = 1
			0								C0 = 0
			1								C0 = 1
		0									SS2 = 0
		1									SS2 = 1

NOTES:
1. When either MAD09 or MAD10 is sensed as a zero, it's output driver is tri-stated, thus allowing these two pins to be tied directly to ground. For example, if 1M DRAMs are used, MAD10 should be tied to ground, since 1M DRAMs only require use of bits 0–9. MD09 should be lightly pulled up (~ 10K).
2. For MAD bits 0–8, any bit that is to be sensed as a one should be lightly pulled up. Any bit that is to be sensed as a zero must be driven low by a tri-state driver that is active while the ABC RESET input is active, and then tri-stated from the falling edge of RESET, as depicted in the figure:



290188–10

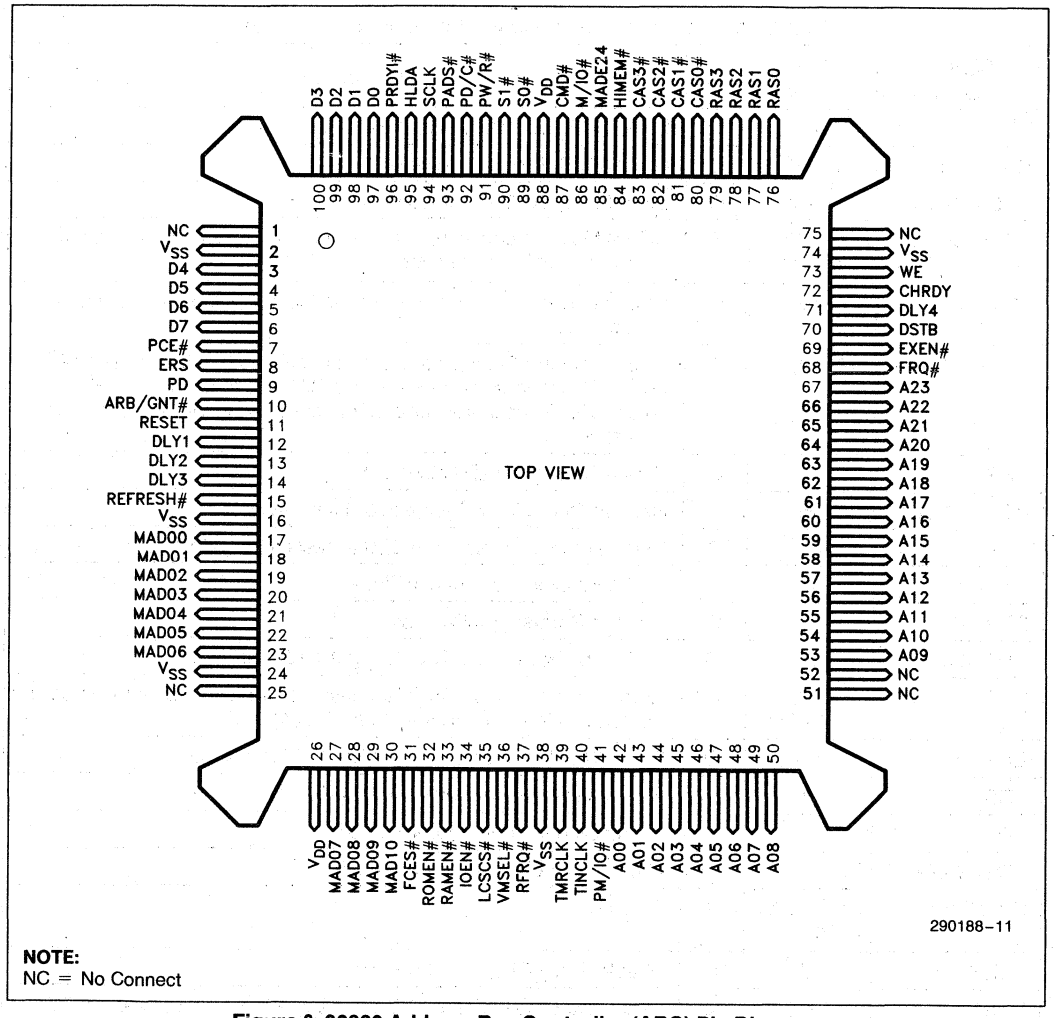
3. MAD4 sensed as a 0 is a reserved state. This bit should be lightly pulled up.
4. MAD0 is typically configured low for an 80386 system, and high for an 80386SX system.
5. MAD1 and MAD8 are respectively the system select bits SS1 and SS2. These bits determine the definition of ABC ports E0, E1 and 103, as described in the section on ABC ports and registers.
6. MAD5, MAD6 and MAD7 are respectively the DRAM performance select bits C2, C1 and C0. The effect of these bits is described in the DRAM control section.

82309 Address Bus Controller Pin Definitions

Signal Name	Pin Number	I/O	Description
A<00:23>	42–50, 53–67	I	Micro Channel Address 0 to 23
HIMEM#	84	I	Micro Channel Address 24 to 31 = FF (Active Low). Used in decoding the top-of-memory mapping of the BIOS EPROMs.
MADE24	85	I	Micro Channel Address 24 to 31 = 00 (Active High)
ROMEN#	32	O	EPROM Decode. In systems that support shadow RAM, if ROM is enabled (bit 1 in port E1), accesses to ROM space actually generate both ROMEN# and RAMEN#. In this mode, reads are from ROM, and writes are to RAM.
RAMEN#	33	O	DRAM Decode
IOEN#	34	O	Motherboard I/O devices decode (Active Low). Decode also includes memory decode of video RAM.
LCSCS#	35	O	Chip Select for the LCS (82306) Chip (Active Low). Decodes address range 0–3FFH when CPU master, or 100–3FFFH when CPU is not master.
VMSEL#	36	O	VGA Memory Space Selected (Active Low) (000A0000–000BFFFF)
S0#	89	I	Micro Channel S0# Signal
S1#	90	I	Micro Channel S1# Signal
PM/IO#	41	I	Microprocessor M/IO# Signal
PW/R#	91	I	Microprocessor W/R# Signal
PD/C#	92	I	Microprocessor D/C# Signal
PADS#	93	I	Microprocessor ADS# Signal
SCLK	94	I	Microprocessor CLK2
HLDA	95	I	HLDA Signal from the Processor
PRDYI#	96	I	READY# Signal from the Processor
M/IO#	86	I	Micro Channel M/IO# Signal
CMD#	87	I	Micro Channel CMD Signal

82309 Address Bus Controller Pin Definitions (Continued)

Signal Name	Pin Number	I/O	Description
WE	73	O	DRAM Write Enable Signal (Active High)
RAS<0:3>	76–79	O	DRAM RAS Strobes (Active High)
CAS# <0:3>	80–83	O	DRAM CAS Strobe Enables (Active Low)
MAD<00:10>	17–23, 27–30	B	DRAM Muxed Address bus. These signals are sampled at reset to determine ABC configuration.
DSTB	70	O	Output to Delay Line. A pulse put into the delay line controls page miss timing.
DLY<1:4>	12–14, 71	I	Inputs from the Delay Line. DLY1 controls CHRDY timing in non-CPU cycles. DLY2 controls RAS active to CAS active timing. DLY3 controls RAS precharge, and DLY4 controls row-to-column address multiplex.
CHRDY	72	O	DRAM Ready Signal (Active High)
REFRESH#	15	I	Refresh Operation in Progress (Active Low)
FCES#	31	O	Request to BC to terminate CPU accesses to system board memory.
TINCLK	40	I	14.3 MHz Clock for Refresh Timer
RFRQ#	37	O	Refresh Request (Active Low)
TMRCLK	39	O	14.3 MHz Clock divided by 12 to get 1.19 MHz.
FRQ#	68	O	Asynchronous Cache Flush Request. Activated in I/O writes to ports E0, E1, or 100–107 (POS Address Space).
EXEN#	69	O	Read/Write Strobe for Ports 00E0–00E7 (Active Low)
ERS	8	I	Sampling Strobe for Ports 00E2–00E7
PD	9	I	Select Signal for POS Register 10X
ARB/GNT#	10	I	Micro Channel ARB/GNT# Signal
D<0:7>	97–100, 3–6	I/O	Data Bus
PCE#	7	O	Enable Parity Checking (MER<0>)
RESET	11	I	Synchronous reset input. RESET falling edge used to synchronize ABC internal clock to CPU phase.
NC	1, 25, 51, 52, 75		No Connect
V_{DD}	26, 88		Power
V_{SS}	2, 16, 24, 38, 74		Ground



NOTE:
NC = No Connect

Figure 8. 82309 Address Bus Controller (ABC) Pin Diagram

82309 PARAMETRICS

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

ABSOLUTE MAXIMUM RATINGS*

Case Temperature under Bias .... −40°C to +85°C
Storage Temperature ..... −65°C to +150°C
Voltage to Any Pin with Respect to Ground ..... −0.3V to (V_{CC}+0.3)V
DC Supply Voltage (V_{CC}) ..... −0.3V to +7.0V
DC Input Current ..... ±10 mA

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

1

D.C. CHARACTERISTICS

T_C = 0°C to +70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Conditions
V_{IL}	Input Low Voltage		0.8	V	
V_{IH}	Input High Voltage	2.0		V	
V_{IL}	Input Low Voltage		0.8	V	SCLK
V_{IH}	Input High Voltage	V_{CC} − 0.8		V	SCLK
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 4 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = 4 mA
I_{CC}	Power Supply Current		180	mA	No DC Loads
I_{LI}	Input Leakage Current		±10	μA	$V_{SS} < V_{IN} < V_{CC}$
I_{OZ}	Tri-State Output Leakage Current		±10	μA	$V_{SS} < V_{OUT} < V_{CC}$

82309 ADDRESS BUS CONTROLLER A.C. SPECS

T_C = 0°C to +70°C, V_{CC} = 5V ± 10%

Symbol	Parameter	Kit 16 MHz		Kit 20 MHz		Kit 25 MHz		CL(PF)	Notes
		Min	Max	Min	Max	Min	Max		
T1	SCLK PERIOD	31.25		25		20			
T2A	SCLK HIGH/LOW TIME (50%)	12		10		8			
T2B	SCLK HIGH/LOW TIME (90%)	8		6.5		6			
T3	RESET SETUP	10		10		10			
T4	RESET HOLD	4		4		4			
T5A	STATUS SETUP TO SCLK	11		11		8			1
T5B	CMD# SETUP TO SCLK	11		11		8			1
T6	PADS#,PW/R#,PD/C#,PM/IO# SETUP	25		22		13			
T7	PADS#,PW/R#,PD/C#,PM/IO# HOLD	4		4		4			
T8	ADDRESS,M/IO#,MADE24,REFRESH# SETUP	10		10		10			
T9	ADDRESS,M/IO#,MADE24,REFRESH# HOLD	12		12		12			
T10	ADDRESS, M/IO# SETUP	40		50		36			3
T11	ADDRESS, M/10# HOLD	8		8		8			3
T12	MADE24 SETUP	32		40		28			3
T13	MADE24 HOLD	8		8		8			3
T14	M/IO#,ARB/GNT# SETUP TO ERS	20		20		20			
T15	M/IO#,ARB/GNT# HOLD FROM ERS	10		10		10			
T16	PRDYI# SETUP	18		18		15			
T17	PRDYI# HOLD	3		3		3			
T18A	ROMEN#,RAMEN#,IOEN#,VMSEL# DLY FRM MADE24 HIMEM#	2	30	2	30	2	30	75	
T18B	ROMEN#,RAMEN#,IOEN#,VMSEL# DLY FRM ADDR	2	38	2	38	2	38	75	14
T18C	ROMEN#,RAMEN#,IOEN#,VMSEL# DLY FRM A20	2	35	2	35	2	35	75	14
T19	LCSCS# DELAY	2	45	2	45	2	45	75	
T20A	FCES# DELAY FROM SCLK	3	45	3	35	3	26	25	5
T20B	FCES# DLY FRM ADDR, M/IO#, MADE24		50					25	2, 5
T21	PD SETUP TO CMD# ↑	100		100		100			
T22	WRITE DATA SETUP	30		30		30			
T23	WRITE DATA HOLD	5		5		5			
T24	READ DATA VALID DELAY		200		200		200	75	
T24A	CMD# ↓ TO READ DATA LOW-Z	25		25		25		75	
T25	READ DATA FLOAT DELAY	2	35	2	35	2	35	75	
T26	EXEN# DELAY (INACTIVE)	2	50	2	50	2	50	50	
T26A	EXEN# DELAY (ACTIVE)	25	150	25	150	25	150	50	
T27A	CHRDY DELAY (FROM ADDR)	2	50	2	50	2	50	50	4, 5, 7
T27B	CHRDY DLY FROM STATUS OR CMD#	0	33	0	33	0	33	50	4, 5, 7
T29	TMRCLK HIGH/LOW TIME	300		300		300		50	
T30	RFRQ# PULSE WIDTH	300		300		300		50	
T31	TINCLK HIGH/LOW TIME	21		21		21			
T32B	CMD# ↑ -RAS ↓ (REFRESH CYCLE ONLY)	0	55	0	55	0	55	75	10
T32C	CMD# ↑ -CAS# ↑	0	38	0	38	0	38	75	
T32E	SCLK-DSTB ↓	4	27	4	27	4	27	50	8
T32F	DLY3 ↑ -DSTB ↑	0	50	0	50	0	50	50	
T32G	DLY3 ↑ -RAS ↑	4	26	4	26	4	26	75	
T32I	SCLK-RAS ↓	8	40	8	40	8	40	75	8
T33	DLY1 ↓ TO CHRDY ↑	5	30	5	30	5	30	50	
T34	DLY2 ↓ TO CAS# ↓	3	27	3	27	3	27	75	

82309 ADDRESS BUS CONTROLLER A.C. SPECS (Continued)

Symbol	Parameter	Kit 16 MHz		Kit 20 MHz		Kit 25 MHz		CL(PF)	Notes
		Min	Max	Min	Max	Min	Max		
T35	CMD# ↓ TO CAS# ↓ (WRITE CYCLES ONLY)	25	115	25	115	25	115	75	6
T39A	WE DLY FROM STATUS	2	30	2	30	2	30	75	13
T39B	WE DLY FROM CMD# ↑	2	30	2	30	2	30	75	13
T41A	CAS# ↓ DELAY FROM SCLK (READS)	2	30	2	30	2	30	75	
T41B	CAS# ↑ DELAY FROM SCLK	5	34	5	34	5	34	75	9
T41C	CAS# ↓ DELAY FROM SCLK (WRITES)	2	38	2	38	2	38	75	
T43A	ADDR TO MAD DELAY (COLUMN ADDR)		45		45		40	75	12
T43B	SCLK TO MAD DELAY (COLUMN ADDR)		36		36		31	75	12
T43C	CMD# TO MAD DELAY (COLUMN ADDR)		38		38		38	75	12
T43D	SCLK TO MAD DELAY (ROW ADDR)		50		50		50	75	12
T44	WE DELAY FROM SCLK	2	42	2	42	2	42	75	
T45	DLY4 ↓ TO MAD	6	32	6	32	6	32	75	
T46	MAX PAGE MODE RAS ACTIVE	15 μs		15 μs		15 μs			10, 11

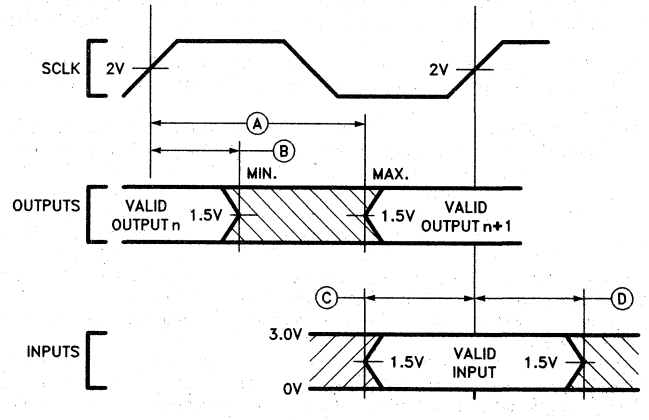
NOTES:
1. Status and CMD# are asychronous inputs. T5 simply guarantees that they are recognized at a particular clock edge.
2. FCES# is speced from address only in 0WS pipelined/1WS non-pipelined memory cycles, which are only supported at 16 MHz.
3. Address, M/IO# and MADE24 setup times are speced relative to the Phase 2 SCLK edge only in 0WS pipelined/1WS non-pipelined memory cycles, which are only supported at 16 MHz. (This Phase 2 edge is in the middle of the first T2 state, or the middle of the T1P state.)
4. The 82309 de-activates CHRDY for motherboard I/O and VGA Memory cycles (as decoded by IOEN#), and then re-activates it when CMD# is activated. The 82309 also de-activates CHRDY for non-CPU (DMA or channel master) accesses to motherboard DRAM that are decoded as page misses. CHRDY is then re-activated according to the appropriate external DRAM control delay line tap (DLY1), or else when CMD# is activated, whichever comes later.
5. FCES# is used to terminate CPU accesses to motherboard DRAM, as these cycles are not broadcast on the Micro Channel. CHRDY is used to terminate DMA and channel master accesses to motherboard DRAM, which are broadcast.
6. The large value for T35 (Min) guarantees that data being written into motherboard DRAM by a channel master or DMA controller has adequate time to propogate through the data buffers between the channel or DMA and memory. T35 (Min) is guaranteed on any non-CPU write, both page hit and page miss. (The Micro Channel specs 0NS of data setup to CMD# active.) T35 (Max) applies only when CMD# ↓-to-CAS# ↓ is indeed the limiting spec; specifically, when neither T34 (Max) nor T41A (Max) limits CAS# activation.
7. The 82309 guarantees that any time it de-asserts CHRDY, it will not re-assert it until after CMD# is activated.
8. These specs are referenced with respect to the causal SCLK edge, which differs in different frequency systems. At 16 MHz, the appropriate edge is one clock phase after the edge that recognizes status in non-CPU cycles, or one phase after the edge that samples PADS# active in CPU cycles. At 20 MHz, the appropriate edge is two clock phases after these events. The 82309 distinguishes 16 MHz from 20 MHz via the memory performance configuration inputs C0, C1 and C2. 16 MHz is assumed anytime these inputs indicate a zero wait state pipelined read page hit. (C0, C1, C2 = 000,001,010).
9. This spec insures a minimum CAS# high time of 25 ns at 16 MHz. (16 MHz is the worst case since the CAS# inactive and CAS# active SCLK edges are only one phase apart. At 20 MHz, these edges are always at least two phases apart.)
10. Refresh cycles are RAS only, and are forced to be page misses. Thus, the refresh interval (typically 15 μs) defines the required page mode RAS active time. RAS is de-activated at the end of a refresh cycle since typical page mode DRAMs spec a maximum RAS active time less than 15 μs for refresh cycles. (Note that the first access to any bank following a refresh cycle is also a forced page miss.)
11. Functional spec only . . . Not tested. Max page mode RAS active is governed by refresh interval.
12. T43A, T43B, and T43C all refer to column address, as the 82309 assumes a page hit as the default case until proven otherwise. In case of a page miss, the row address is muxed onto the MAD lines from the same clock edge that de-activates RAS and fires a pulse (DSTB) into the delay line. The column address is then muxed onto the MAD lines by delay tap DLY4.
13. In Non-CPU cycles, WE (active high from the 82309) is simply an inverted version of channel S0#, which indicates a write cycle when low. This signal is internally latched (transparent latch) by the leading edge of CMD#, and then released by the trailing edge of CMD#, hence the need for T39B.
14. A20 typically has more logic in its path than the other address bits, hence the tighter spec. T18B applies to all bits except A20.

82309SX Micro Channel Address Bus Controller

A.C. SPECIFICATIONS

The following specifications are the only exceptions to the A.C. specifications listed on the previous pages.

Symbol	Parameter	16 MHz and 20 MHz
T10	ADDRESS, M/IO# SETUP	30 ns
T12	MADE24 SETUP	NA
T20B	FCES# DLY FROM ADDR, M/IO#, MADE24	39 ns
T32E	SCLK TO DSTB FALLING EDGE	24 ns
T32G	DLY3 RISING TO RAS# RISING	23 ns
T34	DLY2 FALLING TO CAS# FALLING	25 ns
T41a	CAS# DELAY FROM SCLK (READS)	24 ns
T43A	ADDR TO MAD DELAY (COLUMN ADDR)	38 ns
T43B	SCLK TO MAD DELAY (COLUMN ADDR)	33 ns
T45	DELAY4 FALLING EDGE TO MAD	30 ns



LEGEND:
A. Maximum Output Delay Specification.
B. Minimum Output Delay Specification.
C. Minimum Input Setup Specification.
D. Minimum Input Hold Specification.

Input waveforms have tr ≤ 2.0 ns from 0.8V to 2.0V.

Figure 9. Drive Levels and Measurement Points for A.C. Specifications

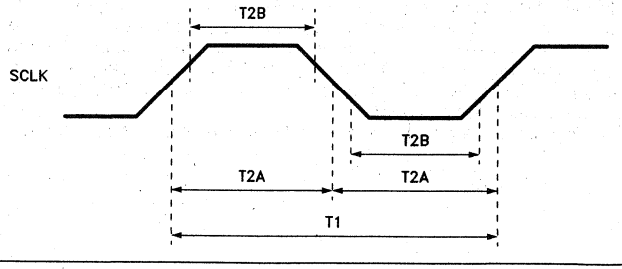


Figure 10. SCLK Waveform

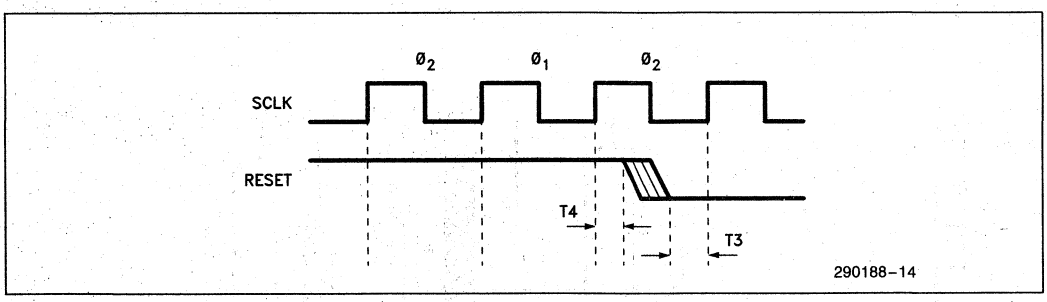


Figure 11. RESET Setup and Hold Diagram

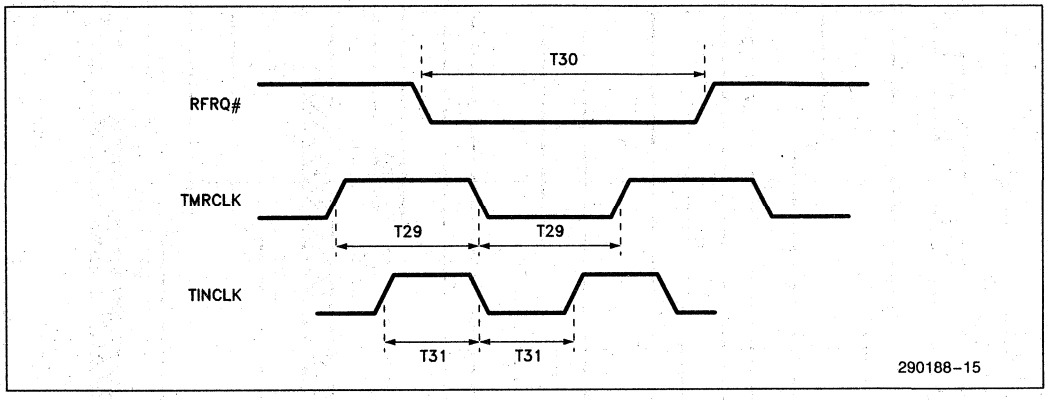


Figure 12. RFRQ, TMRCLK, TINCLK A.C. Timing Diagrams

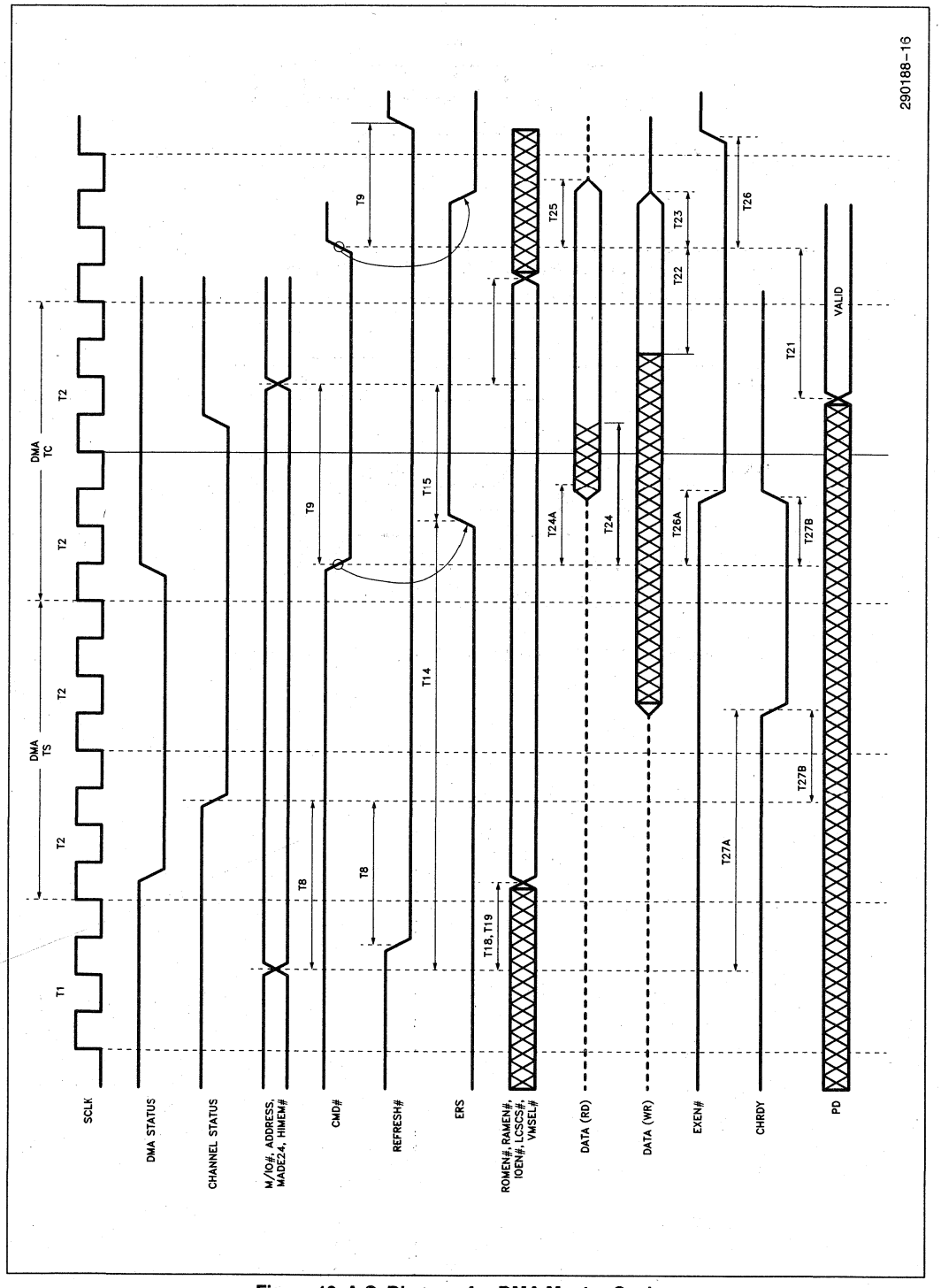
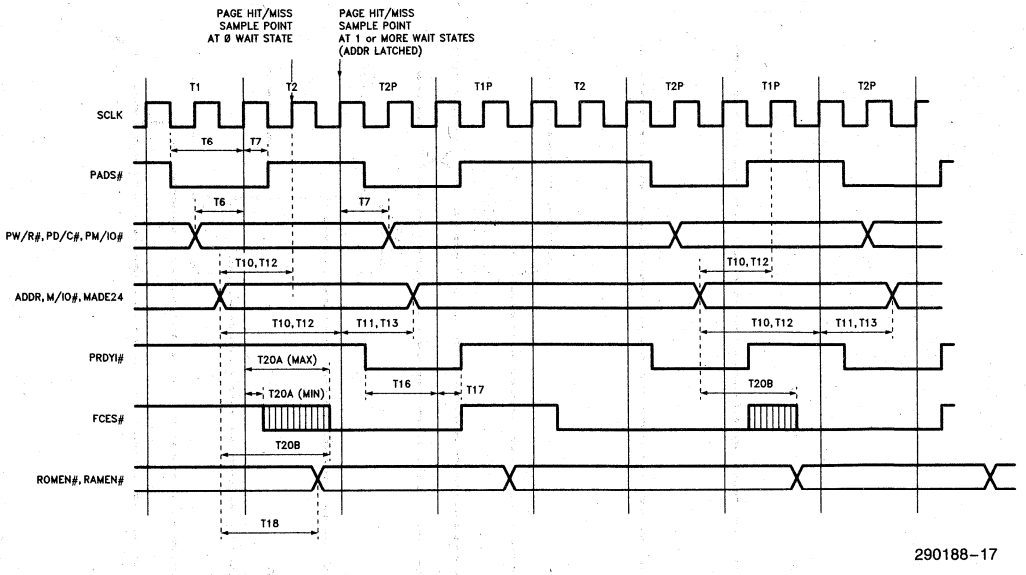


Figure 13. A.C. Diagram for DMA Master Cycles



NOTE:
Once the ABC fires FCES#, it remains active until the ABC detects and end of cycle via its PRDYI# input. This allows the BC or other external devices to extend DRAM cycles if desired.

Figure 14. 82309/80386 Interface Timings

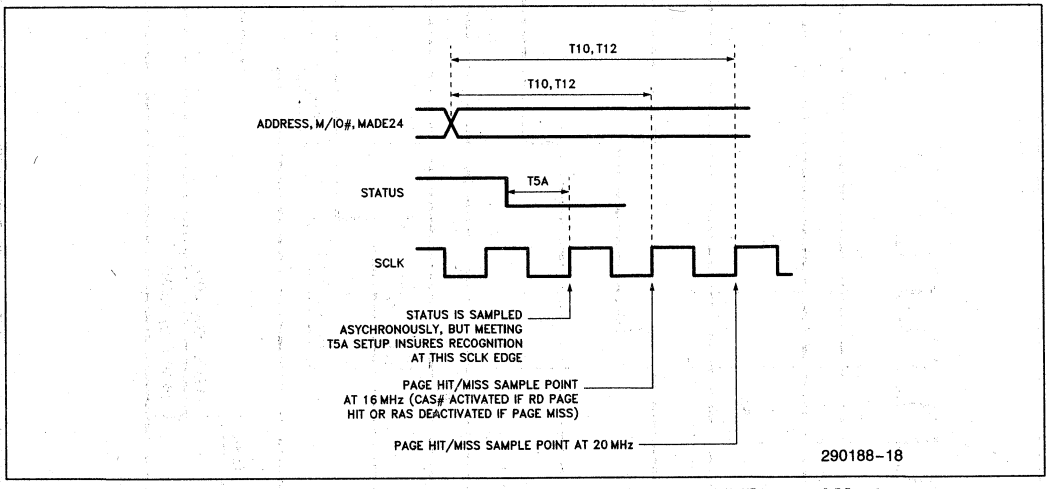


Figure 15. Address, M/IO#, MADE 24 Setup for DMA/Micro Channel Master

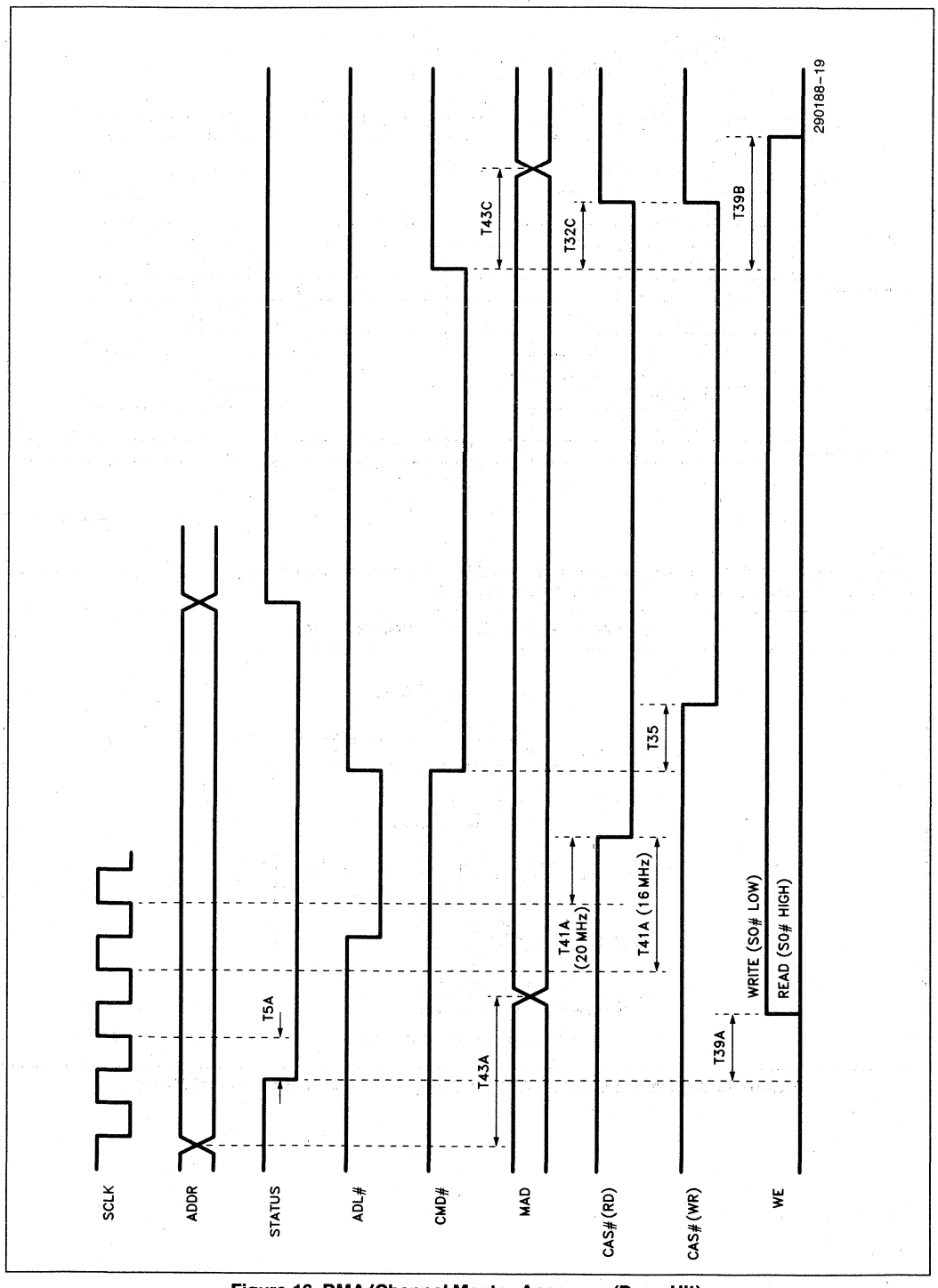


Figure 16. DMA/Channel Master Accesses (Page Hit)

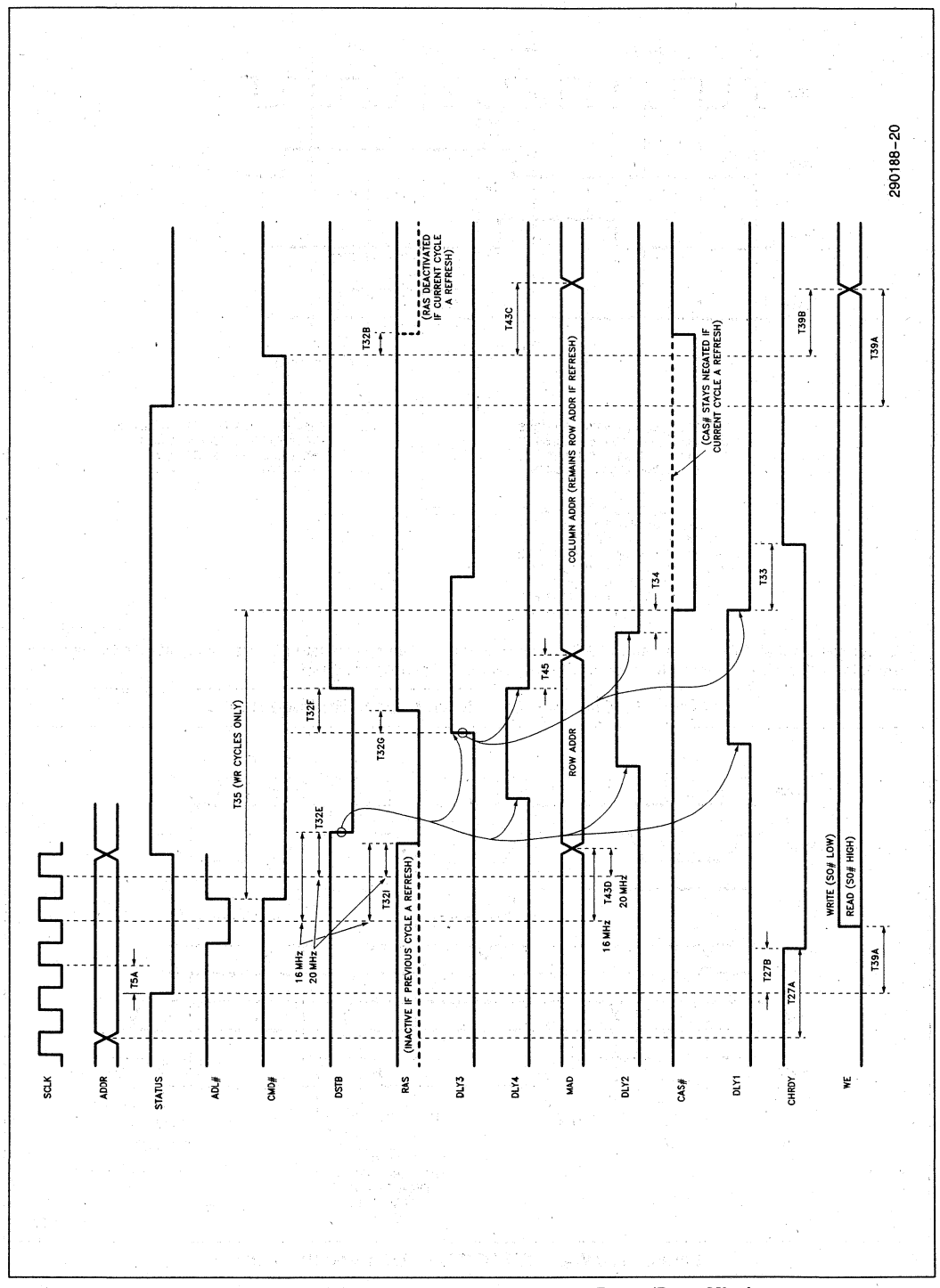
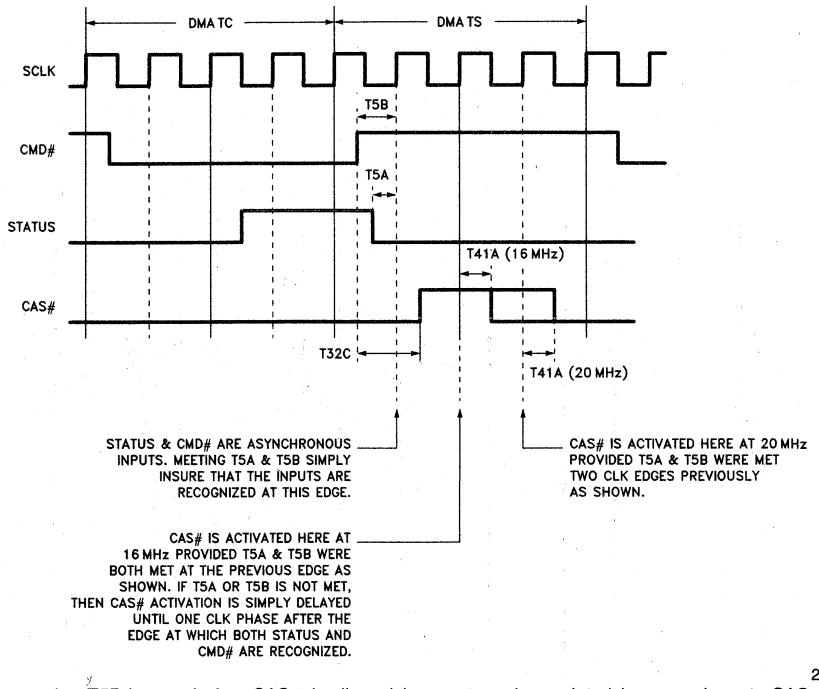


Figure 17. DMA/Channel Master Accesses to Dram (Page Miss)



The requirement that T5B be met before CAS# is allowed (one or two phases later) insures adequate CAS# high time in back-to-back cycles. (One SCLK phase at 16 MHz; Two SCLK phases at 20 MHz.)

Figure 18. DMA Master . . . Back-to-back RD Page Hits

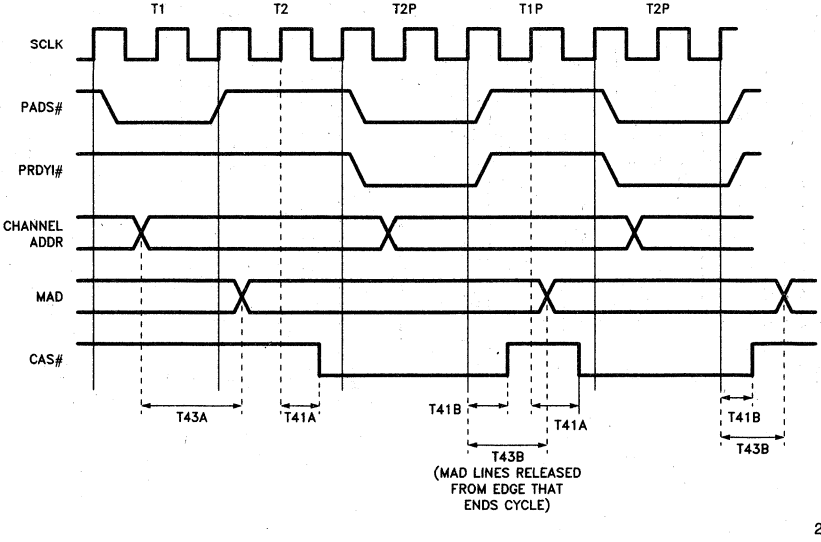


Figure 19. 0 WAIT STATE (Pipelined) Read Page Hits

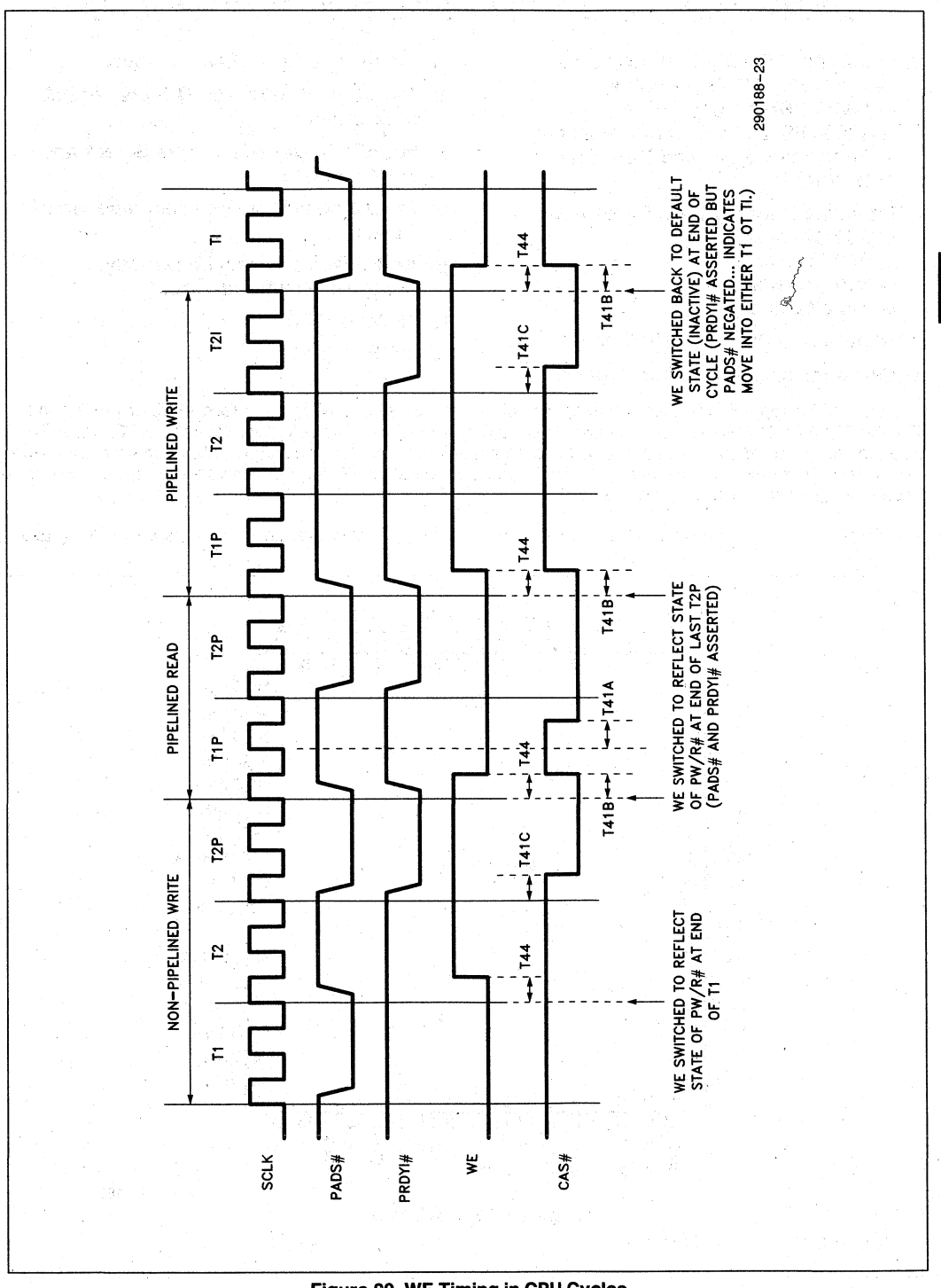


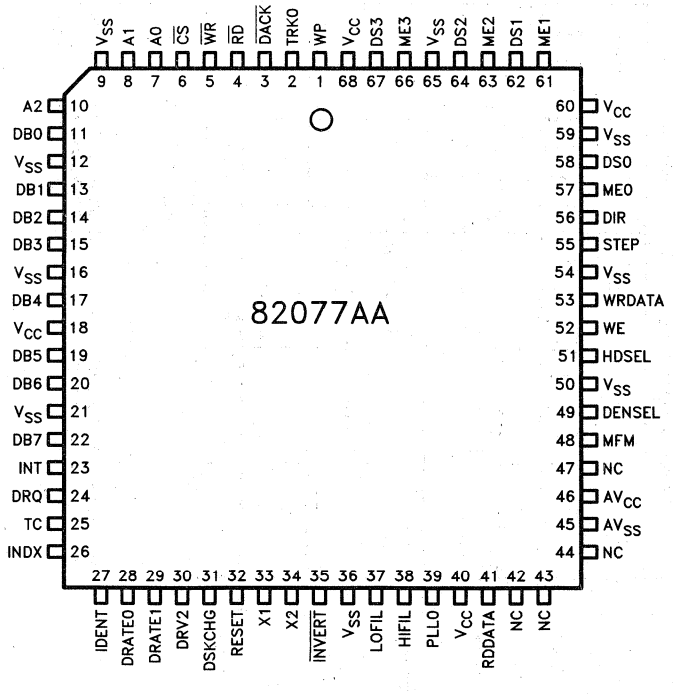
Figure 20. WE Timing in CPU Cycles

82077AA
CHMOS SINGLE-CHIP FLOPPY DISK CONTROLLER

- **Single-Chip Floppy Disk Solution**
  - **— 100% PC AT* Compatible**
  - **— 100% PS/2* Compatible**
  - **— 100% PS/2 Model 30 Compatible**
  - **— Integrated Drive and Data Bus Buffers**
- **Integrated Analog Data Separator**
  - **— 250 Kbits/sec**
  - **— 300 Kbits/sec**
  - **— 500 Kbits/sec**
  - **— 1 Mbits/sec**
- **High Speed Processor Interface**
- **Perpendicular Recording Support**
- **Integrated Tape Drive Support**
- **12 mA Host Interface Drivers, 40 mA Disk Drivers**
- **Four Fully Decoded Drive Select and Motor Signals**
- **Programmable Write Precompensation Delays**
- **Addresses 256 Tracks Directly, Supports Unlimited Tracks**
- **16 Byte FIFO**
- **68-Pin PLCC**

The 82077AA floppy disk controller has completely integrated all of the logic required for floppy disk control. The 82077AA, a 24 MHz crystal, a resistor package and a device chip select implements a PC AT or PS/2 solution. All programmable options default to compatible values. The dual PLL data separator has better performance than most board level/discrete PLL implementations. The FIFO allows better system performance in multi-master systems (e.g. PS/2, EISA).

The 82077AA is fabricated with Intel's CHMOS III technology and is available in a 68-lead PLCC (plastic) package.



290166-1

Figure 1. 82077AA Pinout

*PS/2 and PC AT are trademarks of IBM.

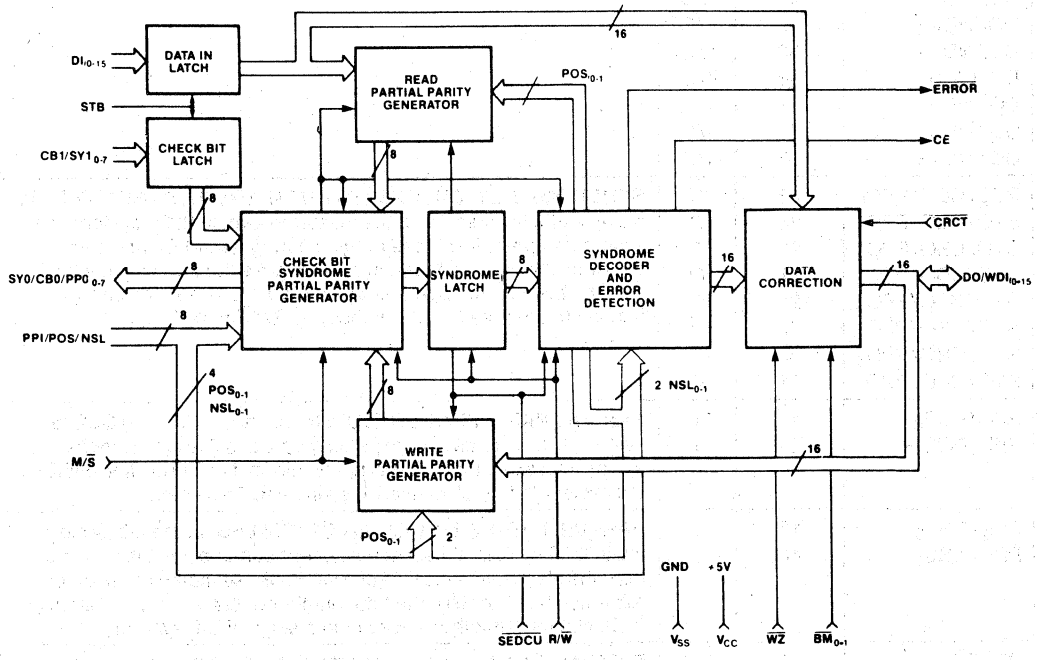
Memory Controllers 2

8206
ERROR DETECTION AND CORRECTION UNIT

- **Detects All Single Bit, and Double Bit and Most Multiple Bit Errors**
- **Corrects All Single Bit Errors**
3 Selections	8206-1	8206
Detection	35 ns	42 ns
Correction	55 ns	67 ns
- **Syndrome Outputs for Error Logging**
- **Automatic Error Scrubbing with 8207**
- **Expandable to Handle 80 Bit Memories**
- **Separate Input and Output Busses—No Timing Strobes Required**
- **Supports Read With and Without Correction, Writes, Partial (Byte) Writes, and Read-Modify-Writes**
- **HMOS III Technology for Low Power**
- **68 Pin Leadless JEDEC Package**
- **68 Pin Grid Array Package**

2

The HMOS 8206 Error Detection and Correction Unit is a high-speed device that provides error detection and correction for memory systems (static and dynamic) requiring high reliability and performance. Each 8206 handles 8 or 16 data bits and up to 8 check bits. 8206's can be cascaded to provide correction and detection for up to 80 bits of data. Other 8206 features include the ability to handle byte writes, memory initialization, and error logging.



205220-1

Figure 1. 8206 Block Diagram

September 1987
Order Number: 205220-008

Table 1. 8206 Pin Description

Symbol	Pin No.	Type	Name and Function
DI_{0-15}	1, 68-61, 59-53	I	**DATA IN:** These inputs accept a 16 bit data word from RAM for error detection and/or correction.
CBI/SYI_0 CBI/SYI_1 CBI/SYI_2 CBI/SYI_3 CBI/SYI_4 CBI/SYI_5 CBI/SYI_6 CBI/SYI_7	5 6 7 8 9 10 11 12	I I I I I I I I	**CHECK BITS IN/SYNDROME IN:** In a single 8206 system,or in the master in a multi-8206 system, these inputs accept the check bits (5 to 8) from the RAM. In a single 8206 16 bit system, CBI_{0-5} are used. In slave 8206's these inputs accept the syndrome from the master.
DO/WDI_0 DO/WDI_1 DO/WDI_2 DO/WDI_3 DO/WDI_4 DO/WDI_5 DO/WDI_6 DO/WDI_7 DO/WDI_8 DO/WDI_9 DO/WDI_{10} DO/WDI_{11} DO/WDI_{12} DO/WDI_{13} DO/WDI_{14} DO/WDI_{15}	51 50 49 48 47 46 45 44 42 41 40 39 38 37 36 35	I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O	**DATA OUT/WRITE DATA IN:** In a read cycle, data accepted by DI_{0-15} appears at these outputs corrected if $\overline{CRCT}$ is low, or uncorrected if $\overline{CRCT}$ is high. The $\overline{BM}$ inputs must be high to enable the output buffers during the read cycle. In a write cycle, data to be written into the RAM is accepted by these inputs for computing the write check bits. In a partial-write cycle, the byte not to be modified appears at either DO_{0-7} if $\overline{BM}_0$ is high, or DO_{8-15} if $\overline{BM}_1$ is high, for writing to the RAM. When $\overline{WZ}$ is active, it causes the 8206 to output all zeros at DO_{0-15}, with the proper write check bits on CBO.
$SYO/CBO/PPO_0$ $SYO/CBO/PPO_1$ $SYO/CBO/PPO_2$ $SYO/CBO/PPO_3$ $SYO/CBO/PPO_4$ $SYO/CBO/PPO_5$ $SYO/CBO/PPO_6$ $SYO/CBO/PPO_7$	23 24 25 27 28 29 30 31	O O O O O O O O	**SYNDROME OUT/CHECK BITS OUT/PARTIAL PARITY OUT:** In a single 8206 system, or in the master in a multi-8206 system, the syndrome appears at these outputs during a read. During a write, the write check bits appear. In slave 8206's the partial parity bits used by the master appear at these outputs. The syndrome is latched (during read-modify-writes) by $R/\overline{W}$ going low.
PPI_0/POS_0 PPI_1/POS_1	13 14	I I	**PARTIAL PARITY IN/POSITION:** In the master in a multi-8206 system, these inputs accept partial parity bits 0 and 1 from the slaves. In a slave 8206 these inputs inform it of its position within the system (1 to 4). Not used in a single 8206 system.
PPI_2/NSL_0 PPI_3/NSL_1	15 16	I I	**PARTIAL PARITY IN/NUMBER OF SLAVES:** In the master in a multi-8206 system, these inputs accept partial parity bits 2 and 3 from the slaves. In a multi-8206 system these inputs are used in slave number 1 to tell it the total number of slaves in the system (1 to 4). Not used in other slaves or in a single 8206 system.
PPI_4CE	17	I/O	**PARTIAL PARITY IN/CORRECTABLE ERROR:** In the master in a multi-8206 system this pin accepts partial parity bit 4. In slave number 1 only, or in a single 8206 system, this pin outputs the correctable error flag. CE is latched by $R/\overline{W}$ going low. Not used in other slaves.

Table 1. 8206 Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
PPI_5 PPI_6 PPI_7	18 19 20	I I I	**PARTIAL PARITY IN:** In the master in a multi-8206 system these pins accept partial parity bits 5 to 7. The number of partial parity bits equals the number of check bits. Not used in single 8206 systems or in slaves.
$\overline{ERROR}$	22	O	**ERROR:** This pin outputs the error flag in a single 8206 system or in the master of a multi-8206 system. It is latched by R/$\overline{W}$ going low. Not used in slaves.
$\overline{CRCT}$	52	I	**CORRECT:** When low this pin causes data correction during a read or read-modify-write cycle. When high, it causes error correction to be disabled, although error checking is still enabled.
$\overline{STB}$	2	I	**STROBE:** STB is an input control used to strobe data at the DI inputs and check-bits at the CBI/SYI inputs. The signal is active high to admit the inputs. The signals are latched by the high-to-low transition of STB.
$\overline{BM}_0$ $\overline{BM}_1$	33 32	I I	**BYTE MARKS:** When high, the Data Out pins are enabled for a read cycle. When low, the Data Out buffers are tristated for a write cycle. $\overline{BM}_0$ controls DO_{0-7}, while $\overline{BM}_1$ controls DO_{8-15}. In partial (byte) writes, the byte mark input is low for the new byte to be written.
R/$\overline{W}$	21	I	**READ/WRITE:** When high this pin causes the 8206 to perform detection and correction (if $\overline{CRCT}$ is low). When low, it causes the 8206 to generate check-bits. On the high-to-low transition the syndrome is latched internally for read-modify-write cycles.
$\overline{WZ}$	34	I	**WRITE ZERO:** When low this input overrides the $\overline{BM}_{0-1}$ and R/$\overline{W}$ inputs to cause the 8206 to output all zeros at DO_{0-15} with the corresponding check-bits at CBO_{0-7}. Used for memory initialization.
M/$\overline{S}$	4	I	**MASTER/SLAVE:** Input tells the 8206 whether it is a master (high) or a slave (low).
$\overline{SEDCU}$	3	I	**SINGLE EDC UNIT:** Input tells the master whether it is operating as a single 8206 (low) or as the master in a multi-8206 system (high). Not used in slaves.
V_{CC}	60	I	**POWER SUPPLY:** +5V
V_{SS}	26	I	**LOGIC GROUND**
V_{SS}	43	I	**OUTPUT DRIVER GROUND**

FUNCTIONAL DESCRIPTION

The 8206 Error Detection and Correction Unit provides greater memory system reliability through its ability to detect and correct memory errors. It is a single chip device that can detect and correct all single bit errors and detect all double bit and some higher multiple bit errors. Some other odd multiple bit errors (e.g., 5 bits in error) are interpreted as single bit errors, and the CE flag is raised. While some even multiple bit errors (e.g., 4 bits in error) are interpreted as no error, most are detected as double bit errors. This error handling is a function of the number of check bits used by the 8206 (see Figure 2) and the specific Hamming code used. Errors in check bits are not distinguished from errors in a word.

For more information on error correction codes, see Intel Application Notes AP-46 and AP-73.

A single 8206 handles 8 or 16 bits of data, and up to 5 8206's can be cascaded in order to handle data paths of 80 bits. For a single 8206 8 bit system, the DI_{8-15}, DO/WDI_{8-15} and $\overline{BM}_1$ inputs are grounded. See the Multi-Chip systems section for information on 24-80 bit systems.

The 8206 has a "flow through" architecture. It supports two kinds of error correction architecture: 1) Flow-through, or correct-always; and 2) Parallel, or check-only. These are two separate 16-pin busses,

Data Word Bits	Check Bits
8	5
16	6
24	6
32	7
40	7
48	8
56	8
64	8
72	8
80	8

Figure 3. Number of Check Bits Used by 8206

one to accept data from the RAM (DI) and the other to deliver corrected data to the system bus (DO/WDI). The logic is entirely combinatorial during a read cycle. This is in contrast to an architecture with only one bus, with bidirectional bus drivers that must first read the data and then be turned around to output the corrected data. The latter architecture typically requires additional hardware (latches and/or transceivers) and may be slower in a system due to timing skews of control signals.

READ CYCLE

With the R/$\overline{W}$ pin high, data is received from the RAM outputs into the DI pins where it is optionally latched by the STB signal. Check bits are generated from the data bits and compared to the check bits read from the RAM into the CBI pins. If an error is detected the $\overline{ERROR}$ flag is activated and the correctable error flag (CE) is used to inform the system whether the error was correctable or not. With the $\overline{BM}$ inputs high, the word appears corrected at the DO pins if the error was correctable, or unmodified if the error was uncorrectable.

If more than one 8206 is being used, then the check bits are read by the master. The slaves generate a partial parity output (PPO) and pass it to the master. The master 8206 then generates and returns the syndrome to the slaves (SYO) for correction of the data.

The 8206 may alternatively be used in a "check-only" mode with the $\overline{CRCT}$ pin left high. With the correction facility turned off, the propagation delay from memory outputs to 8206 outputs is significantly shortened. In this mode the 8206 issues an $\overline{ERROR}$ flag to the CPU, which can then perform one of several options: lengthen the current cycle for correction, restart the instruction, perform a diagnostic routine, etc.

A syndrome word, five to eight bits in length and containing all necessary information about the existence and location of an error, is made available to the system at the SYO_{0-7} pins. Error logging may be accomplished by latching the syndrome and the memory address of the word in error.

WRITE CYCLE

For a full write, in which an entire word is written to memory, the data is written directly to the RAM, bypassing the 8206. The same data enters the 8206 through the WDI pins where check bits are generated. The Byte Mark inputs must be low to tristate the DO drivers. The check bits, 5 to 8 in number, are then written to the RAM through the CBO pins for storage along with the data word. In a multi-chip system, the master writes the check bits using partial parity information from the slaves.

In a partial write, part of the data word is overwritten, and part is retained in memory. This is accomplished by performing a read-modify-write cycle. The complete old word is read into the 8206 and corrected, with the syndrome internally latched by R/$\overline{W}$ going low. Only that part of the word not to be modified is output onto the DO pins, as controlled by the Byte Mark inputs. That portion of the word to be overwrit-

ten is supplied by the system bus. The 8206 then calculates check bits for the new word, using the byte from the previous read and the new byte from the system bus, and writes them to the memory.

READ-MODIFY-WRITE CYCLES

Upon detection of an error the 8206 may be used to correct the bit in error in memory. This reduces the probability of getting multiple-bit errors in subsequent read cycles. This correction is handled by executing read-modify-write cycles.

The read-modify-write cycle is controlled by the R/$\overline{W}$ input. After (during) the read cycle, the system dynamic RAM controller or CPU examines the 8206 $\overline{ERROR}$ and CE outputs to determine if a correctable error occurred. If it did, the dynamic RAM controller or CPU forces R/$\overline{W}$ low, telling the 8206 to latch the generated syndrome and drive the corrected check bits onto the CBO outputs. The corrected data is available on the DO pins. The DRAM controller then writes the corrected data and corresponding check bits into memory.

The 8206 may be used to perform read-modify-writes in one or two RAM cycles. If it is done in two cycles, the 8206 latches are used to hold the data and check bits from the read cycle to be used in the following write cycle. The Intel 8207 Dual Port Dynamic RAM controller allows read-modify-write cycles in one memory cycle. See the System Environment section.

INITIALIZATION

A memory system operating with ECC requires some form of initialization at system power-up in order to set valid data and check bit information in memory. The 8206 supports memory initialization by the write zero function. By activating the $\overline{WZ}$ pin, the 8206 will write a data pattern of zeros and the associated check bits in the current write cycle. By thus writing to all memory at power-up, a controller can set memory to valid data and check bits. Massive memory failure, as signified by both data and check bits all ones or zeros, will be detected as an uncorrectable error.

MULTI-CHIP SYSTEMS

A single 8206 handles 8 or 16 bits of data and 5 or 6 check bits, respectively. Up to 5 8206's can be cascaded for 80 bit memories with 8 check bits.

When cascaded, one 8206 operates as a master, and all others as slaves. As an example, during a read cycle in a 32 bit system with one master and one slave, the slave calculates parity on its portion of the word—"partial parity"—and presents it to the master through the PPO pins. The master combines the partial parity from the slave with the parity it calculated from its own portion of the word to generate the syndrome. The syndrome is then returned by the master to the slave for error correction. In systems with more than one slave the above description continues to apply, except that the partial parity outputs of the slaves must be XOR'd externally. Figure 4 shows the necessary external logic for multi-chip systems. Write and read-modify-write cycles are carried out analogously. See the System Operation section for multi-chip wiring diagrams.

There are several pins used to define whether the 8206 will operate as a master or a slave. Tables 3 and 4 illustrate how these pins are tied.

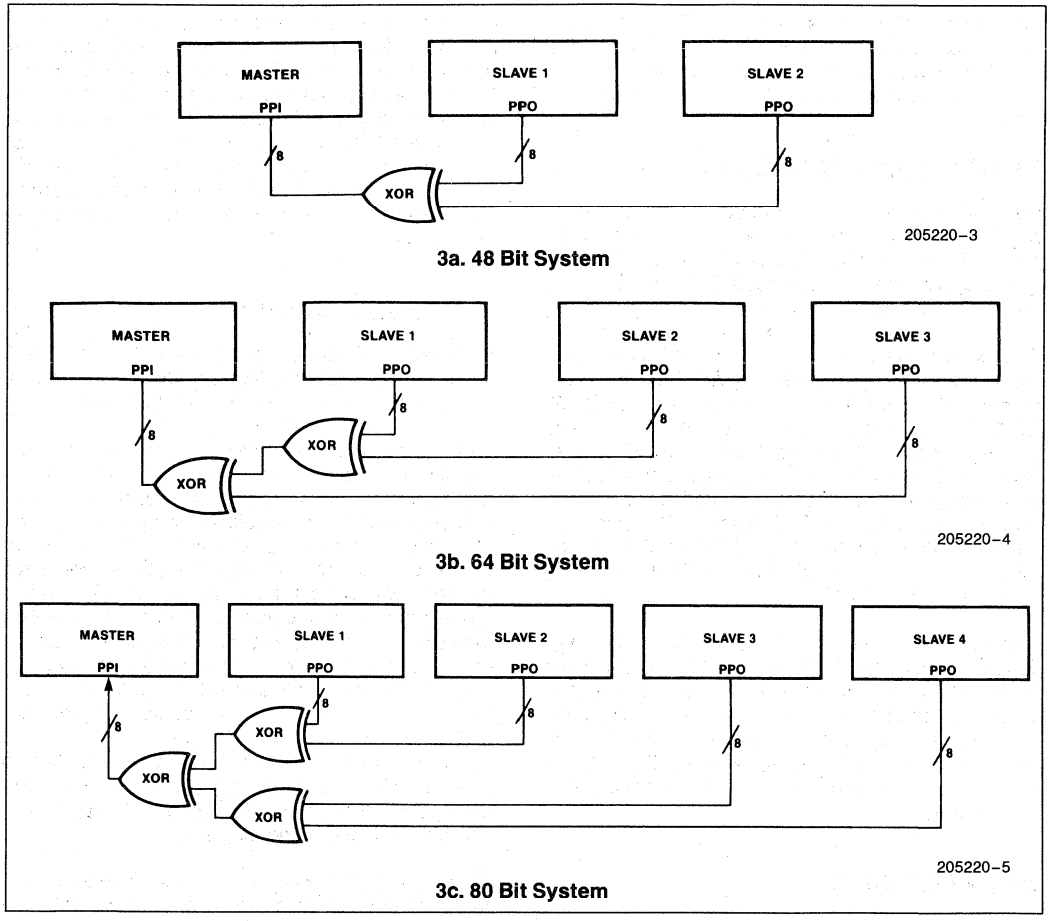


Figure 4. External Logic for Multi-Chip Sytems

Table 3. Master/Slave Pin Assignments

Pin No.	Pin Name	Master	Slave 1	Slave 2	Slave 3	Slave 4
4	$M/\overline{S}$	+5V	gnd	gnd	gnd	gnd
3	$\overline{SEDCU}$	+5V	+5V	+5V	+5V	+5V
13	PPI_0/POS_0	PPI	gnd	+5V	gnd	+5V
14	PPI_1/POS_1	PPI	gnd	gnd	+5V	+5V
15	PPI_2/NSL_0	PPI	*	+5V	+5V	+5V
16	PPI_3/NSL_1	PPI	*	+5V	+5V	+5V

NOTE:
Pins 13, 14, 15, 16 have internal pull-up resistors and may be left as N.C. where specified as connecting to +5V.

Table 4. NSL Pin Assignments for Slave 1

Number of Slaves				
Pin	**1**	**2**	**3**	**4**
PPI_2/NSL_0	GND	+5V	GND	+5V
PPI_3/NSL_1	GND	GND	+5V	+5V

The timing specifications for multi-chip systems must be calculated to take account of the external XOR gating in 3, 4 and 5-chip systems. Let tXOR be the delay for a single external TTL XOR gate. Then the following equations show how to calculate the relevant timing parameters for 2-chip (n = 0), 3-chip (n = 1), 4-chip (n = 2), and 5-chip (n = 2) systems:

Data-in to corrected data-out (read cycle) =

TDVSV + TPVSV + TSVQV + ntXOR

Data-in to error flag (read cycle) =

TDVSV + TPVEV + ntXOR

Data-in to correctable error flag (read cycle) =

TDVSV + TPVSV + TSVCV + ntXOR

Write data to check-bits valid (full write cycle) =

TQVQV + TPVSV + ntXOR

Data-in to check-bits valid (read-mod-write cycle) =

TDVSV + TPVSV + TSVQV + TQVQV + TPVSV + 2ntXOR

Data-in to check-bits valid (non-correcting read-modify-write cycle) =

TDVQU + TQVQV + TPVSV + ntXOR

HAMMING CODE

The 8206 uses a modified Hamming code which was optimized for multi-chip EDCU systems. The code is such that partial parity is computed by all 8206's in parallel. No 8206 requires more time for propagation through logic levels than any other one, and hence no one device becomes a bottleneck in the parity operation. However, one or two levels of external TTL XOR gates are required in systems with three to five chips. The code appears in Table 5. The check bits are derived from the table by XORing or XNORing together the bits indicated by 'X's in each row corresponding to a check bit. For example, check bit 0 in the MASTER for data word 1000110101101011 will be "0". It should be noted that the 8206 will detect the gross-error condition of all lows or all highs.

Error correction is accomplished by identifying the bad bit and inverting it. Table 5 can also be used as an error syndrome table by replacing the 'X's with '1's. Each column then represents a different syndrome word, and by locating the column corresponding to a particular syndrome the bit to be corrected may be identified. If the syndrome cannot be located then the error cannot be corrected. For example, if the syndrome word is 00110111, the bit to be corrected is bit 5 in the slave one data word (bit 21).

The syndrome decoding is also summarized in Tables 6 and 7 which can be used for error logging. By finding the appropriate syndrome word (starting with bit zero, the least significant bit), the result is either: 1) no error; 2) an identified (correctable) single bit error; 3) a double bit error; or 4) a multi-bit uncorrectable error.

Table 5. Modified Hamming Code Check Bit Generation

Check bits are generated by XOR'ing (except for the CB0 and CB1 data bits, which are XNOR'ed in the Master) the data bits in the rows corresponding to the check bits. Note there are 6 check bits in a 16-bit system, 7 in a 32-bit system, and 8 in 48-or-more-bit systems.

BYTE NUMBER		0								1								OPERATION
BIT NUMBER		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
CHECK BITS	CB0 =	x	x	-	x	-	x	x	-	x	-	-	x	-	x	-	-	XNOR
	CB1 =	x	-	x	-	-	x	-	x	-	x	-	x	x	-	x	-	XNOR
	CB2 =	-	x	x	-	x	-	x	x	-	-	x	-	x	-	-	x	XOR
	CB3 =	x	x	x	x	x	-	-	-	x	x	x	-	-	-	-	-	XOR
	CB4 =	-	-	-	x	x	x	x	x	-	-	-	-	-	x	x	x	XOR
	CB5 =	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	XOR
	CB6 =	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	XOR
	CB7 =	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	XOR
DATA BITS		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	
		0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	
		16 BIT OR MASTER																

	2								3								OPERATION
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
	-	x	x	x	-	x	x	-	-	x	x	-	-	x	-	-	XOR
	x	x	x	-	-	x	-	x	x	x	-	-	-	-	-	x	XOR
	-	x	x	x	-	x	x	x	-	-	x	x	-	-	-	-	XOR
	x	x	-	-	x	-	x	x	x	-	-	x	x	-	-	-	XOR
	x	x	-	-	x	x	x	x	-	-	-	-	x	-	x	-	XOR
	-	-	-	x	x	x	x	x	-	-	-	-	-	x	x	x	XOR
	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	XOR
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	XOR
	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3	
	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
	SLAVE #1																

BYTE NUMBER		4								5								6								7								8								9								OPERATION
BIT NUMBER		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
CHECK BITS	CB0 =	x	x	-	x	-	x	x	-	x	-	-	x	-	x	-	-	x	-	x	x	-	-	x	-	x	-	x	x	-	-	x	-	-	x	x	-	x	-	-	x	-	x	-	-	x	-	-	-	XOR
	CB1 =	x	-	x	-	-	x	-	x	-	x	-	x	x	-	x	-	-	x	x	-	-	-	x	x	x	x	x	-	-	x	-	x	-	x	x	x	-	-	x	x	-	-	x	x	-	-	-	-	XOR
	CB2 =	-	x	x	-	x	-	x	x	-	-	x	-	x	-	-	x	-	x	x	x	-	x	x	-	-	x	x	-	-	x	x	-	-	x	x	-	x	x	-	-	-	-	x	x	-	x	-	x	XOR
	CB3 =	x	x	x	x	x	-	-	-	x	x	x	-	-	-	-	-	x	-	x	-	-	x	x	-	x	x	-	-	x	-	-	x	-	x	x	x	x	-	-	x	x	x	-	-	x	-	-	-	XOR
	CB4 =	-	-	-	x	x	x	x	x	-	-	-	-	-	x	x	x	-	-	-	x	x	x	x	x	-	-	-	-	-	x	x	x	-	x	x	-	-	-	x	x	x	x	-	-	-	x	-	XOR
	CB5 =	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	x	-	-	-	x	-	-	-	x	XOR
	CB6 =	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	x	x	-	-	x	x	x	x	-	-	-	-	-	x	-	XOR
	CB7 =	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	XOR
DATA BITS		3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	7						
		2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	
		SLAVE #2																SLAVE #3																SLAVE #4																

Table 6. 8206 Syndrome Decoding

Syndrome Bits 7 6 5 4	0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
	1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
	2 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
	3 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
0 0 0 0	N CB0 CB1 D CB2 D D 18 CB3 D D 0 D 1 2 D
0 0 0 1	CB4 D D 5 D 6 7 D D D 16 D 4 D D 17
0 0 1 0	CB5 D D 11 D 19 12 D D 8 9 D 10 D D 67
0 0 1 1	D 13 14 D 15 D D 21 20 D D 66 D 22 23 D
0 1 0 0	CB6 D D 25 D 26 49 D D 48 24 D 27 D D 50
0 1 0 1	D 52 55 D 51 D D 70 28 D D 65 D 53 54 D
0 1 1 0	D 29 31 D 64 D D 69 68 D D 32 D 33 34 D
0 1 1 1	30 D D 37 D 38 39 D D 35 71 D 36 D D U
1 0 0 0	CB7 D D 43 D 77 44 D D 40 41 D 42 D D U
1 0 0 1	D 45 46 D 47 D D 74 72 D D U D 73 U D
1 0 1 0	D 59 75 D 79 D D 58 60 D D 56 D U 57 D
1 0 1 1	63 D D 62 D U U D D U U D 61 D D U
1 1 0 0	D U U D U D D U 76 D D U D U U D
1 1 0 1	78 D D U D U U D D U U D U D D U
1 1 1 0	U D D U D U U D D U U D U D D U
1 1 1 1	D U U D U D D U U D D U D U U D

N = No Error
CBX = Error in Check Bit X
X = Error in Data Bit X
D = Double Bit Error
U = Uncorrectable Multi-Bit Error

SYSTEM ENVIRONMENT

The 8206 interface to a typical 32 bit memory system is illustrated in Figure 5. For larger systems, the partial parity bits from slaves two to four must be XOR'ed externally, which calls for one level of XOR gating for three 8206's and two levels for four or five 8206's.

The 8206 is designed for direct connection to the Intel 8207 Dynamic RAM Controller. The 8207 has the ability to perform dual port memory control, and Figure 6 illustrates a highly integrated dual port RAM implementation using the 8206 and 8207. The 8206/8207 combination permits such features as automatic scrubbing (correcting errors in memory during refresh), extending RAS and CAS timings for Read-Modify-Writes in single memory cycles, and automatic memory initialization upon reset. Together these two chips provide a complete dual-port, error-corrected dynamic RAM subsystem.

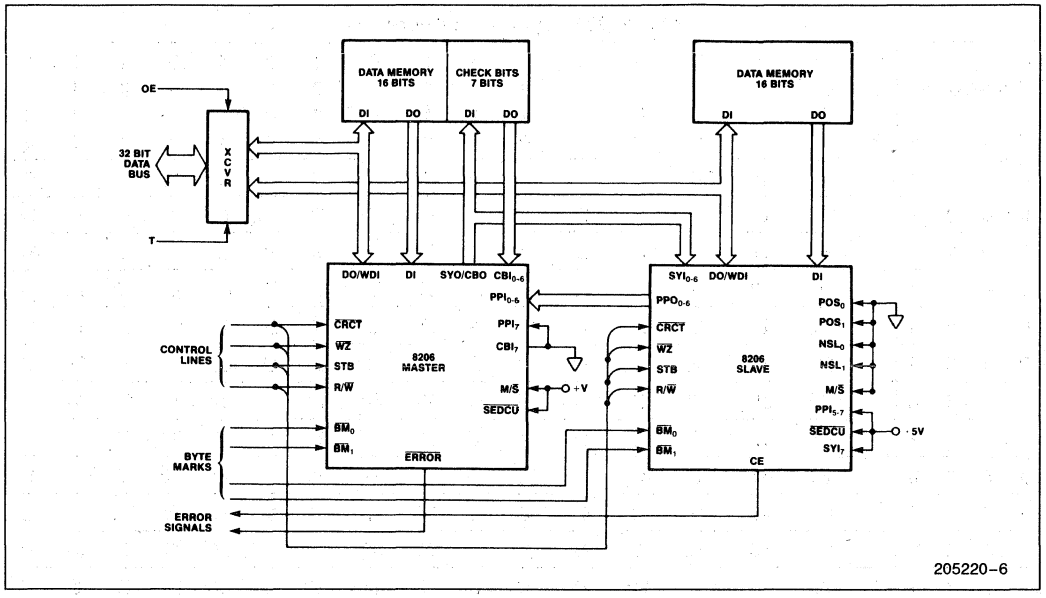


Figure 5. 32-Bit 8206 System Interface

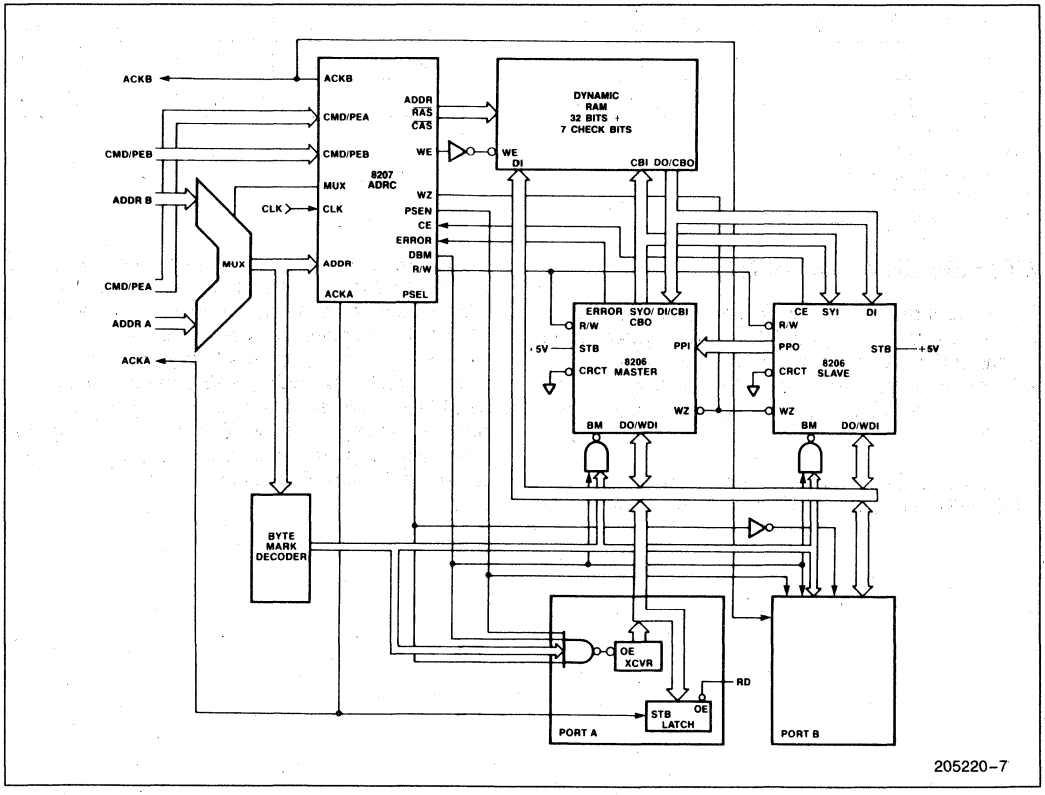


Figure 6. Dual Port RAM Subsystem with 8206/8207 (32-bit bus)

MEMORY BOARD TESTING

The 8206 lends itself to straightforward memory board testing with a minimum of hardware overhead. The following is a description of four common test modes and their implementation.

Mode 0— Read and write with error correction.
Implementation: This mode is the normal 8206 operating mode.

Mode 1— Read and write data with error correction disabled to allow test of data memory.
Implementation: This mode is performed with $\overline{CRCT}$ deactivated.

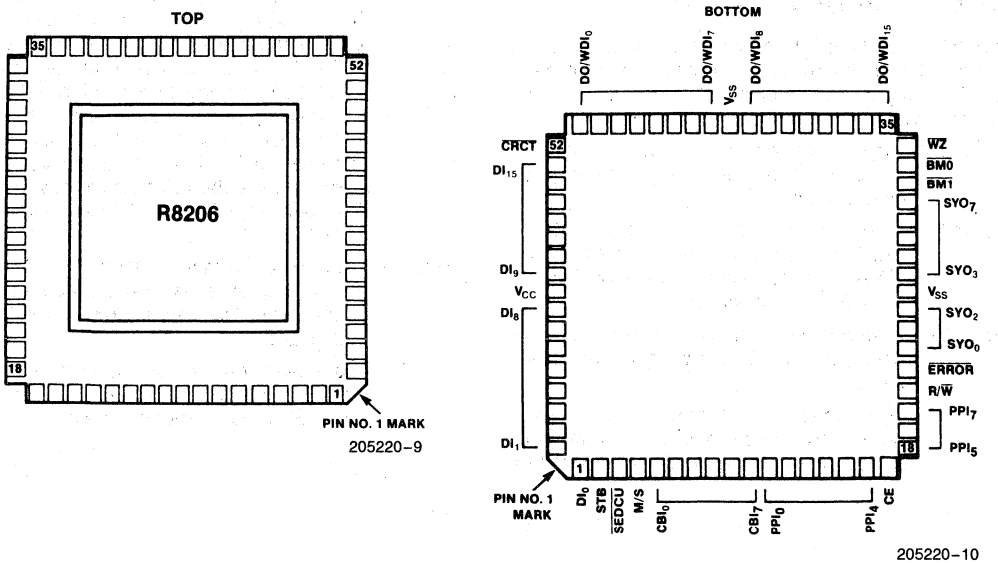
Mode 2— Read and write check bits with error correction disabled to allow test of check bits memory.

Implementation: Any pattern may be written into the check bits memory by judiciously choosing the proper data word to generate the desired check bits, through the use of the 8206 Hamming code. To read out the check bits it is first necessary to fill the data memory with all zeros, which may be done by activating $\overline{WZ}$ and incrementing memory addresses with $\overline{WE}$ to the check bits memory held inactive, and then performing ordinary reads. The check bits will then appear directly at the SYO outputs, with bits CB0 and CB1 inverted.

Mode 3— Write data, without altering or writing check bits, to allow the storage of bit combinations to cause error correction and detection.

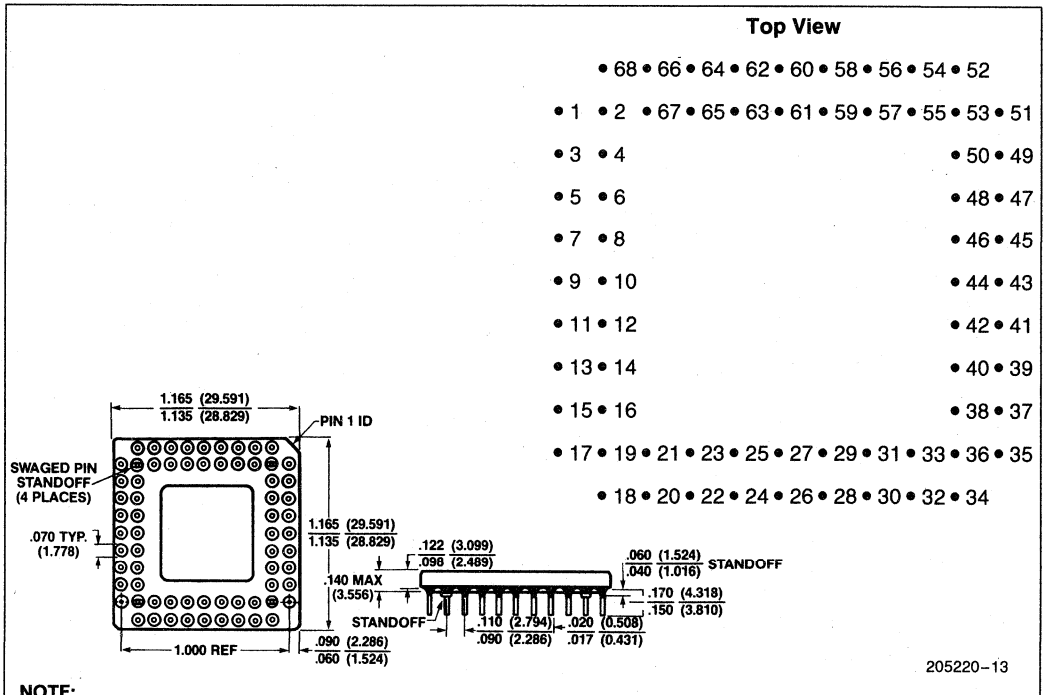
Implementation: This mode is implemented by writing the desired word to memory with $\overline{WE}$ to the check bits array held inactive.

2



NOTE:
68 pin JEDEC TYPE A hermetic chip carrier

Figure 8a. 8206 Leadless Chip Carrier (LCC) Pinout Diagram



NOTE:
68 lead ceramic pin grid array package, type A

Figure 8b. 8206 Pin Grid Array (PGA) Package and Pinout Diagram

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage On Any Pin
with Respect to Ground.....−0.5V to +7V
Power Dissipation.....1.5W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

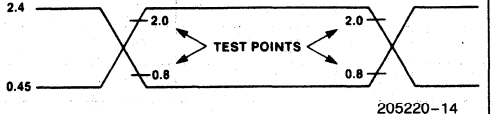
**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5.0V ± 10%, V_{SS} = GND

Symbol	Parameter	Min	Max	Units	Test Conditions
I_{CC}	Power Supply Current —Single 8206 or Slave #1 —Master in Multi-Chip or Slaves #2, 3, 4		270 230	mA mA	
V_{IL}(1)	Input Low Voltage	−0.5	0.8	V	
V_{IH}(1)	Input High Voltage	2.0	V_{CC} + 0.5V	V	
V_{OL}	Output Low Voltage —DO —All Others		0.45 0.45	V V	I_{OL} = 8 mA I_{OL} = 2.0 mA
V_{OH}	Output High Voltage —DO, CBO —All Other Outputs	2.6 2.4		V V	I_{OH} = −2 mA I_{OH} = −0.4 mA
I_{LO}	I/O Leakage Current —PPI_4/CE —DO/WDI_{0-15}		±20 ±10	μA μA	0.45V ≤ $V_{I/O}$ ≤ V_{CC}
I_{LI}	Input Leakage Current —$PPI_{0-3, 5-7}$, CBI_{6-7}, $\overline{SEDCU}$(2) —All Other Input Only Pins		±20 ±10	μA μA	0V ≤ V_{IN} ≤ V_{CC}

NOTES:
1. $\overline{SEDCU}$ (pin 3) and M/$\overline{S}$ (pin 4) are device strapping options and should be tied to V_{CC} or GND. V_{IH} min = V_{CC} −0.5V and V_{IL} max = 0.5V.
2. PPI_{0-7} (pins 13–20) and CBI_{6-7} (pins 11, 12) have internal pull-up resistors and if left unconnected will be pulled to V_{CC}.

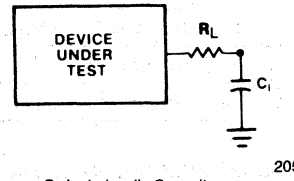
A.C. TESTING INPUT, OUTPUT WAVEFORM



205220-14

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT



205220-15

C_L Includes Jig Capacitance

A.C. CHARACTERISTICS

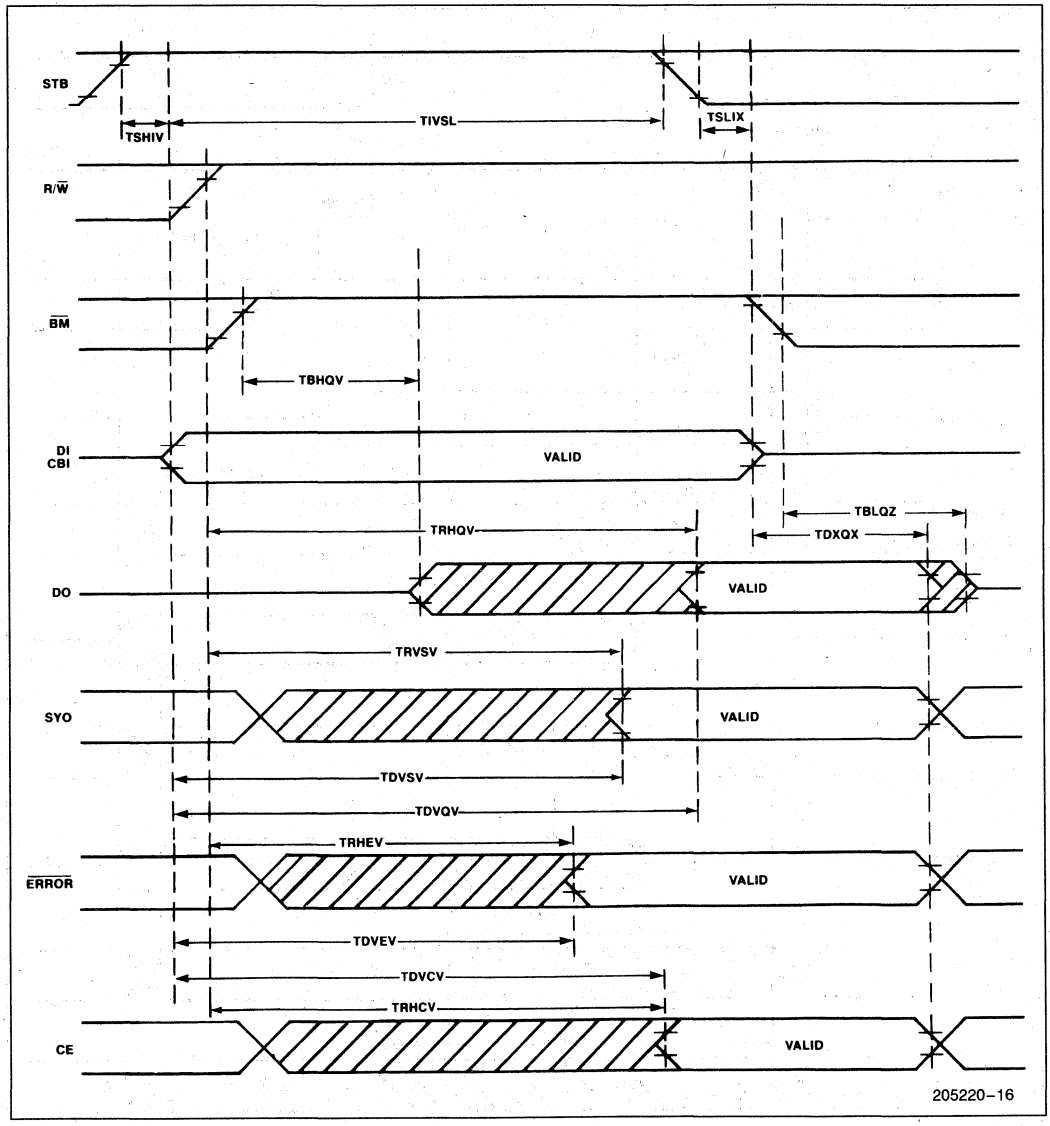
T_A = 0°C to 70°C, V_{CC} = +5V ± 10%, V_{SS} = 0V, R_L = 22Ω, C_L = 50 pF; all times are in ns

Symbol	Parameter	8206-1		8206		Notes
		Min	Max	Min	Max	
T_{RHEV}	$\overline{ERROR}$ Valid from R/$\overline{W}$ ↑		20		25	
T_{RHCV}	CE Valid from R/$\overline{W}$ ↑ (Single 8206)		34		44	
T_{RHQV}	Corrected Data Valid from R$\overline{W}$ ↑		44		54	1
T_{RVSV}	SYO/CBO/PPO Valid from R/$\overline{W}$		32		42	1
T_{DVEV}	$\overline{ERROR}$ Valid from Data/Check Bits In		35		42	
T_{DVCV}	CE Valid from Data/Check Bits In		50		70	
T_{DVQV}	Corrected Data Valid from Data/Check Bits In		55		67	
T_{DVSV}	SYO/PPO Valid from Data/Check Bits In		40		55	
T_{BHQV}	Corrected Data Access Time		35		37	
T_{BXQX}	Hold Time from Data/Check Bits In	0		0		1
T_{BLQZ}	Corrected Data Float Delay	0	25	0	28	1
T_{SHIV}	STB High to Data/Check Bits In Valid	30		30		2
T_{IVSL}	Data/Check Bits In to STB ↓ Set-Up	5		5		
T_{SLIX}	Data/Check Bits In from STB ↓ Hold	15		25		
T_{PVEV}	$\overline{ERROR}$ Valid from Partial Parity In		21		30	3
T_{PVQV}	Corrected Data (Master) from Partial Parity In		46		61	1, 3
T_{PVSV}	Syndrome/Check Bits Out from Partial Parity In		32		43	1, 3
T_{SVQV}	Corrected Data (Slave) Valid from Syndrome		41		51	3
T_{SVCV}	CE Valid from Syndrome (Slave Number 1)		43		48	3
T_{QVQV}	Check Bits/Partial Parity Out from Write Data In		44		64	1
T_{RHSX}	Check Bits/Partial Parity Out from R/W, $\overline{WZ}$ Hold	0		0		1
T_{RLSX}	Syndrome Out from R/$\overline{W}$ Hold	0		0		
T_{QXQX}	Hold Time from Write Data In	0		0		1
T_{SVRL}	Syndrome Out to R/$\overline{W}$ ↓ Set-Up	5		17		3
T_{DVRL}	Data/Check Bits to R/$\overline{W}$ Set-Up	24		39		1
T_{DVQU}	Uncorrected Data Out from Data In		29		32	
T_{TVQV}	Corrected Data Out from $\overline{CRCT}$ ↓		25		30	
T_{WLQL}	$\overline{WZ}$ ↓ to Zero Out		25		30	
T_{WHQX}	Zero Out from $\overline{WZ}$ ↑ Hold	0		0		0

NOTES:
1. A.C. Test Levels for CBO and DO are 2.4V and 0.8V.
2. T_{SHIV} is required to guarantee output delay timings: T_{DVEV}, T_{DVCV}, T_{DVQV}, T_{DVSV}, T_{SHIV} + T_{IVSL} guarantees a min STB pulse width of 35 ns.
3. Not required for 8/16 bit systems.

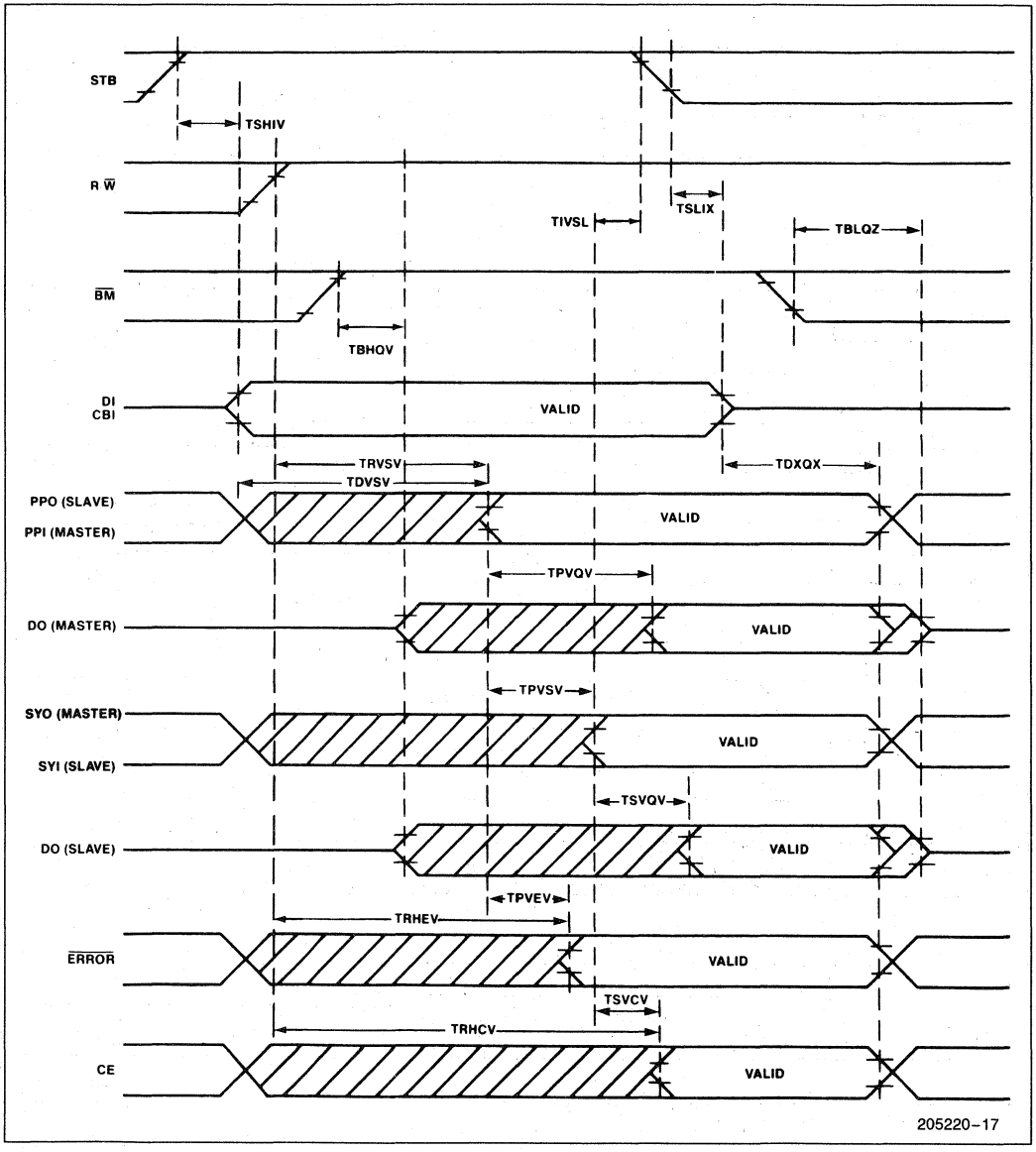
WAVEFORMS

READ



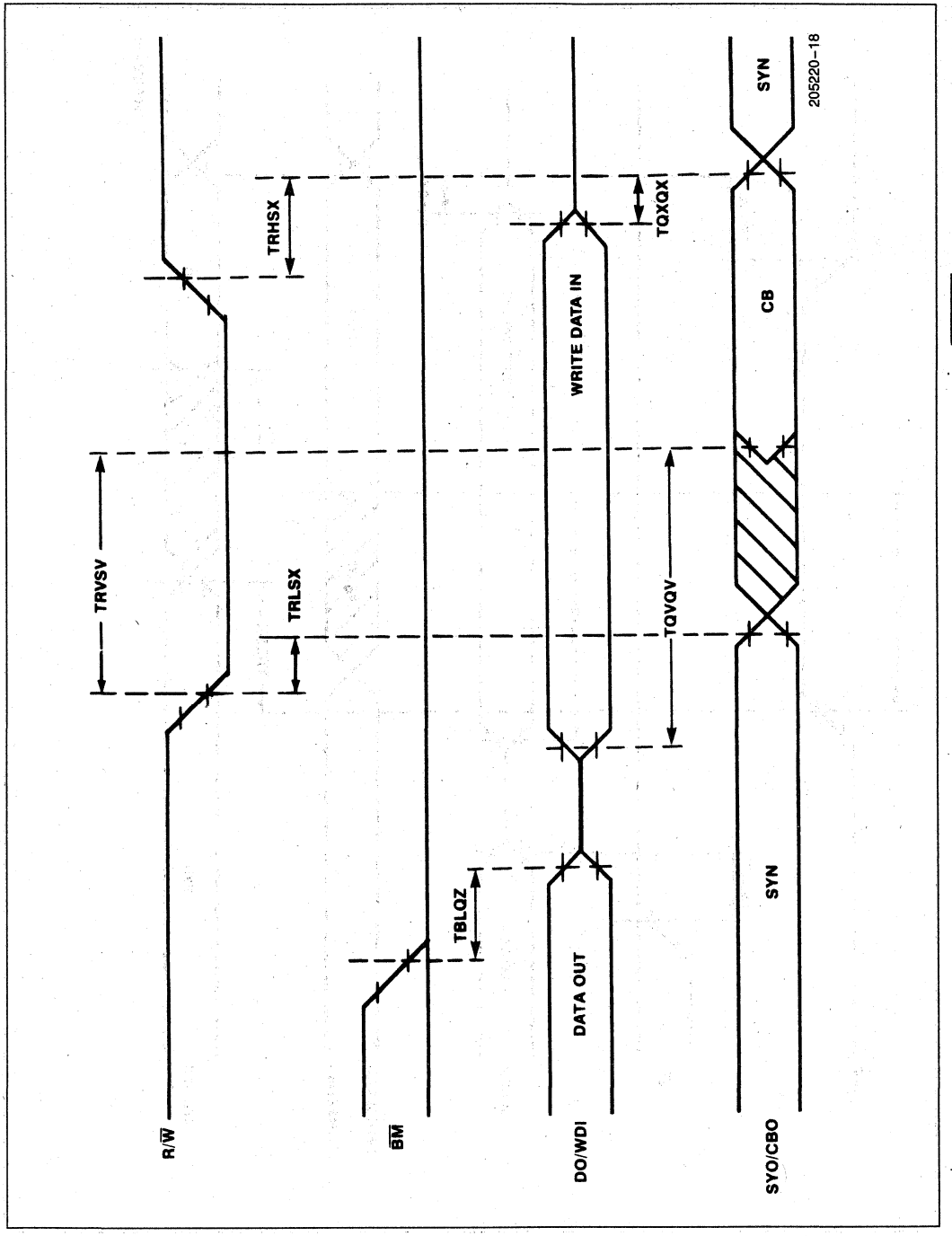
WAVEFORMS (Continued)

READ—MASTER/SLAVE



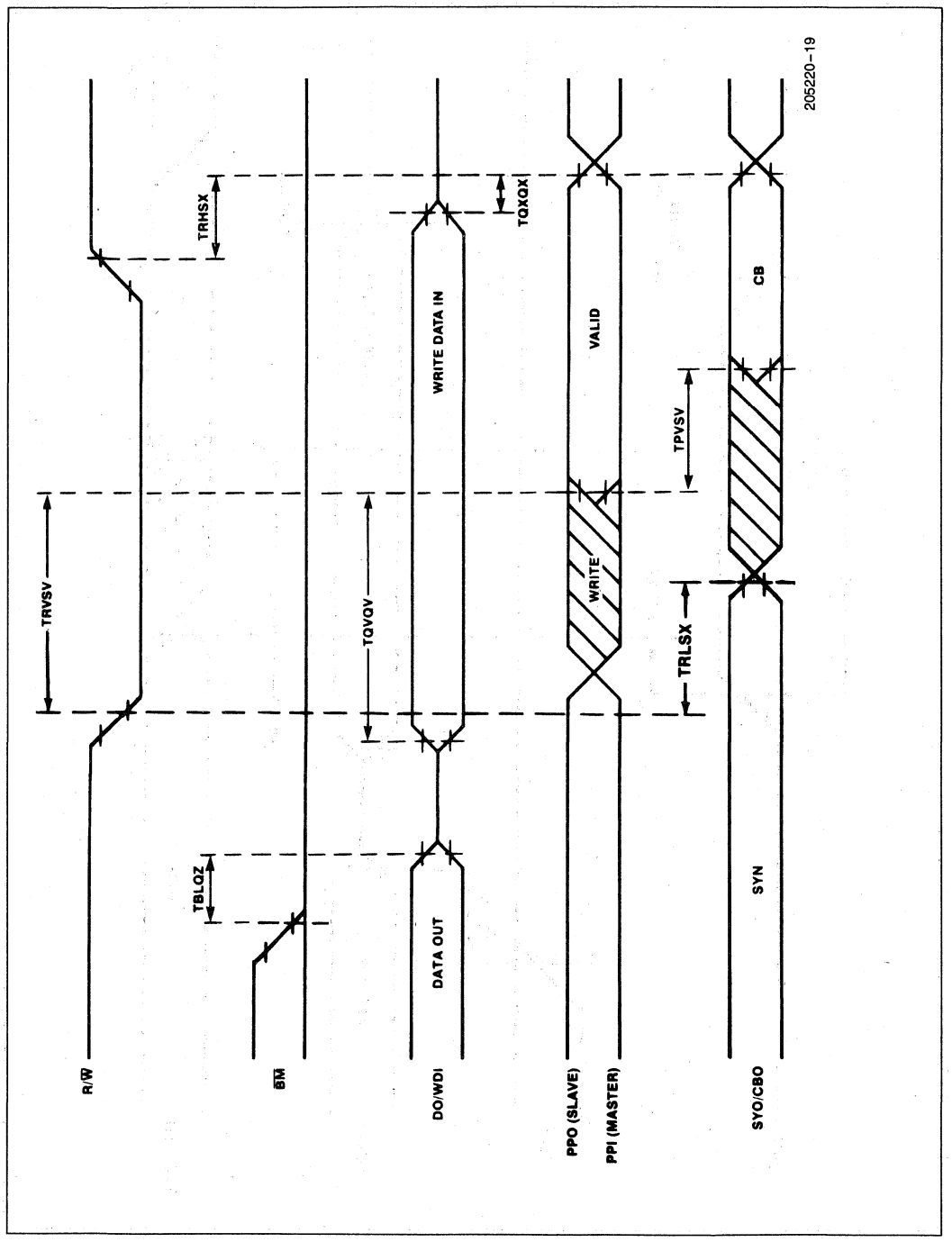
WAVEFORMS (Continued)

FULL WRITE



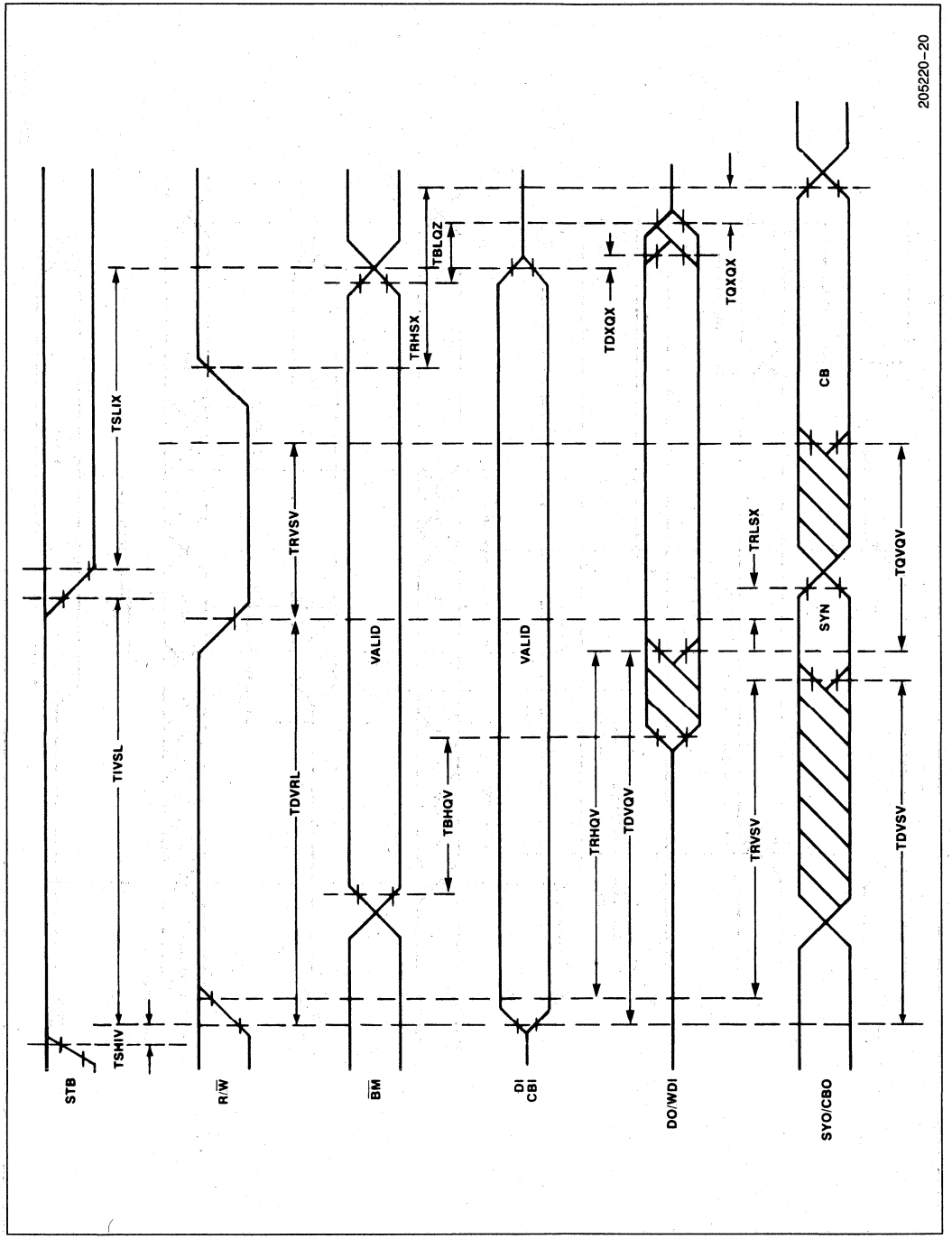
WAVEFORMS (Continued)

FULL WRITE—MASTER/SLAVE



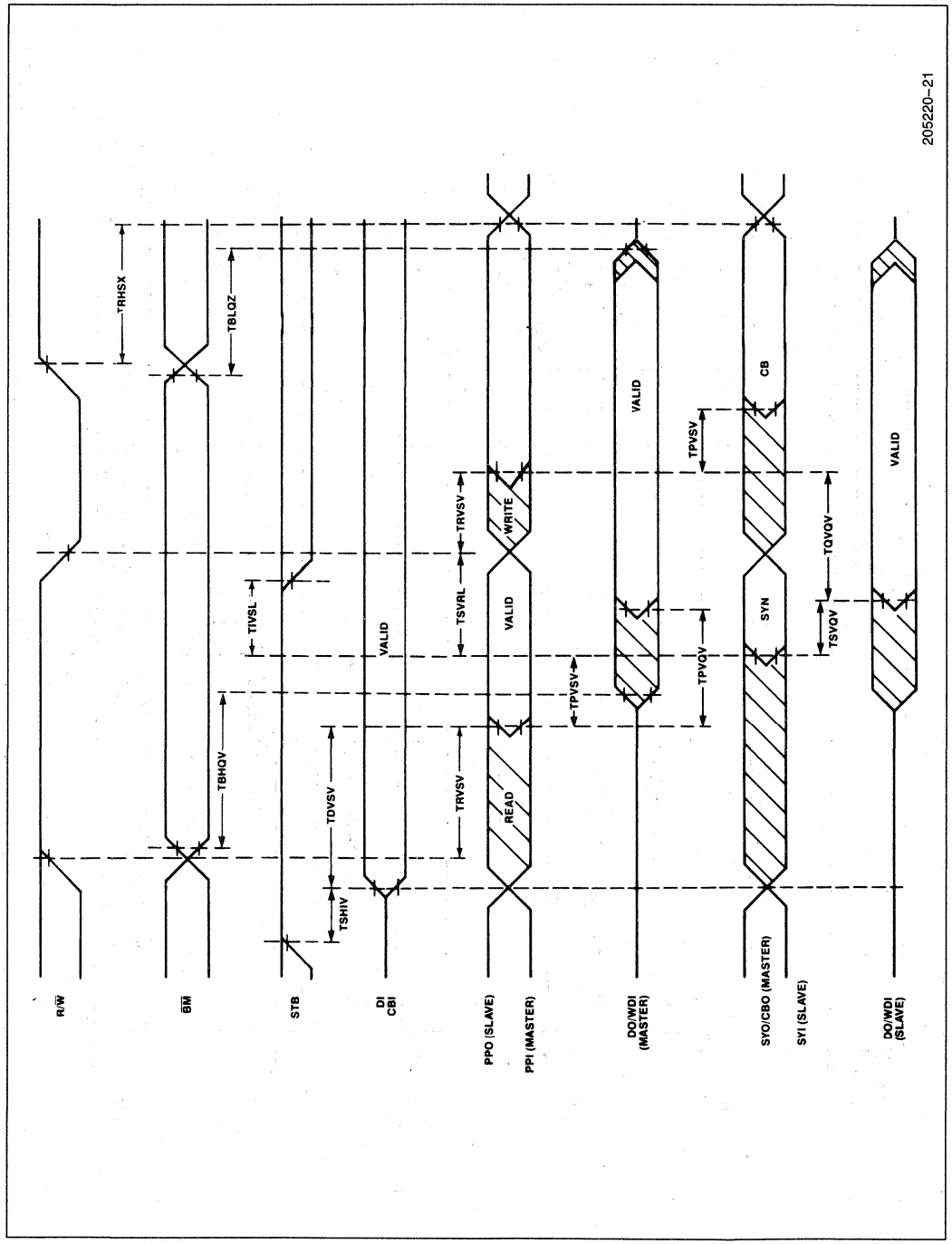
WAVEFORMS (Continued)

READ MODIFY WRITE



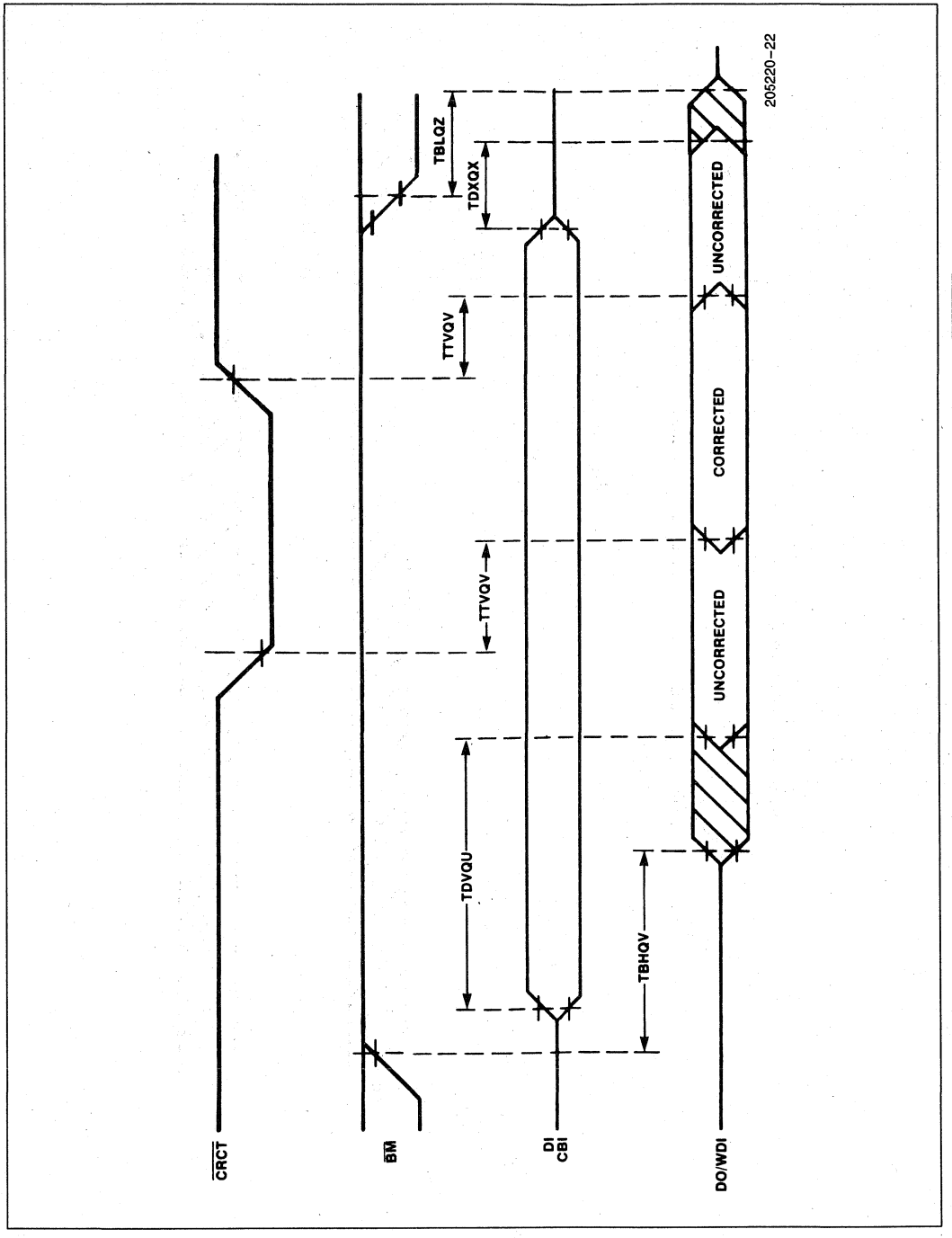
WAVEFORMS (Continued)

READ MODIFY WRITE—MASTER/SLAVE



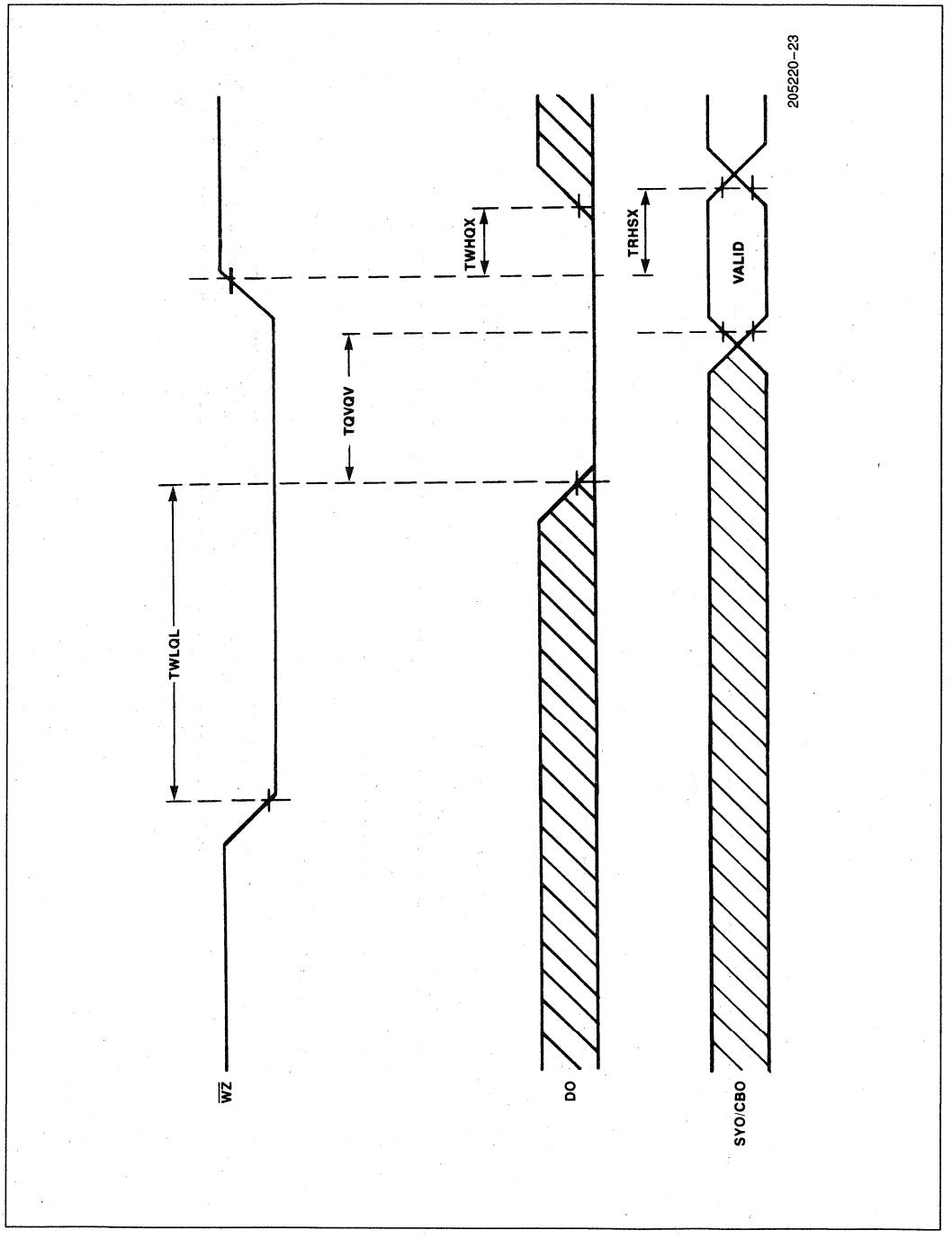
WAVEFORMS (Continued)

NON-CORRECTING READ



WAVEFORMS (Continued)

WRITE ZERO

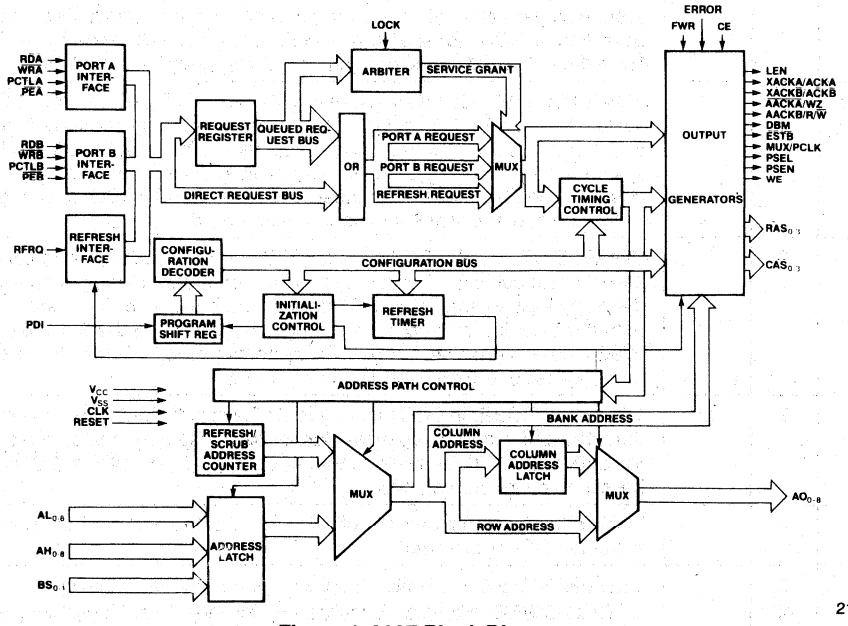


8207
DUAL-PORT DYNAMIC RAM CONTROLLER

- **Provides All Signals Necessary to Control 16K, 64K and 256K Dynamic RAMs**
- **Directly Addresses and Drives up to 2 Megabytes without External Drivers**
- **Supports Single and Dual-Port Configurations**
- **Automatic RAM Initialization in All Modes**
- **Four Programmable Refresh Modes**
- **Transparent Memory Scrubbing in ECC Mode**
- **Fast Cycle Support for 8 MHz 80286 with 8207-16**
- **Slow Cycle Support for 8 MHz, 10 MHz 8086/88, 80186/188 with 8207-8, 8207-10**
- **Provides Signals to Directly Control the 8206 Error Detection and Correction Unit**
- **Supports Synchronous or Asynchronous Operation on Either Port**
- **68 Lead JEDEC Type A Leadless Chip Carrier (LCC) and Pin Grid Array (PGA), Both in Ceramic.**

2

The Intel 8207 Dual-Port Dynamic RAM Controller is a high-performance, systems-oriented, Dynamic RAM controller that is designed to easily interface 16K, 64K and 256K Dynamic RAMs to Intel and other microprocessor systems. A dual-port interface allows two different busses to independently access memory. When configured with an 8206 Error Detection and Correction Unit the 8207 supplies the necessary logic for designing large error-corrected memory arrays. This combination provides automatic memory initialization and transparent memory error scrubbing.



210463-1

Figure 1. 8207 Block Diagram

September 1987
Order Number: 210463-007

Table 1. Pin Description

Symbol	Pin	Type	Name and Function
LEN	1	O	**ADDRESS LATCH ENABLE:** In two-port configurations, when Port A is running with iAPX 286 Status interface mode, this output replaces the ALE signal from the system bus controller of port A and generates an address latch enable signal which provides optimum setup and hold timing for the 8207. This signal is used in Fast Cycle operation only.
$\overline{XACKA}$/ $\overline{ACKA}$	2	O	**TRANSFER ACKNOWLEDGE PORT A/ACKNOWLEDGE PORT A:** In non-ECC mode, this pin is $\overline{XACKA}$ and indicates that data on the bus is valid during a read cycle or that data may be removed from the bus during a write cycle for Port A. $\overline{XACKA}$ is a Multibus-compatible signal. In ECC mode, this pin is $\overline{ACKA}$ which can be configured, depending on the programming of the X program bit, as an $\overline{XACK}$ or $\overline{AACK}$ strobe. The SA programming bit determines whether the $\overline{AACK}$ will be an early $\overline{EAACKA}$ or a late $\overline{LAACKA}$ interface signal.
$\overline{XACKB}$/ $\overline{ACKB}$	3	O	**TRANSFER ACKNOWLEDGE PORT B/ACKNOWLEDGE PORT B:** In non-ECC mode, this pin is XACKB and indicates that data on the bus is valid during a read cycle or that data may be removed from the bus during a write cycle for Port B. $\overline{XACKB}$ is a Multibus-compatible signal. In ECC mode, this pin is $\overline{ACKB}$ which can be configured, depending on the programming of the X program bit, as an $\overline{XACK}$ or $\overline{AACK}$ strobe. The SB programming bit determines whether the $\overline{AACK}$ will be an early $\overline{EAACKB}$ or a late $\overline{LAACKB}$ interface signal.
$\overline{AACKA}$/ $\overline{WZ}$	4	O	**ADVANCED ACKNOWLEDGE PORT A/WRITE ZERO:** In non-ECC mode, this pin is $\overline{AACKA}$ and indicates that the processor may continue processing and that data will be available when required. This signal is optimized for the system by programming the SA program bit for synchronous or asynchronous operation. In ECC mode, after a RESET, this signal will cause the 8206 to force the data to all zeros and generate the appropriate check bits.
$\overline{AACKB}$/ R/$\overline{W}$	5	O	**ADVANCED ACKNOWLEDGE PORT B/READ/WRITE:** In non-ECC mode, this pin is $\overline{AACKB}$ and indicates that the processor may continue processing and that data will be available when required. This signal is optimized for the system by programming the SB program bit for synchronous or asynchronous operation. In ECC mode, this signal causes the 8206 EDCU to latch the syndrome and error flags and generate check bits.
$\overline{DBM}$	6	O	**DISABLE BYTE MARKS:** This is an ECC control output signal indicating that a read or refresh cycle is occurring. This output forces the byte address decoding logic to enable all 8206 data output buffers. In ECC mode, this output is also asserted during memory initialization and the 8-cycle dynamic RAM wake-up exercise. In non-ECC systems this signal indicates that either a read, refresh or 8-cycle warm-up is in progress.
$\overline{ESTB}$	7	O	**ERROR STROBE:** In ECC mode, this strobe is activated when an error is detected and allows a negative-edge triggered flip-flop to latch the status of the 8206 EDCU CE for systems with error logging capabilities. $\overline{ESTB}$ will not be issued during refresh cycles.
LOCK	8	I	**LOCK:** This input instructs the 8207 to lock out the port not being serviced at the time LOCK was issued.
V_{CC}	9 43	I	**DRIVER POWER:** +5 volts. Supplies V_{CC} for the output drivers. **LOGIC POWER:** +5 volts. Supplies V_{CC} for the internal logic circuits.
CE	10	I	**CORRECTABLE ERROR:** This is an ECC input from the 8206 EDCU which instructs the 8207 whether a detected error is correctable or not. A high input indicates a correctable error. A low input inhibits the 8207 from activating WE to write the data back into RAM. This should be connected to the CE output of the 8206.

Table 1. Pin Description (Continued)

Symbol	Pin	Type	Name and Function
$\overline{\text{ERROR}}$	11	I	**ERROR:** This is an ECC input from the 8206 EDCU and instructs the 8207 that an error was detected. This pin should be connected to the ERROR output of the 8206.
MUX/ PCLK	12	O	**MULTIPLEXER CONTROL/PROGRAMMING CLOCK:** Immediately after a RESET this pin is used to clock serial programming data into the PDI pin. In normal two-port operation, this pin is used to select memory addresses from the appropriate port. When this signal is high, port A is selected and when it is low, port B is selected. This signal may change state before the completion of a RAM cycle, but the RAM address hold time is satisfied.
PSEL	13	O	**PORT SELECT:** This signal is used to select the appropriate port for data transfer. When this signal is high port A is selected and when it is low port B is selected.
PSEN	14	O	**PORT SELECT ENABLE:** This signal used in conjunction with PSEL provides contention-free port exchange on the data bus. When PSEN is low, port selection is allowed to change state.
WE	15	O	**WRITE ENABLE:** This signal provides the dynamic RAM array the write enable input for a write operation.
$\overline{\text{FWR}}$	16	I	**FULL WRITE:** This is an ECC input signal that instructs the 8207, in an ECC configuration, whether the present write cycle is normal RAM write (full write) or a RAM partial write (read-modify-write) cycle.
RESET	17	I	**RESET:** This signal causes all internal counters and state flip-flops to be reset and upon release of RESET, data appearing at the PDI pin is clocked in by the PCLK output. The states of the PDI, PCTLA, PCTLB and RFRQ pins are sampled by RESET going inactive and are used to program the 8207. An 8-cycle dynamic RAM warm-up is performed after clocking PDI bits into the 8207.
$\overline{\text{CAS0}}$–$\overline{\text{CAS3}}$	18–21	O	**COLUMN ADDRESS STROBE:** These outputs are used by the dynamic RAM array to latch the column address, present on the AO0–8 pins. These outputs are selected by the BS0 and BS1 as programmed by program bits RB0 and RB1. These outputs drive the dynamic RAM array directly and need no external drivers.
$\overline{\text{RAS0}}$–$\overline{\text{RAS3}}$	22–25	O	**ROW ADDRESS STROBE:** These outputs are used by the dynamic RAM array to latch the row address, present on the AO0–8 pins. These outputs are selected by the BS0 and BS1 as programmed by program bits RB0 and RB1. These outputs drive the dynamic RAM array directly and need no external drivers.
V_{SS}	26 60	I I	**DRIVER GROUND:** Provides a ground for the output drivers. **LOGIC GROUND:** Provides a ground for the remainder of the device.
AO0–AO8	35–27	O	**ADDRESS OUTPUTS:** These outputs are designed to provide the row and column addresses of the selected port to the dynamic RAM array. These outputs drive the dynamic RAM array directly and need no external drivers.
BS0–BS1	36–37	I	**BANK SELECT:** These inputs are used to select one of four banks of the dynamic RAM array as defined by the program bits RB0 and RB1.
AL0–AL8	38–42 44–47	I	**ADDRESS LOW:** These lower-order address inputs are used to generate the row address for the internal address multiplexer.
AH0–AH8	48–56	I	**ADDRESS HIGH:** These higher-order address inputs are used to generate the column address for the internal address multiplexer.

2

Table 1. Pin Description (Continued)

Symbol	Pin	Type	Name and Function
PDI	57	I	**PROGRAM DATA INPUT:** This input programs the various user-selectable options in the 8207. The PCLK pin shifts programming data into the PDI input from optional external shift registers. This pin may be strapped high or low to a default ECC (PDI = Logic "1") or non-ECC (PDI = Logic "0") mode configuration.
RFRQ	58	I	**REFRESH REQUEST:** This input is sampled on the falling edge of RESET. If it is high at RESET, then the 8207 is programmed for internal refresh request or external refresh request with failsafe protection. If it is low at RESET, then the 8207 is programmed for external refresh without failsafe protection or burst refresh. Once programmed the RFRQ pin accepts signals to start an external refresh with failsafe protection or external refresh without failsafe protection or a burst refresh.
CLK	59	I	**CLOCK:** This input provides the basic timing for sequencing the internal logic.
$\overline{RDB}$	61	I	**READ FOR PORT B:** This pin is the read memory request command input for port B. This input also directly accepts the $\overline{S1}$ status line from Intel processors.
$\overline{WRB}$	62	I	**WRITE FOR PORT B:** This pin is the write memory request command input for port B. This input also directly accepts the $\overline{S0}$ status line from Intel processors.
$\overline{PEB}$	63	I	**PORT ENABLE FOR PORT B:** This pin serves to enable a RAM cycle request for port B. It is generally decoded from the port address.
PCTLB	64	I	**PORT CONTROL FOR PORT B:** This pin is sampled on the falling edge of RESET. If low after RESET, the 8207 is programmed to accept memory read and write commands, Multibus commands or iAPX 286 status inputs. If high after RESET, the 8207 is programmed to accept status inputs from iAPX 86 or iAPX 186 processors. The $\overline{S2}$ status line should be connected to this input if programmed to accept iAPX 86 or iAPX 186 status inputs. When programmed to accept commands or iAPX 286 status, it should be tied low or it may be used as a Multibus-compatible inhibit signal.
$\overline{RDA}$	65	I	**READ FOR PORT A:** This pin is the read memory request command input for port A. This input also directly accepts the $\overline{S1}$ status line from Intel processors.
$\overline{WRA}$	66	I	**WRITE FOR PORT A:** This pin is the write memory request command input for port A. This input also directly accepts the $\overline{S0}$ status line from Intel processors.
$\overline{PEA}$	67	I	**PORT ENABLE FOR PORT A:** This pin serves to enable a RAM cycle request for port A. It is generally decoded from the port address.
PCTLA	68	I	**PORT CONTROL FOR PORT A:** This pin is sampled on the falling edge of RESET. If low after RESET, the 8207 is programmed to accept memory read and write commands, Multibus commands or iAPX 286 status inputs. If high after RESET, the 8207 is programmed to accept status inputs from iAPX 86 or iAPX 186 processors. The $\overline{S2}$ status line should be connected to this input if programmed to accept iAPX 86 or iAPX 186 status inputs. When programmed to accept commands or iAPX 286 status, it should be tied low or it may be connected to INHIBIT when operating with Multibus.

GENERAL DESCRIPTION

The Intel 8207 Dual-Port Dynamic RAM Controller is a microcomputer peripheral device which provides the necessary signals to address, refresh and directly drive 16K, 64K and 256K dynamic RAMs. This controller also provides the necessary arbitration circuitry to support dual-port access of the dynamic RAM array.

The 8207 supports several microprocessor interface options including synchronous and asynchronous connection to iAPX 86, iAPX 88, iAPX 186, iAPX 188, iAPX 286 and Multibus.

This device may be used with the 8206 Error Detection and Correction Unit (EDCU). When used with the 8206, the 8207 is programmed in the Error Checking and Correction (ECC) mode. In this mode, the 8207 provides all the necessary control signals for the 8206 to perform memory initialization and transparent error scrubbing during refresh.

FUNCTIONAL DESCRIPTION

Processor Interface

The 8207 has control circuitry for two ports each capable of supporting one of several possible bus structures. The ports are independently configurable allowing the dynamic RAM to serve as an interface between two different bus structures.

Each port of the 8207 may be programmed to run synchronous or asynchronous to the processor clock. (See Synchronous/Asynchronous Mode.) The 8207 has been optimized to run synchronously with Intel's iAPX 86, iAPX 88, iAPX 186, iAPX 188, and iAPX 286. When the 8207 is programmed to run in asynchronous mode, the 8207 inserts the necessary synchronization circuitry for the $\overline{RD}$, $\overline{WR}$, $\overline{PE}$, and PCTL inputs.

The 8207 achieves high performance (i.e., no wait states) by decoding the status lines directly from the iAPX 86, iAPX 88, iAPX 186, iAPX 188 and iAPX 286 processors. The 8207 can also be programmed to receive read or write Multibus commands or commands from a bus controller. (See Status/Command Mode.)

The 8207 may be programmed to accept the clock of the iAPX 86, 88, 186, 188 or 286. The 8207 adjusts its internal timing to allow for the different clock frequencies of these microprocessors. (See Microprocessor Clock Frequency Option.)

Figures 2A and 2B show the different processor interfaces to the 8207 using the synchronous or asynchronous mode and status or command interface.

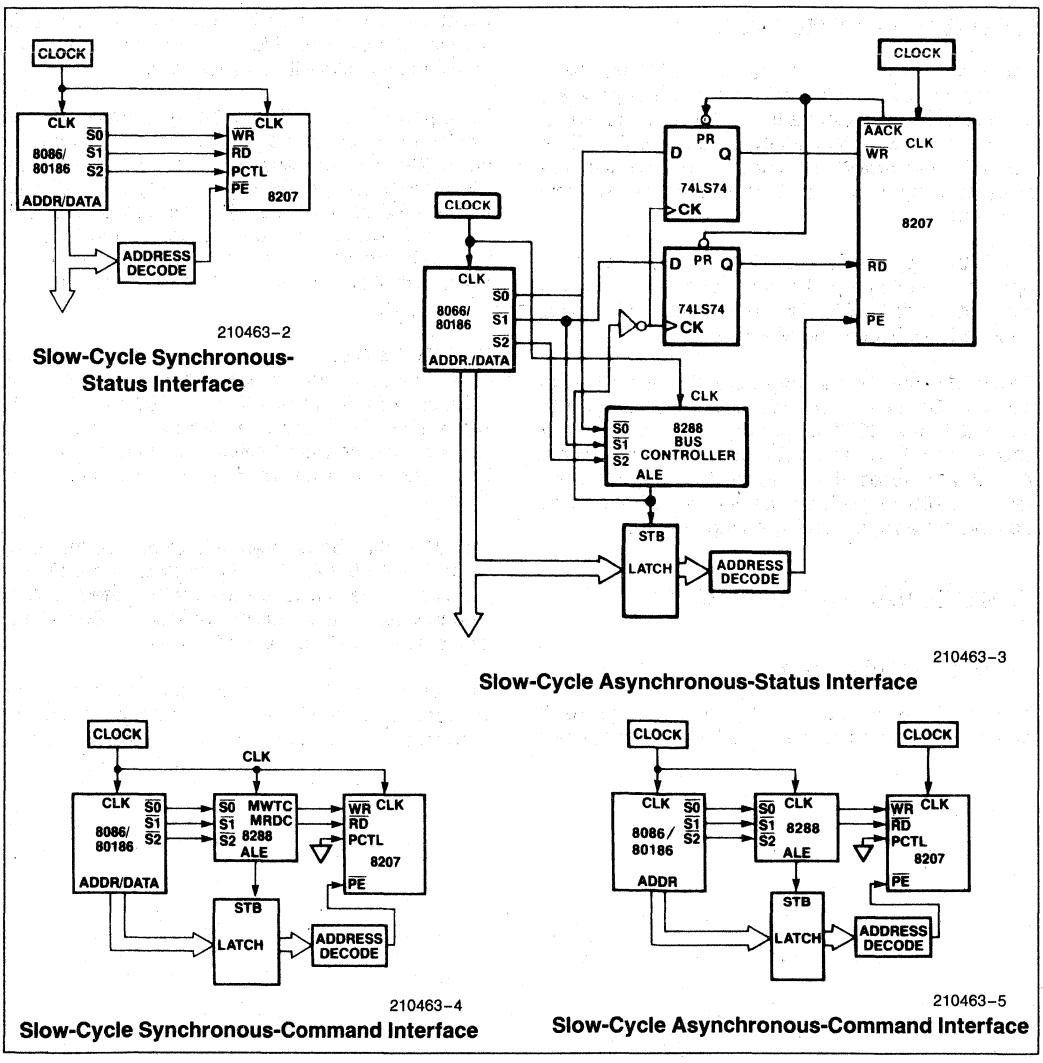


Figure 2A. Slow-Cycle (CFS = 0) Port Interfaces Supported by the 8207

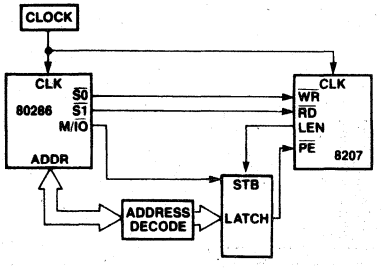
Single-Port Operation

The use of an address latch with the iAPX 286 status interface is not needed since the 8207 can internally latch the addresses with an internal signal similar in behavior to the LEN output. This operation is active only in single-port applications when the processor is interfaced to port A.

Dual-Port Operation

The 8207 provides for two-port operation. Two independent processors may access memory controlled by the 8207. The 8207 arbitrates between each of the processor requests and directs data to or from the appropriate port. Selection is done on a priority concept that reassigns priorities based upon past history. Processor requests are internally queued.

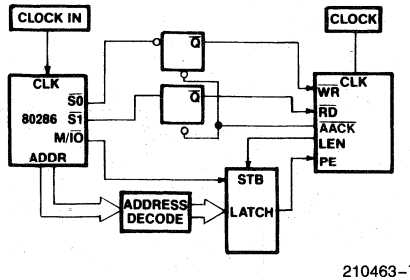
Figure 3 shows a dual-port configuration with two iAPX 86 systems interfacing to dynamic RAM. One of the processor systems is interfaced synchronously using the status interface and the other is interfaced asynchronously also using the status interface.



210463-6

NOTE:
Address latch not required in single-port mode.

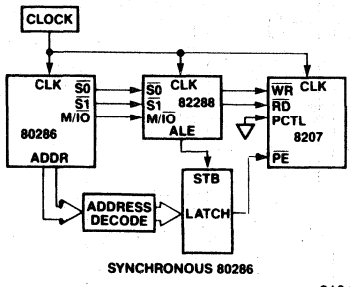
Fast-Cycle Synchronous-Status Interface



210463-7

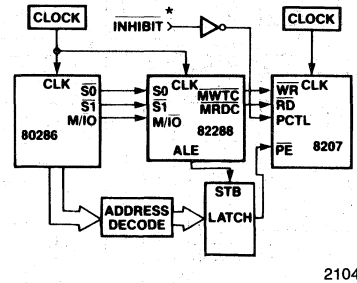
NOTE:
Address latch not required in single-port mode.

Fast-Cycle Asynchronous-Status Interface



210463-8

Fast-Cycle Synchronous-Command Interface



210463-9

*MULTI-BUS Option

Fast-Cycle Asynchronous-Command Interface

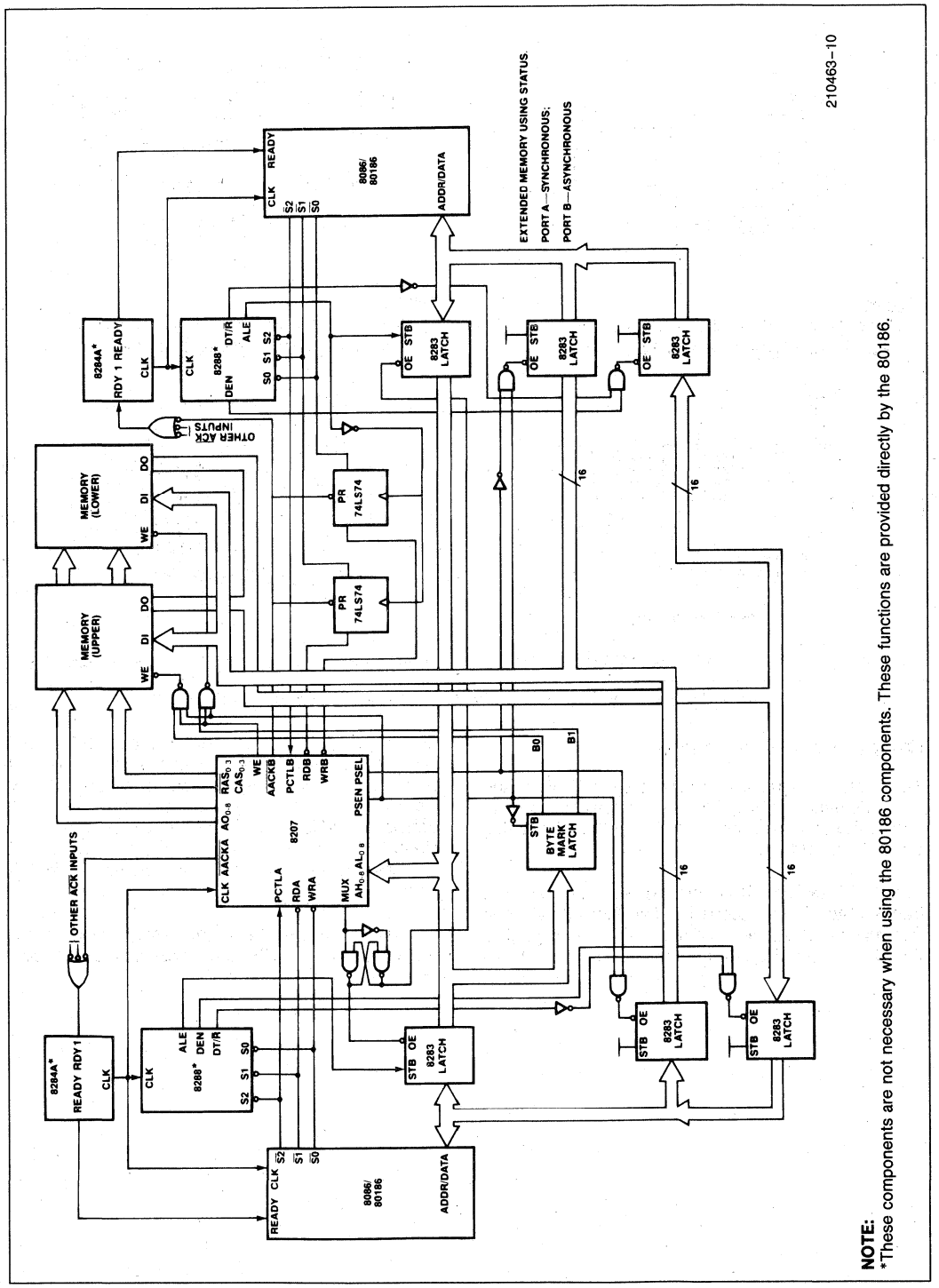
Figure 2B. Fast-Cycle (CFS = 1) Port Interfaces Supported by the 8207

Dynamic RAM Interface

The 8207 is capable of addressing 16K, 64K and 256K dynamic RAMs. Figure 4 shows the connection of the processor address bus to the 8207 using the different RAMs.

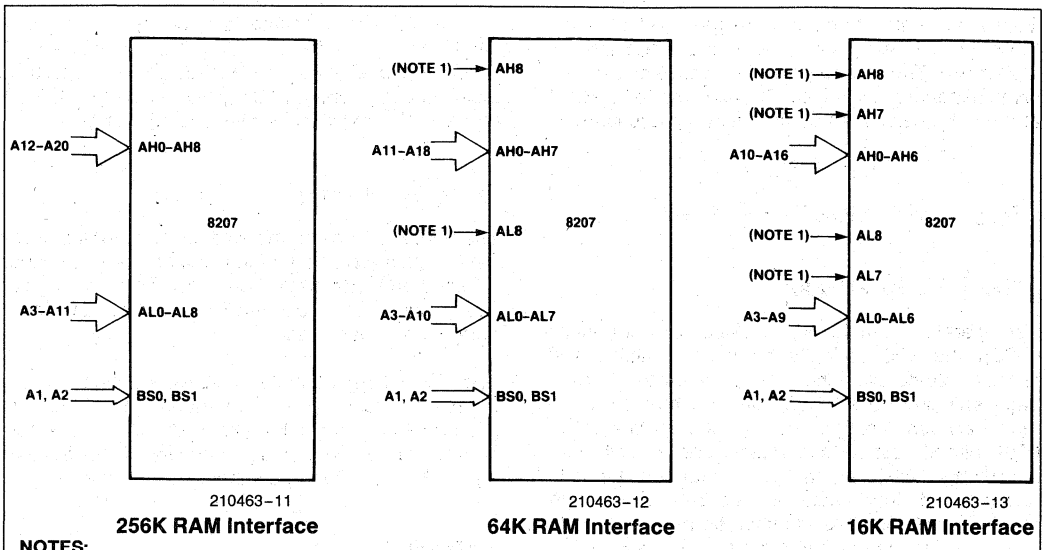
The 8207 divides memory into as many as four banks, each bank having its own Row ($\overline{RAS}$) and Column ($\overline{CAS}$) Address Strobe pair. This organization permits RAM cycle interleaving and permits error scrubbing during ECC refresh cycles. RAM cycle interleaving overlaps the start of the next RAM cycle with the RAM Precharge period of the previous cycle. Hiding the precharge period of one RAM cycle behind the data access period of the next RAM cycle optimizes memory bandwidth and is effective as long as successive RAM cycles occur in alternate banks.

Successive data access to the same bank will cause the 8207 to wait for the precharge time of the previous RAM cycle.



NOTE:
*These components are not necessary when using the 80186 components. These functions are provided directly by the 80186.

Figure 3. 8086/80186 Dual Port System



NOTES:
1. Unassigned address input pins should be strapped high or low.
2. A0 along with BHE are used to select a byte within a processor word.
3. Low order address bits are used as bank select inupts so that consecutive memory access requests are to alternate banks allowing bank interleaving of memory cycles.

Figure 4. Processor Address Interface to the 8207 Using 16K, 64K, and 256K RAMs

If not all RAM banks are occupied, the 8207 reassigns the $\overline{RAS}$ and $\overline{CAS}$ strobes to allow using wider data words without increasing the loading on the $\overline{RAS}$ and $\overline{CAS}$ drivers. Table 2 shows the bank selection decoding and the word expansion, including $\overline{RAS}$ and $\overline{CAS}$ assignments. For example, if only two RAM banks are occupied, then two $\overline{RAS}$ and two $\overline{CAS}$ strobes are activated per bank. Program bits RB1 and RB0 are not used to check the bank select inputs BS1 and BS0. The system design must protect from accesses to "illegal", non-existent banks of memory, by deactivating the PEA, PEB inputs when addressing an illegal bank.

The 8207 can interface to fast or slow RAMs. The 8207 adjusts and optimizes internal timings for either the fast or slow RAMs as programmed. (See RAM Speed Option.)

Memory Initialization

After programming, the 8207 performs eight RAM "warm-up" cycles to prepare the dynamic RAM for proper device operation. During "warm-up" some RAM parameters, such as tRAH, tASC, may not be met. This causes no harm to the dynamic RAM array. If configured for operation with error correction, the 8207 and 8206 EDCU will proceed to initialize all of memory (memory is written with zeros with corresponding check bits).

Table 2. Bank Selection Decoding and Word Expansion

Program Bits		Bank Input		RAS/CAS Pair Allocation
RB1	RB0	BS1	BS0	
0	0	0	0	RAS_{0-3}, CAS_{0-3} to Bank 0
0	0	0	1	Illegal
0	0	1	0	Illegal
0	0	1	1	Illegal
0	1	0	0	$RAS_{0,1}$, $CAS_{0,1}$ to Bank 0
0	1	0	1	$RAS_{2,3}$, $CAS_{2,3}$ to Bank 1
0	1	1	0	Illegal
0	1	1	1	Illegal
1	0	0	0	RAS_0, CAS_0 to Bank 0
1	0	0	1	RAS_1, CAS_1 to Bank 1
1	0	1	0	RAS_2, CAS_2 to Bank 2
1	0	1	1	Illegal
1	1	0	0	RAS_0, CAS_0 to Bank 0
1	1	0	1	RAS_1, CAS_1 to Bank 1
1	1	1	0	RAS_2, CAS_2 to Bank 2
1	1	1	1	RAS_3, CAS_3 to Bank 3

Because the time to initialize memory is fairly long, the 8207 may be programmed to skip initialization in ECC mode. The time required to initialize all of memory is dependent on the clock cycle time to the 8207 and can be calculated by the following equation:

$$T_{INIT} = (2^{23})\ T_{CLCL} \quad (1)$$

if $T_{CLCL} = 125$ ns then $T_{INIT} \simeq 1$ sec.

8206 ECC Interface

For operation with Error Checking and Correction (ECC), the 8207 adjusts its internal timing and changes some pin functions to optimize performance and provide a clean dual-port memory interface between the 8206 EDCU and memory. The 8207 directly supports a master-only (16-bit word plus 6 check bits) system. Under extended operation and reduced clock frequency, the 8207 will support any ECC master-slave configuration up to 80 data bits, which is the maximum set by the 8206 EDCU. (See Extend Option.)

Correctable errors detected during memory read cycles are corrected immediately and then written back into memory.

In a synchronous bus environment, ECC system performance has been optimized to enhance processor throughput, while in an asynchronous bus environment (the Multibus), ECC performance has been optimized to get valid data onto the bus as quickly as possible. Performance optimization, processor throughput or quick data access may be selected via the Transfer Acknowledge Option.

The main difference between the two ECC implementations is that, when optimized for processor throughput, RAM data is always corrected and an advanced transfer acknowledge is issued at a point when, by knowing the processor characteristics, data is guaranteed to be valid by the time the processor needs it.

When optimized for quick data access, (valid for Multibus) the 8206 is configured in the uncorrecting mode where the delay associated with error correction circuitry is transparent, and a transfer acknowledge is issued as soon as valid data is known to exist. If the $\overline{\text{ERROR}}$ flag is activated, then the transfer acknowledge is delayed until after the 8207 has instructed the 8206 to correct the data and the corrected data becomes available on the bus. Figure 5 illustrates a dual-port ECC system.

Figure 6 illustrates the interface required to drive the $\overline{\text{CRCT}}$ pin of the 8206, in the case that one port (PORT A) receives an advanced acknowledge (not Multibus-compatible), while the other port (PORT B) receives $\overline{\text{XACK}}$ (which is Multibus-compatible).

Error Scrubbing

The 8207/8206 performs error correction during refresh cycles (error scrubbing). Since the 8207 must refresh RAM, performing error scrubbing during refresh allows it to be accomplished without additional performance penalties.

Upon detection of a correctable error during refresh, the RAM refresh cycle is lengthened slightly to permit the 8206 to correct the error and for the corrected word to be rewritten into memory. Uncorrectable errors detected during scrubbing are ignored.

Refresh

The 8207 provides an internal refresh interval counter and a refresh address counter to allow the 8207 to refresh memory. The 8207 will refresh 128 rows every 2 milliseconds or 256 rows every 4 milliseconds, which allows all RAM refresh options to be supported. In addition, there exists the ability to refresh 256 row address locations every 2 milliseconds via the Refresh Period programming option.

The 8207 may be programmed for any of four different refresh options: Internal refresh only, External refresh with failsafe protection, External refresh without failsafe protection, Burst Refresh mode, or no refresh. (See Refresh Options.)

It is possible to decrease the refresh time interval by 10%, 20% or 30%. This option allows the 8207 to compensate for reduced clock frequencies. Note that an additional 5% interval shortening is built-in in all refresh interval options to compensate for clock variations and non-immediate response to the internally generated refresh request. (See Refresh Period Options.)

External Refresh Requests after RESET

External refresh requests are not recognized by the 8207 until after it is finished programming and preparing memory for access. Memory preparation includes 8 RAM cycles to prepare and ensure proper

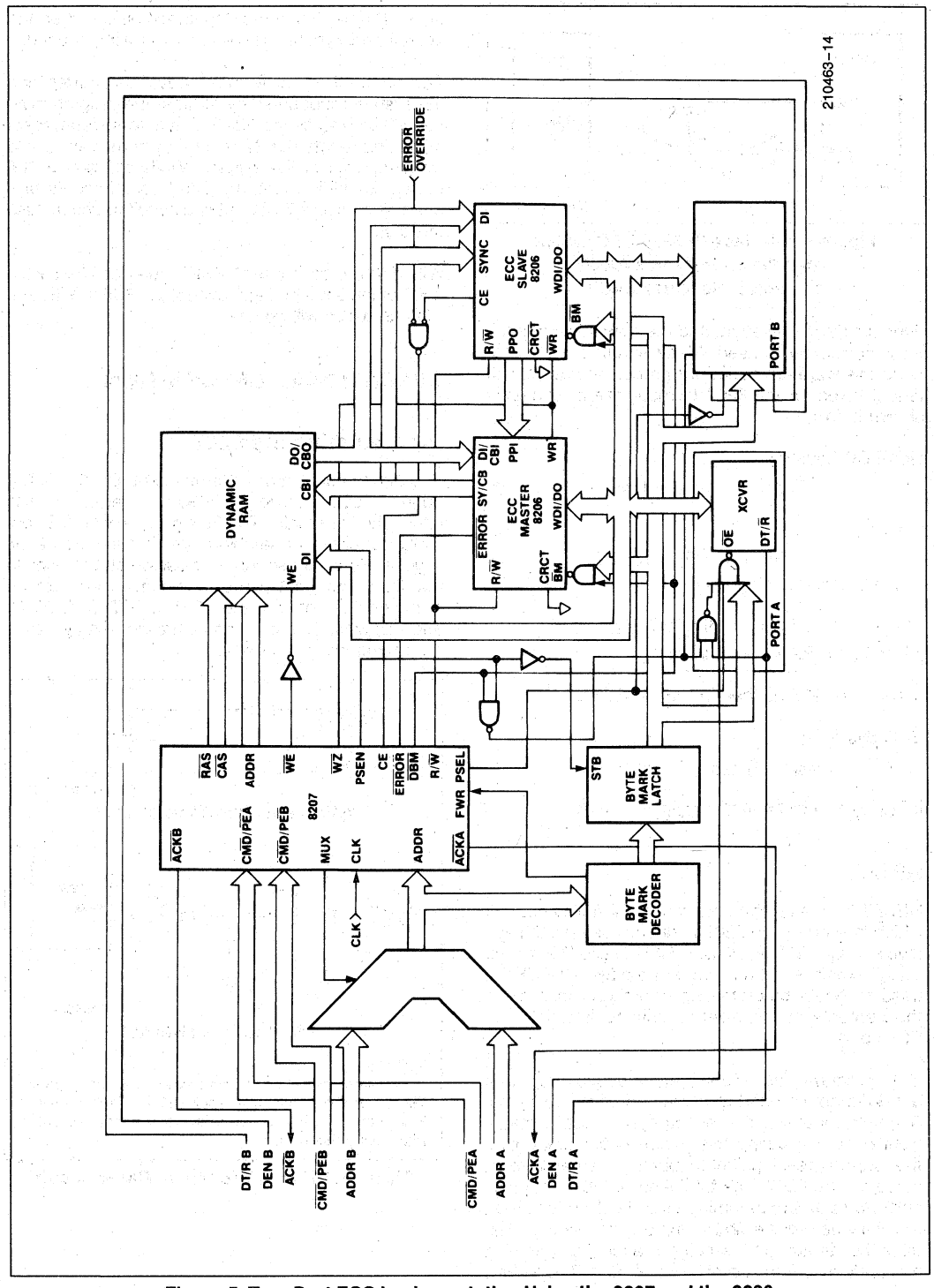


Figure 5. Two-Port ECC Implementation Using the 8207 and the 8206

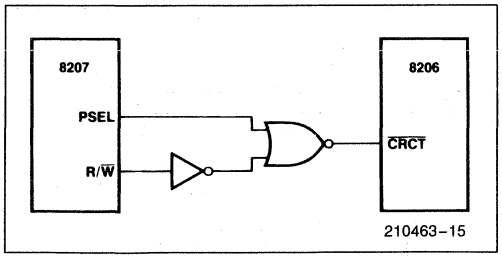


Figure 6. Interface to 8206 $\overline{\text{CRCT}}$ Input when Port A Receives $\overline{\text{AACK}}$ and Port B Receives $\overline{\text{XACK}}$

dynamic RAM operation, and memory initialization if error correction is used. Many dynamic RAMs require this warm-up period for proper operation. The time it takes for the 8207 to recognize a request is shown below.

Non-ECC Systems:

$$T_{RESP} = T_{PROG} + T_{PREP} \quad (2)$$

where:

$$T_{PROG} = (66)\ (T_{CLCL}) \quad (3)$$

which is programming time

$$T_{PREP} = (8)\ (32)\ (T_{CLCL})$$

which is the RAM warm-up time

if $T_{CLCL} = 125$ ns then $T_{RESP} \approx 41\ \mu s$

ECC Systems:

$$T_{RESP} = T_{PROG} + T_{PREP} + T_{INIT} \quad (5)$$

if $T_{CLCL} = 125$ ns then $T_{RESP} \approx 1$ sec.

RESET

RESET is an asynchronous input, the falling edge of which is used by the 8207 to directly sample to logic levels of the PCTLA, PCTLB, RFRQ, and PDI inputs. The internally synchronized falling edge of RESET is used to begin programming operations (shifting in the contents of the external shift register into the PDI input).

Until programming is complete the 8207 registers but does not respond to command or status inputs. A simple means of preventing commands or status from occurring during this period is to differentiate the system reset pulse to obtain a smaller reset pulse for the 8207. The total time of the reset pulse and the 8207 programming time must be less than the time before the first command in systems that alter the default port synchronization programming bits (default is Port A synchronous, Port B asynchronous). Differentiated reset is unnecessary when the default port synchronization programming is used.

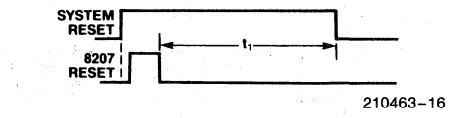
The differentiated reset pulse would be shorter than the system reset pulse by at least the programming period required by the 8207. The differentiated reset pulse first resets the 8207, and system reset would reset the rest of the system. While the rest of the system is still in reset, the 8207 completes its programming. Figure 7 illustrates a circuit to accomplish this task.

Within four clocks after RESET goes active, all the 8207 outputs will go high, except for PSEN, WE, and AO0–2, which will go low.

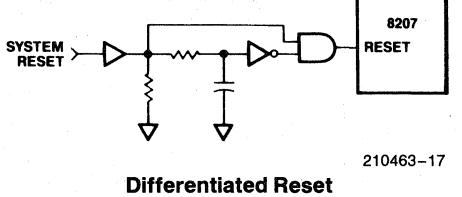
OPERATIONAL DESCRIPTION

Programming the 8207

The 8207 is programmed after reset. On the falling edge of RESET, the logic states of several input pins are latched internally. The falling edge of RESET actually performs the latching, which means that the logic levels on these inputs must be stable prior to that time. The inputs whose logic levels are latched at the end of reset are the PCTLA, PCTLB, REFRQ, and PDI pins. Figure 8 shows the necessary timing for programming the 8207.



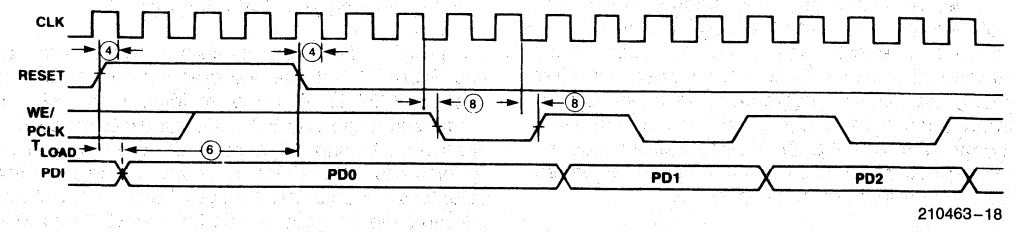
t_I Programming Time of 8207



Differentiated Reset

NOTES:
1. Required only when the port synchronization options (SA & SB) are altered from their initial default values.
2. V_{CC} must be stable before system reset is activated when using this circuit.

Figure 7. 8207 Differentiated Reset Circuit



NOTES:
TRTVCL—Reset is an asynchronous input, if reset occurs before T_1, then it is guaranteed to be recognized.
TPGVCL—Minimum PDI valid time prior to reset going low.
TCLPC—MUX/PCLK delay.
TLOAD—Asynchronous load data propagation delay.

Figure 8. Timing Illustrating External Shift Register Requirements for Programming the 8207

Status/Command Mode

The two processor ports of the 8207 are configured by the states of the PCTLA and PCTLB pins. Which interface is selected depends on the state of the individual port's PCTL pin at the end of reset. If PCTL is high at the end of the reset, the 8086 Status interface is selected; if it is low, then the Command interface is selected.

The status lines of the 80286 are similar in code and timing to the Multibus command lines, while the status code and timing of the 8086 and 8088 are identical to those of the 80186 and 80188 (ignoring the differences in clock duty cycle). Thus there exists two interface configurations, one for the 80286 status or Multibus memory commands, which is called the Command interface, and one for 8086, 8088, 80186 or 80188 status, called the 8086 Status interface. The Command interface can also directly interface to the command lines of the bus controllers for the 8086, 8088, 80186 and the 80286.

The 8086 Status interface allows direct decoding of the status of the iAPX 86, iAPX 88, iAPX 186 and the iAPX 188. Table 3 shows how the status lines are decoded. While in the Command mode the iAPX 286 status can be directly decoded. Microprocessor bus controller read or write commands or Multibus commands can also be directed to the 8207 when in Command mode.

Refresh Options

Immediately after system reset, the state of the REFRQ input pin is examined. If REFRQ is high, the 8207 provides the user with the choice between self-refresh or user-generated refresh with failsafe protection. Failsafe protection guarantees that if the

Table 3A. Status Coding of 8086, 80186 and 80286

Status Code			Function	
$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	8086/80186	80286
0	0	0	Interrupt	Interrupt
0	0	1	I/O Read	I/O Read
0	1	0	I/O Write	I/O Write
0	1	1	Halt	Idle
1	0	0	Instruction Fetch	Halt
1	0	1	Memory Read	Memory Read
1	1	0	Memory Write	Memory Write
1	1	1	Idle	Idle

Table 3B. 8207 Response

8207 Command			Function	
PCTL	$\overline{RD}$	$\overline{WR}$	8086/80186 Status Interface	80286 Status or Command Interface
0	0	0	Ignore	Ignore*
0	0	1	Ignore	Read
0	1	0	Ignore	Write
0	1	1	Ignore	Ignore*
1	0	0	Read	Ignore
1	0	1	Read	Inhibit
1	1	0	Write	Inhibit
1	1	1	Ignore	Ignore

*Illegal with CFS = 0

user does not come back with another refresh request before the internal refresh interval counter times out, a refresh request will be automatically generated. If the REFRQ pin is low immediately after a reset, the 8207 is programmed in a non-failsafe refresh mode. In this mode the refresh cycle is initiated only upon receipt of an external refresh request. The user has the choice of a single external refresh cycle, burst refresh or no refresh.

Internal Refresh Only

For the 8207 to generate internal refresh requests, it is necessary only to strap the REFRQ input pin high.

External Refresh with Failsafe

To allow user-generated refresh requests with failsafe protection, it is necessary to hold the REFRQ input high until after reset. Thereafter, a low-to-high transition on this input causes a refresh request to be generated and the internal refresh interval counter to be reset. A high-to-low transition has no effect on the 8207. A refresh request is not recognized until a previous request has been serviced.

External Refresh without Failsafe

To generate single external refresh requests without failsafe protection, it is necessary to hold REFRQ low until after reset. Thereafter, bringing REFRQ high for one clock period causes refresh request to be generated. A refresh request is not recognized unitl a previous request has been serviced.

Burst Refresh

Burst refresh is implemented through the same procedure as a single external refresh without failsafe (i.e., REFRQ is kept low until after reset). Thereafter, bringing REFRQ high for at least two clock periods causes a burst of up to 128 row address locations to be refreshed.

The ECC-configured systems, 128 locations are scrubbed. Any refresh request is not recognized until a previous request has been serviced (i.e., burst completed).

No Refresh

It is necessary to hold REFRQ low until after reset. This is the same as programming External Refresh without Failsafe. No refresh is accomplished by keeping REFRQ low.

Option Program Data Word

The program data word consists of 16 program data bits, PD0–PD15. If the first program data bit shifted into the 8207 (PD0) is set to logic 1, the 8207 is configured to support ECC. If it is logic 0, the 8207 is configured to support a non-ECC system. The remaining bits, PD1–PD15, may then be programmed to optimize a selected configuration. Figures 9 and 10 show the Program words for non-ECC and ECC operation.

Using an External Shift Register

The 8207 may be configured to use an external shift register with asynchronous load capability such as a 74LS165. The reset pulse serves to parallel load the shift register and the 8207 supplies the clocking signal to shift the data in. Figure 11 shows a sample circuit diagram of an external shift register circuit.

Serial data is shifted into the 8207 via the PDI pin (57), and clock is provided by the MUX/PCLK pin (12), which generates a total of 16 clock pulses. After programming is complete, data appearing at the input of the PDI pin is ignored. MUX/PCLK is a dual-function pin. During programming, it serves to clock the external shift register, and after programming is completed, it reverts to a MUX conrol pin. As the pin changes state to select different port addresses, it continues to clock the shift register. This does not present a problem because data at the PDI pin is ignored after programming. Figure 8 illustrates the timing requirements of the shift register circuitry.

ECC Mode (ECC Program Bit)

The state of PDI (Program Data In) pin at reset determines whether the system is an ECC or non-ECC configuration. It is used internally by the 8207 to begin configuring timing circuits, even before programming is completely finished. The 8207 then begins programming the rest of the options.

Default Programming Options

After reset, the 8207 serially shifts in a program data word via the PDI pin. This pin may be strapped either high or low, or connected to an external shift register. Strapping PDI high causes the 8207 to default to a particular system configuration with error correction, and strapping it low causes the 8207 to default to a particular system configuration without error correction. Table 4 shows the default configurations.

PD15								PD8 PD7							PD0
0	0	TM 1	$\overline{PPR}$	$\overline{FFS}$	EXT	$\overline{PLS}$	CI0	CI1	$\overline{RB1}$	$\overline{RB0}$	$\overline{RFS}$	$\overline{CFS}$	SB	$\overline{SA}$	0

Program Data Bit	Name	Polarity/Function
PD0	ECC	ECC = 0 For Non-ECC Mode
PD1	$\overline{SA}$	$\overline{SA}$ = 0 Port A is Synchronous $\overline{SA}$ = 1 Port A is Asnychronous
PD2	SB	SB = 0 Port B is Asynchronous SB = 1 Port B is Synchronous
PD3	$\overline{CFS}$	$\overline{CFS}$ = 0 Fast-Cycle iAPX 286 Mode $\overline{CFS}$ = 1 Slow-Cycle iAPX 86 Mode
PD4	$\overline{RFS}$	$\overline{RFS}$ = 0 Fast RAM $\overline{RFS}$ = 1 Slow RAM
PD5 PD6	$\overline{RB0}$ $\overline{RB1}$	RAM Bank Occupancy See Table 2
PD7 PD8	CI1 CI0	Count Interval Bit 1; see Table 6 Count Interval Bit 0; see Table 6
PD9	$\overline{PLS}$	$\overline{PLS}$ = 0 Long Refresh Period $\overline{PLS}$ = 1 Short Refresh Period
PD10	EXT	EXT = 0 Not Extended EXT = 1 Extended
PD11	$\overline{FFS}$	$\overline{FFS}$ = 0 Fast CPU Frequency $\overline{FFS}$ = 1 Slow CPU Frequency
PD12	$\overline{PPR}$	$\overline{PPR}$ = 0 Most Recently Used Port Priority $\overline{PPR}$ = 1 Port A Preferred Priority
PD13	TM1	TM1 = 0 Test Mode 1 Off TM1 = 1 Test Mode 1 Enabled
PD14	0	Reserved, Must be Zero
PD15	0	Reserved, Must be Zero

Figure 9. Non-ECC Mode Program Data Word

PD15							PD8 PD7							PD0
$\overline{TM2}$	RB1	RB0	PPR	FFS	$\overline{EXT}$	PLS	$\overline{CI0}$	$\overline{CI1}$	XB	$\overline{XA}$	RFS	CFS	$\overline{SB}$	SA	1

Program Data Bit	Name	Polarity/Function
PD0	ECC	ECC = 1 ECC Mode
PD1	SA	SA = 0 Port A Asynchronous SA = 1 Port A Synchronous
PD2	$\overline{SB}$	$\overline{SB}$ = 0 Port B Synchronous $\overline{SB}$ = 1 Port B Asynchronous
PD3	CFS	CFS = 0 Slow-Cycle iAPX 86 Mode CFS = 1 Fast-Cycle iAPX 286 Mode
PD4	RFS	RFS = 0 Slow RAM RFS = 1 Fast RAM
PD5	$\overline{XA}$	$\overline{XA}$ = 0 MULTIBUS-Compatible ACKA $\overline{XA}$ = 1 Advanced ACKA Not Multibus-Compatible
PD6	XB	Advanced ACKB Not Multibus-Compatible XB = 1 Multibus-Compatible ACKB
PD7 PD8	$\overline{CI1}$ $\overline{CI0}$	Count Interval Bit 1; see Table 6 Count Interval Bit 0; see Table 6
PD9	PLS	PLS = 0 Short Refresh Period PLS = 1 Long Refresh Period
PD10	$\overline{EXT}$	$\overline{EXT}$ = 0 Master and Slave EDCU $\overline{EXT}$ = 1 Master EDCU Only
PD11	FFS	FFS = 0 Slow CPU Frequency FFS = 1 Fast CPU Frequency
PD12	PPR	PPR = 0 Port A Preferred Priority PPR = 1 Most Recently Used Port Priority
PD13 PD14	RB0 RB1	RAM Bank Occupancy See Table 2
PD15	$\overline{TM2}$	$\overline{TM2}$ = 0 Test Mode 2 Enabled $\overline{TM1}$ = 1 Test Mode 2 Off

Figure 10. ECC Mode Program Data Word

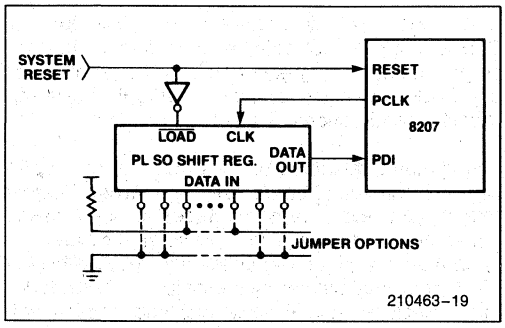


Figure 11. External Shift Register Interface

Table 4A. Default Non-ECC Programming, PDI Pin (57) Tied to Ground

Port A is Synchronous ($\overline{EAACKA}$ and $\overline{XACKA}$)
Port B is Asynchronous ($\overline{LAACKB}$ and $\overline{XACKB}$)
Fast-Cycle Processor Interface (iAPX 286)
Fast RAM
Refresh Interval uses 236 Clocks
128 Row Refresh in 2 ms; 256 Row Refresh in 4 ms
Fast Processor Clock Frequency (16 MHz)
"Most Recently Used" Priority Scheme
4 RAM banks occupied

Table 4B. Default ECC Programming, PDI Pin (57) Tied to V_{CC}

Port A is Synchronous
Port B is Asynchronous
Fast-Cycle Processor Interface (iAPX 286)
Fast RAM
Port A has $\overline{EAACKA}$ strobe (non-multibus)
Port B has $\overline{XACKB}$ strobe (multibus)
Refresh interval uses 236 clocks
128 Row refresh in 2 ms; 256 Row refresh in 4 ms
Master EDCU only (16-bit system)
Fast Processor Clock Frequency (16 MHz)
"Most Recently Used" Priority Scheme
4 RAM banks ocuppied

If further system flexibility is needed, one or two external shift registers can be used to tailor the 8207 to its operating environment.

Synchronous/Asynchronous Mode (SA and SB Program Bits)

Each port of the 8207 may be independently configured to accept synchronous or asynchronous port commands ($\overline{RD}$, $\overline{WR}$, PCTL) and Port Enable ($\overline{PE}$) via the program bits SA and SB. The state of the SA and SB programming bits determine whether their associated ports are synchronous or asynchronous.

While a port may be configured with either the Status or Command interface in the synchronous mode, certain restrictions exist in the asynchronous mode. An asynchronous Command interface using the control lines of the Multibus is supported, and an asynchronous 8086 interface using the control lines of the 8086 is supported, with the use of TTL gates as illustrated in Figure 2. In the 8086 case, the TTL gates are needed to guarantee that status does not appear at the 8207's inputs too much before address, so that a cycle would start before address was valid.

Microprocessor Clock Frequency Option (CFS and FFS Program Bits)

The 8207 can be programmed to interface with slow-cycle microprocessors like the 8086, 8088, 81088 and 80186 or fast-cycle microprocessors like the 80286. The CFS bit configures the microprocessor interface to accept slow or fast cycle signals from either microprocessor group.

The FFS bit is used to select the speed of the microprocessor clock. Table 5 shows the various microprocessor clock frequency options that can be programmed.

Table 5. Microprocessor Clock Frequency Options

Program Bits		Processor	Clock Frequency
CFS	FFS		
0	0	iAPX 86, 88, 186, 188	≤ 6 MHz
0	1	iAPX 86, 88, 186, 188	> 6 MHz
1	0	iAPX 286	≤12 MHz
1	1	iAPX 286	>12 MHz

The external clock frequency must be programmed so that the failsafe refresh repetition circuitry can adjust its internal timing accordingly to produce a refresh request as programmed.

RAM Speed Option (RFS Program Bit)

The RAM Speed programming option determines whether RAM timing will be optimized for a fast or slow RAM.

Refresh Period Options (CI0, CI1 and PLS Program Bits)

The 8207 refreshes with either 128 rows every 2 milliseconds or 256 rows every 4 milliseconds. This translates to one refresh cycle being executed approximately once every 15.6 microseconds. This rate can be changed to 256 rows every 2 milliseconds or a refresh approximately once every 7.8 microseconds via the Period Long/Short, program bit PLS, programming option. The 7.8 microsecond refresh request rate is intended for those RAMs, 64K and above, which may require a faster refresh rate.

In addition to PLS program option, two other programming bits for refresh exist: Count Interval 0 (CI0) and Count Interval 1 (CI1). These two programming bits allow the rate at which refresh requests are generated to be increased in order to permit refresh requests to be generated close to the same 15.6 or 7.8 microsecond period when the 8207 is operating at reduced frequencies. The interval between refreshes is decreased by 0%, 10%, 20%, or 30% as a function of how the count interval bits are programmed. A 5% guardband is built-in to allow for any clock frequency variations. Table 6 shows the refresh period options available.

The numbers tabulated under Count Interval represent the number of clock periods between internal refresh requests. The percentages in parentheses represent the decrease in the interval between refresh requests. Note that all intervals have a built-in 5% (approximately) safety factor to compensate for minor clock frequency deviations and non-immediate response to internal refresh requests.

Extend Option (EXT Program Bit)

The Extend option lengthens the memory cycle to allow longer access time which may be required by the system. Extend alters the RAM timing to compensate for increased loading on the Row and Column Address Strobes, and in the multiplexed Address Out lines.

Port Priority Option and Arbitration (PPR Program Bit)

The 8207 has to internally arbitrate among three ports: Port A, Port B and Port C—the refresh port. Port C is an internal port dedicated to servicing refresh requests, whether they are generated internally by the refresh interval counter, or externally by the user. Two arbitration approaches are available via

Table 6. Refresh Count Interval Table

Ref. Period (μs)	CFS	PLS	FFS	Count Interval CI1, CI0 (8207 Clock Periods)			
				00 (0%)	01 (10%)	10 (20%)	11 (30%)
15.6	1	1	1	236	212	188	164
7.8	1	0	1	118	106	94	82
15.6	1	1	0	148	132	116	100
7.8	1	0	0	74	66	58	50
15.6	0	1	1	118	106	94	82
7.8	0	0	1	59	53	47	41
15.6	0	1	0	74	66	58	50
7.8	0	0	0	37	33	29	25

NOTE:
Refresh period = clock period × refresh count interval.

the Port Priority programming option, program bit PPR. PPR determines whether the most recently used port will remain selected (PPR = 1) or whether Port A will be favored or preferred over Port B (PPR = 0).

A port is selected if the arbiter has given the selected port direct access to the timing generators. The front-end logic, which includes the arbiter, is designed to operate in parallel with the selected port. Thus a request on the selected port is serviced immediately. In contrast, an unselected port only has access to the timing generators through the front-end logic. Before a RAM cycle can start for an unselected port, that port must first become selected (i.e., the MUX output now gates that port's address into the 8207 in the case of Port A or B). Also, in order to allow its address to stabilize, a newly selected port's first RAM cycle is started by the front-end logic. Therefore, the selected port has direct access to the timing generators. What all this means is that a request on a selected port is started immediately, while a request on an unselected port is started two to three clock periods after the request, assuming that the other two ports are idle. Under normal operating conditions, this arbitration time is hidden behind the RAM cycle of the selected port so that as soon as the present cycle is over a new cycle is started. Table 7 lists the arbitration rules for both options.

Port LOCK Function

The LOCK function provides each port with the ability to obtain uninterrupted access to a critical region of memory and, thereby, to guarantee that the opposite port cannot "sneak in" and read from or write to the critical region prematurely.

Only one LOCK pin is present and is multiplexed between the two ports as follows: when MUX is high, the 8207 treats the LOCK input as originating at PORT A, while when MUX is low, the 8207 treats LOCK as originating at PORT B. When the 8207 recognizes a LOCK, the MUX output will remain pointed to the locking port until LOCK is deactivated. Refresh is not affected by LOCK and can occur during a locked memory cycle.

2

Table 7. The Arbitration Rules for the Most Recently Used Port Priority and for Port A Priority Options Are As Follows:

1.	If only one port requests service, then that port—if not already selected—becomes selected.
2a.	When no service requests are pending, the last selected processor port (Port A or B) will remain selected. (Most Recently Used Port Priority Option.)
2b.	When no service requests are pending, Port A is selected whether it requests service or not. (Port A Priority Option.)
3.	During reset initialization only Port C, the refresh port, is selected.
4.	If no processor requests are pending after reset initialization, Port A will be selected.
5b.	If Ports A and B simultaneously(*) request service while Port C is selected, then the next port to be selected is Port A. (Port A Priority Option.)
6.	If a port simultaneously requests service with the currently selected port, service is granted to the selected port.
7.	The MUX output remains in its last state whenever Port C is selected.
8.	If Port C and either Port A or Port B (or both) simultaneously request service, then service is granted to the requester whose port is already selected. If the selected port is not requesting service, then service is granted to Port C.
9.	If during the servicing of one port, the other port requests service before or simultaneously with the refresh port, the refresh port is selected. A new port is not selected before the presently selected port is deactivated.
10.	Activating LOCK will mask off service requests from Port B if the MUX output is high, or from Port A if the MUX output is low.

NOTE:
*By "simultaneous" it is meant that two or more requests are valid at the clock edge at which the internal arbiter samples them.

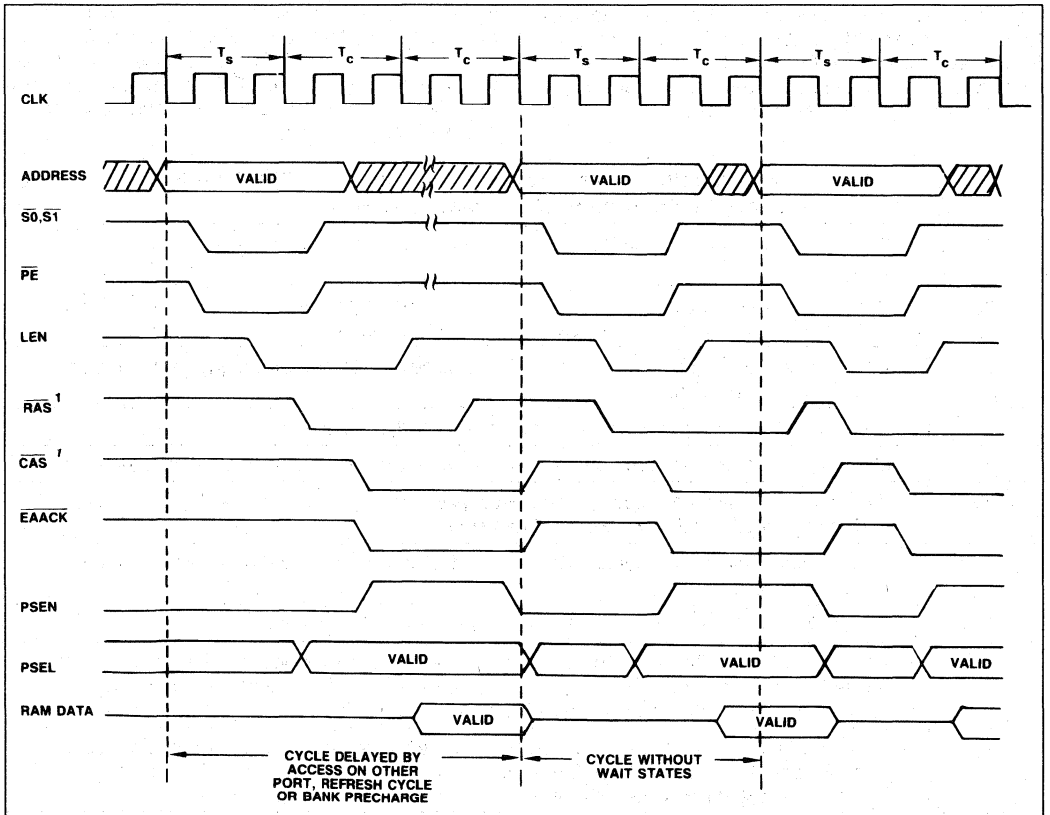
Dual-Port Considerations

For both ports to be operated synchronously, several conditions must be met. The processors must be the same type (Fast or Slow Cycle) as defined by Table 8 and they must have synchronized clocks. Also when processor types are mixed, even though the clocks may be in phase, one frequency may be twice that of the other. So to run both ports synchronous using the status interface, the processors must have related timings (both phase and frequency). If these conditions cannot be met, then one port must run synchronous and the other asynchronous.

Figure 3 illustrates an example of dual-port operation using the processors in the slow cycle group. Note the use of cross-coupled NAND gates at the MUX output for minimizing contention between the two latches, and the use of flip flops on the status lines of the asynchronous processor for delaying the status and thereby guaranteeing RAS will not be issued, even in the worst case, until address is valid.

Processor Timing

In order to run without wait states, $\overline{AACK}$ must be used and connected to the $\overline{SRDY}$ input of the appropriate bus controller. $\overline{AACK}$ is issued relative to a point within the RAM cycle and has no fixed relationship to the processor's request. The timing is such, however, that the processor will run without wait states, barring refresh cycles, bank precharge, and RAM accesses from the other port. In non-ECC fast cycle, fast RAM, non-extended configurations (80286), $\overline{AACK}$ is issued on the next falling edge of



NOTE:
1. The RAS and CAS shown in figure are different banks being accessed.

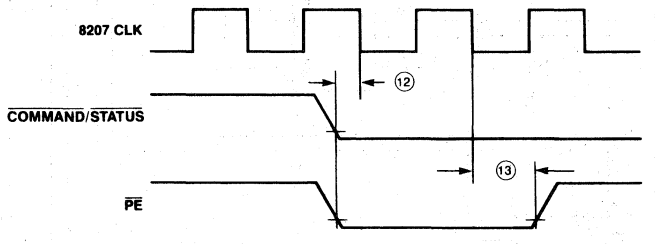
Figure 14. iAPX 286/8207 Synchronous-Status Timing Programmed in non-ECC Mode, C0 Configuration (Read Cycle)

the clock after the edge that issues RAS. In non-ECC, slow cycle, non-extended, or extended with fast RAM cycle configurations (8086, 80188, 80186), $\overline{AACK}$ is issued on the same clock cycle that issues RAS. Figure 14 illustrates the timing relationship between $\overline{AACK}$, the RAM cycle, and the processor cycle for several different situations.

Port Enable ($\overline{PE}$) setup time requirements depend on whether the associated port is configured for synchronous or asynchronous fast or slow cycle operation. In a synchronous fast cycle configuration, $\overline{PE}$ is required to be setup to the same clock edge as the status or commands. If $\overline{PE}$ is true (low), a RAM cycle is started; if not, the cycle is aborted. The memory cycle will only begin when both valid signals ($\overline{PE}$ and $\overline{RD}$ or WR) are recognized at a particular clock edge. In asynchronous operation. $\overline{PE}$ is required to be setup to the same clock edge as the internally synchronized status or commands. Externally, this allows the internal synchronization delay to be added to the status (or command)-to-$\overline{PE}$ delay time, thus allowing for more external decode time that is available in synchronous operation.

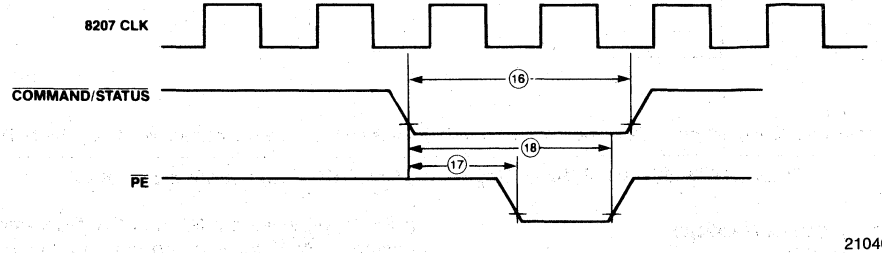
The minimum synchronization delay is the additional amount that $\overline{PE}$ must be held valid. If $\overline{PE}$ is not held valid for the maximum synchronization delay time, it is possible that $\overline{PE}$ will go invalid prior to the status or command being synchronized. In such a case the 8207 aborts the cycle. If a memory cycle intended for the 8207 is aborted, then no acknowledge ($\overline{AACK}$ or $\overline{XACK}$) is issued and the processor locks up in endless wait states. Figure 15 illustrates the status (command) timing requirements for synchronous and asynchronous systems. Figures 16 and 17 show a more detailed hook-up of the 8207 to the 8086 and the 80286, respectively.

2



210463-21

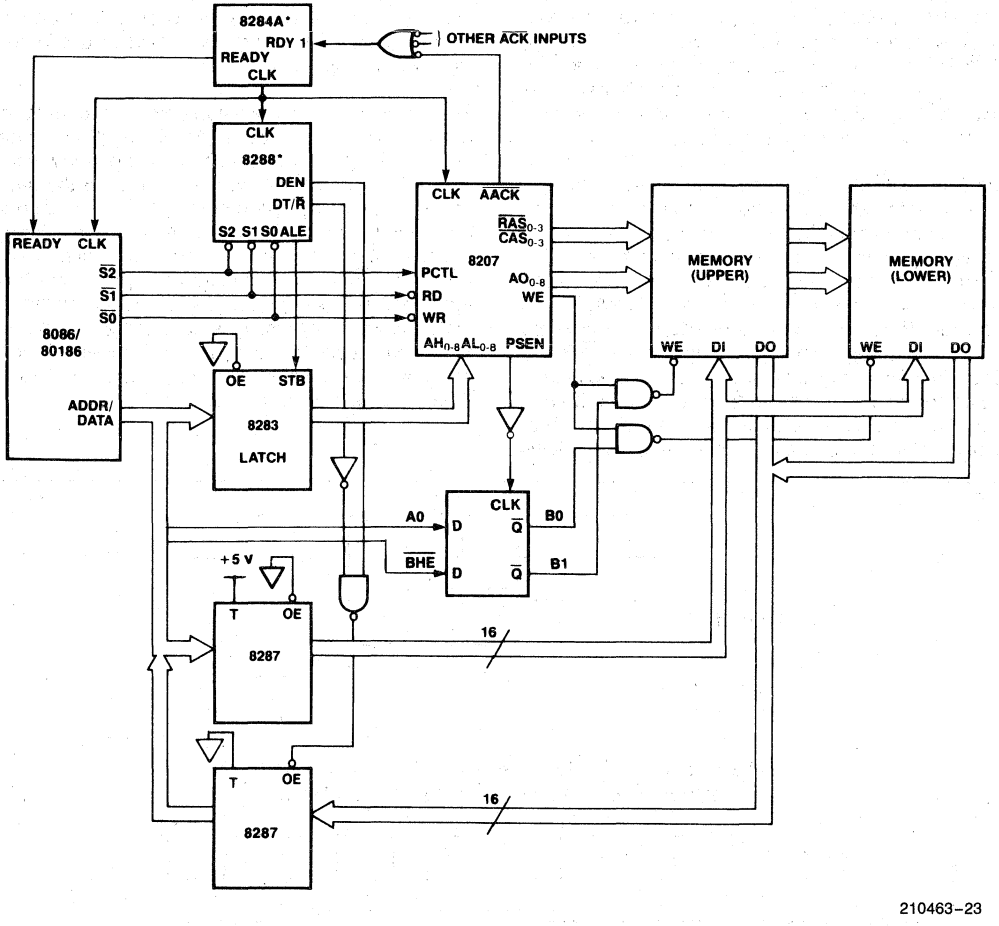
(A) $\overline{PE}$ Set-Up and Hold Time Requirements for Fast Cycle, Synchronous Operation (80286 CMD/Status)



210463-22

(B) $\overline{PE}$ Timing Requirements for Fast or Slow Cycle Asynchronous Operation

Figure 15



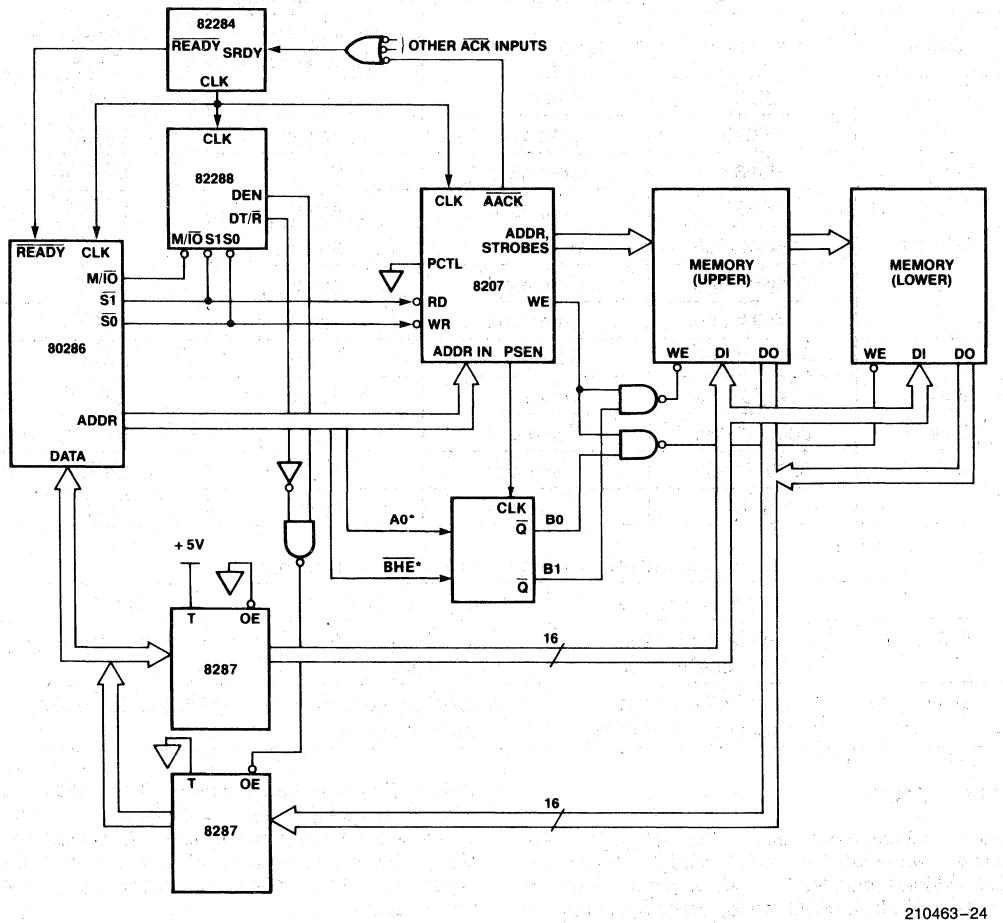
NOTE:
*These components are not necessary when using the 80186. These functions are provided directly by the 80186.

Figure 16. 8086/80186, 8207 Single Port Non-ECC Synchronous Systems

Memory Acknowledge (AACK, XACK)

In system configurations without error correction, two memory acknowledge signals per port are supplied by the 8207. They are the Advanced Acknowledge strobe ($\overline{AACK}$) and the Transfer Acknowledge strobe ($\overline{XACK}$). The CFS programming bit determines for which processor $\overline{AACKA}$ and $\overline{AACKB}$ are optimized, either 80286 (CFS = 1) or 8086/186 (CFS = 0), while the SA and SB programming bits optimize $\overline{AACK}$ for synchronous operation ("early" $\overline{AACK}$) or asynchronous operation ("late" $\overline{AACK}$).

Both the early and late $\overline{AACK}$ strobes are three clocks long for CFS = 1 and two clocks long for CFS = 0. The $\overline{XACK}$ strobe is asserted when data is valid (for reads) or when data may be removed (for writes) and meets the Multibus requirements. $\overline{XACK}$ is removed asynchronously by the command going



NOTE:
While the 8207 does not need the input addresses latched, A0, $\overline{BHE}$ must come from the latched address bus.

Figure 17. 80286 Hook-Up to 8207 Non-ECC Synchronous System-Single Port

inactive. Since in asynchronous operation the 8207 removes read data before late $\overline{AACK}$ or $\overline{XACK}$ is recognized by the CPU, the user must provide for data latching in the system until the CPU reads the data. In synchronous operation, data latching is unnecessary since the 8207 will not remove data until the CPU has read it.

In ECC-based systems there is one memory acknowledge ($\overline{XACK}$ or $\overline{AACK}$) per port and a programming bit associated with each acknowledge. If the X programming bit is active, the strobe is configured as $\overline{XACK}$, while if the bit is inactive, the strobe is configured as $\overline{AACK}$. As in non-ECC, the SA and SB programming bits determine whether the $\overline{AACK}$ strobe is early or late ($\overline{EAACK}$ or $\overline{LAACK}$).

Data will always be valid a fixed time after the occurrence of the advanced acknowledge. Table 9 summarizes the various transfer acknowledge options.

Table 8. Processor Interface/Acknowledge Summary

Cycle	Processor	Request Type	Sync/Async Interface	Acknowledge Type
Fast Cycle CFS = 1	80286	Status	Sync	$\overline{EAACK}$
	80286	Status	Async	$\overline{LAACK}$
	80286	Command	Sync	$\overline{EAACK}$
	80286	Command	Async	$\overline{LAACK}$
	8086/80186	Status	Async	$\overline{LAACK}$
	8086/80186	Command	Async	$\overline{LAACK}$
	Multibus	Command	Async	$\overline{XACK}$
Slow Cycle CFS = 0	8086/80186	Status	Sync	$\overline{EAACK}$
	8086/80186	Status	Async	$\overline{LAACK}$
	8086/80186	Command	Sync	$\overline{EAACK}$
	8086/80186	Command	Async	$\overline{LAACK}$
	Multibus	Command	Async	$\overline{XACK}$

Table 9. Memory Acknowledge Option Summary

	Synchronous	Asynchronous	XACK
Fast Cycle	AACK Optimized for Local 80286	AACK Optimized for Remote 80286	Multibus Compatible
Slow Cycle	AACK Optimized for Local 8086/186	AACK Optimized for Remote 8086/186	Multibus Compatible

Test Modes

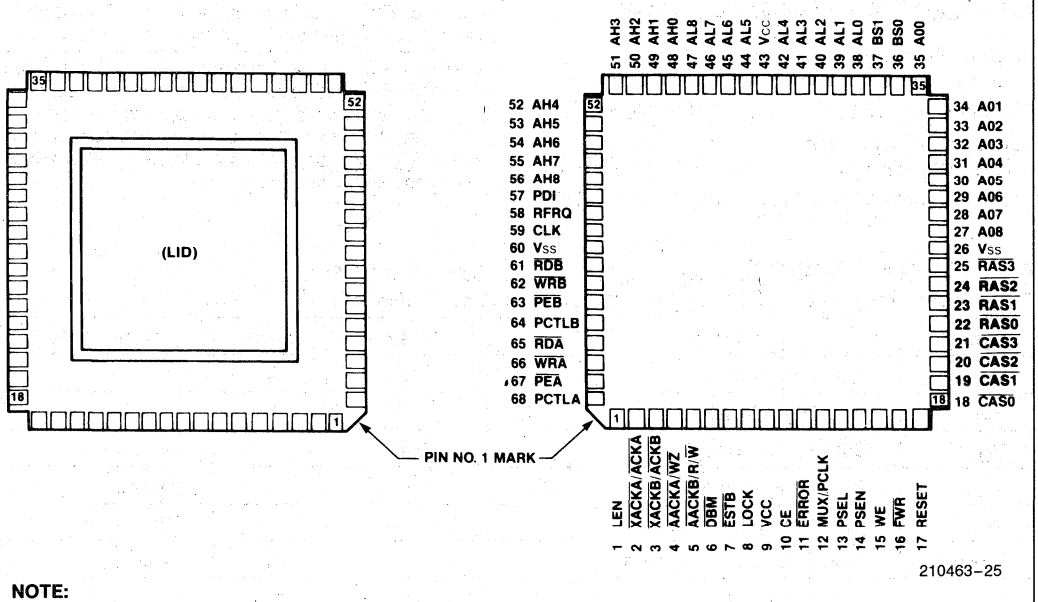
Two special test modes exist in the 8207 to facilitate testing. Test Mode 1 (non-ECC mode) splits the refresh address counter into two separate counters and Test Mode 2 (ECC mode) presets the refresh address counter to a value slightly less than rollover.

Test Mode 1 splits the address counter into two, and increments both counters simultaneously with each refresh address update. By generating external refresh requests, the tester is able to check for proper operation of both counters. Once proper individual counter operation has been established, the 8207 must be returned to normal mode and a second test performed to check that the carry from the first counter increments the second counter. The outputs of the counters are presented on the address out bus with the same timing as the row and column addresses of a normal scrubbing operation. During Test Mode 1, memory initialization is inhibited, since the 8207, be definition, is in non-ECC mode.

Test Mode 2 sets the internal refresh counter to a value slightly less than rollover. During functional testing other than that covered in Test Mode 1, the 8207 will normally be set in Test Mode 2. Test Mode 2 eliminates memory initialization in ECC mode. This allows quick examination of the circuitry which brings the 8207 out of memory initialization and into normal operation.

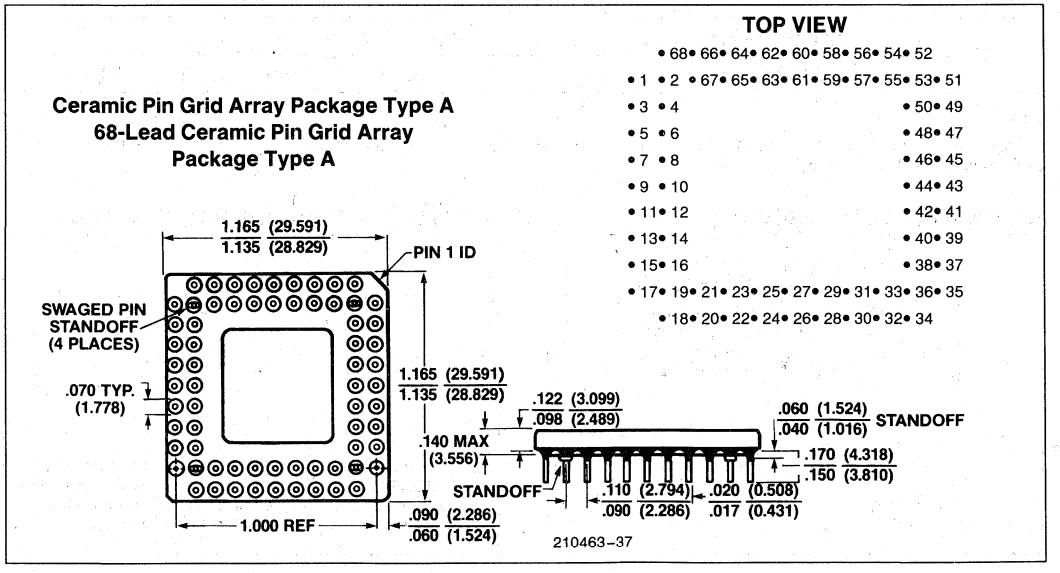
General System Considerations

The RAS_{0-3}, CAS_{0-3}, AO_{0-8}, output buffers were designed to directly drive the heavy capacitive loads associated with dynamic RAM arrays. To keep the RAM driver outputs from ringing excessively in the system environment and causing noise in other output pins it is necessary to match the output impedance of the RAM output buffers with the RAM array by using series resistors and to add series resistors to other control outputs for noise reduction if necessary. Each application may have different impedance characteristics and may require different series resistance values. The series resistance values should be determined for each application. In non-ECC systems unused ECC input pins should be tied high or low to improve noise immunity.



NOTE:
LCC is mounted lid-down into socket.

Figure 19. 8207 Pinout Diagram



8207 Pin Grid Array (PGA) Pin-Out

Packaging

The 8207 is packaged in a 68 lead JEDEC Type A Leadless Chip Carrier (LCC) and in Pin Grid Array (PGA), both in Ceramic. The package designations are R and A respectively.

eg: R 8207-8 LCC, 8 MHz DRAM Controller
eg: A 8207-16 PGA, 16 MHz DRAM Controller

NOTE:
The pin-out of the PGA is the same as the socketed pinout of the LCC.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature
Under Bias ..... −0°C to +70°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground ..... −0.5V to +7V
Power Dissipation (Note 2) 2.5W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

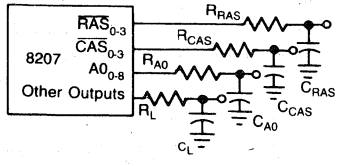
**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS V_{CC} = 5.0V ±10% for 8207-10, 8207-8; T_A = 0°C to 70°C; V_{SS} = GND; V_{CC} = 5.0V ±5% for 8207-16 (Note 2)

Symbol	Parameter	Min	Max	Units	Comments
V_{IL}	Input Low Voltage	−0.5	+0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} +0.5	V	
V_{OL}	Output Low Voltage		0.45	V	(Note 1)
V_{OH}	Output High Voltage	2.4		V	(Note 1)
V_{ROL}	RAM Output Low Voltage		0.45	V	(Note 1)
V_{ROH}	RAM Output High Voltage	2.6		V	(Note 1)
I_{CC}	Supply Current		455	mA	T_A = 0°C
I_{LI}	Input Leakage Current		±10	μA	0V ≤ V_{IN} ≤ V_{CC}
V_{CL}	Clock Input Low Voltage	−0.5	±0.6	V	
V_{CH}	Clock Input High Voltage	3.8	V_{CC} + 0.5	V	
C_{IN}	Input Capacitance		20	pF	fc = 1 MHz(2)

NOTE:
1. I_{OL} = 5 mA and I_{OH} = −0.2 mA (Typically I_{OL} = 10 mA and I_{OH} = −0.88 mA). WE: I_{OL} = 8 mA.
2. Sampled, not 100% tested.

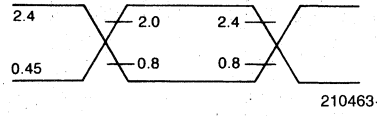
A.C. TESTING LOAD CIRCUIT(2)



210463-26

R_{RAS} = 39Ω
R_{CAS} = 39Ω
R_{A0} = 22Ω
R_L = 39Ω
C_{RAS} = 150 pF
C_{CAS} = 150 pF
C_{A0} = 380 pF
C_L = 100 pF

A.C. TESTING INPUT, OUTPUT WAVEFORM



210463-27

A.C. Testing inputs (except clock) are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0" (clock is driven at 4.0V and 0.45V for Logic "1" and "0" respectively). Timing measurements are made at 2.0V, 2.4V for Logic "1" and 0.8V for Logic "0".

A.C. CHARACTERISTICS

V_{CC} = 5V ±10% for 8207-8; T_A = 0°C to 70°C; V_{CC} = +5V ±5% for 8207-16

Measurements made with respect to RAS_{0-3}, CAS_{0-3}, AO_{0-8}, are a +2.4V and 0.8V. All other pins are measured at 2.0V and 0.8V. All times are ns unless otherwise indicated. Testing done with specified test load.

Ref	Symbol	Parameter	8207-16, -8		8207-10		Units	Notes
			Min	Max	Min	Max		
CLOCK AND PROGRAMMING								
—	tF	Clock Fall Time		10		10	ns	3
—	tR	Clock Rise Time		10		10	ns	3
1	TCLCL	Clock Period 8207-16	62.5	200			ns	1
		8207-10			100	250	ns	2
		8207-8	125	500			ns	2
2	TCL	Clock Low Time 8207-16	15	180			ns	1
		8207-10			TCLCL/2−12		ns	2
		8207-8	TCLCL/2−12				ns	2
3	TCH	Clock High Time 8207-16	20	180			ns	1
		8207-10			TCLCL/3−3		ns	2
		8207-8	TCLCL/3−3				ns	2
4	TRTVCL	Reset to CLK ↓ Setup	20		40		ns	4
5	TRTH	Reset Pulse Width	4TCLCL		4TCLCL		ns	
6	TPGVRTL	PCTL, PDI, RFRQ to RESET ↓ Setup	125		125		ns	5
7	TRTLPGX	PCTL,RFRQ to RESET ↓ Hold	10		10		ns	
8	TCLPC	PCLK from CLK ↓ Delay		45		45	ns	
9	TPDVCL	PDin to CLK ↓ Setup	60		60		ns	
10	TCLPDX	PDin to CLK ↓ Hold	40		40		ns	6
RAM WARM-UP AND INITIALIZATION								
64	TCLWZL	$\overline{WZ}$ from CLK ↓ Delay		40		40	ns	7
SYNCHRONOUS μP PORT INTERFACE								
11	TPEVCL	$\overline{PE}$ to CLK ↓ Setup	27		27		ns	2
12	TKVCL	$\overline{RD}$, $\overline{WR}$, $\overline{PE}$, PCTL to CLK ↓ Setup	20				ns	1
13	TCLKX	$\overline{RD}$, $\overline{WR}$, $\overline{PE}$, PCTL to CLK ↓ Hold	0		0		ns	
14	TKVCH	$\overline{RD}$, $\overline{WR}$, PCTL to CLK ↑ Setup	20		20		ns	2

A.C. CHARACTERISTICS (Continued)

V_{CC} = 5V ±10% for 8207-8; T_A = 0°C to 70°C; V_{CC} = +5V ±5% for 8207-16

Measurements made with respect to RAS_{0-3}, CAS_{0-3}, AO_{0-8}, are a +2.4V and 0.8V. All other pins are measured at 2.0V and 0.8V. All times are ns unless otherwise indicated. Testing done with specified test load.

Ref	Symbol	Parameter	8207-16, -8		8207-10		Units	Notes
			Min	Max	Min	Max		
ASYNCHRONOUS μP PORT INTERFACE								
15	TRWVCL	$\overline{RD}$, $\overline{WR}$ to CLK ↓ Setup	20		20		ns	8, 9
16	TRWL	$\overline{RD}$, $\overline{WR}$ Pulse Width	2TCLCL+30		2TCLCL+30		ns	
17	TRWLPEV	$\overline{PE}$ from $\overline{RD}$, CFS=1 $\overline{WR}$ ↓ Delay CFS=0		TCLCL−20 TCLCL−30		 TCLCL−20	ns ns	1 2
18	TRWLPEX	$\overline{PE}$ to $\overline{RD}$, $\overline{WR}$ ↓ Hold	2TCLCL+30		2TCLCL+30		ns	
19	TRWLPTV	PCTL from $\overline{RD}$, $\overline{WR}$ ↓ Delay		TCLCL−30		TCLCL−30	ns	2
20	TRWLPTX	PCTL to $\overline{RD}$, $\overline{WR}$ ↓ Hold	2TCLCL+30		2TCLCL+30		ns	2
21	TRWLPTV	PCTL from $\overline{RD}$, $\overline{WR}$ ↓ Delay		2TCLCL−20		2TCLCL−30	ns	1
22	TRWLPTX	PCTL to $\overline{RD}$, $\overline{WR}$ ↓ Hold	3TCLCL+30		3TCLCL+40		ns	1
RAM INTERFACE								
23	TAVCL	AL, AH, BS to CLK ↓ Setup	35+tASR		35+tASR		ns	10
24	TCLAX	AL, AH, BS to CLK ↓ Hold	0		0		ns	
25	TCLLN	LEN from CLK ↓ Delay		35			ns	1
26	TCLRSL	$\overline{RAS}$ ↓ from CLK ↓ Delay		35		35	ns	
27	TRCD	$\overline{RAS}$ to $\overline{CAS}$ CFS=1 Delay CFS=0	TCLCL−25 TCLCL/2−25		 25		ns ns ns	1, 14 11, 14
28	TCLRSH	$\overline{RAS}$ ↑ from CLK ↓ Delay		50		50	ns	
29	TRAH	Row A0 to CFS=1 $\overline{RAS}$ Hold CFS=0	TCLCL/2−11 TCLCL/4−11		 18		ns ns	1, 13, 15 11, 15
30	TASR	Row A0 to $\overline{RAS}$ Setup	0		0			10, 18
31	TASC	Column A0 to CFS=1 $\overline{CAS}$ ↓ Setup CFS=0	0 5		 5		ns ns	13, 19, 20 13, 19, 20

A.C. CHARACTERISTICS (Continued)

V_{CC} = 5V ±10% for 8207-8; T_A = 0°C to 70°C; V_{CC} = +5V ±5% for 8207-16

Measurements made with respect to RAS_{0-3}, CAS_{0-3}, AO_{0-8}, are a +2.4V and 0.8V. All other pins are measured at 2.0V and 0.8V. All times are ns unless otherwise indicated. Testing done with specified test load.

Ref	Symbol	Parameter	8207-16, -8		8207-10		Units	Notes
			Min	Max	Min	Max		
RAM INTERFACE (Continued)								
32	TCAH	Column A0 to $\overline{CAS}$ Hold	(See DRAM Interface Tables)					21
33	TCLCSL	$\overline{CAS}$ ↓ from CLK ↓ Delay	TCLCL/4+30	TCLCL/1.8+53	TCLCL/4+30	100	ns	11, 12
34	TCLCSL	$\overline{CAS}$ ↓ from CLK ↓ Delay		35		40	ns	1
35	TCLCSH	$\overline{CAS}$ ↑ from CLK ↓ Delay		50		50	ns	
36	TCLW	WE from CLK ↓ Delay		35		35	ns	
37	TCLTKL	$\overline{XACK}$ ↓ from CLK ↓ Delay		35		35	ns	
38	TRWLTKH	$\overline{XACK}$ ↑ from $\overline{RD}$ ↑, $\overline{WR}$ ↑ Delay		50		50	ns	
39	TCLAKL	$\overline{AACK}$ ↓ from $\overline{CLK}$ ↓ Delay		35		35	ns	
40	TCLAKH	$\overline{AACK}$ ↑ from CLK ↓ Delay		50		50	ns	
41	TCLDL	$\overline{DBM}$ from CLK ↓ Delay		35		35	ns	
ECC INTERFACE								
42	TWRLFV	$\overline{FWR}$ from $\overline{WR}$ ↓ Delay CFS=1 CFS=0		2TCLCL−40 TCLCL+TCL−40		100	ns ns	1, 22 2, 22
43	TFVCL	$\overline{FWR}$ to CLK ↓ Setup	40		30		ns	23
44	TCLFX	$\overline{FWR}$ to CLK ↓ Hold	0		0		ns	24
45	TEVCL	$\overline{ERROR}$ to CLK ↓ Setup	20		20		ns	25, 26
46	TCLEX	$\overline{ERROR}$ to CLK ↓ Hold	0		0		ns	
47	TCLRL	R/$\overline{W}$ ↓ from CLK ↓ Delay		40		40	ns	
48	TCLRH	R/$\overline{W}$ ↑ from CLK ↓ Delay		50		50	ns	
49	TCEVCL	CE to CLK ↓ Setup	20		20		ns	25, 27
50	TCLCEX	CE to CLK ↓ Hold	0		0		ns	
51	TCLES	$\overline{ESTB}$ from CLK ↓ Delay		35		45	ns	

A.C. CHARACTERISTICS (Continued)

V_{CC} = 5V ±10% for 8207-8; T_A = 0°C to 70°C; V_{CC} = +5V ±5% for 8207-16

Measurements made with respect to RAS_{0-3}, CAS_{0-3}, AO_{0-8}, are a +2.4V and 0.8V. All other pins are measured at 2.0V and 0.8V. All times are ns unless otherwise indicated. Testing done with specified test load.

Ref	Symbol	Parameter	8207-16, -8		8207-10		Units	Notes
			Min	Max	Min	Max		
PORT SWITCHING AND LOCK								
52	TCLMV	MUX from CLK ↓ Delay		45		45	ns	
53	TCLPNV	PSEN from CLK ↓ Delay	TCL	TCL+35	TCL	TCL+35	ns	28
54	TCLPSV	PSEL from CLK ↓		35		35	ns	
55	TLKVCL	LOCK to CLK ↓ Setup	30		30		ns	30, 31
56	TCLLKX	LOCK to CLK ↓ Hold	10		10		ns	30, 31
57	TRWLLKV	LOCK from $\overline{RD}$ ↓, $\overline{WR}$ ↓ Delay		2TCLCL−30		2TCLCL−30	ns	31, 32
58	TRWHLKX	LOCK to $\overline{RD}$ ↓, $\overline{WR}$ ↓ Hold	3TCLCL+30		3TCLCL+30		ns	31, 32
REFRESH REQUEST								
59	TRFVCL	RFRQ to CLK ↓ Setup	20		20		ns	
60	TCLRFX	RFRQ to CLK ↓ Hold	10		10		ns	
61	TFRFH	Failsafe RFRQ Pulse Width	TCLCL+30		TCLCL+30		ns	33
62	TRFXCL	Single RFRQ Inactive to CLK ↓ Setup	20		20		ns	34
63	TBRFH	Burst RFRQ Pulse Width	2TCLCL+30		2TCLCL+30		ns	33

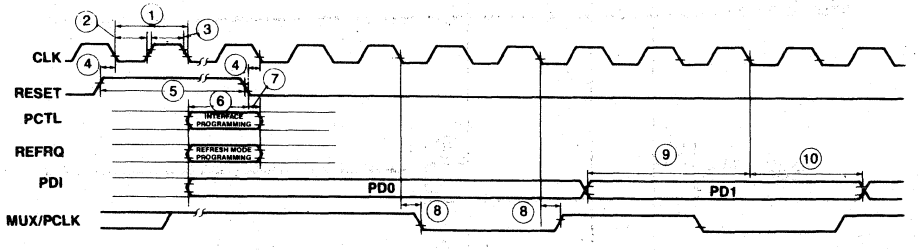
NOTES:
1. Specification when programmed in the Fast Cycle processor mode (iAPX 286 mode).
2. Specification when programmed in the Slow Cycle processor mode (iAPX 186 mode).
3. tR and tF are referenced from the 3.5V and 1.0V levels.
4. RESET is internally synchronized to CLK. Hence a set-up time is required only to guarantee its recognition at a particular clock edge.
5. The first programming bit (PD0) is also sampled by RESET going low.
6. TCLPDX is guaranteed if programming data is shifted using PCLK.
7. WZ is issued only in ECC mode.
8. TRWVCL is not required for an asynchronous command except to guarantee its recognition at a particular clock edge.
9. Valid when programmed in either Fast or Slow Cycle mode.
10. tASR is a user specified parameter and its value should be added accordingly to TAVCL.
11. When programmed in Slow Cycle mode and 125 ns ≤ TCLCL < 200 ns.
12. When programmed in Slow Cycle mode and 200 ns ≤ TCLCL.
13. Specification for Test Load conditions.
14. tRCD (actual) = tRCD (specification) + 0.06 (ΔC_{RAS}) − 0.6 (ΔC_{CAS}) where ΔC = C (test load) − C (actual) in pF (These are first order approximations).
15. tRAH (actual) = tRAH (specification) + 0.06 (ΔC_{RAS}) − 0.022 (ΔC_{A0}) where ΔC = C (test load) − C (actual) in pF. (These are first order approximations.)
18. tASR (actual) = tASR (specification) + 0.06 (ΔC_{A0}) − 0.025 (ΔC_{RAS}) where ΔC = C (test load) − C (actual) in pF. (These are first order approximations.)
19. tASC (actual) = tASC (specification) + 0.06 (ΔC_{A0}) − 0.025 (ΔC_{CAS}) where ΔC = C (test load) − C (actual) in pF. (These are first order approximations.)

20. tASC is a function of clock frequency and thus varies with changes in frequency. A minimum value is specified.
21. See 8207 DRAM Interface Tables 14–18.
22. TWRLFV is defined for both synchronous and asynchronous $\overline{FWR}$. In systems in which $\overline{FWR}$ is decoded directly from the address inputs to the 8207, TCLFV is automatically guaranteed by TCLAV.
23. TFVCL is defined for synchronous $\overline{FWR}$.
24. TCLFV is defined for both synchronous and asynchronous $\overline{FWR}$. In systems in which $\overline{FWR}$ is decoded directly from the address inputs to the 8207 TCLFV is automatically guaranteed by TCLAV.
25. $\overline{ERROR}$ and CE are set-up to CLK ↓ in fast cycle mode and CLK ↑ in slow cycle mode.
26. $\overline{ERROR}$ is set-up to the same edge as R/$\overline{W}$ is referenced to, in RMW cycles.
27. CE is set-up to the same edge as WE is referenced to in RMW cycles.
28. Specification when TCL < 25 ns.
29. Synchronous operation only. Must arrive by the second clock falling edge after the clock edge which recognizes the command in order to be effective.
30. LOCK must be held active for the entire period the opposite port must be locked out. One clock after the release of LOCK the opposite port will be able to obtain access to memory.
31. Asynchronous mode only. In this mode a synchronizer stage is used internally in the 8207 to synchronize up LOCK. TRWLLKV and TRWHLKX are only required for guaranteeing that LOCK will be recognized for the requesting port, but these parameters are not required for correct 8207 operation.
32. TFRFH and TBRFH pertain to asynchronous operation only.
33. Single RFRQ cannot be supplied asynchronously.

2

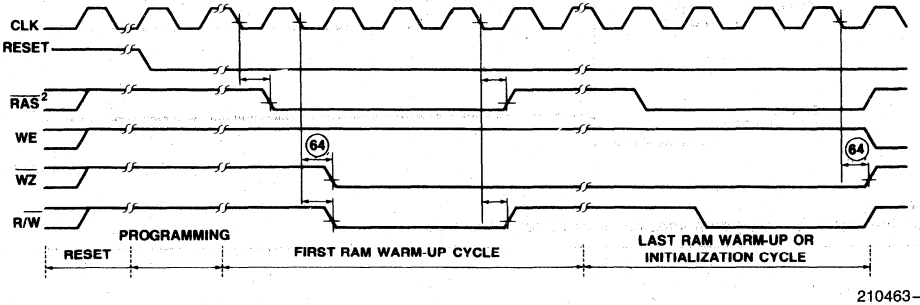
WAVEFORMS

CLOCK AND PROGRAMMING TIMINGS



210463-28

RAM WARM-UP AND MEMORY INITIALIZATION CYCLES

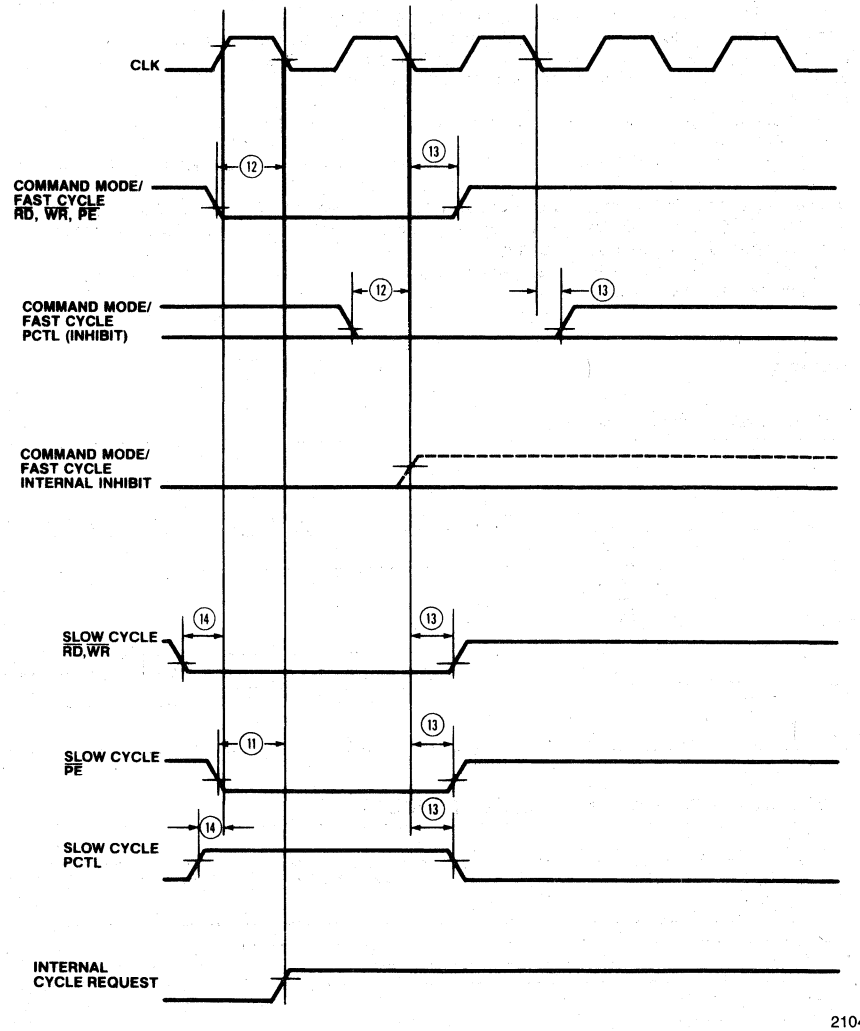


210463-29

NOTES:
1. When in non-ECC mode or in ECC mode with the TM2 programming bit on, there are no initialization cycles, when in ECC mode with TM2 off, the dummy cycles are followed by initialization cycles.
2. The present example assumes a RAS four clocks long.

WAVEFORMS (Continued)

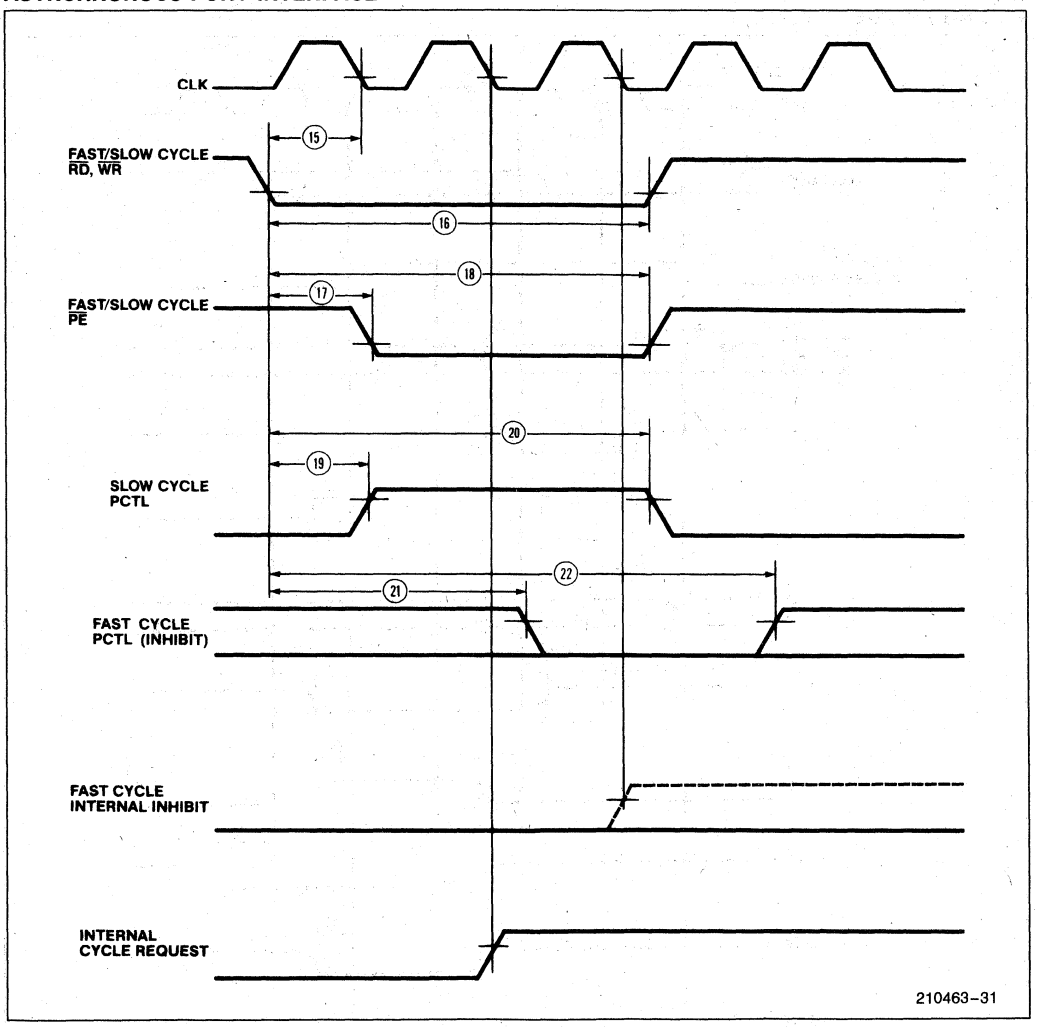
SYNCHRONOUS PORT INTERFACE



NOTE:
Actual transitions are programmable. Refer to Tables 12 and 13.

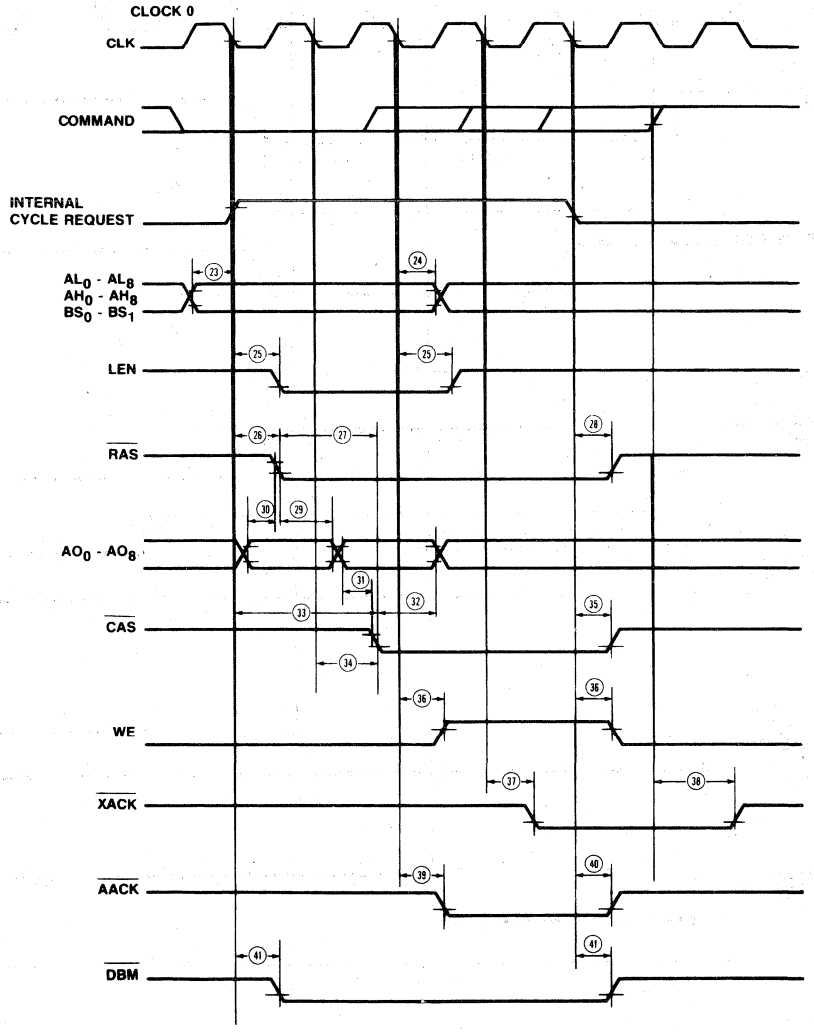
WAVEFORMS (Continued)

ASYNCHRONOUS PORT INTERFACE



WAVEFORMS (Continued)

RAM INTERFACE TIMING ECC AND NON-ECC MODE

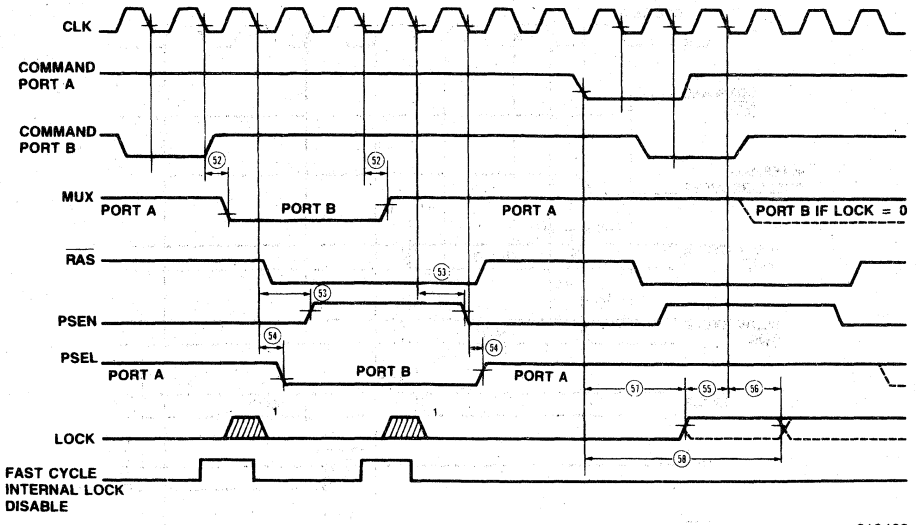


210463-32

NOTE:
Actual transitions are programmable. Refer to Tables 12 and 13.

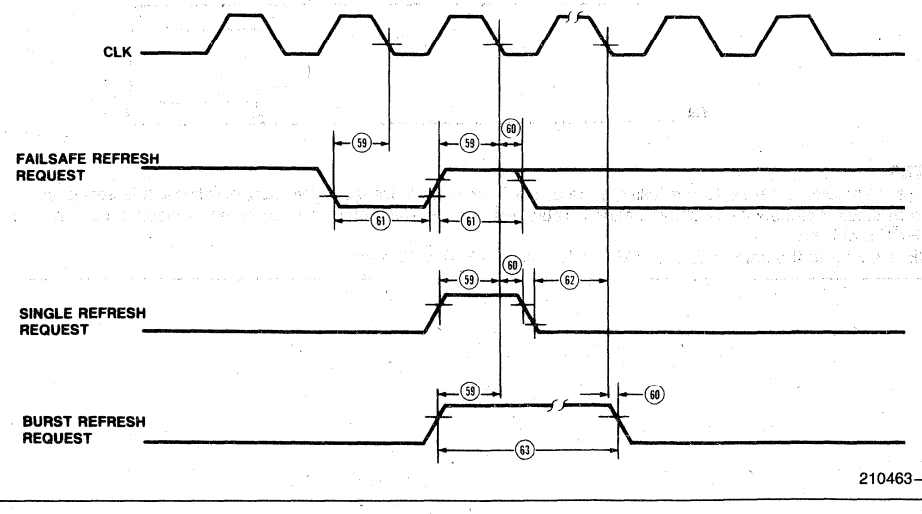
WAVEFORMS (Continued)

PORT SWITCHING AND LOCK TIMING



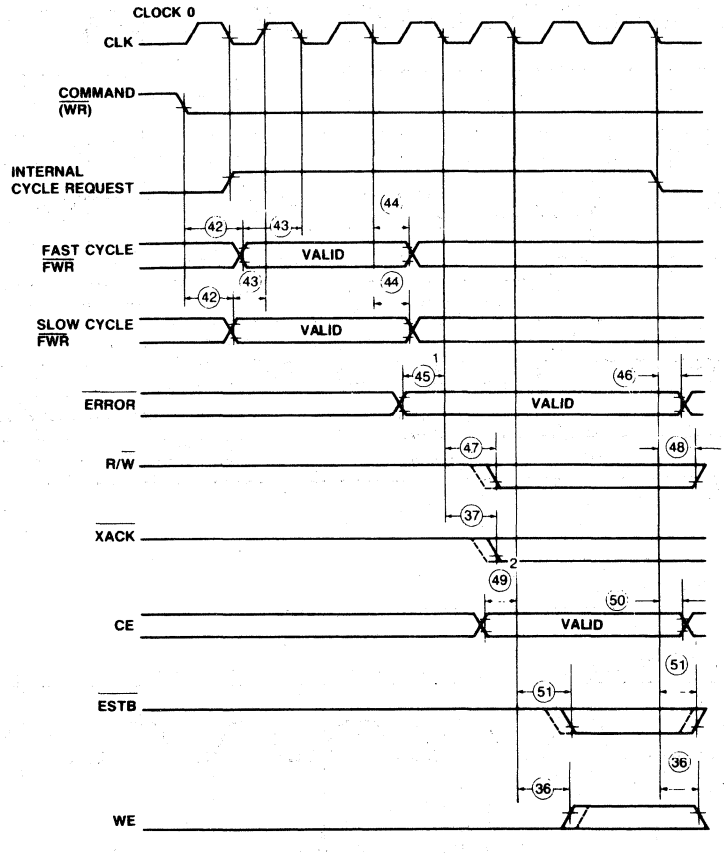
NOTE:
Transients during MUX switching.

REFRESH REQUEST TIMING



WAVEFORMS (Continued)

ECC INTERFACE TIMING



210463-35

NOTES:
1. This parameter is set-up to the falling edge of clock, as shown, for fast cycle configurations. It is set-up to the rising edge of clock if in slow cycle configurations. Table 13A shows which clock and clock edge these signals are set-up in the R/$\overline{W}$ L column.
2. CE is set-up to the same edge as WE is referenced to in RMW cycles.

CONFIGURATION TIMING CHARTS

The timing charts that follow are based on 8 basic system configurations where the 8207 operates.

Tables 10 and 11 give a description of non-ECC and ECC system configurations based on the 8207's PD0, PD3, PD4, PD10 and PD11 programming bits.

Table 10. Non-ECC System Configurations

Non-ECC Mode: PD0 = 0

Timing Conf.	$\overline{CFS}$(PD3)	$\overline{RFS}$(PD4)	EXT(PD10)	$\overline{FFS}$(PD11)
C_0	iAPX286(0)	Fast RAM(0)	Not EXT(0)	12 MHz(1)
C_0	iAPX286(0)	Fast RAM(0)	EXT(1)	12 MHz(1)
C_0	iAPX286(0)	Slow RAM(1)	Not EXT(0)	12 MHz(1)
C_0	iAPX286(0)	Slow RAM(1)	EXT(1)	12 MHz(1)
C_0	iAPX286(0)	Fast RAM(0)	Not EXT(0)	16 MHz(0)
C_1	iAPX286(0)	Slow RAM(1)	Not EXT(0)	16 MHz(0)
C_1	iAPX286(0)	Fast RAM(0)	EXT(1)	16 MHz(0)
C_2	iAPX286(0)	Slow RAM(1)	EXT(1)	16 MHz(0)
C_3	iAPX186(1)	Fast RAM(0)	Not EXT(0)	10, 8 MHz(0)
C_3	iAPX186(1)	Slow RAM(1)	Not EXT(0)	10, 8 MHz(0)
C_3	iAPX186(1)	Fast RAM(0)	EXT(1)	10, 8 MHz(0)
C_3	iAPX186(1)	Fast RAM(0)	Not EXT(0)	6 MHz(1)
C_3	iAPX186(1)	Fast RAM(0)	EXT(1)	6 MHz(1)
C_3	iAPX186(1)	Slow RAM(1)	Not EXT(0)	6 MHz(1)
C_3	iAPX186(1)	Slow RAM(1)	EXT(1)	6 MHz(1)
C_4	iAPX186(1)	Slow RAM(1)	EXT(1)	10, 8 MHz(0)

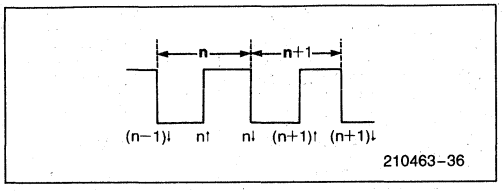
Table 11. ECC System Configurations

ECC Mode: PD0 = 1

Timing Conf.	CFS(PD3)	RFS(PD4)	$\overline{EXT}$(PD10)	FFS(PD11)
C_0	iAPX286(1)	Slow RAM(0)	M/S EDCU(0)	12 MHz(0)
C_0	iAPX286(1)	Slow RAM(0)	M EDCU(1)	12 MHz(0)
C_0	iAPX286(1)	Fast RAM(1)	M/S EDCU(0)	12 MHz(0)
C_0	iAPX286(1)	Fast RAM(1)	M EDCU(1)	12 MHz(0)
C_0	iAPX286(1)	Fast RAM(1)	M EDCU(1)	16 MHz(1)
C_1	iAPX286(1)	Slow RAM(0)	M EDCU(1)	16 MHz(1)
C_2	iAPX286(1)	Fast RAM(1)	M/S EDCU(0)	16 MHz(1)
C_3	iAPX286(1)	Slow RAM(0)	M/S EDCU(0)	16 MHz(1)
C_4	iAPX186(0)	Slow RAM(0)	M/S EDCU(0)	6 MHz(0)
C_4	iAPX186(0)	Fast RAM(1)	M/S EDCU(0)	6 MHz(0)
C_4	iAPX186(0)	Slow RAM(0)	M EDCU(1)	10, 8 MHz(1)
C_4	iAPX186(0)	Fast RAM(1)	M EDCU(1)	10, 8 MHz(1)
C_5	iAPX186(0)	Slow RAM(0)	M/S EDCU(0)	10, 8 MHz(1)
C_5	iAPX186(0)	Fast RAM(1)	M/S EDCU(0)	10, 8 MHz(1)
C_6	iAPX186(0)	Slow RAM(0)	M EDCU(1)	6 MHz(0)
C_6	iAPX186(0)	Fast RAM(1)	M EDCU(1)	6 MHz(0)

Using the Timing Charts

The notation used to indicate which clock edge triggers an output transition is "n ↑ " or "n ↓ ", where "n" is the number of clock periods that have passed since clock 0, the reference clock, and " ↑ " refers to rising edge and " ↓ " to falling edge. A clock period is defined as the interval from a clock falling edge to the following falling edge. Clock edges are defined as shown below.



The clock edges which trigger transitions on each 8207 output are tabulated in Table 12 for non-ECC mode and Table 13 for ECC mode. "H" refers to the high-going transition, and "L" to low-going transition; "V" refers to valid, and "$\overline{V}$" to non-valid.

Clock 0 is defined as the clock in which the 8207 begins a memory cycle, either as a result of a port request which has just arrived, or of a port request which was stored previously but could not be serviced at the time of its arrival because the 8207 was performing another memory cycle. Clock 0 may be identified externally by the leading edge of RAS, which is always triggered on 0 ↓.

Notes for interpreting the timing charts:

1. **PSEL - valid** is given as the latest time it can occur. It is entirely possible for PSEL to become valid before the time given in a refresh cycle. PSEL can switch as defined in the chart, but it has no bearing on the refresh cycle itself, but only on a subsequent cycle for one of the external ports.
2. **LEN - low** is given as the latest time it can occur. LEN is only activated by port A configured in Fast Cycle iAPX286 mode, and thus it is not activated by a refresh cycle, although it may be activated by port A during a refresh cycle.
3. **ADDRESS - col** is the time column address becomes valid.
4. In non-ECC mode the $\overline{CAS}$, $\overline{EAACK}$, $\overline{LAACK}$ and $\overline{XACK}$ outputs are not issued during refresh.
5. In ECC mode there are really seven types of cycles: Read without error, read with error, full write, partial write without error, partial write with error, refresh without error, and refresh with error. These cycles may be derived from the timing chart as follows:
   A. Read without error: Use row marked 'RD, RF'.
   B. Read with error: Use row marked 'RMW', except for $\overline{EAACK}$ and $\overline{LAACK}$, which should be taken from 'RD, RF'. If the error is uncorrectable. WE will not be issued.
   C. Full write: Use row marked 'WR'.
   D. Partial write without error: Use row marked 'RMW', except that $\overline{DBM}$ and $\overline{ESTB}$ will not be issued.
   E. Partial write with error: Use row marked 'RMW', except that $\overline{DBM}$ will not be issued. If the error is uncorrectable, WE will not be issued.
   F. Refresh without error: Use row marked 'RD, RF', except that $\overline{ESTB}$, $\overline{EAACK}$, $\overline{LAACK}$, and $\overline{XACK}$ will not be issued.
   G. Refresh with error: Use row marked 'RMW', except that $\overline{EAACK}$, $\overline{LAACK}$, $\overline{ESTB}$, and $\overline{XACK}$ will not be issued. If the error is uncorrectable WE will not be issued.
6. **XACK - high** is reset asynchronously by command going inactive and not by a clock edge.
7. **MUX - valid** is given as the latest time it can occur.

Table 12A. Timing Chart—Non-ECC Mode

		PSEN		PSEL		$\overline{DBM}$		LEN		$\overline{RAS}$		$\overline{CAS}$		WE	
C_n	Cycle	H	L	V	$\overline{V}$	L	H	L	H	L	H	L	H	H	L
C_0	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓	0↓	2↓	0↓	3↓	1↓	4↓		
	WR	0↓	4↓	0↓	5↓			0↓	2↓	0↓	5↓	1↓	5↓	2↓	5↓
C_1	RD, RF	0↓	5↓	0↓	6↓	0↓	6↓	0↓	2↓	0↓	4↓	1↓	6↓		
	WR	0↓	4↓	0↓	5↓			0↓	2↓	0↓	5↓	1↓	5↓	2↓	5↓
C_2	RD, RF	0↓	5↓	0↓	6↓	0↓	6↓	0↓	2↓	0↓	4↓	1↓	6↓		
	WR	0↓	4↓	0↓	5↓			0↓	2↓	0↓	5↓	1↓	5↓	2↓	5↓
C_3	RD, RF	0↓	2↓	0↓	3↓	0↓	3↓			0↓	3↓	0↓	3↓		
	WR	0↓	3↓	0↓	4↓					0↓	4↓	0↓	4↓	2↑	4↓
C_4	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓			0↓	4↓	0↓	4↓		
	WR	0↓	3↓	0↓	4↓					0↓	4↓	0↓	4↓	2↑	4↓

Table 12B. Timing Chart—Non-ECC Mode

		Col Addr		$\overline{EAACK}$		$\overline{LAACK}$		$\overline{XACK}$		MUX	
C_n	Cycle	V	$\overline{V}$	L	H	L	H	L	H	V	$\overline{V}$
C_0	RD, RF	0↓	2↓	1↓	4↓	2↓	5↓	3↓	$\overline{RD}$	−2↓	2↓
	WR	0↓	2↓	1↓	4↓	1↓	4↓	3↓	$\overline{WR}$	−2↓	2↓
C_1	RD, RF	0↓	3↓	2↓	5↓	2↓	5↓	4↓	$\overline{RD}$	−2↓	2↓
	WR	0↓	3↓	1↓	4↓	1↓	4↓	3↓	$\overline{WR}$	−2↓	2↓
C_2	RD, RF	0↓	3↓	2↓	5↓	3↓	6↓	4↓	$\overline{RD}$	−2↓	2↓
	WR	0↓	3↓	1↓	4↓	1↓	4↓	3↓	$\overline{WR}$	−2↓	2↓
C_3	RD, RF	0↓	2↓	0↓	2↓	1↓	3↓	2↓	$\overline{RD}$	−1↓	2↓
	WR	0↓	2↓	0↓	2↓	1↑	3↑	2↓	$\overline{WR}$	−1↓	2↓
C_4	RD, RF	0↓	2↓	1↓	3↓	1↓	3↓	3↑	$\overline{RD}$	−1↓	2↓
	WR	0↓	2↓	0↓	2↓	1↑	3↑	2↓	$\overline{WR}$	−1↓	2↓

Table 13A. Timing Chart—ECC Mode

		PSEN		PSEL		DBM		LEN		RAS		CAS		R/W		WE	
C_n	Cycle	H	L	V	$\overline{V}$	L	H	L	H	L	H	L	H	L	H	H	L
C_0	RD, RF	0↓	5↓	0↓	6↓	0↓	6↓	0↓	2↓	0↓	4↓	1↓	6↓				
	WR	0↓	5↓	0↓	6↓			0↓	2↓	0↓	6↓	1↓	6↓	1↓	6↓	3↓	6↓
	RMW	0↓	8↓	0↓	9↓	0↓	9↓	0↓	2↓	0↓	9↓	1↓	9↓	4↓	9↓	6↓	9↓
C_1	RD, RF	0↓	5↓	0↓	6↓	0↓	6↓	0↓	2↓	0↓	4↓	1↓	6↓				
	WR	0↓	5↓	0↓	6↓			0↓	2↓	0↓	6↓	1↓	6↓	1↓	6↓	3↓	6↓
	RMW	0↓	8↓	0↓	9↓	0↓	9↓	0↓	2↓	0↓	9↓	1↓	9↓	4↓	9↓	6↓	9↓
C_2	RD, RF	0↓	6↓	0↓	7↓	0↓	7↓	0↓	2↓	0↓	5↓	1↓	7↓				
	WR	0↓	6↓	0↓	7↓			0↓	2↓	0↓	7↓	1↓	7↓	1↓	7↓	4↓	7↓
	RMW	0↓	10↓	0↓	11↓	0↓	11↓	0↓	2↓	0↓	11↓	1↓	11↓	5↓	11↓	8↓	11↓
C_3	RD, RF	0↓	6↓	0↓	7↓	0↓	7↓	0↓	2↓	0↓	5↓	1↓	7↓				
	WR	0↓	6↓	0↓	7↓			0↓	2↓	0↓	7↓	1↓	7↓	1↓	7↓	4↓	7↓
	RMW	0↓	10↓	0↓	11↓	0↓	11↓	0↓	2↓	0↓	11↓	1↓	11↓	5↓	11↓	8↓	11↓
C_4	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓			0↓	3↓	0↓	4↓				
	WR	0↓	4↓	0↓	5↓					0↓	5↓	0↓	5↓	1↑	5↓	3↑	5↓
	RMW	0↓	6↓	0↓	7↓	0↓	7↓			0↓	7↓	0↓	7↓	3↑	7↓	5↑	7↓
C_5	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓			0↓	3↓	0↓	4↓				
	WR	0↓	4↓	0↓	5↓					0↓	5↓	0↓	5↓	1↑	5↓	3↑	5↓
	RMW	0↓	6↓	0↓	7↓	0↓	7↓			0↓	7↓	0↓	7↓	3↑	7↓	5↑	7↓
C_6	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓			0↓	3↓	0↓	4↓				
	WR	0↓	3↓	0↓	4↓					0↓	4↓	0↓	4↓	1↑	4↓	2↑	4↓
	RMW	0↓	4↓	0↓	5↓	0↓	5↓			0↓	5↓	0↓	5↓	2↑	5↓	3↑	5↓

Table 13B. Timing Chart—ECC Mode

		Col Addr		$\overline{\text{ESTB}}$		$\overline{\text{EAACK}}$		$\overline{\text{LAACK}}$		$\overline{\text{XACK}}$		$\overline{\text{MUX}}$	
C_n	Cycle	V	$\overline{V}$	L	H	L	H	L	H	L	H	V	$\overline{V}$
C_0	RD, RF	0↓	2↓			2↓	5↓	3↓	6↓	4↓	RD	−2↓	2↓
	WR	0↓	2↓			2↓	5↓	2↓	5↓	4↓	WR	−2↓	2↓
	RMW	0↓	2↓	6↓	8↓	5↓	8↓	5↓	8↓	7↓	WR	−2↓	2↓
C_1	RD, RF	0↓	3↓			3↓	6↓	3↓	6↓	4↓	RD	−2↓	2↓
	WR	0↓	3↓			2↓	5↓	2↓	5↓	4↓	WR	−2↓	2↓
	RMW	0↓	3↓	6↓	8↓	5↓	8↓	5↓	8↓	7↓	WR	−2↓	2↓
C_2	RD, RF	0↓	3↓			4↓	7↓	4↓	7↓	5↓	RD	−2↓	2↓
	WR	0↓	3↓			3↓	6↓	3↓	6↓	5↓	WR	−2↓	2↓
	RMW	0↓	3↓	8↓	10↓	7↓	10↓	7↓	10↓	9↓	WR	−2↓	2↓
C_3	RD, RF	0↓	3↓			4↓	7↓	5↓	8↓	5↓	RD	−2↓	2↓
	WR	0↓	3↓			3↓	6↓	3↓	6↓	5↓	WR	−2↓	2↓
	RMW	0↓	3↓	8↓	10↓	7↓	10↓	7↓	10↓	9↓	WR	−2↓	2↓
C_4	RD, RF	0↓	2↓			1↓	3↓	2↑	4↑	3↑	RD	−1↓	2↓
	WR	0↓	2↓			1↓	3↓	2↑	4↑	3↓	WR	−1↓	2↓
	RMW	0↓	2↓	5↑	6↑	3↓	5↓	4↑	6↑	5↓	WR	−1↓	2↓
C_5	RD, RF	0↓	2↓			2↓	4↓	3↑	5↑	3↑	RD	−1↓	2↓
	WR	0↓	2↓			1↓	3↓	2↑	4↑	3↓	WR	−1↓	2↓
	RMW	0↓	2↓	5↑	6↑	3↓	5↓	4↑	6↑	5↓	WR	−1↓	2↓
C_2	RD, RF	0↓	2↓			1↓	3↓	1↑	3↑	2↑	RD	−1↓	2↓
	WR	0↓	2↓			1↓	3↓	1↑	3↑	2↓	WR	−1↓	2↓
	RMW	0↓	2↓	3↑	4↑	1↓	3↓	2↑	4↑	3↓	WR	−1↓	2↓

8207—DRAM Interface Parameter Equations

Several DRAM parameters, but not all, are a direct function of 8207 timings, and the equations for these parameters are given in the following tables. The following is a list of those DRAM parameters which have NOT been included in the following tables, with an explanation for their exclusion.

READ, WRITE, READ-MODIFY-WRITE & REFRESH CYCLES

tRAC: response parameter.

tCAC: response parameter.

tREF: See "Refresh Period Options"

tCRP: must be met only if $\overline{CAS}$-only cycles, which do not occur with 8207, exist.

tRAH: See "A.C. Characteristics"

tRCD: See "A.C. Characteristics"

tASC: See "A.C. Characteristics"

tASR: See "A.C. Characteristics"

tOFF: response parameter.

READ & REFRESH CYCLES

tRCH: WE always goes active after $\overline{CAS}$ goes active, hence tRCH is guaranteed by tCPN.

WRITE CYCLE

tRC: guaranteed by tRWC.

tRAS: guaranteed by tRRW.

tCAS: guaranteed by tCRW.

tWCS: WE always activated after $\overline{CAS}$ is activated, except in memory initialization, hence tWCS is always negative (this is important for RMW only) except in memory initialization; in memory initialization tWCS is positive and has several clocks of margin.

tDS: system-dependent parameter.

tDH: system-dependent parameter.

tDHR: system-dependent parameter.

READ-MODIFY-WRITE CYCLE

tRWD: don't care in 8207 write cycles, but tabulated for 8207 RMW cycles.

tCWD: don't care in 8207 write cycles, but tabulated for 8207 RMW cycles.

Table 14. Non-ECC Mode—RD, RF Cycles

Parameter	Fast Cycle Configurations			Slow Cycle Configurations		Notes
	C_0	C_1	C_2	C_3	C_4	
tRP	3TCLCL−T26	4TCLCL−T26	4TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tCPN	3TCLCL−T35	3TCLCL−T35	3TCLCL−T35	2.5TCLCL−T35	2.5TCLCL−T35	1
tRSH	2TCLCL−T34	3TCLCL−T34	3TCLCL−T34	3TCLCL−T34	4TCLCL−T34	1
tCSH	4TCLCL−T26	6TCLCL−T26	6TCLCL−T26	3TCLCL−T26	4TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	3/30	2
tRC	6TCLCL	8TCLCL	8TCLCL	5TCLCL	6TCLCL	1
tRAS	3TCLCL−T26	4TCLCL−T26	4TCLCL−T26	3TCLCL−T26	4TCLCL−T26	1
tCAS	3TCLCL−T34	5TCLCL−T34	5TCLCL−T34	3TCLCL−T34	4TCLCL−T34	1
tRCS	2TCLCL−TCL −T36−TBUF	2TCLCL−TCL −T36−TBUF	2TCLCL−TCL −T36−TBUF	1.5TCLCL−TCL −T36−TBUF	1.5TCLCL−TCL −T36−TBUF	1

Table 15. Non-ECC Mode—WR Cycle

Parameter	Fast Cycle Configurations			Slow Cycle Configurations		Notes
	C_0	C_1	C_2	C_3	C_4	
tRP	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tCPN	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	2.5TCLCL−T35	2.5TCLCL−T35	1
tRSH	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	1
tCSH	5TCLCL−T26	5TCLCL−T26	5TCLCL−T26	4TCLCL−T26	4TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	3/30	2
tRWC	8TCLCL	8TCLCL	8TCLCL	6TCLCL	6TCLCL	1
tRRW	5TCLCL−T26	5TCLCL−T26	5TCLCL−T26	4TCLCL−T26	4TCLCL−T26	1
tCRW	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	1
tWCH	3TCLCL+TCL −T34	3TCLCL+TCL −T34	3TCLCL+TCL −T34	3TCLCL+TCL −T34	3TCLCL+TCL −T34	1, 3
tWCR	4TCLCL+TCL −T26	4TCLCL+TCL −T26	4TCLCL+TCL −T26	3TCLCL+TCL −T26	3TCLCL+TCL −T26	1, 3
tWP	2TCLCL+TCL −T36−TBUF	2TCLCL+TCL −T36−TBUF	2TCLCL+TCL −T36−TBUF	2TCLCL−T36 −TBUF	2TCLCL−T36 −TBUF	1
tRWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1
tCWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1

Table 16A. ECC Mode—RD, RF Cycles

Parameter	Fast Cycle Mode				Notes
	C_0	C_1	C_2	C_3	
tRP	4TCLCL−T26	4TCLCL−T26	4TCLCL−T26	4TCLCL−T26	1
tCPN	3TCLCL−T35	3TCLCL−T35	3TCLCL−T35	3TCLCL−T35	1
tRSH	3TCLCL−T34	3TCLCL−T34	4TCLCL−T34	4TCLCL−T34	1
tCSH	6TCLCL−T26	6TCLCL−T26	7TCLCL−T26	7TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	2
tRC	8TCLCL	8TCLCL	9TCLCL	9TCLCL	1
tRAS	4TCLCL−T26	4TCLCL−T26	5TCLCL−T26	5TCLCL−T26	1
tCAS	5TCLCL−T34	5TCLCL−T34	6TCLCL−T34	6TCLCL−T34	1
tRCS	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	1

Table 16B. ECC Mode—RD, RF Cycles

Parameter	Slow Cycle Mode			Notes
	C_4	C_5	C_6	
tRP	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tCPN	1.5TCLCL−T35	1.5TCLCL−T35	1.5TCLCL−T35	1
tRSH	3TCLCL−T34	3TCLCL−T34	3TCLCL−T34	1
tCSH	4TCLCL−T26	4TCLCL−T26	4TCLCL−T26	1
tCAH	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tT	3/30	3/30	3/30	2
tRC	5TCLCL	5TCLCL	5TCLCL	1
tRAS	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tCAS	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	1
tRCS	0.5TCLCL−T36 −TBUF	0.5TCLCL−T36 −TBUF	0.5TCLCL−T36 −TBUF	1

Table 17A. ECC Mode—WR Cycle

Parameter	Fast Cycle Mode				Notes
	C_0	C_1	C_2	C_3	
tRP	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tCPN	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	1
tRSH	5TCLCL−T34	5TCLCL−T34	6TCLCL−T34	6TCLCL−T34	1
tCSH	6TCLCL−T26	6TCLCL−T26	7TCLCL−T26	7TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	2
tRWC	9TCLCL	9TCLCL	10TCLCL	10TCLCL	1
tRRW	6TCLCL−T26	6TCLCL−T26	7TCLCL−T26	7TCLCL−T26	1
tCRW	5TCLCL−T34	5TCLCL−T34	6TCLCL−T34	6TCLCL−T34	1
tWCH	5TCLCL−T34	5TCLCL−T34	6TCLCL−T34	6TCLCL−T34	1, 4
tWCR	6TCLCL−T26	6TCLCL−T26	7TCLCL−T26	7TCLCL−T26	1, 4
tWP	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1
tRWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1
tCWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1

Table 17B. ECC Mode—WR Cycle

Parameter	Slow Cycle Mode			Notes
	C_4	C_5	C_6	
tRP	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tCPN	2.5TCLCL−T35	2.5TCLCL−T35	2.5TCLCL−T35	1
tRSH	5TCLCL−T34	5TCLCL−T34	4TCLCL−T34	1
tCSH	5TCLCL−T26	5TCLCL−T26	4TCLCL−T26	1
tCAH	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tT	3/30	3/30	3/30	2
tRWC	7TCLCL	7TCLCL	6TCLCL	1
tRRW	5TCLCL−T26	5TCLCL−T26	4TCLCL−T26	1
tCRW	5TCLCL−T34	5TCLCL−T34	4TCLCL−T34	1
tWCH	5TCLCL−T34	5TCLCL−T34	4TCLCL−T34	1, 4
tWCR	5TCLCL−T26	5TCLCL−T26	4TCLCL−T26	1, 4
tWP	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1
tRWL	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1
tCWL	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1

Table 18A. ECC Mode—RMW

Parameter	Fast Cycle Mode				Notes
	C_0	C_1	C_2	C_3	
tRP	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tCPN	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	1
tRSH	8TCLCL−T34	8TCLCL−T34	10TCLCL−T34	10TCLCL−T34	1
tCSH	9TCLCL−T26	9TCLCL−T26	11TCLCL−T26	11TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	2
tRWC	12TCLCL	12TCLCL	14TCLCL	14TCLCL	1
tRRW	9TCLCL−T26	9TCLCL−T26	11TCLCL−T26	11TCLCL−T26	1
tCRW	8TCLCL−T34	8TCLCL−T34	10TCLCL−T34	10TCLCL−T34	1
tRCS	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	1
tRWD	6TCLCL−T26	6TCLCL−T26	8TCLCL−T26	8TCLCL−T26	1, 4
tCWD	5TCLCL−T34	5TCLCL−T34	7TCLCL−T34	7TCLCL−T34	1
tWP	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1
tRWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1
tCWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1

Table 18B. ECC Mode—RMW

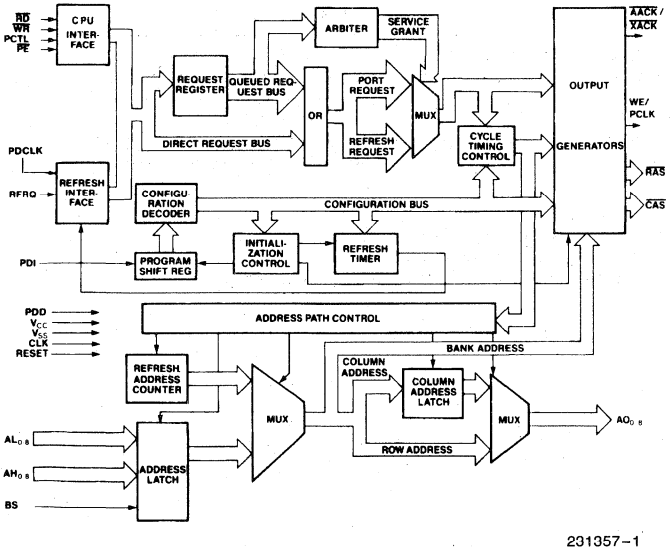
Parameter	Slow Cycle Mode			Notes
	C_4	C_5	C_6	
tRP	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tCPN	2.5TCLCL−T35	2.5TCLCL−T35	2.5TCLCL−T35	1
tRSH	7TCLCL−T34	7TCLCL−T34	5TCLCL−T34	1
tCSH	7TCLCL−T26	7TCLCL−T26	5TCLCL−T26	1
tCAH	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tT	3/30	3/30	3/30	2
tRWC	9TCLCL	9TCLCL	7TCLCL	1
tRRW	7TCLCL−T26	7TCLCL−T26	5TCLCL−T26	1
tCRW	7TCLCL−T34	7TCLCL−T34	5TCLCL−T34	1
tRCS	0.5TCLCL−T36 −TBUF	0.5TCLCL−T36 −TBUF	0.5TCLCL−T36 −TBUF	1
tRWD	4TCLCL+TCL −T26	4TCLCL+TCL −T26	2TCLCL+TCL −T26	1
tCWD	4TCLCL+TCL −T34	4TCLCL+TCL −T34	2TCLCL+TCL −T34	1
tWP	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1
tRWL	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1
tCWL	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1

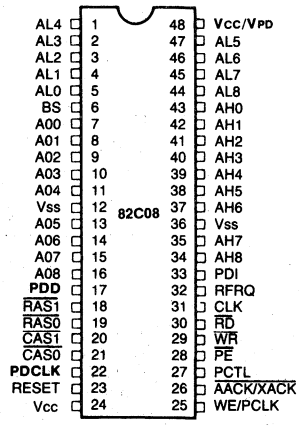
NOTES:
1. Minimum.
2. Value on right is maximum; value on left is minimum.
3. Applies to the eight warm-up cycles during initialization only.
4. Applies to the eight warm-up cycles and to the memory initialization cycles during initialization only.
5. TP = TCLCL
   T26 = TCLRSL
   T34 = TCLCSL
   T35 = TCLCSH
   T36 = TCLW
   TBUF = TTL Buffer delay.

82C08
CHMOS DYNAMIC RAM CONTROLLER

- **0 Wait State with INTEL μProcessors**
- **iAPX 286 (10, 8 MHz) } 82C08-20 20 MHz**
  82C08-16 16 MHz
  iAPX 186/88 } 82C08-10 10 MHz
  86/88 } 82C08-8 8 MHz
- **Supports 64K and 256K DRAMs (256K x 1 and 256K x 4 Organizations)**
- **Power Down Mode with Programmable Memory Refresh using Battery Backup**
- **Directly Addresses and Drives up to 1 Megabyte without External Drivers**
- **Microprocessor Data Transfer and Advance Acknowledge Signals**
- **Five Programmable Refresh Modes**
- **Automatic RAM Warm-up**
- **Pin-Compatible with 8208**
- **48 Lead Plastic DIP; 68 Lead PLCC**
  (See Intel Packaging; Order Number: 231369-001)
- **Compatible with Normal Modes of Static Column and Ripplemode DRAMs**

The Intel 82C08 Dynamic RAM Controller is a CMOS, high performance, systems oriented, Dynamic RAM controller that is designed to easily interface 64K and 256K Dynamic RAMs to Intel and other microprocessors. The 82C08 also has a power down mode where only the refresh logic is activated using battery backup.





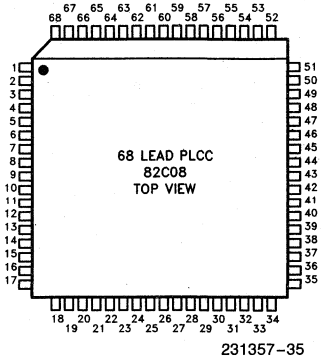


Figure 1. Block Diagram and Pinout Diagrams

September 1990
Order Number: 231357-008

Table 1. Pin Description

Symbol	DIP Pin	PLCC	Type	Name and Function
AL0 AL1 AL2 AL3 AL4 AL5 AL6 AL7 AL8	5 4 3 2 1 47 46 45 44	55 56 57 58 59 63 64 66 67	I I I I I I I I I	**ADDRESS LOW:** These lower order address inputs are used to generate the column address for the internal address multiplexer. In iAPX 286 mode (CFS = 1), these addresses are latched internally.
AH0 AH1 AH2 AH3 AH4 AH5 AH6 AH7 AH8	43 42 41 40 39 38 37 35 34	2 3 4 5 6 7 8 12 13	I I I I I I I I I	**ADDRESS HIGH:** These higher order address inputs are used to generate the row address for the internal address multiplexer. In iAPX 286 mode, these addresses are latched internally.
BS	6	50	I	**BANK SELECT:** This input is used to select one of the two banks of the dynamic RAM array.
AO0 AO1 AO2 AO3 AO4 AO5 AO6 AO7 AO8	7 8 9 10 11 13 14 15 16	49 48 47 46 45 41 40 39 38	O O O O O O O O O	**ADDRESS OUTPUTS:** These outputs are designed to provide the row and column addresses, of either the CPU or the refresh counter, to the dynamic RAM array. These outputs drive the dynamic RAM array directly and need no external drivers. However, they typically need series resistors to match impedances.
$\overline{RAS0}$ $\overline{RAS1}$	19 18	33 36	O O	**ROW ADDRESS STROBE:** These outputs are used by the dynamic RAM array to latch the row address, present on the AO0–8 pins. These outputs are selected by the BS pin. These outputs drive the dynamic RAM array directly and need no external drivers.
$\overline{CAS0}$ $\overline{CAS1}$	21 20	30 31	O O	**COLUMN ADDRESS STROBE:** These outputs are used by the dynamic RAM array to latch the column address, present on the AO0–8 pins, These outputs are selected by the BS pin. These outputs drive the dynamic RAM array directly and need no external drivers.
RESET	23	28	I	**RESET:** This active high signal causes all internal counters to be reset. Upon release of RESET, data appearing at the PDI pin is clocked-in by the PCLK output. The states of the PDI, PCTL, and RFRQ pins are sampled by RESET going inactive and are used to program the 82C08. An 8-cycle dynamic RAM warm-up is performed after clocking PDI bits into the 82C08.
WE/ PCLK	25	24	O	**WRITE ENABLE/PROGRAMMING CLOCK:** Immediately after a RESET this pin becomes PCLK and is used to clock serial programming data into the PDI pin. After the 82C08 is programmed this active high signal provides the dynamic RAM array the write enable input for a write operation.

Table 1. Pin Description (Continued)

Symbol	DIP Pin	PLCC	Type	Name and Function
$\overline{AACK}$/ $\overline{XACK}$	26	23	O	**ADVANCE ACKNOWLEDGE/TRANSFER ACKNOWLEDGE:** When the X programming bit is set to logic 0 this pin is $\overline{AACK}$ and indicates that the processor may continue processing and that data will be available when required. This signal is optimized for the system by programming the S program-bit for synchronous or asynchronous operation. The S programming bit determines whether this strobe will be early or late. If another dynamic RAM cycle is in progress at the time of the new request, the $\overline{AACK}$ is delayed. When the X programming bit is set to logic 1 this pin is $\overline{XACK}$ and indicates that data on the bus is valid during a read cycle or that data may be removed from the bus during a write cycle. $\overline{XACK}$ is a MULTIBUS compatible signal.
PCTL	27	22	I	**PORT CONTROL:** This pin is sampled on the falling edge of RESET. It configures the 82C08 to accept command inputs or processor status inputs. If PCTL is low after RESET the 82C08 is programmed to accept bus/multibus command inputs or iAPX 286 status inputs. If PCTL is high after RESET the 82C08 is programmed to accept status inputs from iAPX 86 or iAPX 186 type processors. The S2 status line should be connected to this input if programmed to accept iAPX 86 or iAPX 186 inputs. When programmed to accept bus commands or iAPX 286 status inputs, it should be tied low or it may be connected to INHIBIT when operating with MULTIBUS.
$\overline{PE}$	28	21	I	**PORT ENABLE:** This pin serves to enable a RAM cycle request. It is generally decoded from the address bus.
$\overline{WR}$	29	20	I	**WRITE:** This pin is the write memory request command input. This input also directly accepts the $\overline{S0}$ status line from Intel processors.
$\overline{RD}$	30	19	I	**READ:** This pin is the read memory request command pin. This input also directly accepts the $\overline{S1}$ status line from Intel processors.
CLK	31	16	I	**CLOCK:** This input provides the basic timing for sequencing the internal logic.
RFRQ	32	15	I	**REFRESH REQUEST:** This input is sampled on the falling edge of RESET. If RFRQ is high at RESET then the 82C08 is programmed for internal-refresh request or external-refresh request with failsafe protection. If RFRQ is low at RESET then the 82C08 is programmed for external-refresh without failsafe protection or burst refresh. Once programmed the RFRQ pin accepts signals to start an external-refresh with failsafe protection or external-refresh without failsafe protection or a burst refresh. RFRQ is also sampled when PDD is activated. When RFRQ = 1 it will cause 3 burst refresh cycles.
PDI	33	14	I	**PROGRAM DATA INPUT:** This input is sampled by RESET going low. It programs the various user selectable options in the 82C08. The PCLK pin shifts programming data into the PDI input from an external shift register. This pin may be strapped low to a default iAPX 186 mode configuration or high to a default iAPX 286 mode configuration.
*PDD	17	37	I	**POWER DOWN DETECT:** This input is sampled before every memory cycle to inform the 82C08 of system detection of power failure. When active, the 82C08 remains in power down mode and performs memory refresh only ($\overline{RAS}$-only refresh). In power down mode the 82C08 uses PDCLK for timing and VPD for power.

Table 1. Pin Description (Continued)

Symbol	DIP Pin	PLCC	Type	Name and Function
*PDCLK	22	29	I	**POWER DOWN CLOCK:** This pin is used as a clock for internal refresh circuits during power down. The input can be asynchronous to pin 31. Extended refresh is achieved by slowing down this clock. This pin should be grounded if not used.
*V_{CC}/V_{PD}	48	61, 62	I	**POWER:** Power supply for internal logic. This should be held active during power down, and normal operation.
V_{CC}	24	26, 27	I	**POWER:** Supply for drivers. Need not be held active during power down.
V_{SS}	12 36	9, 10, 11, 42, 43, 44	I I	GROUND GROUND
NC	—	17, 18, 1, 25, 32, 34, 35, 51, 53, 54, 60, 65, 68		
V_{CCS}		52		Connect to V_{PD}, Pins 61–62 for PLCC package.

*Different function than the HMOS 8208.

GENERAL DESCRIPTION

The Intel 82C08 Dynamic RAM Controller is a microcomputer peripheral device which provides the necessary signals to address, refresh, and directly drive 64K and 256K dynamic RAMs. It is compatible with static column or ripple mode DRAMs in the normal mode. It does not support the fast transfer mode of these DRAMs.

The 82C08 supports several microprocessor interface options including synchronous and asynchronous operations for iAPX 86, iAPX 186, iAPX 286, and MULTIBUS. The 82C08 will also interface to non-Intel microprocessors.

The 82C08 is a CHMOS version of the 8208 and is pin compatible with it. Three pins—17, 22, and 48—of the 82C08 are different from the 8208. They provide a power down mode that allows the system to run at a much lower ICC. In this mode, the 82C08 refreshes the DRAM using battery backup. The power down current (I_{PD}) that is drawn by the 82C08 is very small compared to the I_{CC} which allows memory to be kept alive with a battery. A separate refresh clock, pin 22, allows the designer to take advantage of RAMs that permit extended memory refresh.

The 82C08 also has some timing changes versus the 8208. In order to eliminate the external bus latches, both WE and $\overline{CAS}$ timings are shortened. These timing changes are backwards-compatible for 8208 designs.

FUNCTIONAL DESCRIPTION

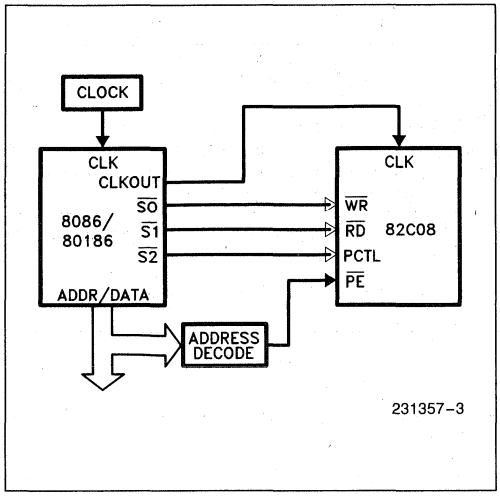
Processor Interface

The 82C08 has control circuitry capable of supporting one of several possible bus structures. The 82C08 may be programmed to run synchronous or asynchronous to the processor clock. The 82C08 has been optimized to run synchronously with Intel's iAPX 86, iAPX 88, iAPX 186/188 and iAPX 286. When the 82C08 is programmed to run in asynchronous mode, the 82C08 inserts the necessary synchronization circuitry for the $\overline{RD}$, $\overline{WR}$ inputs.

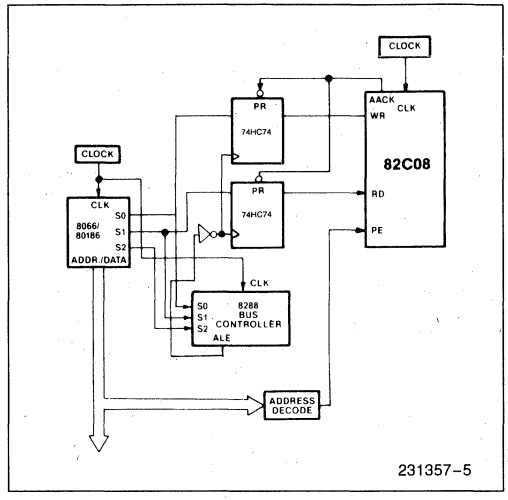
The 82C08 achieves high performance (i.e. no wait states) by decoding the status lines directly from the processor. The 82C08 can also be programmed to receive read or write MULTIBUS commands or commands from a bus controller.

The 82C08 may be programmed to operate synchronously to the processor. It can also be programmed to run at various frequencies. (See Microprocessor Clock Frequency Option.)

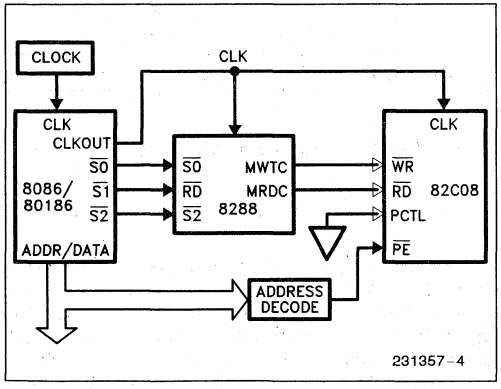
Figure 2 shows the different processor interfaces to the 82C08 using the synchronous or asynchronous mode and status or command interface. Figure 3 shows detailed interfaces to the iAPX 186 and iAPX 286 processors.



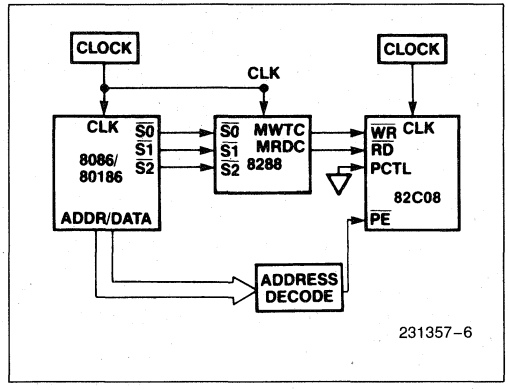
Slow-Cycle Synchronous-Status Interface



Slow-Cycle Asynchronous-Status Interface

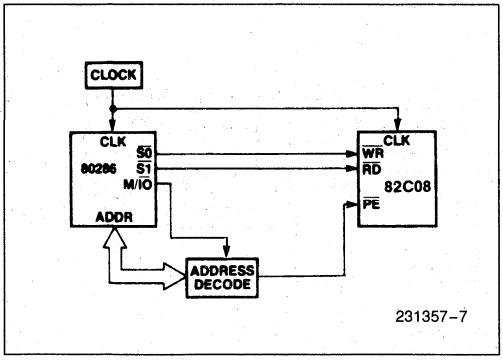


Slow-Cycle Synchronous-Command Interface

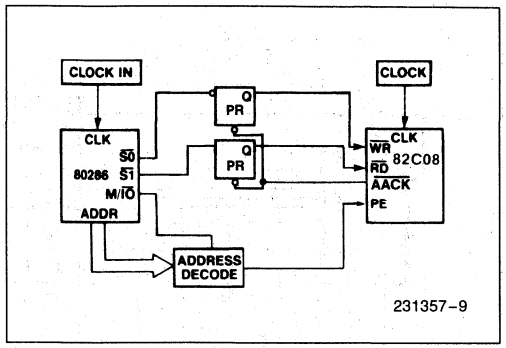


Slow-Cycle Asynchronous-Command Interface

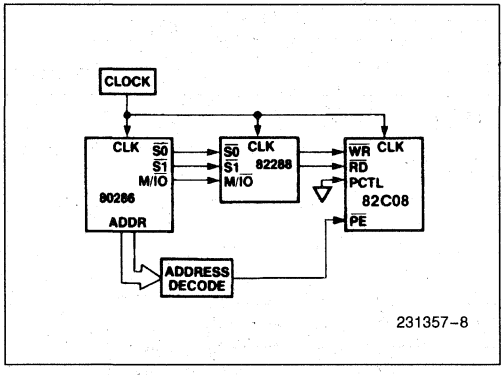
Figure 2A. Slow-cycle (CFS = 0) Port Interfaces Supported by the 82C08



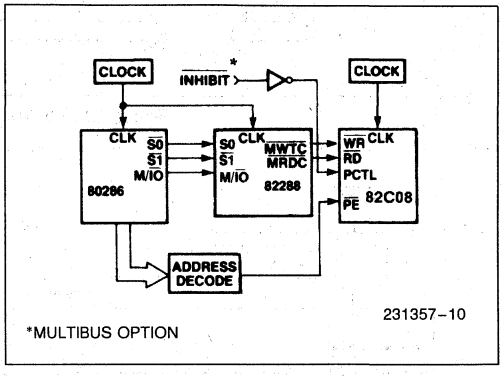
Fast-Cycle Synchronous-Status Interface



Fast-Cycle Asynchronous-Status Interface



Fast-Cycle Synchronous-Command Interface



Fast-Cycle Asynchronous-Command Interface

Figure 2B. Fast-cycle (CFS = 1) Port Interfaces Supported by the 82C08

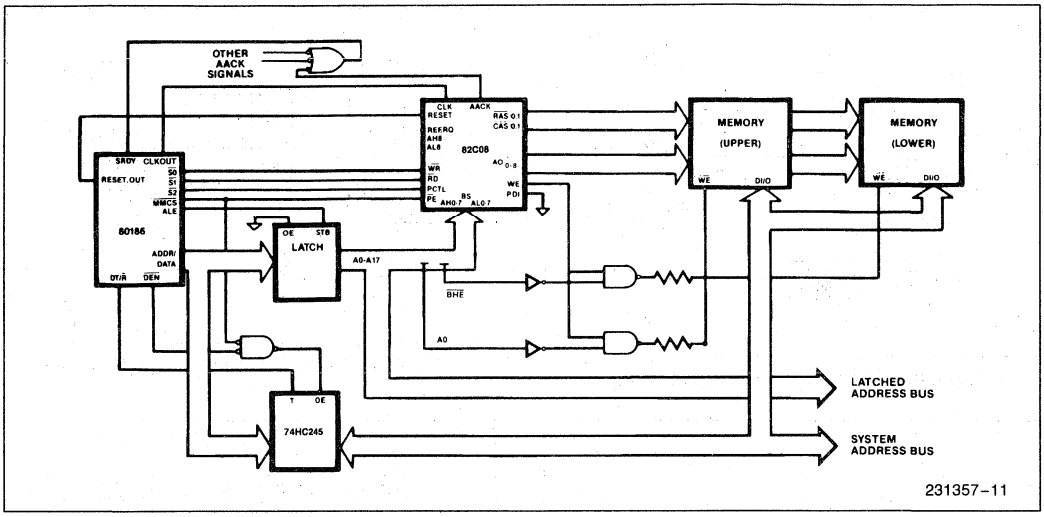


Figure 3A. 82C08 Interface to an 80186

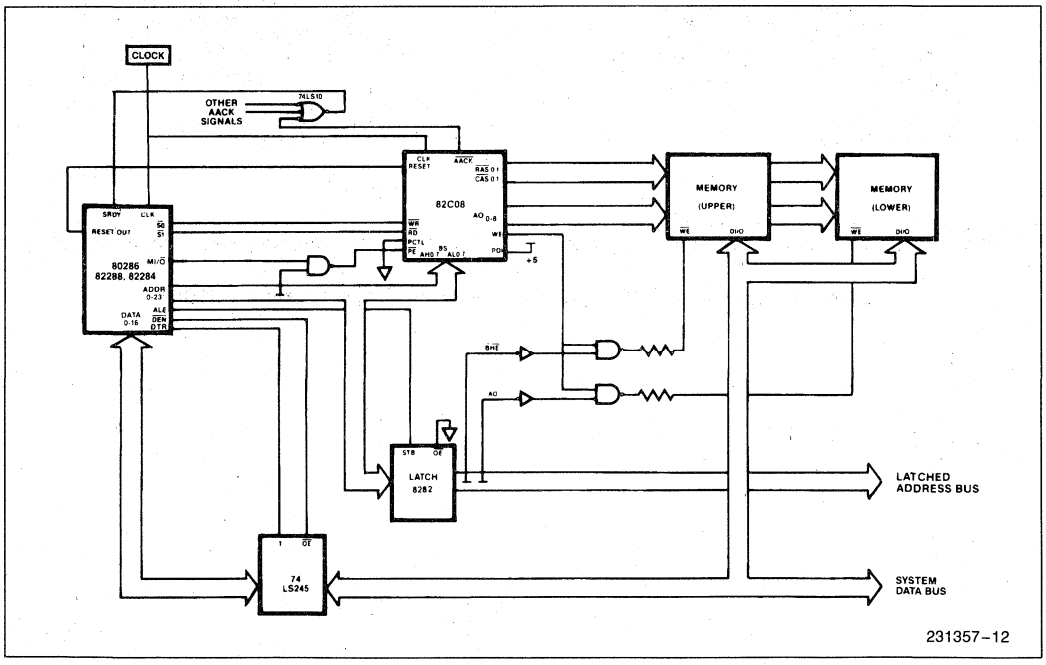
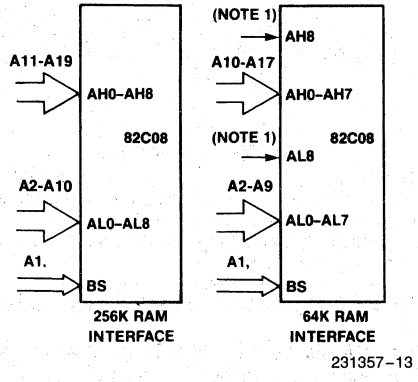


Figure 3B. 82C08 Interface to an 80286

Dynamic RAM Interface

The 82C08 is capable of addressing 64K and 256K dynamic RAMs. Figure 3 shows the connection of the processor address bus to the 82C08 using the different RAMs.



NOTES:
1. Unassigned address input pins should be strapped high.
2. A0 along with BHE are used to select a byte within a processor word.
3. Low order address bit is used as a bank select input so that consecutive memory access requests are to alternate banks allowing bank interleaving of memory cycles.

Figure 3. Processor Address Interface to the 82C08 Using 64K, and 256K RAMS

The 82C08 divides memory into two banks, each bank having its own Row ($\overline{RAS}$) and Column ($\overline{CAS}$) Address Strobe pair. This organization permits RAM cycle interleaving. RAM cycle interleaving overlaps the start of the next RAM cycle with the RAM precharge period of the previous cycle. Hiding the precharge period of one RAM cycle behind the data access period of the next RAM cycle optimizes memory bandwidth and is effective as long as successive RAM cycles occur in the alternate banks.

Successive data access to the same bank cause the 82C08 to wait for the precharge time of the previous RAM cycle. But when the 82C08 is programmed in an iAPX 186 synchronous configuration, consecutive cycles to the same bank do not result in additional wait states (i.e. 0 wait state).

If not all RAM banks are occupied, the 82C08 can be programmed to reassign the $\overline{RAS}$ and $\overline{CAS}$ strobes to allow using wider data words without increasing the loading on the $\overline{RAS}$ and $\overline{CAS}$ drivers.

Table 2 shows the bank selection decoding and the corresponding $\overline{RAS}$ and $\overline{CAS}$ assignments. For example, if only one RAM bank is occupied, then the two $\overline{RAS}$ and $\overline{CAS}$ strobes are activated with the same timing.

Table 2. Bank Selection Decoding and Word Expansion

Program Bit RB	Bank Input BS	82C08 $\overline{RAS}/\overline{CAS}$ Pair Allocation
0	0	$\overline{RAS}_{0,1}$, $\overline{CAS}_{0,1}$ to Bank 0
0	1	Illegal
1	0	$\overline{RAS}_0$, $\overline{CAS}_0$ to Bank 0
1	1	$\overline{RAS}_1$, $\overline{CAS}_1$ to Bank 1

Program bit RB is not used to check the bank select input BS. The system design must protect from accesses to "illegal", non-existent banks of memory by deactivating the PE input when addressing an "illegal", non-existent bank of memory.

The 82C08 adjusts and optimizes internal timings for either the fast or slow RAMs as programmed. (See RAM Speed Option.)

Memory Initialization

After programming, the 82C08 performs eight RAM "wake-up" cycles to prepare the dynamic RAM for proper device operation.

Refresh

The 82C08 provides an internal refresh interval counter and a refresh address counter to allow the 82C08 to refresh memory. The 82C08 has a 9-bit internal refresh address counter which will refresh 128 rows every 2 milliseconds, 256 rows every 4 milliseconds or 512 rows every 8 milliseconds, which allows all RAM refresh options to be supported. In addition, there exists the ability to refresh 256 row address locations every 2 milliseconds via the Refresh Period programming option.

The 82C08 may be programmed for any of five different refresh options: Internal refresh only, External refresh with failsafe protection, External refresh without failsafe protection, Burst refresh modes, or no refresh. (See Refresh Options.)

It is possible to decrease the refresh time interval by 10%, 20% or 30%. This option allows the 82C08 to compensate for reduced clock frequencies. Note

that an additional 5% interval shortening is built-in in all refresh interval options to compensate for clock variations and non-immediate response to the internally generated refresh request. (See Refresh Period Options.)

External Refresh Requests after RESET

External refresh requests are not recognized by the 82C08 until after it is finished programming and preparing memory for access. Memory preparation includes 8 RAM cycles to prepare and ensure proper dynamic RAM operation. The time it takes for the 82C08 to recognize a request is shown below.

eg. 82C08 System Response:

TRESP = TPROG + TPREP

where: TPROG = (40) (TCLCL) programming time

TPREP = (8) (32) (TCLCL) RAM warm-up time

if TCLCL = 125 ns then TRESP = 37 μs

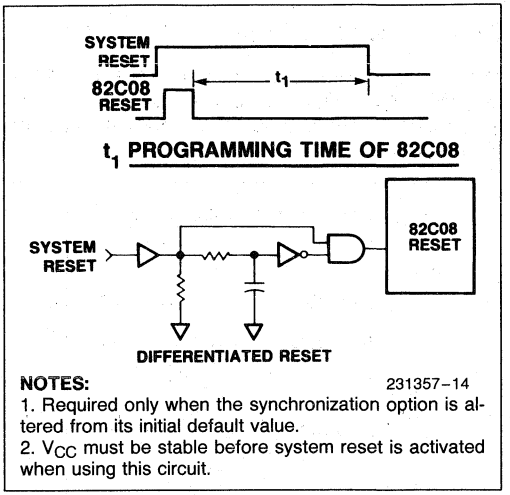
Reset

RESET is an asynchronous input, its falling edge is used by the 82C08 to directly sample the logic levels of the PCTL, RFRQ, and PDI inputs. The internally synchronized falling edge of reset is used to begin programming operations (shifting in the contents of the external shift register, if needed, into the PDI input).

Differentiated reset is unnecessary when the default synchronization programming is used.

Until programming is complete the 82C08 latches but does not respond to command or status inputs. A problem may occur if the S bit is programmed inconsistently from the Command which was latched before programming was completed. A simple means of preventing commands or status from occurring during this period is to differentiate the system reset pulse to obtain a smaller reset pulse for the 82C08.

The differentiated reset pulse would be shorter than the system reset pulse by at least the programming period required by the 82C08. The differentiated reset pulse first resets the 82C08, and system reset would reset the rest of the system. While the rest of the system is still in reset, the 82C08 completes its programming. Figure 4 illustrates a circuit to accomplish this task.



NOTES:
1. Required only when the synchronization option is altered from its initial default value.
2. V_{CC} must be stable before system reset is activated when using this circuit.

Figure 4. 82C08 Differentiated Reset Circuit

Within four clocks after RESET goes active, all the 82C08 outputs will go high, except for AO0–2, which will go low.

OPERATIONAL DESCRIPTION

Programming the 82C08

The 82C08 is programmed after reset. On the falling edge of RESET, the logic states of several input pins are latched internally. The falling edge of RESET actually performs the latching, which means that the logic levels on these inputs must be stable prior to that time. The inputs whose logic levels are latched at the end of reset are the PCTL, RFRQ, and PDI pins.

Status/Command Mode

The processor port of the 82C08 is configured by the states of the PCTL pin. Which interface is selected depends on the state of the PCTL pin at the end of reset. If PCTL is high at the end of reset, the 8086/80186 Status interface is selected; if it is low, then the MULTIBUS or Command interface is selected.

The status lines of the 80286 are similar in code and timing to the Multibus command lines, while the status code and timing of the 8086 and 8088 are identical to those of the 80186 and 80188 (ignoring the differences in clock duty cycle). Thus there exists two interface configurations, one for the 80286 status or Multibus memory commands, which is called the Command interface, and one for 8086,

8088, 80186 or 80188 status, called the 8086 Status interface. The Command interface can also directly interface to the command lines of the bus controllers for the 8086, 8088, 80186 and the 80286.

The 80186 Status interface allows direct decoding of the status lines for the iAPX 86, iAPX 88, iAPX 186 and the iAPX 188. Table 3 shows how the status lines are decoded.

Table 3A. Status Coding of 8086, 80186 and 80286

Status Code			Function	
$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	8086/80186	80286*
0	0	0	INTERRUPT	INTERRUPT
0	0	1	I/O READ	I/O READ
0	1	0	I/O WRITE	I/O WRITE
0	1	1	HALT	IDLE
1	0	0	INSTRUCTION FETCH	HALT
1	0	1	MEMORY READ	MEMORY READ
1	1	0	MEMORY WRITE	MEMORY WRITE
1	1	1	IDLE	IDLE

* Refer to 80286 pin description table

Table 3B. 82C08 Response

82C08 Command			Function	
PCTL	$\overline{RD}$	$\overline{WR}$	8086/80186 Status Interface	80286 Status or Command Interface
0	0	0	IGNORE	IGNORE*
0	0	1	IGNORE	READ
0	1	0	IGNORE	WRITE
0	1	1	IGNORE	IGNORE
1	0	0	READ	IGNORE
1	0	1	READ	INHIBIT
1	1	0	WRITE	INHIBIT
1	1	1	IGNORE	IGNORE

*Illegal with CFS = 0

Refresh Options

Immediately after system reset, the state of the RFRQ input pin is examined. If RFRQ is high, the 82C08 provides the user with the choice between self-refresh and user-generated refresh with failsafe protection. Failsafe protection guarantees that if the user does not come back with another refresh request before the internal refresh interval counter times out, a refresh request will be automatically generated. If the RFRQ pin is low immediately after a reset, then the user has the choice of a single external refresh cycle without failsafe, burst refresh or no refresh.

Internal Refresh Only

For the 82C08 to generate internal refresh requests, it is necessary only to strap the RFRQ input pin high.

External Refresh with Failsafe

To allow user-generated refresh requests with failsafe protection, it is necessary to hold the RFRQ input high until after reset. Thereafter, a low-to-high transition on this input causes a refresh request to be generated and the internal refresh interval counter to be reset. A high-to-low transition has no effect on the 82C08. A refresh request is not recognized until a previous request has been serviced.

External Refresh without Failsafe

To generate single external refresh requests without failsafe protection, it is necessary to hold RFRQ low until after reset. Thereafter, bringing RFRQ high for one clock period will cause a refresh request to be generated. A refresh request is not recognized until a previous request has been serviced.

Burst Refresh

Burst refresh is implemented through the same procedure as a single external refresh without failsafe (i.e., RFRQ is kept low until after reset). Thereafter, bringing RFRQ high for at least two clock periods will cause a burst of up to 128 row address locations to be refreshed. A refresh request is not recognized until a previous request has been serviced (i.e. burst is completed).

No Refresh

It is necessary to hold RFRQ low until after reset. This is the same as programming External Refresh without Failsafe. No refresh is accomplished by keeping RFRQ low.

Option Program Data Word

PROGRAMMING FOR SLOW CYCLE

The program data word consists of 9 program data bits, PD0–PD8. If the first program data bit, PD0 is

set to logic 0, the 82C08 is configured to support iAPX 186, 188, 86, or 88 systems. The remaining bits, PD1–PD8, may then be programmed to optimize a selected system configuration. A default of all zeros in the remaining program bits optimizes the 82C08 timing for 8 MHz Intel CPUs using 150 ns (or faster) dynamic RAMs with no performance penalty.

PROGRAMMING FOR FAST CYCLE

If the first program data bit is set to logic 1, the 82C08 is configured to support iAPX 286 systems (Command mode). A default of all ones in the program bits optimizes the 82C08 timing for an 8 MHz 286 using 120 ns DRAMs at zero wait states. Note that the programming bits PD1–8 change polarity according to PD0. This ensures the same choice of options for both default modes.

Table 4A shows the various options that can be programmed into the 82C08.

Table 4A. Program Data Word

Program Data Bit	Name		Polarity/Function
	PD0 = 0	PD0 = 1	
PD0	CFS	CFS	CFS = 0 SLOW CYCLE CFS = 1 FAST CYCLE
PD1	$\overline{S}$	S	$\overline{S}$ = 0 SYNCHRONOUS* $\overline{S}$ = 1 ASYNCHRONOUS
PD2	$\overline{RFS}$	RFS	$\overline{RFS}$ = 0 FAST RAM* $\overline{RFS}$ = 1 SLOW RAM
PD3	$\overline{RB}$	RB	RAM BANK OCCUPANCY SEE TABLE 2
PD4	CI1	$\overline{CI1}$	COUNT INTERVAL BIT 1; SEE TABLE 6
PD5	CI0	$\overline{CI0}$	COUNT INTERVAL BIT 0; SEE TABLE 6
PD6	$\overline{PLS}$	PLS	$\overline{PLS}$ = 0 LONG REFRESH PERIOD* $\overline{PLS}$ = 1 SHORT REFRESH PERIOD
PD7	$\overline{FFS}$	FFS	$\overline{FFS}$ = 0 FAST CPU FREQUENCY* $\overline{FFS}$ = 1 SLOW CPU FREQUENCY
PD8	X	$\overline{X}$	X = 0 $\overline{AACK}$* X = 1 $\overline{XACK}$

* Default in both modes

Using an External Shift Register

The 82C08 may be programmed by using an external shift register with asynchronous load capability such as a 74HC165. The reset pulse serves to parallel load the shift register and the 82C08 supplies the clocking signal (PCLK) to shift the data into the PDI programming pin. Figure 6 shows a sample circuit diagram of an external shift register circuit.

Serial data is shifted into the 82C08 via the PDI pin (33), and clock is provided by the WE/PCLK pin (25), which generates a total of 9 clock pulses.

WE/PCLK is a dual function pin. During programming, it serves to clock the external shift register, and after programming is completed, it reverts to the write enable RAM control output pin. As the pin changes state to provide the write enable signal to the dynamic RAM array, it continues to clock the shift register. This does not present a problem because data at the PDI pin is ignored after programming. Figure 7 illustrates the timing requirements of the shift register.

Default Programming Options

After reset, the 82C08 serially shifts in a program data word via the PDI pin. This pin may be strapped low or high, or connected to an external shift register. Strapping PDI low causes the 82C08 to default to the iAPX 186 system configuration, while high causes a default to the iAPX 286 configuration. Table 4B shows the characteristics of the default configuration for Fast Cycle (PDI = 1) and Slow Cycle (PDI = 0). If further system flexibility is needed, one external shift register, like a 74HC165, can be used to tailor the 82C08 to its operating environment.

Table 4B. Default Programming

Synchronous interface
Fast RAM (Note 1)
2 RAM banks occupied
128 row refresh in 2 ms; 256 in 4 ms, 512 in 8 ms
Fast processor clock frequency
Advanced ACK strobe

NOTE:
1. For iAPX 86/186 systems either slow or fast (150 or 100 ns) RAMS will run at 8 MHz with zero wait states.

Synchronous/Asynchronous Mode (S program bit)

The 82C08 may be configured to accept synchronous or asynchronous commands ($\overline{RD}$, $\overline{WR}$, PCTL) and Port Enable ($\overline{PE}$) via the S program bit. The state of the S programming bit determines whether the interface is synchronous or asynchronous.

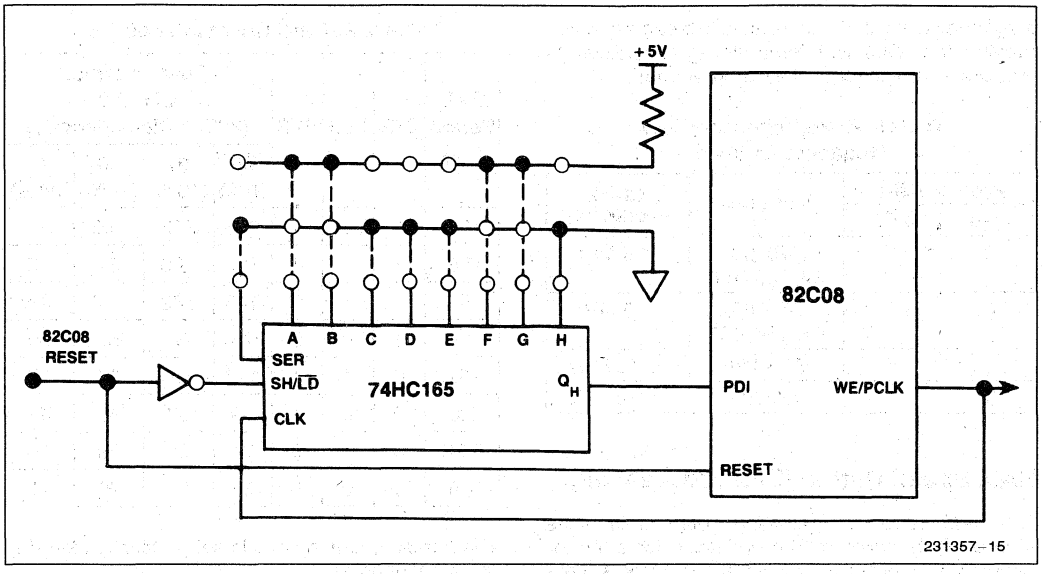
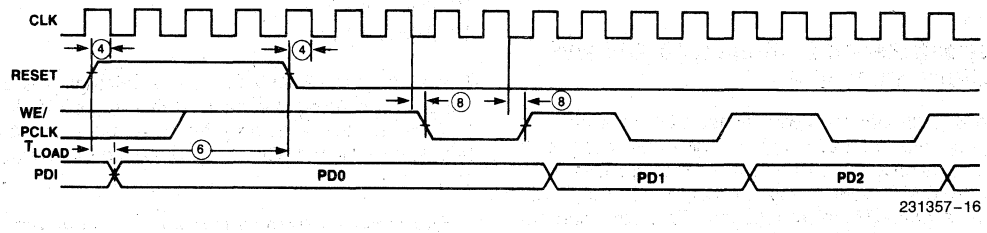


Figure 6. External Shift Register Interface



NOTES:
TRTVCL — Reset is an asychronous input, if reset occurs before TRTVCL, then it is guaranteed to be recognized.
TPGVCL — Minimum PDI valid time prior to reset going low.
TCLPC — MUX/PCLK delay.
TLOAD — Asychronous load data propagation delay.

Figure 7. Timing Illustrating External Shift Register Requirements for Programming the 82C08

While the 82C08 may be configured with either the Status or Command (MULTIBUS) interface in the Synchronous mode, certain restrictions exist in the Asynchronous mode. An Asynchronous-Command interface is directly supported. An Asynchronous-80186/80286 Status interface using the status lines of the 80186/80286 is supported with the use of TTL gates as illustrated in Figure 2. In the 80186 case, the TTL gates are needed to guarantee that status does not appear at the 82C08's inputs too much before address, so that a cycle would start before address was valid. In the case of the 80286, the TTL gates are used for lengthening the Status pulse, as required by the TRWL timing.

Microprocessor Clock Cycle Option (CFS and FFS program bits)

The 82C08 is programmed to interface with microprocessors with "slow cycle" timing like the 8086, 8088, 80186, and 80188, and with "fast cycle" microprocessors like the 80286. The CFS bit is used to select the appropriate timing.

The FFS option is used to select the speed of the microprocessor clock. Table 5 shows the various microprocessor clock frequency options that can be programmed. The external clock frequency must be

programmed so that the failsafe refresh repetition circuitry can adjust its internal timing accordingly to produce a refresh request as programmed.

Table 5. Microprocessor Clock Frequency Options

Program Bits		Processor	Clock Frequency
CFS	FFS		
0	0	iAPX 86, 88, 186, 188	≤ 5 MHz
0	1	iAPX 86, 88, 186, 188	> 5 MHz
1	0	iAPX 286	≤ 10 MHz
1	1	iAPX 286	> 10 MHz

RAM Speed Option (RFS program bit)

The RAM Speed programming option determines whether RAM timing will be optimized for a fast or slow RAM. Whether a RAM is fast or slow is measured relative to 100 ns DRAMs (fast) or 150 ns DRAMs (slow). This option is only a factor in Fast cycle Mode (CFS = 1).

Refresh Period Options (CI0, CI1 and PLS program bits)

The 82C08 refreshes with either 128 rows every 2 milliseconds, with 256 rows every 4 milliseconds or 512 rows every 8 milliseconds. This translates to one refresh cycle being executed approximately once every 15.6 microseconds. This rate can be changed to 256 rows every 2 milliseconds or a refresh approximately once every 7.8 microseconds via the Period Long/Short, program bit PLS, programming option.

The Count Interval 0 (CI0) and Count Interval 1 (CI1) programming options allow the rate at which refresh requests are generated to be increased in order to permit refresh requests to be generated close to the 15.6 or 7.8 microsecond period when the 82C08 is operating at reduced frequencies. The interval between refreshes is decreased by 0%, 10%, 20%, or 30% as a function of how the count interval bits are programmed. A 5% guardband is built-in to allow for any clock frequency variations. Table 6 shows the refresh period options available.

The numbers tabulated under Count Interval represent the number of clock periods between internal refresh requests. The percentages in parentheses represent the decrease in the interval between refresh requests.

Table 6. Refresh Count Interval Table

Ref. Period (μs)	CFS	PLS	FFS	Count Interval CI1, CI0 (82C08 Clock Periods)			
				00 (0%)	01 (10%)	10 (20%)	11 (30%)
15.6	1	1	1	236	212	188	164
7.8	1	0	1	118	106	94	82
15.6	1	1	0	148	132	116	100
7.8	1	0	0	74	66	58	50
15.6	0	1	1	118	106	94	82
7.8	0	0	1	59	53	47	41
15.6	0	1	0	74	66	58	50
7.8	0	0	0	37	33	29	25

The refresh count interval is set up for the following basic frequencies:

5 MHz slow cycle

8 MHz slow cycle

10 MHz fast cycle

16 MHz fast cycle

Example: Best 12 MHz fast cycle performance can be achieved using the basic frequency of 16 MHz (CFS = 1, FFS = 1) and the appropriate count interval bits (CI1 = 1, CI0 = 1) to reduce the frequency.

clock period × refresh count interval = refresh period

i.e. 83.3 ns × 164 = 13.6 μs

Example: 10 MHz slow cycle

CFS = 0, FFS = 1, CI1 = 0, CI0 = 0

i.e. 100 ns × 118 = 11.8 μs

Processor Timing

In order to run without wait states, $\overline{AACK}$ must be used and connected to the $\overline{SRDY}$ input of the appropriate bus controller. $\overline{AACK}$ is issued relative to a point within the RAM cycle and has no fixed relationship to the processors's request. The timing is such, however, that the processor will run without wait states, barring refresh cycles. In slow cycle, fast RAM configurations (8086, 80186), $\overline{AACK}$ is issued on the same clock cycle that issues $\overline{RAS}$.

Port Enable ($\overline{PE}$) set-up time requirements depend on whether the 82C08 is configured for synchronous

or asynchronous, fast or slow cycle operation. In a synchronous fast cycle configuration, $\overline{PE}$ is required to be set-up to the same clock edge as the commands. If $\overline{PE}$ is true (low), a RAM cycle is started; if not, the cycle is not started until the $\overline{RD}$ or $\overline{WR}$ line goes inactive and active again.

In asychronous operation, $\overline{PE}$ is required to be set-up to the same clock edge as the internally synchronized status or commands. Externally, this allows the internal synchronization delay to be added to the status (or command) -to-$\overline{PE}$ delay time, thus allowing for more external decode time than is available in synchronous operation.

The minimum synchronization delay is the additional amount that $\overline{PE}$ must be held valid. If $\overline{PE}$ is not held valid for the maximum synchronization delay time, it is possible that $\overline{PE}$ will go invalid prior to the status or command being synchronized. In such a case the 82C08 may not start a memory cycle. If a memory cycle intended for the 82C08 is not started, then no acknowledge ($\overline{AACK}$ or $\overline{XACK}$) is issued and the processor locks up in endless wait states.

Memory Acknowledge ($\overline{AACK}$, $\overline{XACK}$)

Two types of memory acknowledge signals are supplied by the 82C08. They are the Advanced Acknowledge strobe ($\overline{AACK}$) and the Transfer Acknowledge strobe ($\overline{XACK}$). The S programming bit optimizes $\overline{AACK}$ for synchronous operation ("early" $\overline{AACK}$) or asynchronous operation ("late" $\overline{AACK}$). Both the early and late $\overline{AACK}$ strobes are two clocks long for CFS = 0 and three clocks long for CFS = 1.

The $\overline{XACK}$ strobe is asserted when data is valid (for reads) or when data may be removed (for writes) and meets the MULTIBUS requirements. $\overline{XACK}$ is removed asynchronously by the command going inactive.

Since in an asynchronous operation the 82C08 removes read data before late $\overline{AACK}$ or $\overline{XACK}$ is recognized by the CPU, the user must provide for data latching in the system until the CPU reads the data. In synchronous operation data latching is unnecessary, since the 82C08 will not remove data until the CPU has read it.

If the X programming bit is high, the strobe is configured as $\overline{XACK}$, while if the bit is low, the strobe is configured as $\overline{AACK}$.

Data will always be valid a fixed time after the occurrence of the advanced acknowledge. Thus, the advanced acknowledge may also serve as a RAM cycle timing indicator.

General System Considerations

1. The $\overline{RAS}$0, 1, $\overline{CAS}$0, 1, and AO0–8 output buffers are designed to directly drive the heavy capacitive loads of the dynamic RAM arrays. To keep the RAM driver outputs from ringing excessively in the system environment it is necessary to match the output impedance with the RAM array by using series resistors. Each application may have different impedance characteristics and may require different series resistance values. The series resistance values should be determined for each application.
2. Although the 82C08 has programmable options, in practice there are only a few choices the designer must make. For iAPX 86/186 systems (CFS = 0) the C2 default mode (pin 33 tied low) is the best choice. This permits zero wait states at 8 and 10 MHz with 150 ns DRAMs. The only consideration is the refresh rate, which must be programmed if the CPU is run at less than 8 MHz.

   For iAPX 286 systems (CFS = 1) the designer must choose between configuration C0 ($\overline{RFS}$ = 0) and C1 ($\overline{RFS}$ = 1, $\overline{FFS}$ = 0). C0 permits zero wait state, 8 MHz iAPX 286 operation with 120 ns DRAMs. However, for consecutive reads, this performance depends on interleaving between two banks. The C1 configuration trades off 1 wait state performance for the ability to use 150 ns DRAMs. 150 ns DRAMs can be supported by the C0 configuration using 7 MHz iAPX 286.
3. For non-Intel microprocessors, the asynchronous command mode would be the best choice, since Intel status lines are not available. To minimize the synchronization delay, the 82C08 should use a 16 MHz clock. The preferred timing configuration is C0.

Table 7. Memory Acknowledge Summary

	Synchronous	Asynchronous	XACK
Fast Cycle	AACK Optimized for Local 80286 (early)	AACK Optimized for Remote 80286 (late)	Multibus Compatible
Slow Cycle	AACK Optimized for Local 8086/186 (early)	AACK Optimized for Remote 8086/186 (late)	Multibus Compatible

POWER DOWN

During Power Down (PD) mode, the 82C08 will perform refresh cycles to preserve the memory content. Two pins are dedicated to this feature, PDD (Power Down Detect) and PDCLK (Power Down Clock). PDD is used to inform the 82C08 of a system power failure, and will remain active as long as the power is down. It is the system's responsibility to detect power failure and to supply this signal. PDCLK is used to supply the clock during power down for the 82C08 refresh circuits. It is the system's responsibility to supply this clock.

Power Supplies

Power down is achieved by eliminating the clock from all the 82C08 circuits that are not participating in the refresh generation. The 82C08 has two power pins (V_{CC}'s), one supplies power to the output buffers and the other, to 82C08 logic. All the active circuits during power down are connected to the logic V_{CC}, including the active output buffers. Therefore, it is the user's choice to connect only the logic V_{CC} pin to the back-up power supply, or to connect both pins to it. It is recommended, however, to connect both pins to the same power supply in order to simplify and to shorten the power up time.

Extended Refresh at Power Down (PD)

To reduce power dissipation during PD, 82C08 will support the extended refresh cycle of the Intel 51CXXL (e.g. 51C64L). In this mode, the refresh period can be extended up to 64 milliseconds versus 4 milliseconds in non-extended cycles. This is achieved by slowing down the PDCLK frequency.

The user should take into consideration that when supporting extended refresh during PD, the dynamic RAM must be refreshed completely within 4 milliseconds, without active cycles, both before going into and after coming out of extended refresh. The 82C08 has the option of performing burst refresh of all the memory whenever the user cannot guarantee the 4 milliseconds idle interval. This is achieved by performing 3 consecutive burst refresh cycles, activated internally by the 82C08.

The option of refreshing all the memory is enabled in failsafe mode configuration (RFRQ input high at reset). When 82C08 detects power down, (high level at PDD) it examines the RFRQ input. High level at the RFRQ input will cause 3 PD burst refresh cycles to be performed. The user should supply the power and the system clock during the time interval of the 3 PD burst cycles, e.g. 4700 (fast cycle) or 3100 (slow cycle) clock cycles after activating PDD. Low level at RFRQ input enables the 82C08 to enter power down immediately without executing any bursts.

Power Down Procedure

The 82C08 will preserve the memory content during the entire period of the system operation. Upon detection of power down, the 82C08 will save internally its configuration status and the refresh address counter content, execute 3 burst refresh cycles. (If it is programmed to failsafe mode and the RFRQ input level is high), it will switch the internal clock from the system clock (CLK) to the power down clock (PDCLK) and will continue the refresh to the next address location. (See Figure 11.)

When power is up again (PDD input deactivated), the 82C08 will issue internal reset which will not reprogram the device and will not clear the refresh address counter, and therefore, refresh will continue to the next address location. After the internal reset, 82C08 performs 3 PD burst refresh cycles which refresh the whole memory, as at entering extended PD. This is done to give the 82C08 enough time to wake up. Notice, at the time interval of 4700 (fast cycle) or 3100 (slow cycle) clocks after power recovering no memory access will be performed.

82C08 Outputs on Power Down

Four of the 82C08 outputs are not activated during power down, $\overline{AACK}$, $\overline{CAS}0-1$ and WE. All these outputs will be forced to a non-active state, $\overline{AACK}$ and $\overline{CAS}0-1$ will be forced high and WE will be forced low (External NAND buffer is used to drive the $\overline{WE}$ DRAM inputs, hence a high level on the DRAM inputs). The other 82C08 outputs, AO0–9 and $\overline{RAS}0-1$, will switch to perform the memory refresh in a "$\overline{RAS}$-ONLY REFRESH CYCLE." The $\overline{RAS}$ outputs internal pull-ups assure high levels on these outputs, as close as possible to V_{CC}, for low DRAM power. The size of the output buffers, in power down, is smaller than the normal size, and therefore, the speed of these buffers is slower. It is done in order to reduce the speed of charging and discharging the outputs and hence reduce spikes on the power lines. It is required especially in power down, since there is only one power supply pin active which drives the output buffers as well as the internal logic.

All the device inputs, beside PDD, PDCLK, and RESET will be ignored during power down.

During power down burst refresh the 82C08 performs up to 256 refreshes. Whereas during standard burst refresh the 82C08 performs up to 128 refreshes. The power down burst refresh feature allows the 82C08 to support extended refreshes of some DRAMs, configured as 512 rows.

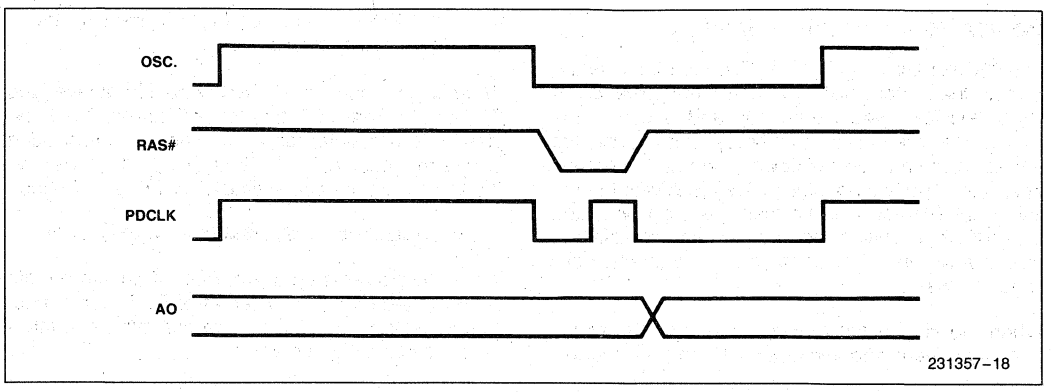


Figure 8

Power Down Detect

As previously mentioned, the PDD input will be supplied by the system to inform the 82C08 of a power failure. It can be asynchronous since the 82C08 synchronizes it internally. The PDD input will be sampled by the 82C08 before the beginning of every memory cycle but only after the termination of programming and initialization period. The user should guarantee V_{CC} and CLK stable during the programming and initialization period (300 clocks after RESET). If the whole memory refresh is required (for extended refresh) then V_{CC} and system clock should be available 4700 (fast cycle) or 3100 (slow cycle) clocks after activating PDD. If it isn't required then 82C08 should wait for present memory cycle completion and synchronization time which will take about 25 system clock cycles.

With PDD going inactive, the 82C08 synchronizes the clock back to the CLK clock, issuing internal reset and will perform 3 PD burst refresh cycles.

NOTE:
The power supplies and the CLK should go up before the PDD is deactivated. All CPU requests will be ignored when PDD is active.

Refresh during Power Down

The 82C08 has two clock pins, CLK is the system clock and PDCLK is the power down clock. PDCLK should be an independent clock which has its own crystal oscillator. When entering power down, the 82C08 will disable the system clock internally and will run with the PDCLK. The system clock will be enabled and the PDCLK will be disabled when power is up. The CLK and PDCLK will be switched internally for the refresh circuits.

During power down, '$\overline{\text{RAS}}$-ONLY REFRESH' will be performed by the 82C08. The time interval between refreshes is 5 PDCLKs and this is fixed for all applications. However, the 82C08 can support the extended refresh (up to 64 ms) by slowing down the PDCLK frequency.

During the power down refresh cycle, $\overline{\text{RAS}}$ will be activated for one PDCLK cycle only. In extended refresh, the PDCLK frequency will be below 50 kHz and this will cause a long duration of the $\overline{\text{RAS}}$ signal which will increase the DRAM's current rapidly. To minimize the $\overline{\text{RAS}}$ low pulse, the two RC networks shown in Figure 9 are designed to insert one very fast (1 μs) cycle whenever $\overline{\text{RAS}}$ is low (see Figure 8). The time constant of RC1 and RC2 should be centered around 300 ns and 100 ns respectively.

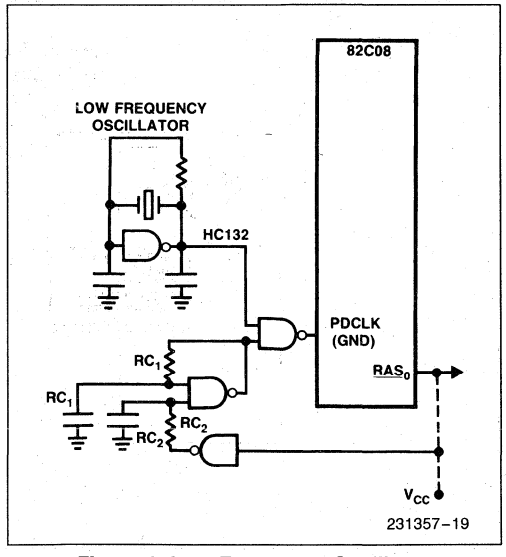


Figure 9. Low Frequency Oscillator

Power Down Synchronization

The 82C08 main clock (MCLK) is generated internally, from the system clock (CLK) and the power down clock (PDCLK) (see Figure 10), and is driving the circuits that are active at all times, i.e.: circuits that are active both in power down mode and in normal operation. The system clock (CLK) is driving the circuits that are active in normal operation only, and the PDCLK is driving the circuits that are active in power down only. The operation of the three clocks is as follows:

When entering power down mode, and the whole memory refresh is required, the CLK minimum active time after PDD is activated is 4700 (fast cycle) or 3100 (slow cycle) clocks.

When it isn't required, PDCLK should be active, and CLK should remain active for at least 20 clock cycles + synchronization time. The synchronization time is the ratio of PDCLK and CLK + 1. Therefore, the CLK minimum active time after PD is activated:

20 + [CLK(MHz) / PDCLK(MHz) + 1] clock cycles

When the power is up again, PDCLK should remain active at least 4 clock cycles after PD is going inactive, to assure completion of refresh cycle and internal synchronization time.

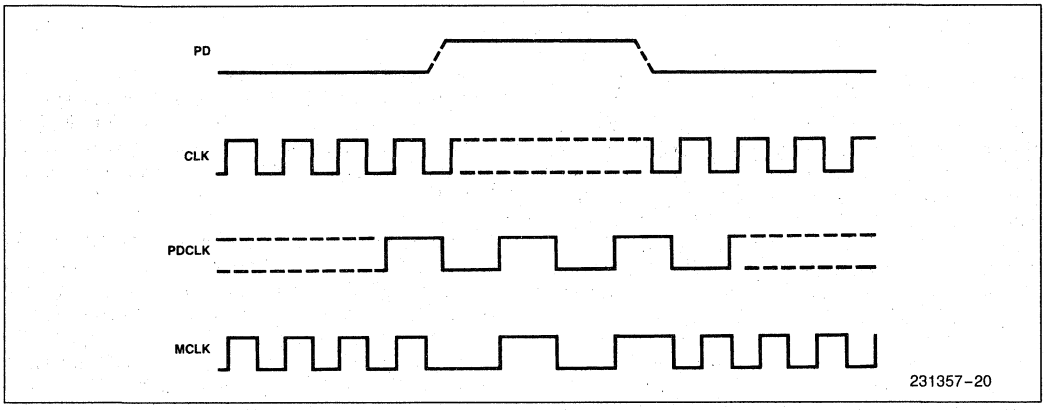


Figure 10

POWER DOWN FLOW

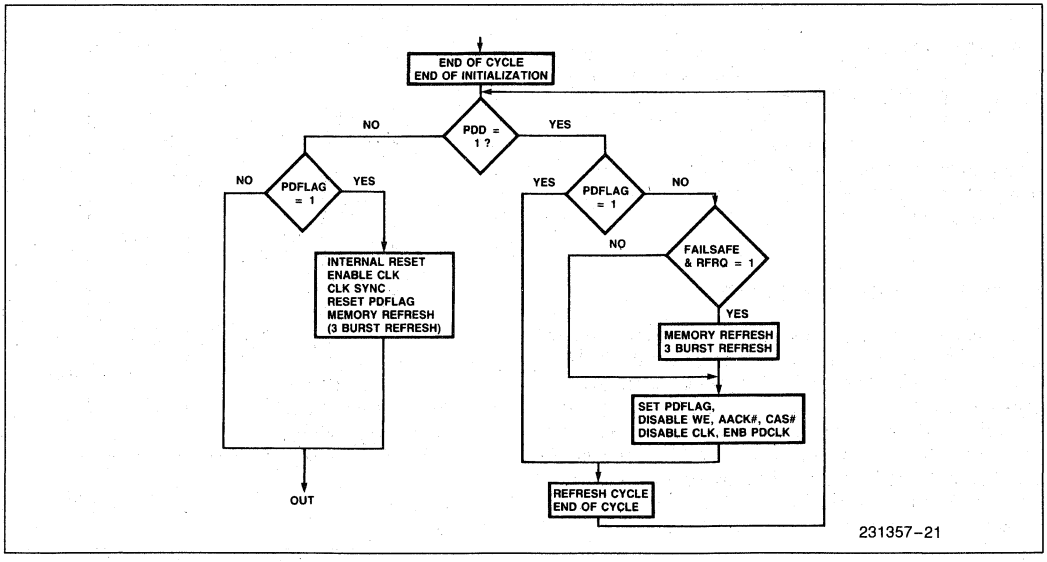


Figure 11

Differences Between 8208 and 82C08

The differences between the HMOS 8208 and the CHMOS 82C08 represent forward compatible enhancements. The 82C08 can be plugged into an 8208 socket without changes.

LOGICAL DIFFERENCES

1. 82C08 has one new feature:

   Power Down (PD)

2. 82C08 supports CMOS DRAMs with T_{RAC} 100, 150

3. Address Mapping:

Outputs	9 Most Significant Bits	9 Least Significant Bits
8208	column address	row address
82C08	row address	column address

4. Slow cycle shortening:

   1). The write cycle is two clocks shorter so consecutive writes will be executed without wait states.

   2) The WE output is two clocks shorter. Therefore, an external latch on the WE output is not necessary.

   3) CAS output is shorter by one clock on the read cycle. This reduces one level of buffers for address/data bus needed in 8208 designs. Read access margins are improved to support non-Intel spec. RAMs.

   4) The address outputs switch from row to column address one clock cycle later in the 82C08 as compared to 8208.

5. Fast cycle shortening:

   1) The write cycle in C0 configuration is shortened by one clock.

   2) For both C0 and C1 synchronous configuration, the CAS signal is shorter by one clock and the activation of RAS is tied to the Φ2 cycle of the 80286. This prevents contention on the data bus.

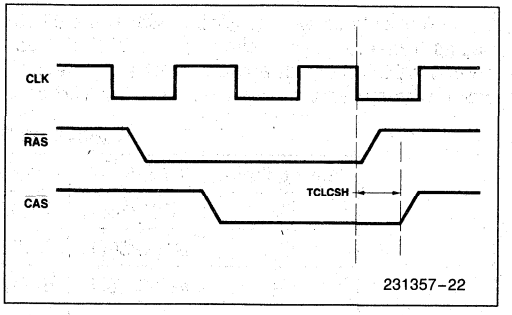
6. Supports Static Column or Ripplemode DRAMs.

ELECTRICAL DIFFERENCES

1. AC parameters:

   1) $\overline{CAS}$ delay: In C2 synchronous read cycle, the $\overline{CAS}$ is deactivated by some delay from clock falling edge (TCLCSH timing) as in the following diagram:

   In C2 write cycles the $\overline{CAS}$ activation is triggered by the clock falling edge with a delay of 35 ns from the clock. For 8208 the delay is TP/1.8 + 53.



   2) 82C08 has an additional timing parameter TARH column address to $\overline{RAS}$ ↑ hold time.

2. DC parameters: The difference is in the current consumption.

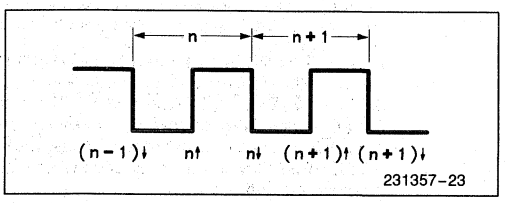
	8208	82C08
I_{CC}	300 mA	30 mA (typical) [10 + 2f] mA (max)
IPD	—	1 mA (max)
I_{SB}	—	2 mA (max)

Configuration Charts

The 82C08 operates in three basic configurations—C0, C1, C2—depending upon the programming of CFS (PD0), $\overline{RFS}$ (PD2), and $\overline{FFS}$ (PD7). Table 8 shows these configurations. These modes determine the clock edges for the 82C08's programmable signals, as shown in Table 9. Finally, Table 10 gives the programmable AC parameters of the 82C08 as a function of configuration. The non-programmable parameters are listed under AC Characteristics.

Using the Timing Charts

The notation used to indicate which clock edge triggers an output transition is "n ↑" or "n ↓", where "n" is the number of clock periods that have passed since clock 0, the reference clock, and "↑" refers to rising edge and "↓" to falling edge. A clock period is defined as the interval from a clock falling edge to the following falling edge. Clock edges are defined as shown below.



The clock edges which trigger transitions on each 82C08 output are tabulated in Table 9. "H" refers to the high-going transition, and "L" to low-going transition.

Clock 0 is defined as the clock in which the 82C08 begins a memory cycle, either as a result of a port request which has just arrived, or of a port request which was stored previously but could not be serviced at the time of its arrival because the 82C08 was performing another memory cycle. Clock 0 is identified externally by the leading edge of RAS, which is always triggered on 0 ↓.

Table 8. 82C08 Configurations

Timing Conf.	CFS(PD0)	RFS(PD2)	FFS(PD7)	Wait States*
C_0	iAPX286(1)	FAST RAM(1)	20 MHz(1)	0
C_0	iAPX286(1)	FAST RAM(1)	16 MHz(1)	0
C_1	iAPX286(1)	SLOW RAM(0)	16 MHz(1)	1
C_0	iAPX286(1)	FAST RAM(1)	10 MHz (0)	0
C_0	iAPX286(1)	SLOW RAM(0)	10 MHz (0)	0
C_2	iAPX186(0)	DON'T CARE	DON'T CARE	0

* Using $\overline{\text{EAACK}}$ (synchronous mode)

Table 9a. Timing Chart — Synchronous Mode

		$\overline{\text{RAS}}$		ADDRESS		$\overline{\text{CAS}}$		WE		$\overline{\text{EAACK}}$	
Cn	Cycle	L	H	Col	Row*	L	H	H	L	L	H
0	RD,RF	0↓	3↓	0↓	3↓	1↓	3↓			1↓	4↓
	WR	0↓	4↓	0↓	3↓	2↓	4↓	1↓	4↓	1↓	4↓
1	RD,RF	0↓	4↓	0↓	4↓	1↓	5↓			2↓	5↓
	WR	0↓	5↓	0↓	4↓	2↓	5↓	1↓	5↓	2↓	5↓
2	RD,RF	0↓	2↓	0↓	3↓	0↓	2↓			0↓	2↓
	WR	0↓	2↓	0↓	3↓	1↓	3↓	0↓	2↓	0↓	2↓

Table 9b. Timing Chart — Asynchronous Mode

		$\overline{\text{RAS}}$		ADDRESS		$\overline{\text{CAS}}$		WE		$\overline{\text{LAACK}}$		$\overline{\text{XAACK}}$	
Cn	Cycle	L	H	Col	Row*	L	H	H	L	L	H	L	H
0	RD,RF	0↓	3↓	0↓	3↓	1↓	4↓			2↓	5↓	3↓	$\overline{\text{RD}}$
	WR	0↓	4↓	0↓	3↓	2↓	4↓	1↓	4↓	1↓	4↓	3↓	$\overline{\text{WR}}$
1	RD,RF	0↓	4↓	0↓	4↓	1↓	6↓			2↓	5↓	4↓	$\overline{\text{RD}}$
	WR	0↓	5↓	0↓	4↓	2↓	5↓	1↓	5↓	1↓	4↓	3↓	$\overline{\text{WR}}$
2	RD,RF	0↓	2↓	0↓	3↓	0↓	3↓			1↓	3↓	2↓	$\overline{\text{RD}}$
	WR	0↓	2↓	0↓	3↓	1↓	3↓	0↓	2↓	1↑	3↑	2↓	$\overline{\text{WR}}$

The only difference between the two tables is the trailing edge of $\overline{\text{CAS}}$ for all read cycle configurations. In asynchronous mode, $\overline{\text{CAS}}$ trailing edge is one clock later than in synchronous mode.

NOTES FOR INTERPRETING THE TIMING CHART:

1. COLUMN ADDRESS is the time column address becomes valid.
2. The $\overline{\text{CAS}}$, $\overline{\text{EAACK}}$, $\overline{\text{LAACK}}$ and $\overline{\text{XACK}}$ outputs are not issued during refresh.
3. $\overline{\text{XACK}}$—high is reset asynchronously by command going inactive and not by a clock edge.
4. $\overline{\text{EAACK}}$ is used in synchronous mode, $\overline{\text{LAACK}}$ and $\overline{\text{XACK}}$ in asynchronous mode.
5. ADDRESS-Row is the clock edge where the 82C08 A0 switches from current column address to the next row address.
6. If a cycle is inhibited by PCTL = 1 (Multibus I/F mode) then $\overline{\text{CAS}}$ is not activated during write cycle and $\overline{\text{XACK}}$ is not activated in either read or write cycles.

*Column addresses switch to row addresses for next memory cycle. The row address buffer is transparent following this clock edge. 'TRAH' specification is guaranteed as per data sheet.

82C08—DRAM Interface Parameter Equations

Several DRAM parameters, but not all, are a direct function of 82C08 timings, and the equations for these parameters are given in the following tables. The following is a list of those DRAM parameters which have NOT been included in the following tables, with an explanation for their exclusion.

WRITE CYCLE

tDS: system-dependent parameter.
tDH: system-dependent parameter.
tDHR: system-dependent parameter.

READ, WRITE REFRESH CYCLES

tRAC: response parameter.
tCAC: response parameter.
tREF: See "Refresh Period Options".
tCRP: must be met only if CAS-only cycles, which do not occur with 82C08, exist.
tRAH: See "A.C. Characteristics"
tRCD: See "A.C. Characteristics"
tASC: See "A.C. Characteristics"
tASR: See "A.C. Characteristics"
tOFF: response parameter.

Table 10. Programmable Timings

Read and Refresh Cycles

Parameter	C2-Slow Cycle	C0-Fast Cycle	C1-Fast Cycle	Notes
tRP	2TCLCL-T27	3TCLCL-T27	3TCLCL-T27	1
tCPN	1.5TCLCL-T34	3TCLCL-T34	2TCLCL-T34	1, 5
tCPN	2.5TCLCL-T34	4TCLCL-T34	3TCLCL-T34	1, 4
tRSH	2TCLCL-T32	2TCLCL-T32	3TCLCL-T32	1
tCSH	3TCLCL-T25	4TCLCL-T25	6TCLCL-T25	1, 5
tCSH	2TCLCL+T34(min)-T25	3TCLCL-T25	5TCLCL-T25	1, 4
tCAH	3TCLCL-T32	2TCLCL-T32	3TCLCL-T32	1
tAR	3TCLCL-T25	3TCLCL-T25	4TCLCL-T25	1
tT	3/30	3/30	3/30	2
tRC	4TCLCL	6TCLCL	7TCLCL	1
tRAS	2TCLCL-T25	3TCLCL-T25	4TCLCL-T25	1
tCAS	3TCLCL-T32	3TCLCL-T32	5TCLCL-T32	1, 5
tCAS	2TCLCL+T34(min)-T32	2TCLCL-T25	4TCLCL-T32	1, 4
tRCS	TCLCL + T32(min)-T35 + TBUF	TCLCL-T35 + TBUF	2TCLCL-T35 + TBUF	1
tRCH	TCLCL + T36(min)-T34 + TBUF	TCLCL-T34 + TBUF	2TCLCL-T34 + TBUF	1

Write Cycles

Parameter	C2-Slow Cycle	C0-Fast Cycle	C1-Fast Cycle	Notes
tRP	2TCLCL-T27	3TCLCL-T27	3TCLCL-T27	1
tCPN	2TCLCL-T34	4TCLCL-T34	4TCLCL-T34	1
tRSH	TCLCL-T32	2TCLCL-T32	3TCLCL-T32	1
tCSH	3TCLCL-T25	4TCLCL-T25	5TCLCL-T25	1
tCAH	2TCLCL-T32	TCLCL-T32	2TCLCL-T32	1
tAR	3TCLCL-T25	3TCLCL-T25	4TCLCL-T25	1
tT	3/30	3/30	3/30	2
tRC	4TCLCL	7TCLCL	8TCLCL	1
tRAS	2TCLCL-T25	4TCLCL-T25	5TCLCL-T25	1
tCAS	2TCLCL-T32 + TBUF	2TCLCL-T32	3TCLCL-T32	1
tWCH	TCLCL-T32 + TBUF	2TCLCL-T32 + TBUF	3TCLCL-T32 + TBUF	1,3
tWCR	2TCLCL-T25 + TBUF	4TCLCL-T25 + TBUF	5TCLCL-T25 + TBUF	1,3
tWP	2TCLCL-T36-TBUF	3TCLCL-T36-TBUF	4TCLCL-T36-TBUF	1
tRWL	2TCLCL-T36-TBUF	3TCLCL-T36-TBUF	4TCLCL-T36-TBUF	1
tCWL	3TCLCL-T36-TBUF	3TCLCL-T36-TBUF	4TCLCL-T36-TBUF	1
tWCS	TCLCL+T36-T31-TBUF	TCLCL−T36-TBUF	TCLCL−T36-TBUF	1

NOTES:
1. Minimum.
2. Value on right is maximum; value on left is minimum.
3. Applies to the eight warm-up cycles during initialization.
4. For synchronous mode only.
5. For asynchronous mode only.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias	−0°C to +70°C
Storage Temperature	−65°C to +150°C
Voltage On Any Pin With Respect to Ground	−0.5V to +7V
Power Dissipation	0.5W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

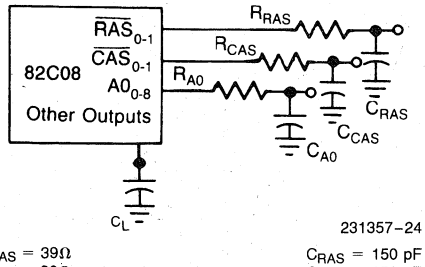
**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C; V_{CC} = 5.0V ±10%; V_{SS} = GND

Symbol	Parameter	Min	Max	Units	Comments
V_{IL}	Input Low Voltage	−0.5	+0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.45	V	(Note 1)
V_{OH}	Output High Voltage	2.6		V	(Note 1)
I_{CC}	Supply Current		10 + 2f	mA	(Note 3)
I_{LI}	Input Leakage Current		±10	μA	$0V \le V_{IN} \le V_{CC}$
V_{CL}	Clock Input Low Voltage	−0.5	+0.6	V	
V_{CH}	Clock Input High Voltage	3.8	V_{CC} + 0.5	V	
C_{IN}	Input Capacitance		20	pF	fc = 1 MHz(6)
V_{OHPD}	RAS Output High Power Down	V_{CC} − 0.5		V	(Note 2)
I_{PD}	Power Down Supply Current	—	5.0	mA	(Note 5)
I_{SB}	Standby Current	—	2.0	mA	(Note 4)

NOTE:
1. I_{OL} = 5 mA and I_{OH} = −0.32 mA WE: I_{OL} = 8 mA
2. $\overline{RAS}$ Output voltage during power down.
3. Typical value. Where f is freq. in MHz.
for CMOS: V_{IL} max = 0.5V; V_{IH} min = (V_{CC} − 0.5V)
for TTL: I_{CC} will be higher by 30 mA
4. Measured at V_{IL} = 0V and V_{IH} = V_{CC} with no loads connected.
5. IPD = 1 mA at 32 KHz with no loads connected.
6. Sampled, not 100% tested. T_A = 25°C.

A.C. Testing Load Circuit

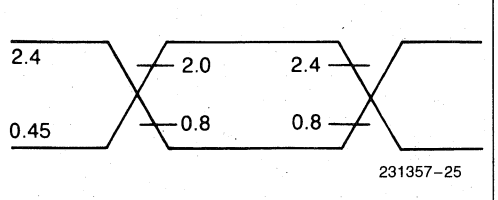


231357-24

R_{RAS} = 39Ω
R_{CAS} = 39Ω
R_{A0} = 22Ω

C_{RAS} = 150 pF
C_{CAS} = 150 pF
C_{A0} = 200 pF
C_L = 50 pF

A.C. Testing Input, Output Waveform



231357-25

A.C. Testing inputs (except clock) are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0" (clock* is driven at 4.0V and 0.45V for Logic "1" and "0" respectively). Timing measurements are made at 2.4V for Logic "1" and 0.8V for Logic "0".

* also PDCLK

A.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = +5V ±10%, V_{SS} = 0V)

Measurements made with respect to RAS_{0-1}, CAS_{0-1}, AO_{0-8}, are at +2.4V and 0.8V CLK at 3V, 1V. All other pins are measured at 2.0V and 0.8V. All times are ns unless otherwise indicated. Testing done with specified test load.

Ref	Symbol	Parameter	Min	Max	Units	Notes
CLOCK AND PROGRAMMING						
	tF	Clock Fall Time		12	ns	3
	tR	Clock Rise Time		12	ns	3
1	TCLCL	Clock Period				
		82C08-20	50	250	ns	1
		82C08-16	62.5	250	ns	1
		82C08-10	100	500	ns	2
		82C08-8	125	500	ns	2
2	TCL	Clock Low Time				
		82C08-20	12	230	ns	1
		82C08-16	15	230	ns	1
		82C08-10	44		ns	2
		82C08-8	TCLCL/2-12		ns	2
3	TCH	Clock High Time				
		82C08-20	16	230	ns	1
		82C08-16	20	230	ns	1
		82C08-10	44		ns	2
		82C08-8	TCLCL/3+2		ns	2
4	TRTVCL	Reset to CLK ↓ Setup	40		ns	4
5	TRTH	Reset Pulse Width	4TCLCL		ns	
6	TPGVRTL	PCTL, PDI, RFRQ to RESET ↓ Setup	125		ns	5
7	TRTLPGX	PCTL, RFRQ to RESET ↓ Hold	10		ns	
8	TCLPC	PCLK from CLK ↓ Delay		45	ns	
9	TPDICL	PDI to CLK ↓ Setup	60		ns	
10	TCLPDX	PDI to CLK ↓ Hold	40		ns	6
SYNCHRONOUS μP PORT INTERFACE						
11	TPEVCL	PE to CLK ↓ Setup	30			2
12	TKVCL	$\overline{RD}$, $\overline{WR}$, $\overline{PE}$, PCTL to CLK ↓ Setup	20		ns	1
13	TCLKX	$\overline{RD}$, $\overline{WR}$, $\overline{PE}$, PCTL to CLK ↓ Hold	0		ns	
14	TKVCH	RD, WR, PCTL to CLK ↑ Setup	20		ns	2

A.C. CHARACTERISTICS (Continued)

Ref	Symbol	Parameter	Min	Max	Units	Notes
ASYNCHRONOUS μP PORT INTERFACE						
15	TRWVCL	$\overline{RD}$, $\overline{WR}$ to CLK ↓ Setup	20		ns	8.9
16	TRWL	$\overline{RD}$, $\overline{WR}$ Pulse Width	2TCLCL+30		ns	
17	TRWLPEV	$\overline{PE}$ from $\overline{RD}$, $\overline{WR}$ ↓ Delay CFS=1 CFS=0		 TCLCL-20 TCLCL-30	 ns ns	 1 2
18	TRWLPEX	$\overline{PE}$ to $\overline{RD}$, $\overline{WR}$ ↓ Hold	2TCLCL+30		ns	
19	TRWLPTV	PCTL from $\overline{RD}$, $\overline{WR}$ ↓ Delay		TCLCL-30	ns	2
20	TRWLPTX	PCTL to $\overline{RD}$, $\overline{WR}$ ↓ Hold	2TCLCL+30		ns	2
21	TRWLPTV	PCTL from $\overline{RD}$, $\overline{WR}$ ↓ Delay		2TCLCL-20	ns	1
22	TRWLPTX	PCTL to $\overline{RD}$, $\overline{WR}$ ↓ Hold	3TCLCL+30		ns	1
RAM INTERFACE						
23	TAVCL	AL, AH, BS to CLK ↓ Set-up 82C08-20 82C08-16	35+tASR 50+tASR 45+tASR		ns ns ns	2
24	TCLAX	AL, AH, BS to CLK ↓ Hold	0		ns	
25	TCLRSL	RAS ↓ from CLK ↓ Delay		25 35 60	ns ns ns	1 2 24
26	TRCD	$\overline{RAS}$ to $\overline{CAS}$ Delay CFS=1 CFS=0 CFS=0 CFS=0	 TCLCL-25 30 TCLCL/2-30 60		 ns ns ns ns	 1, 14 23 2, 11, 14 2, 12, 14
27	TCLRSH	$\overline{RAS}$ ↑ from CLK ↓ Delay		25 60	ns ns	 24

A.C. CHARACTERISTICS (Continued)

Ref	Symbol	Parameter	Min	Max	Units	Notes
RAM INTERFACE (Continued)						
28	TRAH	CFS = 1 CFS = 0 CFS = 0	18 TCLCL/4-10 18		 ns ns	1, 13, 15 2, 11, 15 23
29	TASR	Row A0 $\overline{RAS}\downarrow$ Setup				10, 16
30	TASC	Column A0 to $\overline{CAS}\downarrow$ Setup CFS = 1 CFS = 0 CFS = 0	 2 5 5		 ns ns ns	 1, 13, 17, 18 2, 13, 17, 18 23
31	TCAH	Column A0 to $\overline{CAS}$ Hold	(See DRAM Interface Tables)			
32	TCLCSL	$\overline{CAS}\downarrow$ from CLK $\downarrow$ Delay CFS = 0 CFS = 0 CFS = 0 CFS = 1	 TCLCL/4 + 30 50 8	 TCLCL/1.8 + 56 105 35 35	 ns ns ns ns	 2, 26 23, 26 2, 23, 27 1
34	TCLCSH	$\overline{CAS}\uparrow$ from CLK $\downarrow$ Delay	 TCLCL/4	50 $\frac{TCLCL}{3.2}+50$	ns ns	 22
35	TCLWL	WE $\downarrow$ from CLK $\downarrow$ Delay		35	ns	
36	TCLWH	WE $\uparrow$ from CLK $\downarrow$ Delay CFS = 0 CFS = 1 CFS = 0	 TCLCL/4 + 30 50	 TCLCL/1.8 + 53 35 100	 ns ns	 2 1 23
37	TCLTKL	$\overline{XACK}\downarrow$ from CLK $\downarrow$ Delay		35	ns	
38	TRWLTKH	$\overline{XACK}\uparrow$ from $\overline{RD}\uparrow$, $\overline{WR}\uparrow$ Delay		50	ns	
39	TCLAKL	$\overline{AACK}\downarrow$ from $\overline{CLK}\downarrow$ Delay		35	ns	
40	TCLAKH	$\overline{AACK}\uparrow$ from CLK $\downarrow$ Delay		50	ns	
49	TARH	Column Address to $\overline{RAS}\uparrow$ Hold Time	2			1

A.C. CHARACTERISTICS (Continued)

Ref	Symbol	Parameter	Min	Max	Units	Notes
REFRESH REQUEST						
41	TRFVCL	RFRQ to CLK ↓ Setup	20		ns	
42	TCLRFX	RFRQ to CLK ↓ Hold	10		ns	
43	TFRFH	Failsafe RFRQ Pulse Width	TCLCL + 30		ns	19
44	TRFXCL	Single RFRQ Inactive to CLK ↓ Setup	20		ns	20
45	TBRFH	Burst RFRQ Pulse Width	2TCLCL + 30		ns	19
46	TPDDVCL	PDD Setup Time	20		ns	24, 25
47	TPDHRFX	RFRQ Valid after PDD Active	4TCLCL + 20			24
48	TRFVPDH	RFRQ Setup Time to PDD Active	20			24

The following RC loading is assumed:

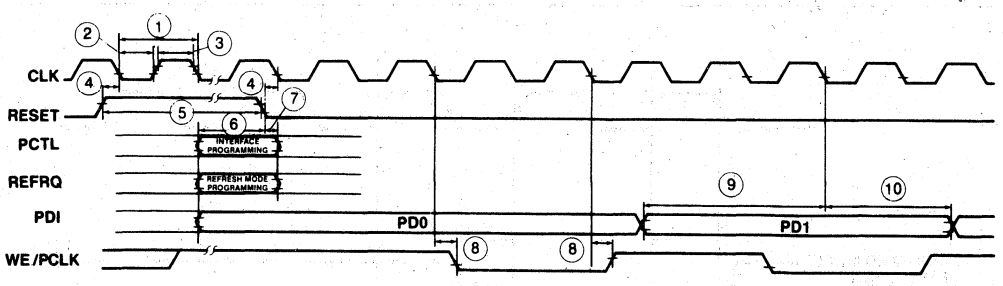
$A0_{0-8}$	R = 22Ω	C = 200 pF
$\overline{RAS}_{0-1}$, $\overline{CAS}_{0-1}$	R = 39Ω	C = 150 pF
$\overline{AACK}$, WE/PCLK		C = 50 pF

NOTES:

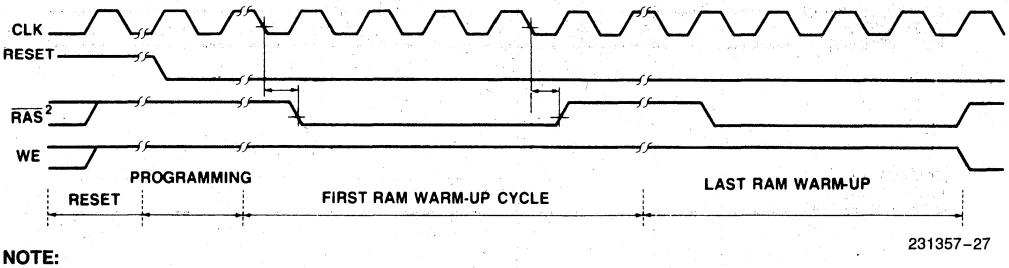
1. Specification when programmed in the Fast Cycle processor mode (iAPX 286 mode). 82C08-20, -16.
2. Specification when programmed in the Slow Cycle processor mode (iAPX 186 mode). 82C08-10, 82C08-8.
3. tR and tF are referenced from the 3.5V and 1.0V levels.
4. RESET is internally synchronized to CLK. Hence a set-up time is required only to guarantee its recognition at a particular clock edge.
5. The first programming bit (PD0) is also sampled by RESET going low.
6. TCLPDX is guaranteed if programming data is shifted using PCLK.
8. TRWVCL is not required for an asynchronous command except to guarantee its recognition at a particular clock edge.
9. Valid when programmed in either Fast or Slow Cycle mode.
10. tASR is a user specified parameter and its value should be added accordingly to TAVCL.
11. When programmed in Slow Cycle mode and 125 ns ≤ TCLCL < 200 ns.
12. When programmed in Slow Cycle mode and 200 ns ≤ TCLCL.
13. Specification for Test Load conditions.
14. tRCD (actual) = tRCD (specification) +0.06 (ΔC_{RAS}) − 0.06(ΔC_{CAS}) where ΔC = C (test load) − C (actual) in pF. (These are first order approximations.)
15. tRAH (actual) = tRAH (specification) + 0.06 (ΔC_{RAS}) − 0.022 (ΔC_{A0}) where ΔC = C (test load) − C (actual) in pF. (These are first order approximations.)
16. tASR (actual) = tASR (specification) +0.06 (ΔC_{A0}) − 0.025 (ΔC_{RAS}) where ΔC = C (test load) − C (actual) in pF. (These are first order approximations.)
17. tASC (actual) = tASC (specification) +0.06 (ΔC_{A0}) − 0.025 (ΔC_{CAS}) where ΔC (test load) − C (actual) in pF. (These are first order approximations.)
18. tASC is a function of clock frequency and thus varies with changes in frequency. A minimum value is specified.
19. TFRFH and TBRFH pertain to asynchronous operation only.
20. Single RFRQ should be supplied synchronously to avoid burst refresh.
22. CFS = 0, synchronous mode, Read cycle.
23. For 10 MHz Slow Cycle only.
24. Power down mode.
25. PDD is internally synchronized. A setup time is required only to guarantee its recognition at a particular clock edge.
26. Slow Cycle Read only.
27. Slow Cycle Write only.

WAVEFORMS

CLOCK AND PROGRAMMING TIMINGS

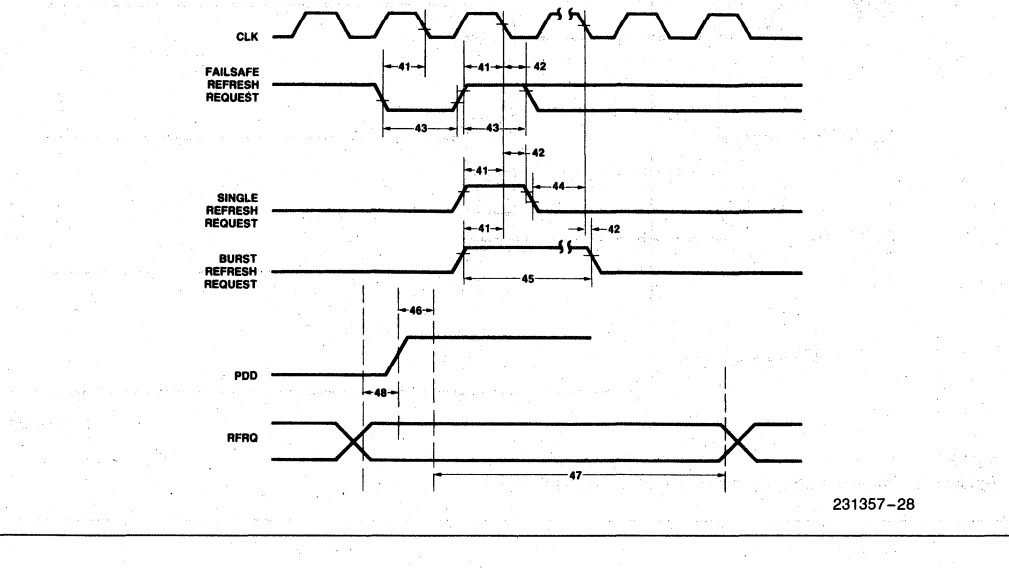


RAM WARM-UP CYCLES



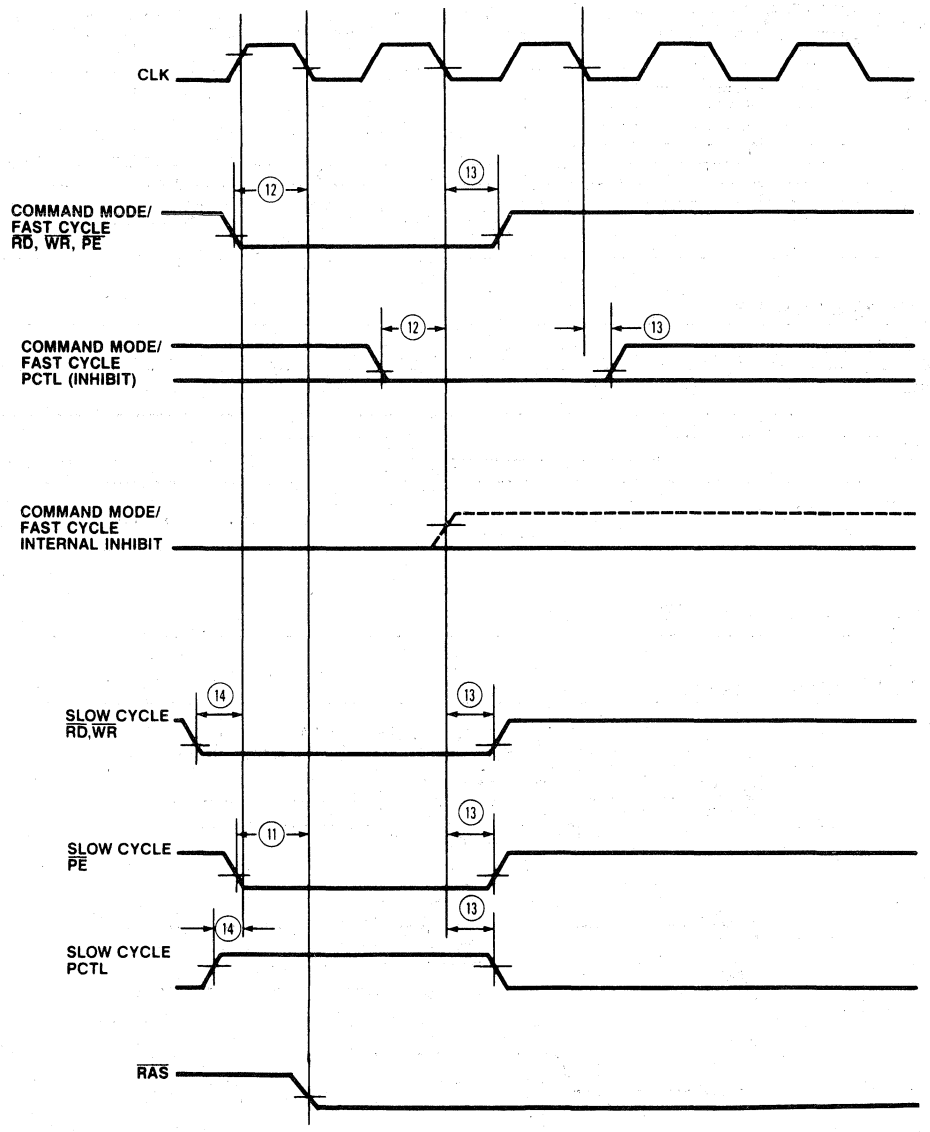
NOTE:
The present example assumes a $\overline{RAS}$ four clocks long.

REFRESH REQUEST TIMING



WAVEFORMS (Continued)

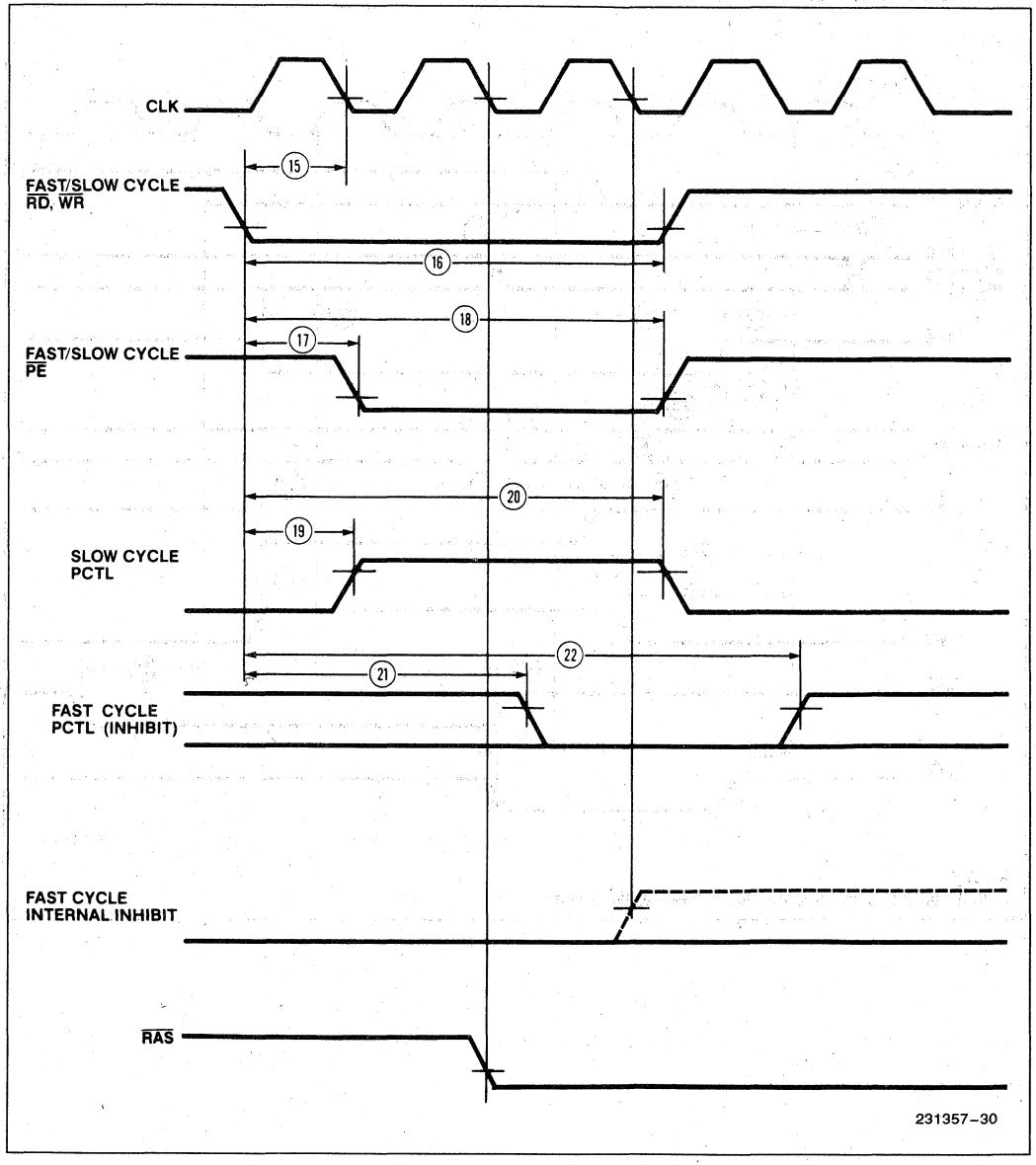
SYNCHRONOUS PORT INTERFACE



NOTE:
Actual transitions are programmable. Refer to Tables 8 and 9.

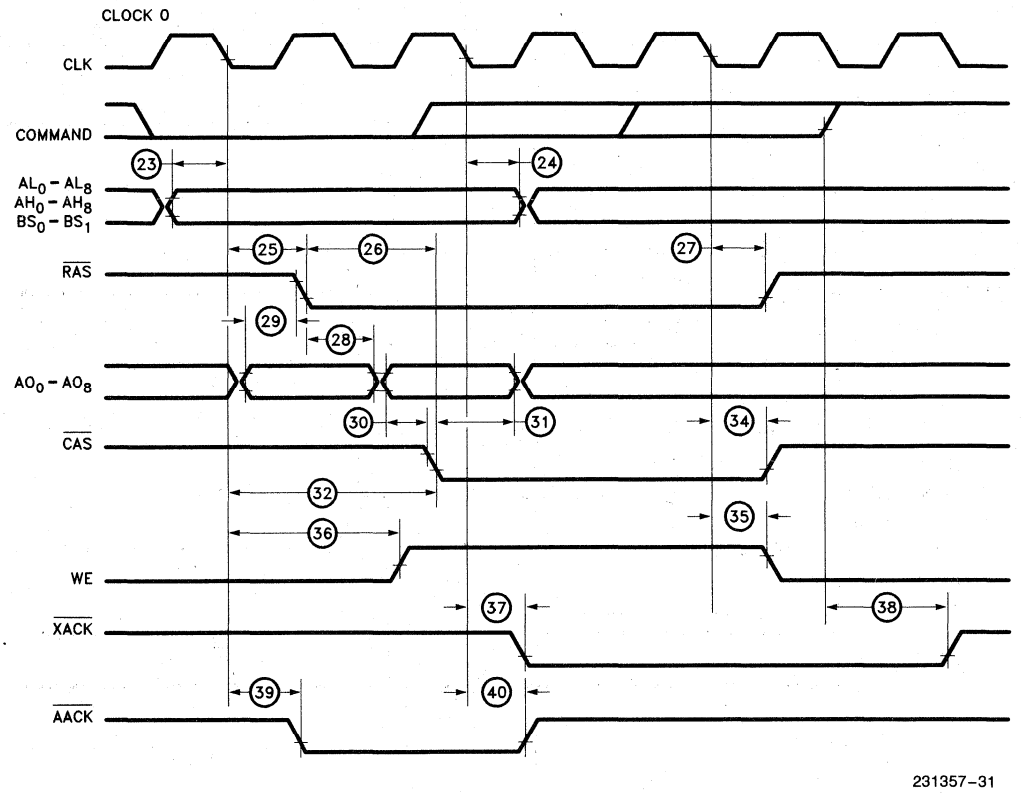
WAVEFORMS (Continued)

ASYNCHRONOUS PORT INTERFACE



WAVEFORMS (Continued)

RAM INTERFACE TIMING



NOTE:
Actual transitions are programmable. See Tables 8 and 9.

intel®

APPLICATION NOTE

AP-167

September 1989

2

Interfacing the 8207 Dynamic RAM Controller to the iAPX 186

Order Number: 230809-001

INTERFACING THE 8207 DYNAMIC RAM CONTROLLER TO THE iAPX 186

CONTENTS

PAGE

INTRODUCTION 2-101

DESIGN GOALS 2-101

USING THE 8207 2-101

8207 Programming 2-101

Acknowledge Setup Time 2-102

Read Access Margin 2-102

Write Data Setup and Hold Margin 2-103

Microprocessor Interface 2-103

RAM Array 2-103

HARDWARE DESIGN 2-103

8207 Command Setup 2-103

Address Setup 2-104

SUMMARY 2-104

INTRODUCTION

Most microprocessor based workstation designs today use large amounts of DRAM for program storage. A drawback to DRAMs is the many critical timings that must be met. This control function could easily equal the area of the DRAM array if implemented with discrete logic.

The VLSI 8207 Advanced Dynamic RAM Controller (ADRC) performs complete DRAM timing and control. This includes the normal RAM 8 warm-up cycles, various refresh cycles and frequencies, address multiplexing, and address strobe timing. The 8207's system interface and RAM timing and control are programmable to permit it to be used in most applications.

Integrating all of the above functions (plus a dual port and error correcting interfaces) allows the user to realize significant cost savings over discrete logic. For example, comparing the 8207 to the iSBC012B 512K byte RAM board (where the DRAM control is done entirely with TTL), an 8207 design saved board space 3 in^2 vs 10 in^2); required less power (420 mA vs 1220 mA); and generated less heat. Moreover, design time was reduced, and increased margins were achieved due to less skewing of critical timings. This comparison is based on a single port design and did not include the 8207's RAM warm-up, dual-port and error correcting features. If these features were fully implemented, there would be no change to the 8207 figures, listed above, while the TTL figures would easily double.

This Application Note will illustrate an iAPX design with the 8207 controlling the dynamic RAM array. The reader should be familiar with the 82097 data sheet, the 80186 data sheet, and a RAM data sheet*.

DESIGN GOALS

The main objective of this design is for the 80186 to run with no wait states with a Dynamic RAM array. The design uses one port of the 8207. The dual port and error correcting interfaces of the 8207 are covered in separate Application Notes.

The size of the RAM array is 4 banks of 64K RAMs or 512K bytes. The memory is to be interfaced locally to the 80186.

USING THE 8207

The three areas to be considered when designing in the 8207 are:

- 8207 programming logic
- Microprocessor interface
- RAM array

8207 Programming

The 8207 requires up to two 74LS165 shift registers for programming. This design needs one 8 bit shift register, as shown in Figure 1. The 16 bits in the Program Data Word are set as shown in Figure 2. Refresh is done internally, so the REFRQ input must be tied high. The memory commands are iAPX 86 status, so the timing

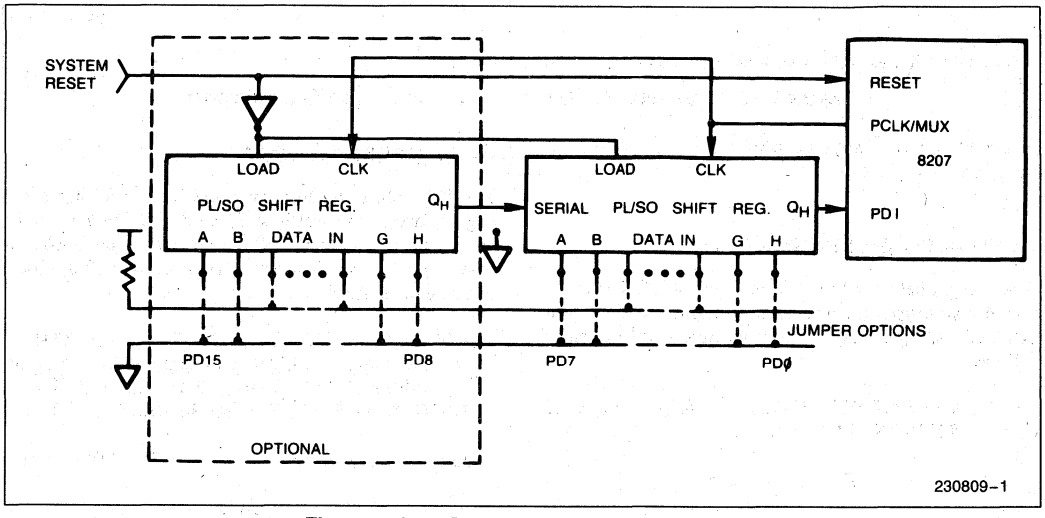
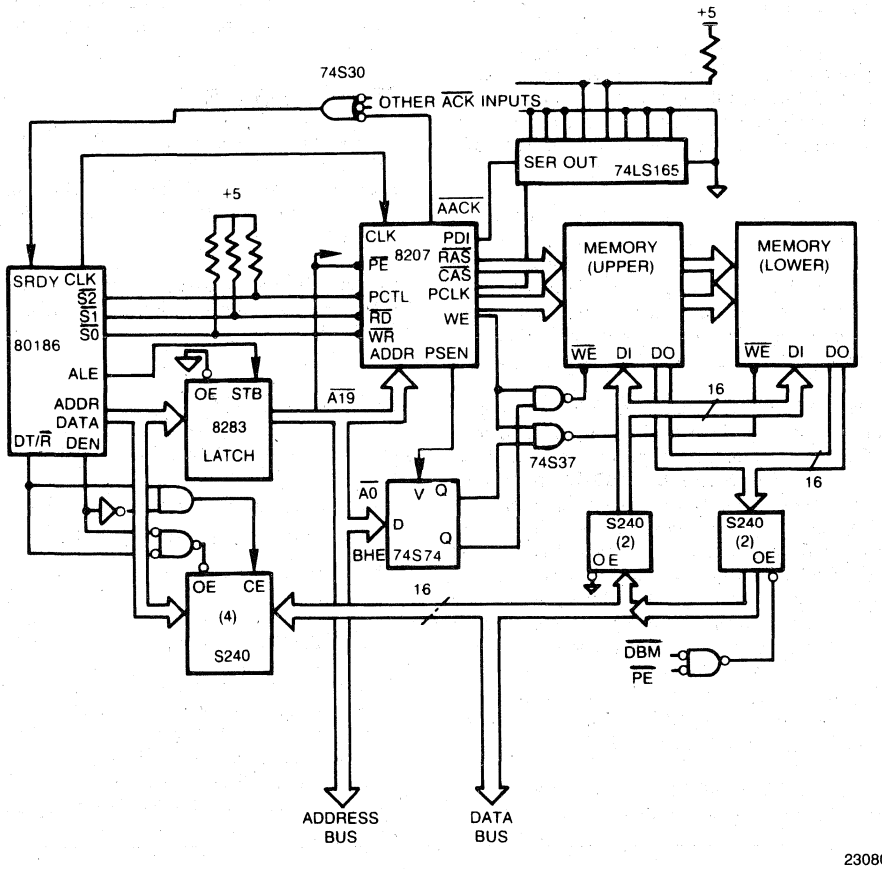


Figure 1. 8207 Programming Shift Registers

*All RAM references in this Application Note are based on Intel's 2164A 64K Dynamic RAM.



NOTE:
The 8207 requires series resistors on all outputs to RAM.

Figure 3. 80186 to 8207, non-ECC, synchronous system single port

of EAACK will always guarantee 2 clocks of address hold time from RAS.

Acknowledge Setup Time

The margin between the 8207 issuing EAACK and the 80186 ready input for no wait states minus delays from clock edges, logic delays, and setup time is calculated as follows.

1 clock − 8207 TCLAKL max − 74S30 t_{PLH} @ 15 pf − 80186 TSRYCL ≥ 0

125 ns − 35 − 22 − 35 = 32 ns

Read Access Margin

The 8207 starts a memory cycle on the falling clock edge between the 80186's T1 and T2. Data must be valid within 2 clocks. Valid data from the RAMs is based upon the CAS access period minus buffer, clock, setup requirements.

2 TCLCL − 8207 TCLCSL @ 150 pF (t34) − DRAM tCAC − 74S240 propagation delay @ 50 pF − additional bus loading delay (250 pF)(1) − 74S240 delay @ 50 pF − 80186 TDVCL ≥ 0

250 ns − 122 − 85 − 7 − 7 − 7 − 20 = 2 ns

NOTE:
(1) 74STTL logic derated by 0.05 ns/pF. 74STTL buffers (240, 37) derated by 0.025 ns/pF.

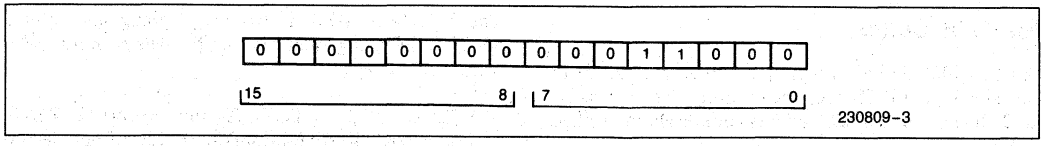


Figure 2. Program Data Word

Write Data Setup and Hold Margin

Data from the processor must be valid when WE is issued by the 8207 to meet the RAM specification tDS (2164A = 0 ns), and then held for a minimum of 30 ns.

The PCTLA input must be high when RESET goes inactive.

The differential reset circuit shown in the Data Sheet is necessary only to ensure that memory commands are not received by the 8207 when Port A is changed from synchronous to asynchronous (vice versa for Port B). This design keeps Port A synchronous so no differential reset circuit is needed.

Microprocessor Interface

To achieve no wait states, the 8207 must connect directly to the microprocessor's CLKOUT and status lines. The 8207 Acknowledge (EAACK) must connect to the SRDY input of the 80186.

When the 80186 is reset, it tristates the status lines. The 8207 PCTLA input requires a high to decode the proper memory commands. This is accomplished by using a pull-up resistor or some component that incorporates a pull-up on S2.

The 8207 address inputs are connected directly to the latched/demultiplexed address bus.

RAM Array

The 8207 provides complete control of all RAM timings, warm up cycles, and refresh cycles. All write cycles are "late writes." During write cycles, the data out lines go active. This requires separate data in/out lines in the RAM array.

To operate the 80186 with no wait states, it is necessary to choose sufficiently fast DRAMs. The 150 ns version of the 2164A allows operating the 80186 at 8 MHz, and the 200 ns version up to 7 MHz.

HARDWARE DESIGN

Figure 3 shows a block diagram of the design, and Figure 4 is a timing diagram showing the relationship between the 8207 and the 80186.

8207 Command Setup

Two events must occur for a command to be recognized by the 8207. The 80186 status outputs are sampled by a rising clock edge and Port Enable (PE) is sampled by the next falling edge (refer to the Data Sheet wave forms).

The command timing is determined by the period between the status being issued and the first rising clock edge of the 8207, minus setup and delays.

80186 status valid to 8207 rising clock − status from clock delay − 8207 setup to clock ≥ 0

1 TCLCL − 80186 TCHSV max − 8207 TKVCH min ≥ 0

125 ns − 55 − 20 = 50 ns

PE is a chip select for a valid address range. It can be generated from the address bus or from the 80186's programmable memory selects. This design uses an inverted A19. The timing is determined by the interval between the address becoming valid and the falling clock edge, minus setup and delays.

80186 address valid to 8207 falling clock edge − 80186 address from clock delay − 8283 latch delays − 8207 PE setup ≥ 0

1 TCLCL − 80186 TCLAV max − 8283 IVOV @ 300 pF − 8207 TPEVCL ≥ 0

125 ns − 44 − 22 − 30 − 29 ns

The hold times are 0 ns and are met.

Address Setup

For an 81086 design, the 8207 requires the address to be stable before RAS goes active, and to remain stable for 2 clocks. Unused 8207 address inputs should be tied to V_{CC}.

tASR is a RAM specification. If it is greater than zero, this must be added to the address setup time of the 807. Address setup is the interval between addresses being issued and RAS going active, minus appropriate delays.

80186 address valid to 8207 RAS active − 80186 address from clock delay − bus delays − (8207 setup + RAM t_{ASR}) ≥ 0

TCLCL + 8207 TCLRSL min @ 150 pf(1) − 80186 TCLAV max − 8283 IVOV max @ 300 pf − (8207 TAVCL min + DRAM t_{ASR}) ≥ 0

125 ns + 0 − 44 − 22 − (35 + 0) = 24 ns

The address hold time of 2 clocks + 0 ns is always met, since the addresses are latched by the 8282/3. Even when the processor is in wait states (for refresh),

NOTE:
(1) Not specified—use 0 ns.

TCLCL + TCLCH + 8207 TCLW min(1) + 74S37 delay tPHL min @ 50 pf + additional loading (142 pf) − 81086 TCVCTV − 74S240tPZL − bus delays (250 pf) − 74S240 delay − 2164A tDS ≥ 0

125 + 62.5 + 0 + 6.5 + 3.5 − 70 − 15 − 7 − 7 − 0 = 98.5 ns.

The hold time, tDH, is from WE going low to the 80186 DEN going high plus buffer delays minus WE from clock delays.

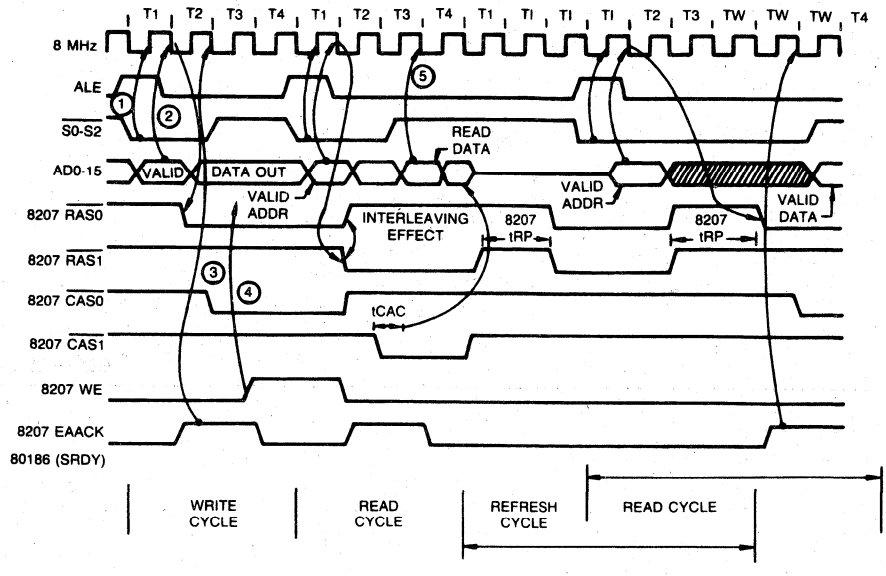
TCLCL − 80186 TCVCTX min + 74S32 tPD(2) min + 74S240 tPHZ (min)(2) + 250 pf bus delays + 74S240 propagation delay min − 8207 TCLW max − 74S37 tPHL @ 50 pf − 142 pf loading delays − DRAM tDH ≥ 0

62.5 ns + 10 + 2 + 3 + 7 + 3.5 − 35 − 3.5 − 30 = 19.5 ns

All margins are actually better by about 10–20 ns. No improvement in timing was allowed for lower capacitive loads when additional buffers are used (i.e. the 80186 address out delay is at 200 pf, but the 8283 latch only loads these lines with about 20 pf).

SUMMARY

The 8207 supports the 80186 microprocessor running with no wait states. The 8207 interfaces easily between the microprocessor and dynamic RAM. There are no difficult timings to be resolved by the designer using external logic.



NOTES:
1. Command Setup Margin
2. PE Setup Margin
3. EAACK Setup Margin
4. Data Setup Margin
5. Read Access Margin

Figure 4. 8207/80186 Timing Relationship

NOTE:
1. Not specified, use 0 ns.
2. Not specified, use one half of typical value.

intel®

APPLICATION NOTE

AP-168

September 1989

Interfacing the 8207 Advanced Dynamic RAM Controller to the iAPX 286

Order Number: 230862-001

INTERFACING THE 8207 ADVANCED DYNAMIC RAM CONTROLLER TO THE iAPX 286

CONTENTS

PAGE

INTRODUCTION 2-108

DESIGN GOALS 2-108

8207 INTERFACE 2-108
Programming the 8207 2-108
RAM Array 2-109
Microprocessor Array 2-109

DESIGNING THE HARDWARE 2-109
8207 Command Setup Margin 2-109
Address Setup Margin 2-112
Acknowledge Setup Margin 2-112
Read Access Margin 2-112
RAS Access Margin 2-113
CAS Access Margin 2-113
Write Data Setup and Hold Margin 2-113

SUMMARY 2-113

INTRODUCTION

The 80286 high speed microprocessor pushes microprocessor based systems to new performance levels. However, its high speed bus requires special design considerations to utilize that performance. Interfacing the 80286 to a dynamic RAM array require many timings to be analyzed, refresh cycle effects on bus timing examined, minimum and maximum signal widths noted, and the list continues.

The 8207 Advanced Dynamic RAM Controller was specifically designed to solve all interfacing issues for the 80286, provide complete control and timing for the DRAM array, plus achieve optimum system performance. This includes the normal RAM 8 warmup cycles, various refresh cycles and frequencies, address multiplexing, and address strobe timings. The 8207 Dynamic RAM Controller's system interface and RAM timing and control are programmable to permit it to be used in most applications.

Integrating these functions (plus dual port and error correcting interfaces) allows the user to realize significant savings in both engineering design time, PC board space and product cost. For example, in comparing the 8207 to the ISBC012B 512k byte RAM board (where the DRAM timing and control is done entirely with TTL), the 8207 design saved board space (3 in^2 vs 10 in^2); used less power (420 mA vs 1220 mA); reduced the design time; and increased margins due to less skewing of timings. The comparison is based upon a single port 8207 design and does not include its RAM warm-up, dual port, error correcting, and error scrubbing or RAM interleaving features.

This Application Note will detail an 80286 and 8207 design. The reader should have read the 8207 and the 80286 data sheets, a DRAM data sheet*, and have them available for reference.

DESIGN GOALS

The main objective of this design is to run the RAM array without wait states, to maximize the 80286's performance, and to use as little board space as possible. The 80286 will interface synchronously to Port A of the 8207 and the 8207 will control 512k bytes of RAM (4 banks using 64k DRAMs). The dual port and error correcting features of the 8207 are covered in separate Application Notes.

8207 INTERFACE

The 8207 Memory design can be subdivided into three sections:

- Programming the 8207.
- The 82086/8207 interface.
- The Dynamic RAM array.

Programming the 8207

The RAM timing is configured via the 16 bit program word that the 8207 shifts-in when reset. This can require two 74LS165 shift registers to provide complete DRAM configurability. The 8207 defaults to the configuration shown in Table 1 when PDI is connected to ground. This design does not need the flexibility the shift registers would allow since standard 8207/80286 clock frequencies, DRAM speeds and refresh rates are used. Table 1 details the 8207/80286 configuration and Table 10 in the Data Sheet identifies "CO" as the configuration of the 8207 all timings will be referenced to (80286 mode at 16 MHz using fast RAMs = CO).

Table 1. Default Non-ECC Programming, PD1 Pin (57) Tied to Ground

Port A is Synchronous (EAACKA and XACKA)
Port B is Asynchronous (LAACKB and XACKB)
Fast-cycle Processor Interface (10 or 16 MHz)
Fast RAM 100/120 ns RAM
Refresh Interval uses 236 clocks
128 Row refresh in 2 ms; 256 Row refresh in 4 ms
Fast Processor Clock Frequency (16 MHz)
"Most Recently Used" Priority Scheme
4 RAM banks occupied

The 8207 will accept 80286 status inputs when the PCTLA pin is sampled low at reset. This pin is not necessary for an 80286 design (besides programming) and is tied to ground.

Refresh is the final option to be programmed. If the Refresh pin is sampled high at reset, an internal timer

*All RAM references in this Application Note are based upon Intel's CMOS 51C64-12 64k Dynamic RAM. Any DRAM with similar timings will function. Refer to section 4.4.

is enabled, and if low at reset, this timer is disabled. The first method is the easiest to implement, so the RFRQ pin is tied to V_{CC}.

The differential reset circuit shown in the Data Sheet is necessary only to ensure that memory commands are not received by the 8207 when Port A is changed from synchronous to asynchronous (vice versa for Port B). This design keeps Port A synchronous so no differential reset circuit is needed.

RAM Array

The 8207 completely controls all RAM timings, warm-up cycles, and refresh cycles. To determine if a particular RAM will work with the 8207, calculate the margins provided by the 8207 (Table 15, 16—8207 Data Sheet) and ensure they are greater than the RAM requirement. An additional consideration is the access times of the RAMs. The access time of the system is dependent upon the number of data buffers between the 80286 and the DRAMs. To operate the 80286 at zero wait states requires access times of 100–120 ns. Slower RAMs can be used (150 ns) by either adding a wait state (programming the 8207 for "C1") or reducing the clock frequency (to 14.9 MHz approximately and maintaining the CO configuration).

All write cycles are "late writes" and the data out lines of the RAM will go active. This will require separate data in and out lines in the RAM array. Another consideration for the RAM array is the proper layout of the RAM, and impedance matching resistors on the 8207 outputs. Proper layout is covered in Intel's RAM Data Sheets and Application Notes.

Microprocessor Array

To achieve no wait state operation, the 8207's clock input must be connected to the 80286's clock input. The EAACK (early acknowledge) output of the 8207 must connect to the SRDY input of the 82284. The 8207's address inputs connect directly to the 80286 address outputs and the addresses are latched internally. This latch is strobed by an internal signal with the same timing as LEN (which is for dual port 82086 designs). Figure 2 shows the timing relationship between LEN and the 80286.

LEN will fall from high to low, which latches the bus address internally, when a valid command is received. LEN can go high in two clock cycles if the RAM cycle started (RAS going low) at the same time LEN went low. If the 8207 is doing a refresh cycle, the 80286 will be put into wait states until the memory cycle can start. LEN will then go high two clocks after RAS starts, since addresses are no longer needed for the current RAM cycle. Thus the low period of LEN could be much longer than listed in the Data Sheet.

DESIGNING THE HARDWARE

Figure 1 shows a detailed block diagram of the design and Figure 2 shows the timing relationship between the 8207 and the 80286.

The following analysis of six parameters will confirm that the design will work. These six system parameters are generally considered the most important in any microprocessor—Dynamic RAM design.

2

8207 Command Setup Margin

Two events must occur for the 8207 to start a memory cycle. Either RD or WR active (low) and PE must be low when the 8207 samples these pins on a falling clock edge. If PE is not valid at the same clock edge that samples RD or WR active, the memory cycle will be aborted and no acknowledge will be issued.

The command setup time is based upon the status being valid at the first falling clock edge.

80286 status valid to 8207 falling clock − 80286 status from clock delay − 8207 command setup to clock ≤ 0

TCLCL − 80286 t12 (max) − 8207 TKVCL (min) ≤ 0

62.5 − 40 ns − 20 ns = 2.5 ns

PE is decoded from the address bus and must be set up to the same falling clock edge that recognizes the RD, WR inputs. This margin is determined from the clock edge that issues the address and the clock edge that will recognize RD or WR, minus decoding logic delays.

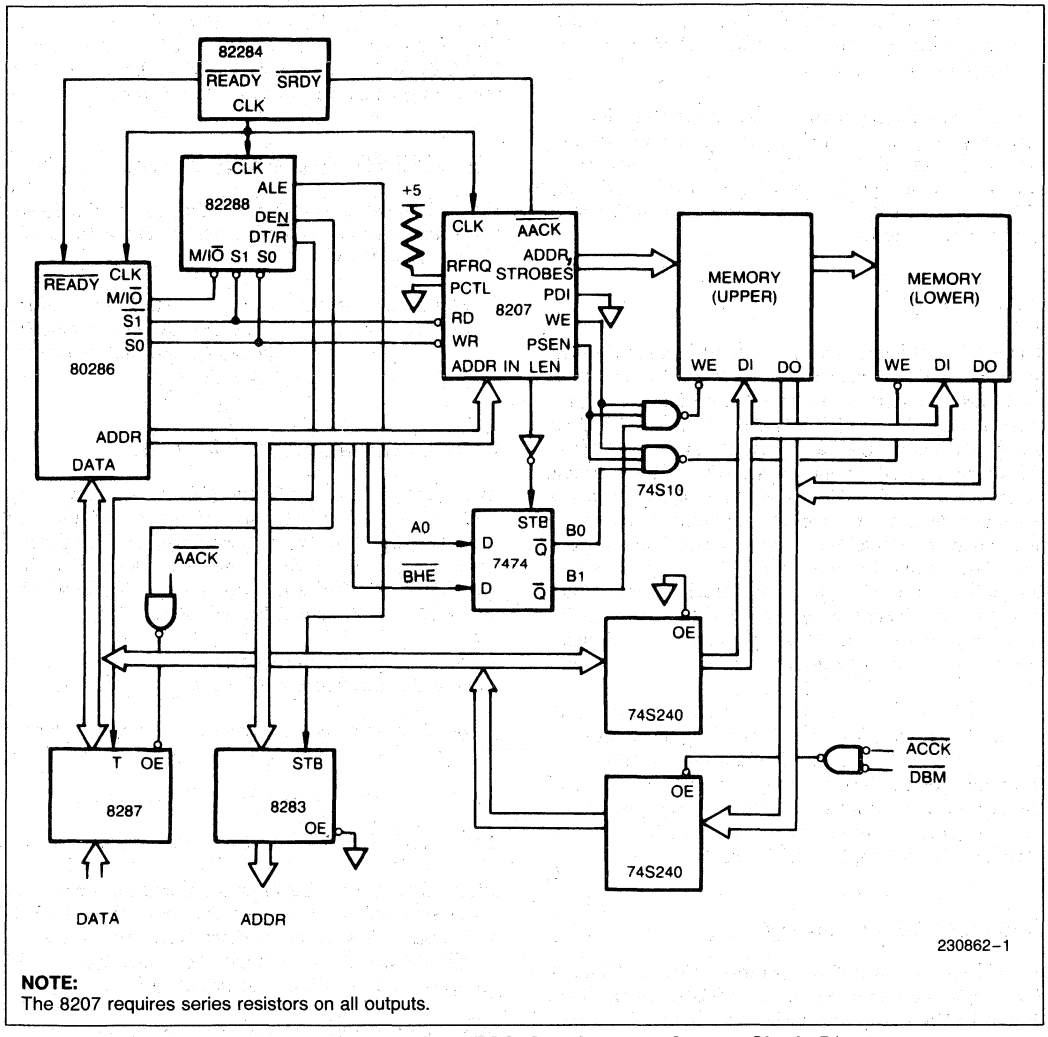
There are 2 clocks between addresses being issued by the 80286 and PE being sampled by the 8207. Then the 80286 address delay from the clock edge and decoding logic delays are subtracted from this interval. This margin must be greater than 0.

2TCLCL − 80286 t13 (max) − 8207 TPEVCL (min) ≤ 0

125 − 60 − 30 = 35 ns

The address decode logic must use no more than 35 ns (and less is better). Figure 3 shows an easy implementation which uses a maximum of 12 ns.

The 8207 requires a zero ns hold time and is always met.



NOTE:
The 8207 requires series resistors on all outputs.

Figure 1. 80286 to 8207, non-ECC, Synchronous System Single Port

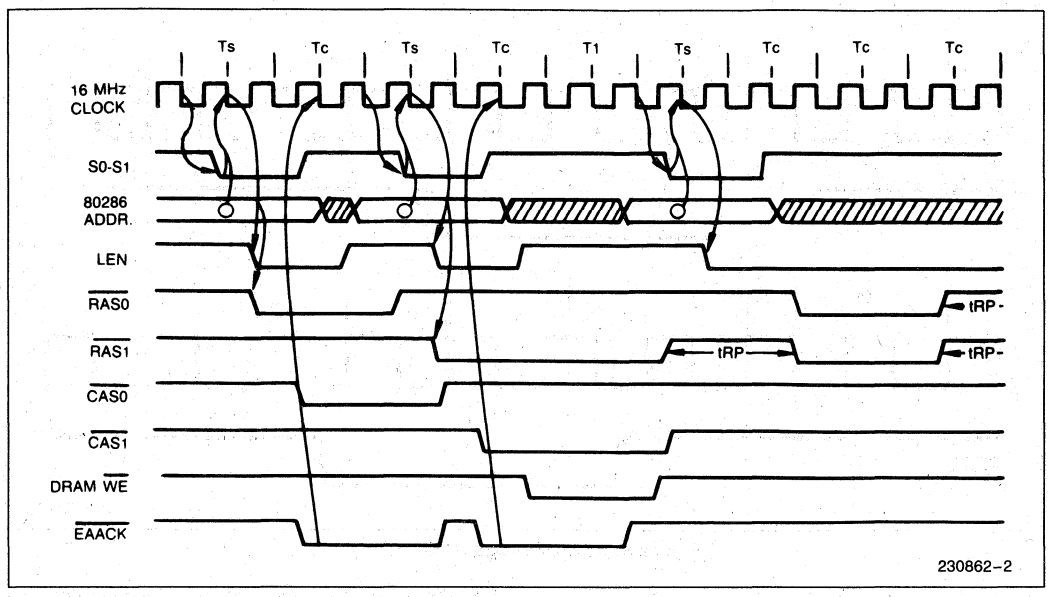


Figure 2. 80286/8207 Timing—"CO"

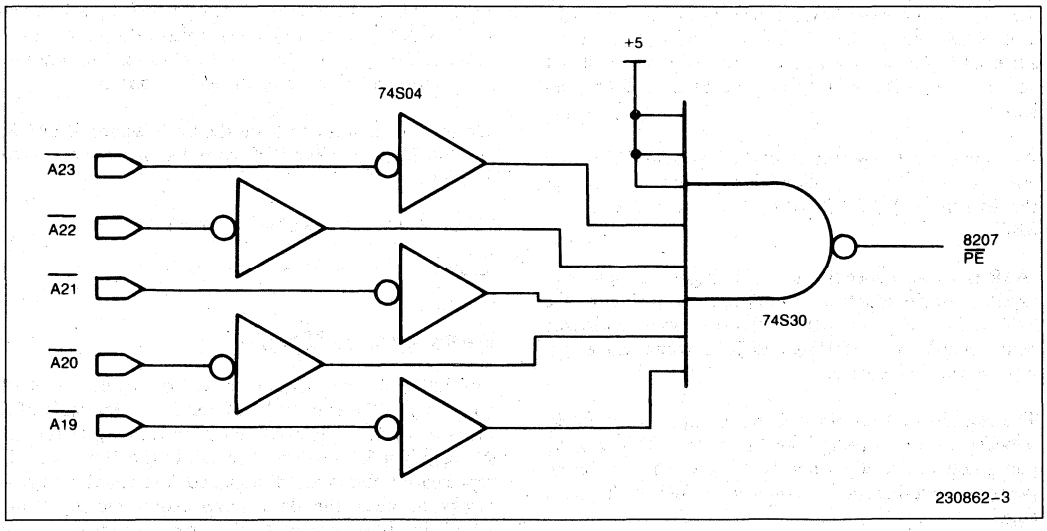


Figure 3. Address Decode Logic

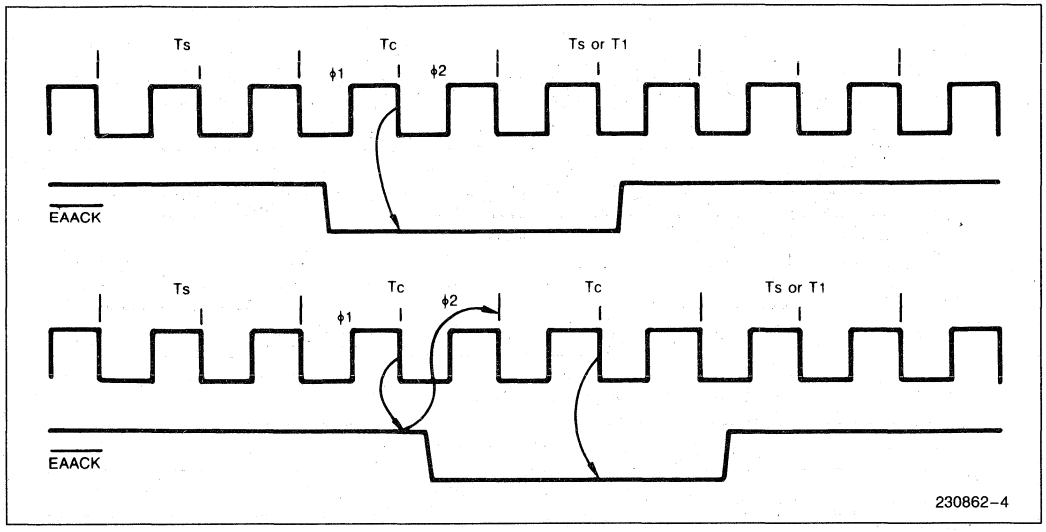


Figure 4. Acknowledge to the 82284

Address Setup Margin

The 8207 must have stable addresses up to two clocks after RAS goes active. This is of no concern to the user, since LEN latches the address internally and will not admit a new address until two clocks after RAS goes active.

Addresses must be stable at least 35 ns (tAVCL) before RAS goes active to allow for propagation delays through the 8207, if a RAM cycle is not delayed by the 8207.

tASR is a RAM specification. If it is greater than zero, tASR must be added to the address setup time of the 8207. Address setup is the interval between addresses being issued, by the 80286, and RAS going active, minus appropriate delays.

The margin is determined from the number of clocks between addresses being issued from the 80286 to RAS going active. Exactly when RAS goes active is unimportant, since here we are only interested in the clock edge.

2TCLCL − 80286 t13 (max) − 8207 TAVCL (min) ≤ 0

125 ns − 60 ns − 35 ns = 30 ns

Acknowledge Setup Margin

The 8207 acknowledge (EAACK) can be issued at any point in the 80286 bus cycle (end of $\phi 1$ or $\phi 2$ of Ts or Tc). If EAACK is issued at the end of $\phi 2$ (Ts or Tc), the 80286 will complete the current bus cycle. If EAACK is issued at the end of $\phi 1$ of Tc, the 82284 will not generate READY to the 80286 in time to end the current bus cycle. A new Tc would then be generated and EAACK would now be sampled in time to terminate the bus cycle. EAACK is 3 clocks long in order to meet setup and hold times for either condition.

We need the margin between the 8207 issuing EAACK and the 82284 needing it. Figure 4 shows a worst case example.

TCLCL − 8207 TCLAKL max − 82284 t11 ≤ 0

62.5 ns − 35 ns − 15 ns = 12.5 ns

Read Access Margin

The 8207 will typically start a memory cycle (i.e. RAS goes low) at the end of $\phi 1$ of Ts. But if the start of a memory cycle is delayed (by a refresh cycle for instance), then RAS will be delayed. In the first case, this represents 3 clocks and the second case could require 4 clocks to meet the data setup requirements of the 80286. In either case, data must be valid at the end of Tc. The 8207 holds CAS active long enough to ensure valid data is received by the 80286 in either case.

DRAMs specify two access times, RAS access (tRAC) and CAS access (tCAC) Both access periods must be calculated and the one with the least margin used. Also the number of data buffers should be kept to a minimum. Too many buffers would require either faster (more expensive) DRAMs, or a reduction in the performance of the CPU (by adding wait states).

RAS Access Margin

3TCLCL − 8207 TCLRSL max @ 150 pF − DRAM tRAC − 74S240 propagation delay max @ 50 pF − 80286 t8 ≤ 0

187.5 ns − 35 ns − 120 ns − 7 ns − 10 ns = 15.5 ns

CAS Access Margin

2TCLCL − 820-7 TCLCSL max @ 150 pF − DRAM tCAA (or tCAC − 74S240 tplh max @ 50 pF − 80286 t8) ≤ 0

125 ns − 35 ns − 60 ns − 7 ns − 10 ns = 13 ns

By solving each equation for tRAC and tCAC, the speed requirement of the RAM can be determined.

DRAM tRAC = 3 TCLCL − 8207 TCLRSL − 74S240 tplh − 82086 t8 = 135.5 ns

DRAM tCAC = 2 TCLCL − 8207 TCLCSL − 74S240 tplh − 80286 t8 = 73 ns

NOTES:
1. Not specified. Assume no delay for worst case analysis.
2. STTL derated by 0.05 ns/pF.

So any DRAM that has a RAS access period less than 135 ns, a CAS access period less than 73 ns, and meets all requirements in the DRAM Interface Timing (Table 15, 16—8207 Data Sheet), will work.

Write Data Setup and Hold Margin

Write data from the processor must be valid when the 8207 issues WE to meet the DRAM specification tDS and then held to meet the tDH requirement. Some write cycles will be byte writes and the information to determine which byte is decoded from A0 and BHE/. Since the 80286's address bus is pipelined, these two signals can change before the RAM cycle starts, hence they must be latched by LEN. PSEN is used in the WE term to shorten the WE pulse. Its use is not essential.

Data must be set up to the falling edge of WE, since WE occurs after CAS. The 2 clocks between valid write data and WE going active (at the RAM's) minus propagation delays determines the margin.

2 TCLCL − 80286 t14 (max) @ 100 pF − 74S240 tplh + 8207 TCLW (min)[1] + 74S10 tphl @ 192 pF[2] − DRAM tDS = 0

125 ns − 50 ns − 7 ns + 0 ns + 14 ns − 0 ns = 82 ns

The timing of the 8207's acknowledge is such that data will be kept valid by the 80286, for more than two clocks after WE goes active. This easily meets all RAM tDH specifications.

SUMMARY

The 8207 complements the 80286's performance and high integration with its own performance, integration and ease of use. No critical timings or logic design has been left to the designer. The 80286/8207 combination allows users to realize maximum performance from their simpler design.

Support Peripherals 3

8231A
ARITHMETIC PROCESSING UNIT

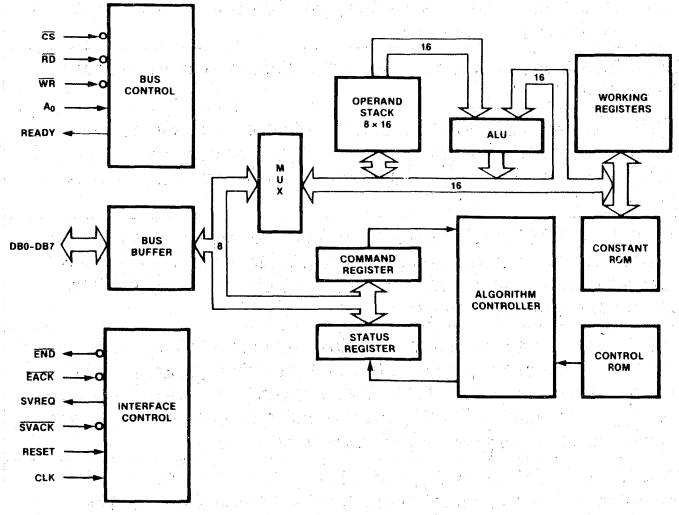
- **Fixed Point Single and Double Precision (16/32 Bit)**
- **Floating Point Single Precision (32 Bit)**
- **Binary Data Formats**
- **Add, Subtract, Multiply and Divide**
- **Trignometric and Inverse Trigonometric Functions**
- **Square Roots, Logarithms, Exponentiation**
- **Float to Fixed and Fixed to Float Conversions**
- **Stack Oriented Operand Storage**
- **Compatible with all Intel and most other Microprocessor Families**
- **Direct Memory Access or Programmed I/O Data Transfers**
- **End of Execution Signal**
- **General Purpose 8-Bit Data Bus Interface**
- **Standard 24 Pin Package**
- **+12V and +5V Power Supplies**
- **Advanced N-Channel Silicon Gate HMOS Technology**

The Intel® 8231A Arithmetic Processing Unit (APU) is a monolithic HMOS LSI device that provides high performance fixed and floating point arithmetic and floating point trigonometric operations. It may be used to enhance the mathematical capability of a wide variety of processor-oriented systems. Chebyshev polynomials are used in the implementation of the APU algorithms.

3

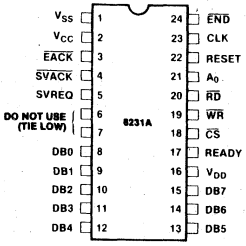
All transfers, including operand, result, status and command information, take place over an 8-bit bidirectional data bus. Operands are pushed onto an internal stack and commands are issued to perform operations on the data and the stack. Results are then available to be retrieved from the stack.

Transfers to and from the APU may be handled by the associated processor using conventional programmed I/O, or may be handled by a direct memory access controller for improved performance. Upon completion of each command, the APU issues an end of execution signal that may be used as an interrupt by the CPU to help coordinate program execution.



231305-1

Figure 1. Block Diagram



231305-2

Figure 2. Pin Configuration

September 1987
Order Number: 231305-002

Table 1. Pin Description

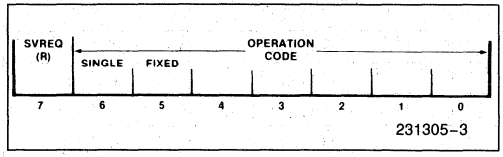
Symbol	Pin No.	Type	Name and Function
V_{CC}	2		**POWER:** +5V power supply.
V_{DD}	16		**POWER:** +12V power supply.
V_{SS}	1		**GROUND.**
CLK	23	I	**CLOCK:** An external, TTL compatible, timing source is applied to the CLK pin.
RESET	22	I	**RESET:** The active high reset signal provides initialization for the chip. RESET also terminates any operation in progress. RESET clears the status register and places the 8231A into the idle state. Stack contents and command registers are not affected (5 clock cycles).
$\overline{CS}$	18	I	**CHIP SELECT:** $\overline{CS}$ is an active low input signal which selects the 8231A and enables communication with the data bus.
A_0	21	I	**ADDRESS:** In conjunction with the $\overline{RD}$ and WR signals, the A_0 control line establishes the type of communication that is to be performed with the 8231A as shown below:

A_0	$\overline{RD}$	$\overline{WR}$	Function
0	1	0	Enter data byte into stack
0	0	1	Read data byte from stack
1	1	0	Enter command
1	0	1	Read status

$\overline{RD}$	20	I	**READ:** This active low input indicates that data or status is to be read from the 8231A if CS is low.
$\overline{WR}$	19	I	**WRITE:** This active low input indicates that data or a command is to be written into the 8231A if CS is low.
$\overline{EACK}$	3	I	**END OF EXECUTION:** This active low input clears the end of execution output signal ($\overline{END}$. If $\overline{EACK}$ is tied low, the $\overline{END}$ output will be a pulse that is one clock period wide.
$\overline{SVACK}$	4	I	**SERVICE REQUEST:** This active low input clears the service request output (SVREQ).
$\overline{END}$	24	O	**END:** This active low, open-drain output indicates that execution of the previously entered command is complete. It can be used as an interrupt request and is cleared by $\overline{EACK}$, RESET or any read or write access to the 8231.
SVREQ	5	O	**SERVICE REQUEST:** This active high output signal indicates that command execution is complete and that post execution service was requested in the previous command byte. It is cleared by $\overline{SVACK}$, the next command output to the device, or by RESET.
READY	17	O	**READY:** This active high output indicates that the 8231A is able to accept communication with the data bus. When an attempt is made to read data, write data or to enter a new command while the 8231A is executing a command, READY goes low until execution of the current command is complete (See READY Operation, p. 6).
DB0–DB7	8-15	I/O	**DATA BUS:** These eight bidirectional lines provide for transfer of commands, status and data between the 8231A and the CPU. The 8231A can drive the data bus only when CS and RD are low.

COMMAND STRUCTURE

Each command entered into the 8231A consists of a single 8-bit byte having the format illustrated below:



Bits 0–4 select the operation to be performed as shown in the table. Bits 5–6 select the data format appropriate to the selected operation. If bit 5 is a 1, a fixed point data format is specified. If bit 5 is 0, floating point format is specified. Bit 6 selects the precision of the data to be operated upon by fixed point commands only (if bit 5 = 0, bit 6 must be 0). If bit 6 is a 1, single-precision (16-bit) operands are assumed. If bit 6 is a 0, double-precision (32-bit) operands are indicated. Results are undefined for all illegal combinations of bits in the command byte. Bit 7 indicates whether a service request is to be issued after the command is executed. If bit 7 is a 1, the service request output (SVREQ) will go high at the conclusion of the command and will remain high until reset by a low level on the service acknowledge pin ($\overline{SVACK}$) or until completion of execution of the succeeding command where service request (bit 7) is 0. Each command issued to the 8231A requests post execution service based upon the state of bit 7 in the command byte. When bit 7 is a 0, SVREQ remains low.

Table 2. 32-Bit Floating Point Instructions

Instruction	Description	Hex(1) Code	Stack Contents(2) After Execution A B C D	Status Flags(4) Affected
ACOS	Inverse Cosine of A	06	R U U U	S, Z, E
ASIN	Inverse Sine of A	05	R U U U	S, Z, E
ATAN	Inverse Tangent of A	07	R B U U	S, Z
CHSF	Sign Change of A	15	R B C D	S, Z
COS	Cosine of A (radians)	03	R B U U	S, Z
EXP	e^A Function	0A	R B U U	S, Z, E
FADD	Add A and B	10	R C D U	S, Z, E
FDIV	Divide B by A	13	R C D U	S, Z, E
FLTD	32-Bit Integer to Floating Point Conversion	1C	R B C U	S, Z
FLTS	16-Bit Integer to Floating Point Conversion	1D	R B C U	S, Z
FMUL	Multiply A and B	12	R C D U	S, Z, E
FSUB	Subtract A from B	11	R C D U	S, Z, E
LOG	Common Logarithm (base 10) of A	08	R B U U	S, Z, E
LN	Natural Logarithm of A	09	R B U U	S, Z, E
POPF	Stack Pop	18	B C D A	S, Z
PTOF	Stack Push	17	A A B C	S, Z
PUPI	Push π onto Stack	1A	R A B C	S, Z
PWR	B^A Power Function	0B	R C U U	S, Z, E
SIN	Sine of A (radians)	02	R B U U	S, Z
SQRT	Square Root of A	01	R B C U	S, Z, E
TAN	Tangent of A (radians)	04	R B U U	S, Z, E
XCHF	Exchange A and B	19	B A C D	S, Z

3

Table 3. 32-Bit Integer Instructions

Instruction	Description	Hex(1) Code	Stack Contents(2) After Execution A B C D	Status Flags(4) Affected
CHSD	Sign Change of A	3 4	R B C D	S, Z, O
DADD	Add A and B	2 C	R C D A	S, Z, C, E
DDIV	Divide B by A	2 F	R C D U	S, Z, E
DMUL	Multiply A and B (R = lower 32-bits)	2 E	R C D U	S, Z, O
DMUU	Multiply A and B (R = upper 32-bits)	3 6	R C D U	S, Z, O
DSUB	Subtract A from B	2 D	R C D A	S, Z, C, O
FIXD	Floating Point to Integer Conversion	1 E	R B C U	S, Z, O
POPD	Stack Pop	3 8	B C D A	S, Z
PTOD	Stack Push	3 7	A A B C	S, Z
XCHD	Exchange A and B	3 9	B A C D	S, Z

Table 4. 16-Bit Integer Instructions

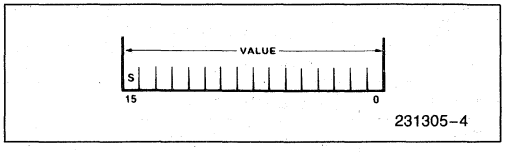
Instruction	Description	Hex(1) Code	Stack Contents(3) After Execution A_U A_L B_U B_L C_U C_L D_U D_L	Status Flags(4) Affected
CHSS	Change Sign of A_U	7 4	R A_L B_U B_L C_U C_L D_U D_L	S, Z, O
FIXS	Floating Point to Integer Conversion	1 F	R B_U B_L C_U C_L U U U	S, Z, O
POPS	Stack Pop	7 8	A_L B_U B_L C_U C_L D_U D_L A_U	S, Z
PTOS	Stack Push	7 7	A_U A_U A_L B_U B_L C_U C_L D_U	S, Z
SADD	Add A_U and A_L	6 C	R B_U B_L C_U C_L D_U D_L A_U	S, Z, C, E
SDIV	Divide A_L by A_U	6 F	R B_U B_L C_U C_L D_U D_L U	S, Z, E
SMUL	Multiply A_L by A_U (R = lower 16-bits)	6 E	R B_U B_L C_U C_L D_U D_L U	S, Z, E
SMUU	Multiply A_L by A_U (R = upper 16-bits)	7 6	R B_U B_L C_U C_L D_U D_L U	S, Z, E
SSUB	Subtract A_U from A_L	6 D	R B_U B_L C_U C_L D_U D_L A_U	S, Z, C, E
XCHS	Exchange A_U and A_L	7 9	A_L A_L B_U B_L C_U C_L D_U D_L	S, Z
NOP	No Operation	0 0	A_U A_L B_U B_L C_U C_L D_U D_L	

NOTES:
1. In the hex code column, SVREQ is a 0.
2. The stack initially is composed of four 32-bit numbers (A, B, C, D). A is equivalent to Top Of Stack (TOS) and B is Next On Stack (NOS). Upon completion of a command the stack is composed of: the result (R); undefined (U); or the initial contents (A, B, C, or D).
3. The stack initially is composed of eight 16-bit numbers (A_U, A_L, B_U, B_L, C_U, C_L, D_U, D_L). A_U is the TOS and A_L is NOS. Upon completion of a command the stack is composed of: the result (R); undefined (U); or the initial contents (A_U, A_L, B_U, B_L . . .).
4. Nomenclature: Sign (S); Zero (Z); Overflow (O); Carry (C); Error Code Field (E).

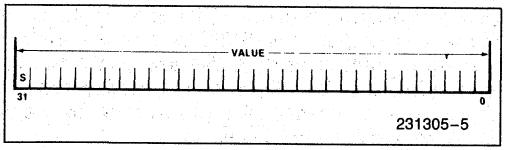
DATA FORMATS

The 8231A arithmetic processing unit handles operands in both fixed point and floating point formats. Fixed point operands may be represented in either single (16-bit operands) or double precision (32-bit operands), and are always represented as binary, two's complement values.

Single Precision Fixed Point Format



Double Precision Fixed Point Format



The sign (positive or negative) of the operand is located in the most significant bit (MSB). Positive values are represented by a sign bit of zero (S = 0). Negative values are represented by the two's complement of the corresponding positive value with a sign bit equal to 1 (S = 1). The range of values that may be accommodated by each of these formats is −32,768 to +32,767 for single precision and −2,147,483,648 to +2,147,483,647 for double precision.

Floating point binary values are represented in a format that permits arithmetic to be performed in a fashion analogous to operations with decimal values expressed in scientific notation.

$$(5.83 \times 10^{2})\ (8.16 \times 10^{1}) = (4.75728 \times 10^{4})$$

In the decimal system, data may be expressed as values between 0 and 10 times 10 raised to a power that effectively shifts the implied decimal point right or left the number of places necessary to express the result in conventional form (e.g., 47,572.8). The value-portion of the data is called the mantissa. The exponent may be either negative or positive.

The concept of floating point notation has both a gain and a loss associated with it. The gain is the ability to represent the significant digits of data with values spanning a large dynamic range limited only by the capacity of the exponent field. For example, in decimal notation in the exponent field is two digits wide, and the mantissa is five digits, a range of values (positive or negative) from 1.0000×10^{-99} to $9.9999 \times 10^{+99}$ can be accommodated. The loss is that only the significant digits of the value can be represented. Thus there is no distinction in this representation between the values 123451 and 123452, for example since each would be expressed as: 1.2345×10^{5}. The sixth digit has been discarded. In most applications where the dynamic range of values to be represented in large, the loss of significance, and hence accuracy of results, is a minor consideration. For greater precision a fixed point format could be chosen, although with a loss of potential dynamic range.

The 8231A is a binary arithmetic processor and requires that floating point data be represented by a fractional mantissa value between 0.5 and 1 multiplied by 2 raised to an appropriate power. This is expressed as follows:

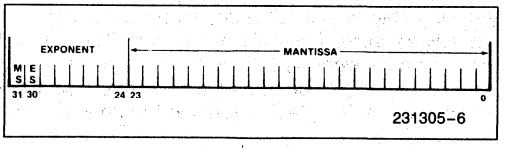
$$\text{value} = \text{mantissa} \times 2^{\text{exponent}}$$

For example, the value 100.5 expressed in this form is 0.1100 1001 $\times 2^{7}$. The decimal equivalent of this value may be computed by summing the components (powers of two) of the mantissa and then multiplying by the exponent as shown below:

$$\begin{aligned}\text{value} &= (2^{-1} + 2^{-2} + 2^{-5} + 2^{-8}) \times 2^{7} \\ &= 0.5 + 0.25 + 0.03125 + 0.00290625) \times 128 \\ &= 0.78515625 \times 128 \\ &= 100.5\end{aligned}$$

FLOATING POINT FORMAT

The format for floating point values in the 8231A is given below. The mantissa is expressed as a 24-bit (fractional) value; the exponent is expressed as a two's complement 7-bit value having a range of −64 to +63. The most significant bit is the sign of the mantissa (0 = positive, 1 = negative), for a total of 32 bits. The binary point is assumed to be the left of the most significant mantissa bit (bit 23). All floating point data values must be normalized. Bit 23 must be equal to 1, except for the value zero, which is represented by all zeros.

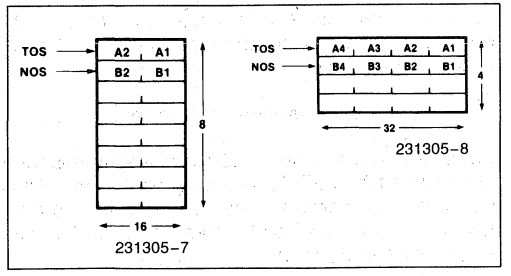


The range of values that can be represented in this format is $\pm(2.7 \times 10^{-20}$ to $9.2 \times 10^{18})$ and zero.

FUNCTIONAL DESCRIPTION

STACK CONTROL

The user interface to the 8231A includes access to an 8 level 16-bit wide data stack. Since single precision fixed point operands are 16-bits in length, eight such values may be maintained in the stack. When using double precision fixed point or floating point formats four values may be stored. The stack in these two configurations can be visualized as shown below:



Data are written onto the stack, eight bits at a time, in the order shown (A1, A2, A3, . . .). Data are removed from the stack in reverse byte order (A4, A3, A2 . . .). Data should be entered onto the stack in multiples of the number of bytes appropriate to the chosen data format.

DATA ENTRY

Data entry is accomplished by bringing the chip select ($\overline{CS}$), the command/data line (A_0), and $\overline{WR}$ low, as shown in the timing diagram. The entry of each new data word "pushes down" the previously entered data and places the new byte on the top of stack (TOS). Data on the bottom of the stack prior to a stack entry are lost.

DATA REMOVAL

Data are removed from the stack in the 8231A by bringing chip select ($\overline{CS}$), command/data (A_0), and $\overline{RD}$ low as shown in the timing diagram. The removal of each data world redefines TOS so that the next successive byte to be removed becomes TOS. Data removed from the stack rotates to the bottom of the stack.

COMMAND ENTRY

After the appropriate number of bytes of data have been entered onto the stack, a command may be issued to perform an operation on that data. Commands which require two operands for execution (e.g., add) operate on the TOS and NOS values. Single operand commands operate only on the TOS.

Commands are issued to the 8231A by bringing the chip select ($\overline{CS}$) line low, command data (A_0) line high, and $\overline{WR}$ line low as indicated by the timing diagram. After a command is issued, the CPU can continue execution of its program concurrently with the 8231A command execution.

COMMAND COMPLETION

The 8231A signals the completion of each command execution by lowering the End Execution line ($\overline{END}$). Simultaneously, the busy bit in the status register is cleared and the Service Request bit of the command register is checked. If it is a "1" the service request output level (SVREQ) is raised. $\overline{END}$ is cleared on receipt of an active low End Acknowledge ($\overline{EACK}$) pulse. Similarly, the service request line is cleared by recognition of an active low Service Acknowledge ($\overline{SVACK}$) pulse.

READY OPERATION

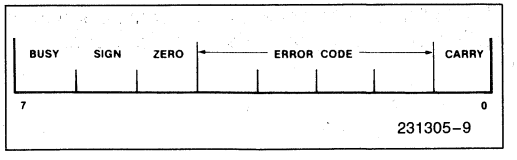
An active high ready (READY) is provided. This line is high in its quiescent state and is pulled low by the 8231A under the following conditions:

1. A previouly initiated operation is in progress (device busy) and Command Entry has been attempted. In this case, the READY line will be pulled low and remain low until completion of the current command execution. It will then go high, permitting entry of the new command.
2. A previously initiated operation is in progress and stack access has been attempted. In this case, the READY line will be pulled low, will remain in that state until execution is complete, and will then be raised to permit completion of the stack access.
3. The 8231A is not busy, and data removal has been requested. READY will be pulled low for the length of time necessary to transfer the byte from the top of stack to the interface latch, and will then go high, indicating availability of the data.
4. The 8231A is not busy, and a data entry has been requested. READY will be pulled low for the length of time required to ascertain if the preceding data byte, if any, has been written to the stack. If so READY will immediately go high. If not, READY will remain low until the interface latch is free and will then go high.
5. When a status read has been requested, READY will be pulled low for the length of time necessary to transfer the status to the interface latch, and will then be raised to permit completion of the status read. Status may be read whether or not the 8231A is busy.

When READY goes low, the APU expects the bus control signals present at the time to remain stable until READY goes high.

DEVICE STATUS

Device status is provided by means of an internal status register whose format is shown below:



Busy: Indicates that 8231A is currently executing a command (1 = Busy)

Sign: Indicates that the value on the top of stack is negative (1 = Negative)

Zero: Indicates that the value on the top of stack is zero (1 = Value is zero)

Error Code: This field contains an indication of the validity of the result of the last operation. The error codes are:

0000—No error

1000—Divide by zero

0100—Square root or log of negative number

1100—Argument of inverse sine, cosine, or e^x too large

XX10—Underflow

XX01—Overflow

Carry: Previous operation resulted in carry or borrow from most significant bit. (1 = Carry/Borrow, 0 = No Carry/No Borrow).

If the BUSY bit in the status register is a one, the other status bits are not defined; if zero, indicating not busy the operation is complete and the other status bits are defined as given above.

READ STATUS

The 8231A status register can be read by the CPU at any time (whether an operation is in progress or not) by bringing the chip select ($\overline{CS}$) low, the command/data line (A_0) high, and lowering $\overline{RD}$. The status register is then gated onto the data bus and may be input by the CPU.

EXECUTION TIMES

Timing for execution of the 8231A command set is contained below. All times are given in terms of clock cycles. Where substantial variation of execution times is possible, the minimum and maximum values are quoted; otherwise, typical values are given. Variations are data dependent.

Total execution times may require allowances for operand transfer into the APU, command execution, and result retrieval from the APU. Except for command execution, these times will be heavily influenced by the nature of the data, the control interface used, the speed of memory, the CPU used, the priority allotted to DMA and interrupt operations, the size and number of operands to be transferred, and the use of chained calculations, etc.

3

DERIVED FUNCTION DISCUSSION

Computer approximations of transcendental functions are often based on some form of polynomial equation, such as:

$$F(X) = A_0 + A_1X + A_2X^2 + A_3X^3 + A_4X^4 \ldots \quad (1\text{-}1)$$

Table 5. Command Execution Times

Command Mnemonic	Clock Cycles	Command Mnemonic	Clock Cycles	Command Mnemonic	Clock Cycles	Command Mnemonic	Clock Cycles
SADD	17	FADD	54-368	LN	4298-6956	POPF	12
SSUB	30	FSUB	70-370	EXP	3794-4878	XCHS	18
SMUL	84-94	FMUL	146-168	PWR	8290-12032	XCHD	26
SMUU	80-98						
SDIV	84-94	FDIV	154-184	NOP	4	XCHF	26
DADD	21	SORT	800	CHSS	23	PUPI	16
DSUB	38	SIN	4464	CHSD	27		
DMUL	194-210	COS	4118	CHSF	18		
DMUU	182-218						
DDIV	208	TAN	5754	PTOS	16		
FIXS	92-216	ASIN	7668	PTOD	20		
FIXD	100-346	ACOS	7734	PTOF	20		
FLTS	98-186	ATAN	6006	POPS	10		
FLTD	98-378	LOG	4474-7132	POPD	12		

The primary shortcoming of an approximation is this form is that it typically exhibits very large errors when the magnitude of |X| is large, although the errors are small when |X| is small. With polynomials in this form, the error distribution is markedly uneven over any arbitrary interval.

A set of approximating functions exists that not only minimizes the maximum error but also provides an even distribution of errors within the selected data representation interval. These are known as Chebyshev Polynomials and are based upon cosine functions. These functions are defined as follows:

$$T_n(X) = \text{Cos } n\theta; \text{ where } n = 0, 1, 2 \ldots \quad (1\text{-}2)$$

$$\theta = \text{Cos}^{-1} X$$

The various terms of the Chebyshev series can be computed as shown below:

$$T_0(X) = \text{Cos } (0 \times \theta) = \text{Cos } (0) = 1 \quad (1\text{-}4)$$

$$T_1(X) = \text{Cos } (\text{Cos}^{-1} X) = X \quad (1\text{-}5)$$

$$T_2(X) = \text{Cos } 2\theta = 2\text{Cos}^2\theta - 1 = 2\text{Cos}^2 (\text{Cos}^{-1} X) - 1 \quad (1\text{-}6)$$

$$= 2X^2 - 1$$

In general, the next term in the Chebyshev series can be recursively derived from the previous term as follows:

$$T_n(X) = 2X [T_n - 1(X)] - T_n - 2(X); n \geq 2 \quad (1\text{-}7)$$

Common logarithms are computed by multiplication of the natural logarithm by the conversion factor 0.43429448 and the error function is therefore the same as that for natural logarithm. The power function is realized by combination of natural log and exponential functions according to the equation:

$$X^Y = e^{Y \text{Ln} x}.$$

The error for the power function is a combination of that for the logarithm and exponential functions.

Each of the derived functions is an approximation of the true function. Thus the result of a derived function will have an error. The absolute error is the difference between the function's result and the true result. A more useful measure of the function's error is relative error (absolute error/true result). This gives a measurement of the significant digits of algorithm accuracy. For the derived functions except LN, LOG, and PWR the relative error is typically 4×10^{-7}. For PWR the relative error is the summation of the EXP and LN errors, 7×10^{-7}. For LN and LOG, the absolute error is 2×10^{-7}.

APPLICATION INFORMATION

The diagram in Figure 4 shows the interface connections for the APU with operand transfers handled by an 8237 DMA controller, and CPU coordination handled by an Interrupt Controller. The APU interrupts the CPU to indicate that a command has been completed. When the performance enhancements provided by the DMA and Interrupt operations are not required, the APU interface can be simplified as shown in Figure 3. The 8231A APU is designed with a general purpose 8-bit data bus and interface control so that it can be conveniently used with any general 8-bit processor.

In many systems it will be convenient to use the microcomputer system clock to drive the APU clock input. In the case of 8080A systems it would be the ϕ2TTL signal. Its cycle time will usually fall in the range of 250 ns to 1000 ns, depending on the system speed.

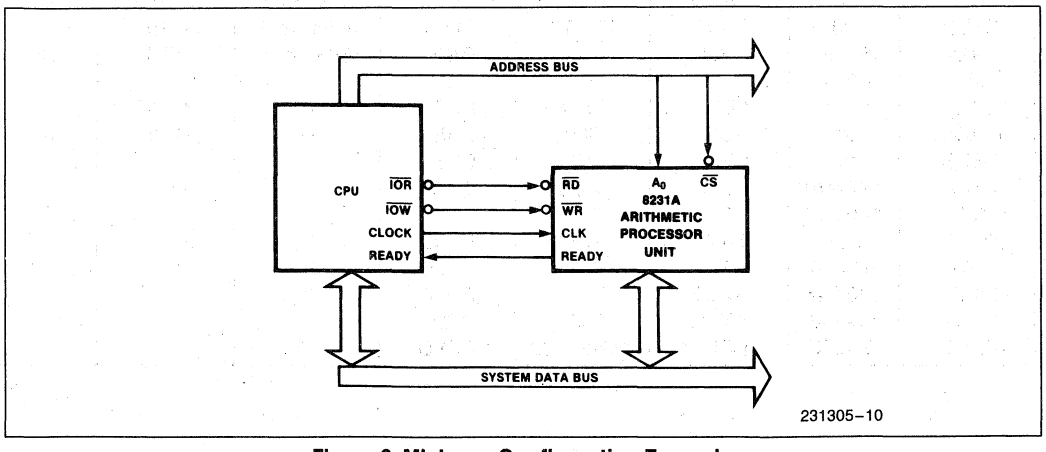


Figure 3. Minimum Configuration Example

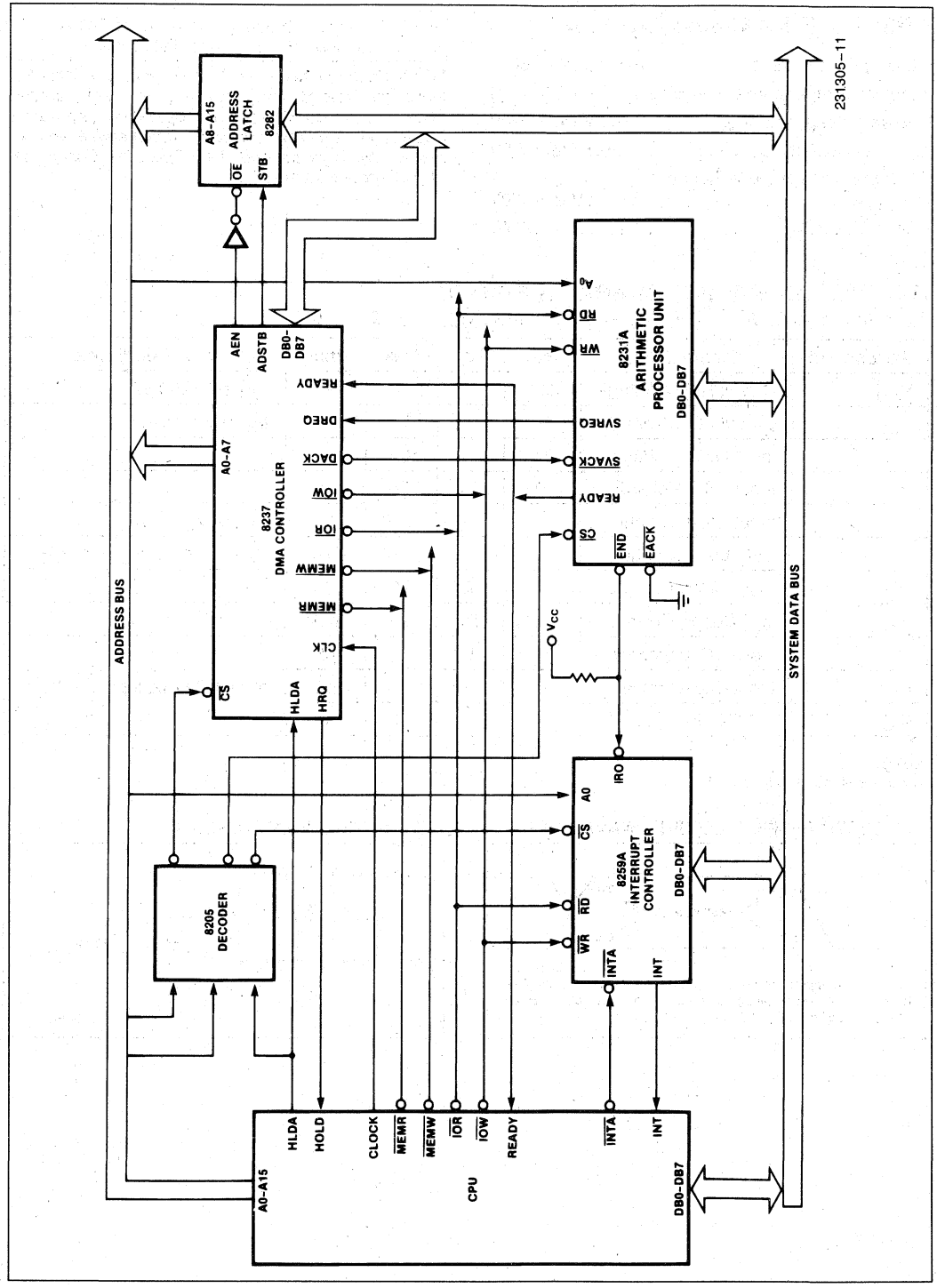


Figure 4. High Performance Configuration Example

ABSOLUTE MAXIMUM RATINGS*

Storage Temperature ..... −65°C to +150°C
Ambient Temperature Under Bias ..... 0°C to 70°C
V_{DD} with Respect to V_{SS} ..... −0.5V to +15.0V
V_{CC} with Respect to V_{SS} ..... −0.5V to +7.0V
All Signal Voltages with Respect to V_{SS} ..... −0.5V to +7.0V
Power Dissipation 2.0W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

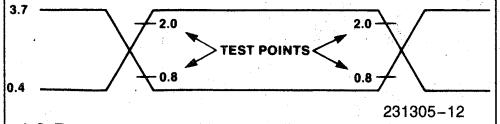
D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to 70°C, V_{SS} = 0V, V_{CC} = +5V ±10%, V_{DD} = +12V ±10%

Parameters	Description	Min	Typ	Max	Units	Test Conditions
V_{OH}	Output HIGH Voltage	3.7			V	I_{OH} = −200 μA
V_{OL}	Output LOW Voltage			0.4	V	I_{OL} = 3.2 mA
V_{IH}	Input HIGH Voltage	2.0		V_{CC}	V	
V_{IL}	Input LOW Voltage	−0.5		0.8	V	
I_{IL}	Input Load Current			±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{OFL}	Data Bus Leakage			±10	μA	V_{SS} +0.45 ≤ $V_{OUT} \le V_{CC}$
I_{CC}	V_{CC} Supply Current		50	95	mA	
I_{DD}	V_{DD} Supply Current		50	95	mA	
C_O	Output Capacitance		8		pF	fc = 1.0 MHz, Inputs = 0V(1)
C_I	Input Capacitance		5		pF	
C_{IO}	I/O Capacitance		10		pF	

NOTE:
1. Sampled, not 100% tested.

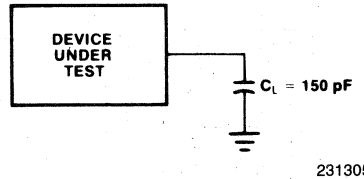
A.C. TESTING INPUT, OUTPUT WAVEFORM



231305-12

A.C. Testing: Inputs are driven at 3.7V for a logical "1" and 0.4V for a logical "0". Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".

A.C. TESTING LOAD CIRCUIT



231305-13

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{SS} = 0V, V_{CC} = +5V ±10%, V_{DD} = +12V ±10%

READ OPERATION

Symbol	Parameter		8231A-8		8231A		Units
			Min	Max	Min	Max	
t_{AR}	A_0, $\overline{CS}$ Setup to $\overline{RD}$		0		0		ns
t_{RA}	A_0, $\overline{CS}$ Hold from $\overline{RD}$		0		0		ns
t_{RY}	READY ↓ from $\overline{RD}$ ↓ Delay (Note 2)			150		100	ns
t_{YR}	Ready ↑ to $\overline{RD}$ ↑		0		0		ns
t_{RRR}	READY Pulse Width (Note 3)	Data	3.5 t_{CY} + 50		3.5 t_{CY} + 50		ns
		Status	1.5 t_{CY} + 50		1.5 t_{CY} + 50		ns
t_{RDE}	Data Bus Enable from $\overline{RD}$ ↓		50		50		ns
t_{DRY}	Data Valid to READY ↑		0		0		ns
t_{DF}	Data Float after $\overline{RD}$ ↑		50	200	50	100	ns

WRITE OPERATION

Symbol	Parameter		8231A-8		8231A		Units
			Min.	Max.	Min.	Max.	
t_{AW}	A_0, $\overline{CS}$ Setup to $\overline{WR}$		0		0		ns
t_{WA}	A_0, $\overline{CS}$ Hold after $\overline{WR}$		60		25		ns
t_{WY}	READY ↓ from $\overline{WR}$ ↓ Delay (Note 2)			150		100	ns
t_{YW}	READY ↑ to $\overline{WR}$ ↑		0		0		ns
t_{RRW}	READY Pulse Width (Note 4)			50		50	ns
t_{WI}	Write Inactive Time (Note 4)	Command	4 t_{CY}		4 t_{CY}		ns
		Data	5 t_{CY}		5 t_{CY}		ns
t_{DW}	Data Setup to $\overline{WR}$		150		100		ns
t_{WD}	Data Hold after $\overline{WR}$		20		20		ns

3

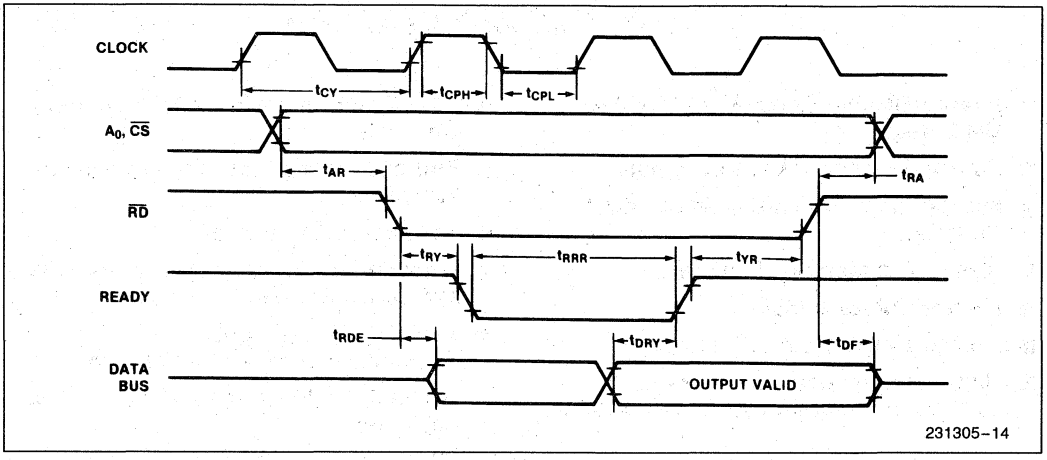
OTHER TIMINGS

Symbol	Parameter	8231A-8		8231A		Units
		Min	Max	Min	Max	
t_{CY}	Clock Period	480	5000	250	2500	ns
t_{CPH}	Clock Pulse High Width	200		100		ns
t_{CPL}	Clock Pulse Low Width	240		120		ns
t_{EE}	$\overline{END}$ Pulse Width (Note 5)	400		200		ns
t_{EAE}	$\overline{EACK}$ ↓ to $\overline{END}$ ↑ Delay		200		150	ns
t_{AA}	$\overline{EACK}$ Pulse Width	100		50		ns
t_{SA}	$\overline{SVACK}$ ↓ to SVREQ ↓ Delay		300		150	ns
t_{SS}	$\overline{SVACK}$ Pulse Width	100		50		ns

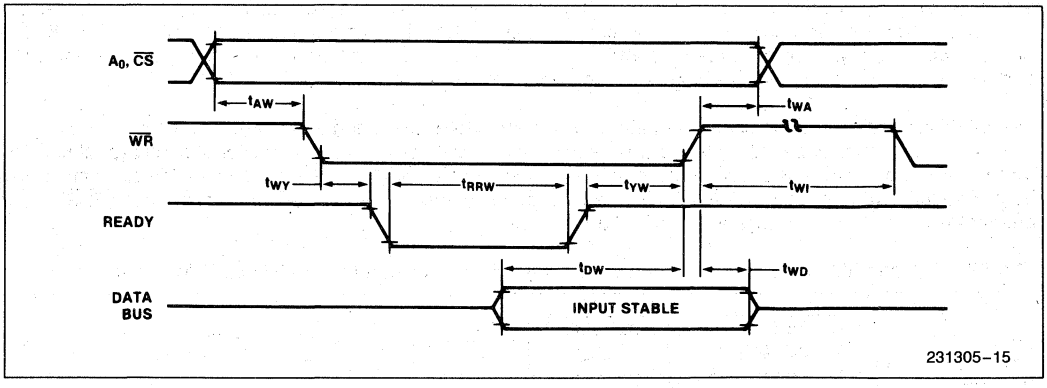
NOTES:
1. Typical values are for T_A = 25°C, nominal supply voltages processing parameters.
2. READY is pulled low for both command and data operations.
3. Minimum values shown assume no previously entered command is being executed for the data access. If a previously entered command is being executed, READY low pulse width is the time to complete execution plus the time shown. Status may be read at any time without exceeding the time shown.
4. READY low pulse width is less than 50 ns when writing into the data port or the control port as long as the duty cycle requirement (t_{WI}) is observed and no previous command is being executed. t_{WI} may be safely violated as long as the extended t_{RRW} that results is observed. If a previously entered command is being executed, READY low pulse width is the time to complete execution plus the time shown. These timings refer specifically to the 8231A.
5. $\overline{END}$ low pulse width is specified for $\overline{EACK}$ tied to VSS. Otherwise t_{EAE} applies.

WAVEFORMS

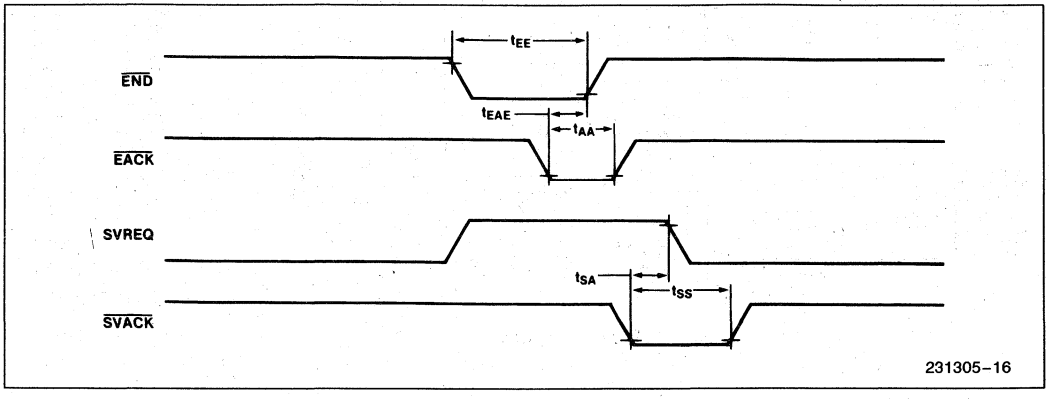
READ OPERATION



WRITE OPERATION



INTERRUPT OPERATION



8237A
HIGH PERFORMANCE PROGRAMMABLE DMA CONTROLLER (8237A, 8237A-4, 8237A-5)

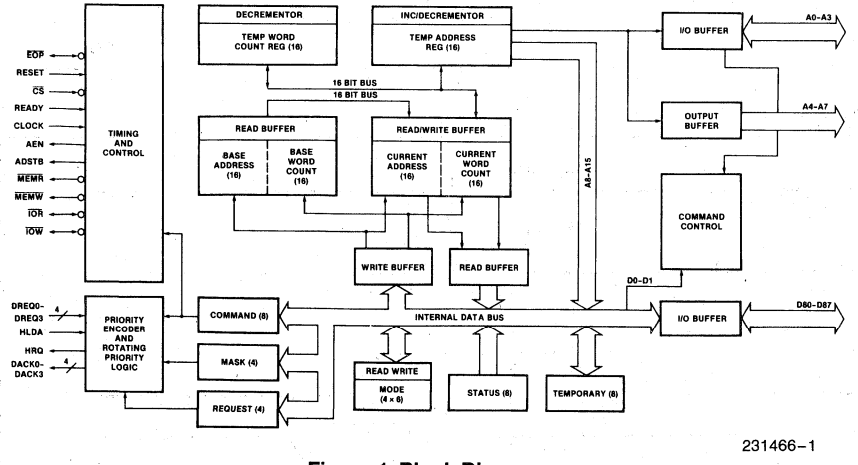
- **Enable/Disable Control of Individual DMA Requests**
- **Four Independent DMA Channels**
- **Independent Autoinitialization of All Channels**
- **Memory-to-Memory Transfers**
- **Memory Block Initialization**
- **Address Increment or Decrement**
- **High Performance: Transfers up to 1.6M Bytes/Second with 5 MHz 8237A-5**
- **Directly Expandable to Any Number of Channels**
- **End of Process Input for Terminating Transfers**
- **Software DMA Requests**
- **Independent Polarity Control for DREQ and DACK Signals**
- **Available in EXPRESS — Standard Temperature Range**
- **Available in 40-Lead Cerdip and Plastic Packages**

(See Packaging Spec, Order #231369)

The 8237A Multimode Direct Memory Access (DMA) Controller is a peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information from the system memory. Memory-to-memory transfer capability is also provided. The 8237A offers a wide variety of programmable control features to enhance data throughput and system optimization and to allow dynamic reconfiguration under program control.

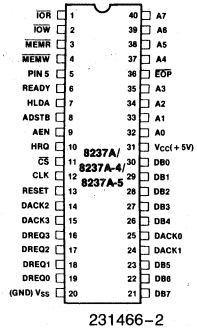
The 8237A is designed to be used in conjunction with an external 8-bit address latch. It contains four independent channels and may be expanded to any number of channels by cascading additional controller chips. The three basic transfer modes allow programmability of the types of DMA service by the user. Each channel can be individually programmed to Autoinitialize to its original condition following an End of Process ($\overline{EOP}$). Each channel has a full 64K address and word count capability.

The 8273A-4 and 8237A-5 are 4 MHz and 5 MHz versions of the standard 3 MHz 8237A respectively.



231466-1

Figure 1. Block Diagram



231466-2

Figure 2. Pin Configuration

October 1989
Order Number: 231466-004

Table 1. Pin Description

Symbol	Type	Name and Function
V_{CC}		**POWER:** +5V supply.
V_{SS}		**GROUND:** Ground.
CLK	I	**CLOCK INPUT:** Clock Input controls the internal operations of the 8237A and its rate of data transfers. The input may be driven at up to 3 MHz for the standard 8237A and up to 5 MHz for the 8237A-5.
$\overline{CS}$	I	**CHIP SELECT:** Chip Select is an active low input used to select the 8237A as an I/O device during the Idle cycle. This allows CPU communication on the data bus.
RESET	I	**RESET:** Reset is an active high input which clears the Command, Status, Request and Temporary registers. It also clears the first/last flip/flop and sets the Mask register. Following a Reset the device is in the Idle cycle.
READY	I	**READY:** Ready is an input used to extend the memory read and write pulses from the 8237A to accommodate slow memories or I/O peripheral devices. Ready must not make transitions during its specified setup/hold time.
HLDA	I	**HOLD ACKNOWLEDGE:** The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system busses.
DREQ0–DREQ3	I	**DMA REQUEST:** The DMA Request lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In fixed Priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. A request is generated by activating the DREQ line of a channel. DACK will acknowledge the recognition of DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high. DREQ must be maintained until the corresponding DACK goes active.
DB0–DB7	I/O	**DATA BUS:** The Data Bus lines are bidirectional three-state signals connected to the system data bus. The outputs are enabled in the Program condition during the I/O Read to output the contents of an Address register, a Status register, the Temporary register or a Word Count register to the CPU. The outputs are disabled and the inputs are read during an I/O Write cycle when the CPU is programming the 8237A control registers. During DMA cycles the most significant 8 bits of the address are output onto the data bus to be strobed into an external latch by ADSTB. In memory-to-memory operations, data from the memory comes into the 8237A on the data bus during the read-from-memory transfer. In the write-to-memory transfer, the data bus outputs place the data into the new memory location.
$\overline{IOR}$	I/O	**I/O READ:** I/O Read is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to read the control registers. In the Active cycle, it is an output control signal used by the 8237A to access data from a peripheral during a DMA Write transfer.
$\overline{IOW}$	I/O	**I/O WRITE:** I/O Write is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to load information into the 8237A. In the Active cycle, it is an output control signal used by the 8237A to load data to the peripheral during a DMA Read transfer.

Table 1. Pin Description (Continued)

Symbol	Type	Name and Function
$\overline{EOP}$	I/O	**END OF PROCESS:** End of Process is an active low bidirectional signal. Information concerning the completion of DMA services is available at the bidirectional $\overline{EOP}$ pin. The 8237A allows an external signal to terminate an active DMA service. This is accomplished by pulling the $\overline{EOP}$ input low with an external $\overline{EOP}$ signal. The 8237A also generates a pulse when the terminal count (TC) for any channel is reached. This generates an $\overline{EOP}$ signal which is output through the $\overline{EOP}$ line. The reception of $\overline{EOP}$, either internal or external, will cause the 8237A to terminate the service, reset the request, and, if Autoinitialize is enabled, to write the base registers to the current registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel by $\overline{EOP}$ unless the channel is programmed for Autoinitialize. In that case, the mask bit remains unchanged. During memory-to-memory transfers, $\overline{EOP}$ will be output when the TC for channel 1 occurs. $\overline{EOP}$ should be tied high with a pull-up resistor if it is not used to prevent erroneous end of process inputs.
A0–A3	I/O	**ADDRESS:** The four least significant address lines are bidirectional three-state signals. In the Idle cycle they are inputs and are used by the CPU to address the register to be loaded or read. In the Active cycle they are outputs and provide the lower 4 bits of the output address.
A4–A7	O	**ADDRESS:** The four most significant address lines are three-state outputs and provide 4 bits of address. These lines are enabled only during the DMA service.
HRQ	O	**HOLD REQUEST:** This is the Hold Request to the CPU and is used to request control of the system bus. If the corresponding mask bit is clear, the presence of any valid DREQ causes 8237A to issue the HRQ.
DACK0–DACK3	O	**DMA ACKNOWLEDGE:** DMA Acknowledge is used to notify the individual peripherals when one has been granted a DMA cycle. The sense of these lines is programmable. Reset initializes them to active low.
AEN	O	**ADDRESS ENABLE:** Address Enable enables the 8-bit latch containing the upper 8 address bits onto the system address bus. AEN can also be used to disable other system bus drivers during DMA transfers. AEN is active HIGH.
ADSTB	O	**ADDRESS STROBE:** The active high, Address Strobe is used to strobe the upper address byte into an external latch.
$\overline{MEMR}$	O	**MEMORY READ:** The Memory Read signal is an active low three-state output used to access data from the selected memory location during a DMA Read or a memory-to-memory transfer.
$\overline{MEMW}$	O	**MEMORY WRITE:** The Memory Write is an active low three-state output used to write data to the selected memory location during a DMA Write or a memory-to-memory transfer.
PIN5	I	**PIN5:** This pin should always be at a logic HIGH level. An internal pull-up resistor will establish a logic high when the pin is left floating. It is recommended however, that PIN5 be connected to V_{CC}.

FUNCTIONAL DESCRIPTION

The 8237A block diagram includes the major logic blocks and all of the internal registers. The data interconnection paths are also shown. Not shown are the various control signals between the blocks. The 8237A contains 344 bits of internal memory in the form of registers. Figure 3 lists these registers by name and shows the size of each. A detailed description of the registers and their functions can be found under Register Description.

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	6 bits	4
Mask Register	4 bits	1
Request Register	4 bits	1

Figure 3. 8237A Internal Registers

The 8237A contains three basic blocks of control logic. The Timing Control block generates internal timing and external control signals for the 8237A. The Program Command Control block decodes the various commands given to the 8237A by the microprocessor prior to servicing a DMA Request. It also decodes the Mode Control word used to select the type of DMA during the servicing. The Priority Encoder block resolves priority contention between DMA channels requesting service simultaneously.

The Timing Control block derives internal timing from the clock input. In 8237A systems, this input will usually be the $\phi 2$ TTL clock from an 8224 or CLK from an 8085AH or 8284A. 33% duty cycle clock generators, however, may not meet the clock high time requirement of the 8237A of the same frequency. For example, 82C84A-5 CLK output violates the clock high time requirement of 8237A-5. In this case 82C84A CLK can simply be inverted to meet 8237A-5 clock high and low time requirements. For 8085AH-2 systems above 3.9 MHz, the 8085 CLK(OUT) does not satisfy 8237A-5 clock LOW and HIGH time requirements. In this case, an external clock should be used to drive the 8237A-5.

DMA OPERATION

The 8237A is designed to operate in two major cycles. These are called Idle and Active cycles. Each device cycle is made up of a number of states. The 8237A can assume seven separate states, each composed of one full clock period. State I (SI) is the inactive state. It is entered when the 8237A has no valid DMA requests pending. While in SI, the DMA controller is inactive but may be in the Program Condition, being programmed by the processor. State S0 (S0) is the first state of a DMA service. The 8237A has requested a hold but the processor has not yet returned an acknowledge. The 8237A may still be programmed until it receives HLDA from the CPU. An acknowledge from the CPU will signal that DMA transfers may begin. S1, S2, S3 and S4 are the working states of the DMA service. If more time is needed to complete a transfer than is available with normal timing, wait states (SW) can be inserted between S2 or S3 and S4 by the use of the Ready line on the 8237A. Note that the data is transferred directly from the I/O device to memory (or vice versa) with $\overline{IOR}$ and $\overline{MEMW}$ (or $\overline{MEMR}$ and $\overline{IOW}$) being active at the same time. The data is not read into or driven out of the 8237A in I/O-to-memory or memory-to-I/O DMA transfers.

Memory-to-memory transfers require a read-from and a write-to-memory to complete each transfer. The states, which resemble the normal working states, use two digit numbers for identification. Eight states are required for a single transfer. The first four states (S11, S12, S13, S14) are used for the read-from-memory half and the last four states (S21, S22, S23, S24) for the write-to-memory half of the transfer.

IDLE CYCLE

When no channel is requesting service, the 8237A will enter the Idle cycle and perform "SI" states. In this cycle the 8237A will sample the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device will also sample $\overline{CS}$, looking for an attempt by the microprocessor to write or read the internal registers of the 8237A. When $\overline{CS}$ is low and HLDA is low, the 8237A enters the Program Condition. The CPU can now establish, change or inspect the internal definition of the part by reading from or writing to the internal registers. Address lines A0–A3 are inputs to the device and select which registers will be read or written. The $\overline{IOR}$ and $\overline{IOW}$ lines are used to select and time reads or writes. Due to the number and size of the internal registers, an internal flip-flop is used to generate an additional bit of address. This bit is used to determine the upper or lower byte of the 16-bit Address and Word Count registers. The flip-flop is reset by Master Clear or Reset. A separate software command can also reset this flip-flop.

Special software commands can be executed by the 8237A in the Program Condition. These commands are decoded as sets of addresses with the $\overline{CS}$ and $\overline{IOW}$. The commands do not make use of the data bus. Instructions include Clear First/Last Flip-Flop and Master Clear.

ACTIVE CYCLE

When the 8237A is in the Idle cycle and a non-masked channel requests a DMA service, the device will output an HRQ to the microprocessor and enter the Active cycle. It is in this cycle that the DMA service will take place, in one of four modes:

Single Transfer Mode—In Single Transfer mode the device is programmed to make one transfer only. The word count will be decremented and the address decremented or incremented following each transfer. When the word count "rolls over" from zero to FFFFH, a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so.

DREQ must be held active until DACK becomes active in order to be recognized. If DREQ is held active throughout the single transfer, HRQ will go inactive and release the bus to the system. It will again go active and, upon receipt of a new HLDA, another single transfer will be performed. In 8080A, 8085AH, 8088, or 8086 system, this will ensure one full machine cycle execution between DMA transfers. Details of timing between the 8237A and other bus control protocols will depend upon the characteristics of the microprocessor involved.

Block Transfer Mode—In Block Transfer mode the device is activated by DREQ to continue making transfers during the service until a TC, caused by word count going to FFFFH, or an external End of Process ($\overline{EOP}$) is encountered. DREQ need only be held active until DACK becomes active. Again, an Autoinitialization will occur at the end of the service if the channel has been programmed for it.

Demand Transfer Mode—In Demand Transfer mode the device is programmed to continue making transfers until a TC or external $\overline{EOP}$ is encountered or until DREQ goes inactive. Thus transfers may continue until the I/O device has exhausted its data capacity. After the I/O device has had a chance to catch up, the DMA service is re-established by means of a DREQ. During the time between services when the microprocessor is allowed to operate, the intermediate values of address and word count are stored in the 8237A Current Address and Current Word Count registers. Only an $\overline{EOP}$ can cause an Autoinitialize at the end of the service. $\overline{EOP}$ is generated either by TC or by an external signal. DREQ has to be low before S4 to prevent another Transfer.

Cascade Mode—This mode is used to cascade more than one 8237A together for simple system expansion. The HRQ and HLDA signals from the additional 8237A are connected to the DREQ and DACK signals of a channel of the initial 8237A. This allows the DMA requests of the additional device to propagate through the priority network circuitry of the preceding device. The priority chain is preserved and the new device must wait for its turn to acknowledge requests. Since the cascade channel of the initial 8237A is used only for prioritizing the additional device, it does not output any address or control

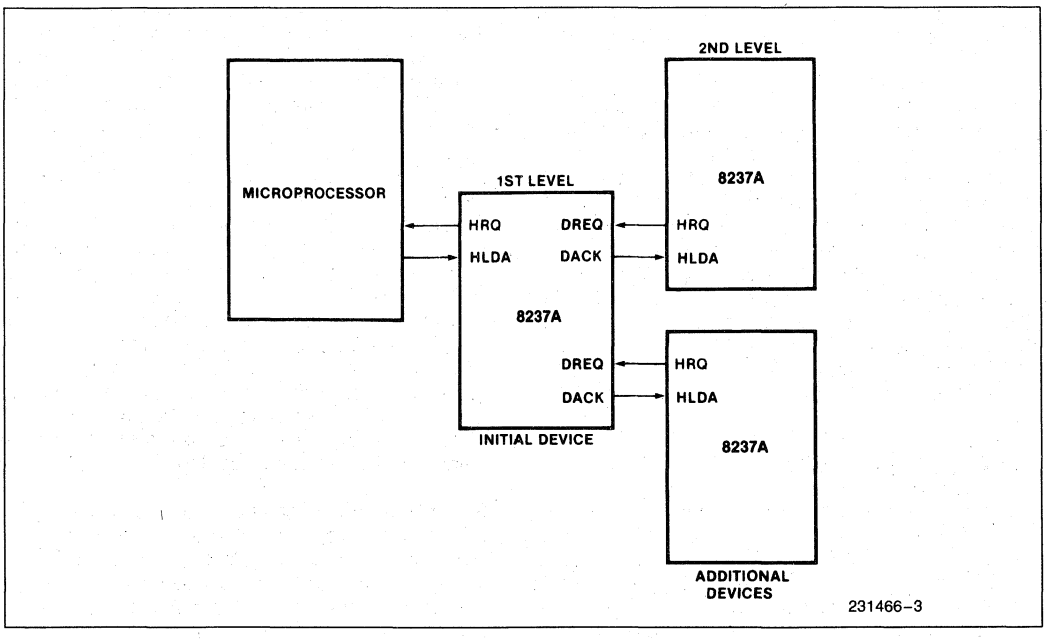


Figure 4. Cascaded 8237As

signals of its own. These could conflict with the outputs of the active channel in the added device. The 8237A will respond to DREQ and DACK but all other outputs except HRQ will be disabled. The ready input is ignored.

Figure 4 shows two additional devices cascaded into an initial device using two of the previous channels. This forms a two level DMA system. More 8237As could be added at the second level by using the remaining channels of the first level. Additional devices can also be added by cascading into the channels of the second level device, forming a third level.

TRANSFER TYPES

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write and Verify. Write transfers move data from an I/O device to the memory by activating $\overline{MEMW}$ and $\overline{IOR}$. Read transfers move data from memory to an I/O device by activating $\overline{MEMR}$ and $\overline{IOW}$. Verify transfers are pseudo transfers. The 8237A operates as in Read or Write transfers generating addresses, and responding to EOP, etc. However, the memory and I/O control lines all remain inactive. The ready input is ignored in verify mode.

Memory-to-Memory—To perform block moves of data from one memory address space to another with a minimum of program effort and time, the 8237A includes a memory-to-memory transfer feature. Programming a bit in the Command register selects channels 0 and 1 to operate as memory-to-memory transfer channels. The transfer is initiated by setting the software DREQ for channel 0. The 8237A requests a DMA service in the normal manner. After HLDA is true, the device, using four state transfers in Block Transfer mode, reads data from the memory. The channel 0 Current Address register is the source for the address used and is decremented or incremented in the normal manner. The data byte read from the memory is stored in the 8237A internal Temporary register. Channel 1 then performs a four-state transfer of the data from the Temporary register to memory using the address in its Current Address register and incrementing or decrementing it in the normal manner. The channel 1 current Word Count is decremented. When the word count of channel 1 goes to FFFFH, a TC is generated causing an $\overline{EOP}$ output terminating the service.

Channel 0 may be programmed to retain the same address for all transfers. This allows a single word to be written to a block of memory.

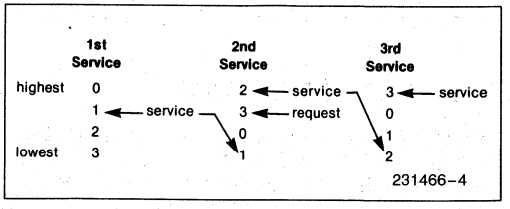
The 8237A will respond to external $\overline{EOP}$ signals during memory-to-memory transfers. Data comparators in block search schemes may use this input to terminate the service when a match is found. The timing of memory-to-memory transfers is found in Figure 12. Memory-to-memory operations can be detected as an active AEN with no DACK outputs.

Autoinitialize—By programming a bit in the Mode register, a channel may be set up as an Autoinitialize channel. During Autoinitialize initialization, the original values of the Current Address and Current Word Count registers are automatically restored from the Base Address and Base Word count registers of that channel following $\overline{EOP}$. The base registers are loaded simultaneously with the current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not altered when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to perform another DMA service, without CPU intervention, as soon as a valid DREQ is detected. In order to Autoinitialize both channels in a memory-to-memory transfer, both word counts should be programmed identically. If interrupted externally, $\overline{EOP}$ pulses should be applied in both bus cycles.

Priority—The 8237A has two types of priority encoding available as software selectable options. The first is Fixed Priority which fixes the channels in priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0. After the recognition of any one channel for service, the other channels are prevented from interfering with that service until it is completed.

After completion of a service, HRQ will go inactive and the 8237A will wait for HLDA to go low before activating HRQ to service another channel.

The second scheme is Rotating Priority. The last channel to get service becomes the lowest priority channel with the others rotating accordingly.



With Rotating Priority in a single chip DMA system, any device requesting service is guaranteed to be recognized after no more than three higher priority services have occurred. This prevents any one channel from monopolizing the system.

Compressed Timing—In order to achieve even greater throughput where system characteristics permit, the 8237A can compress the transfer time to two clock cycles. From Figure 11 it can be seen that state S3 is used to extend the access time of the read pulse. By removing state S3, the read pulse width is made equal to the write pulse width and a transfer consists only of state S2 to change the address and state S4 to perform the read/write. S1 states will still occur when A8–A15 need updating (see Address Generation). Timing for compressed transfers is found in Figure 14.

Address Generation—In order to reduce pin count, the 8237A multiplexes the eight higher order address bits on the data lines. State S1 is used to output the higher order address bits to an external latch from which they may be placed on the address bus. The falling edge of Address Strobe (ADSTB) is used to load these bits from the data lines to the latch. Address Enable (AEN) is used to enable the bits onto the address bus through a three-state enable. The lower order address bits are output by the 8237A directly. Lines A0–A7 should be connected to the address bus. Figure 11 shows the time relationships between CLK, AEN, ADSTB, DB0–DB7 and A0–A7.

During Block and Demand Transfer mode services, which include multiple transfers, the addresses generated will be sequential. For many transfers the data held in the external address latch will remain the same. This data need only change when a carry or borrow from A7 to A8 takes place in the normal sequence of addresses. To save time and speed transfers, the 8237A executes S1 states only when updating of A8–A15 in the latch is necessary. This means for long services, S1 states and Address Strobes may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

REGISTER DESCRIPTION

Current Address Register—Each channel has a 16-bit Current Address register. This register holds the value of the address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written or read by the microprocessor in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialize takes place only after an $\overline{EOP}$.

Current Word Register—Each channel has a 16-bit Current Word Count register. This register determines the number of transfers to be performed. The actual number of transfers will be one more than the number programmed in the Current Word Count register (i.e., programming a count of 100 will result in 101 transfers). The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes from zero to FFFFH, a TC will be generated. This register is loaded or read in successive 8-bit bytes by the microprocessor in the Program Condition. Following the end of a DMA service it may also be reinitialized by an Autoinitialization back to its original value. Autoinitialize can occur only when an $\overline{EOP}$ occurs. If it is not Autoinitialized, this register will have a count of FFFFH after TC.

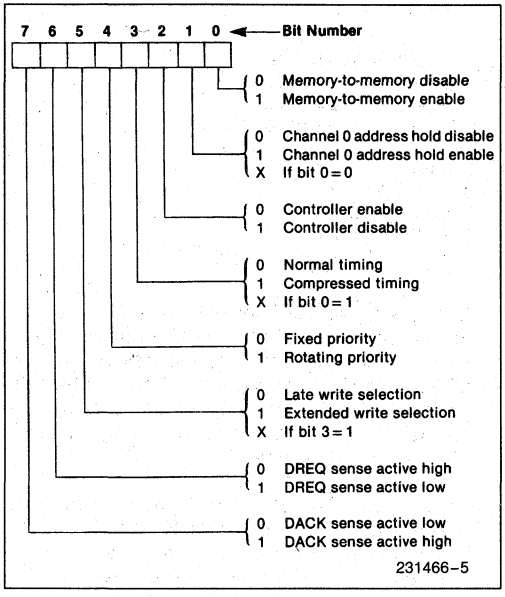
Base Address and Base Word Count Registers—Each channel has a pair of Base Address and Base Word Count registers. These 16-bit registers store the original value of their associated current registers. During Autoinitialize these values are used to restore the current registers to their original values. The base registers are written simultaneously with their corresponding current register in 8-bit bytes in the Program Condition by the microprocessor. These registers cannot be read by the microprocessor.

Command Register—This 8-bit register controls the operation of the 8237A. It is programmed by the microprocessor in the Program Condition and is cleared by Reset or a Master Clear instruction. The following table lists the function of the command bits. See Figure 6 for address coding.

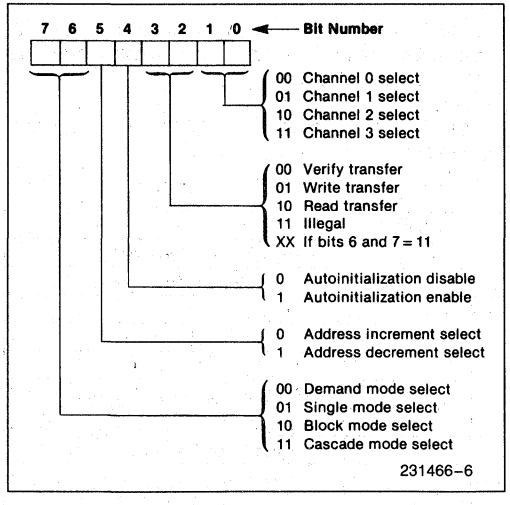
Mode Register—Each channel has a 6-bit Mode register associated with it. When the register is being written to by the microprocessor in the Program Condition, bits 0 and 1 determine which channel Mode register is to be written.

Request Register—The 8237A can respond to requests for DMA service which are initiated by software as well as by a DREQ. Each channel has a request bit associated with it in the 4-bit Request register. These are non-maskable and subject to prioritization by the Priority Encoder network. Each register bit is set or reset separately under software control or is cleared upon generation of a TC or external $\overline{EOP}$. The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word. See Figure 5 for register address coding. In order to make a software request, the channel must be in Block Mode.

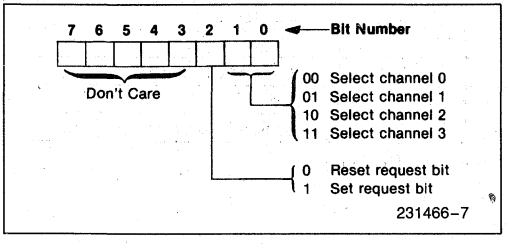
Command Register



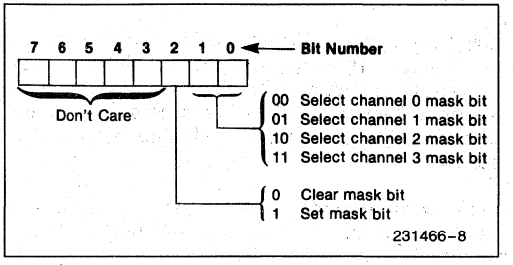
Mode Register



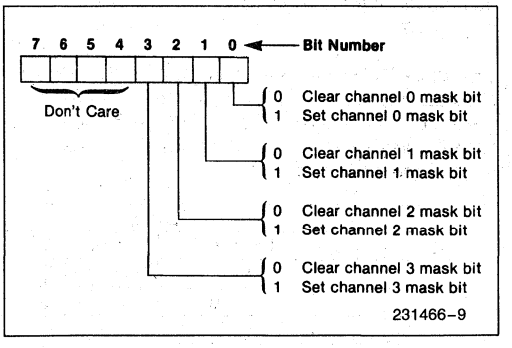
Request Register



Mask Register—Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an $\overline{EOP}$ if the channel is not programmed for Autoinitialize. Each bit of the 4-bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. See Figure 5 for instruction addressing.



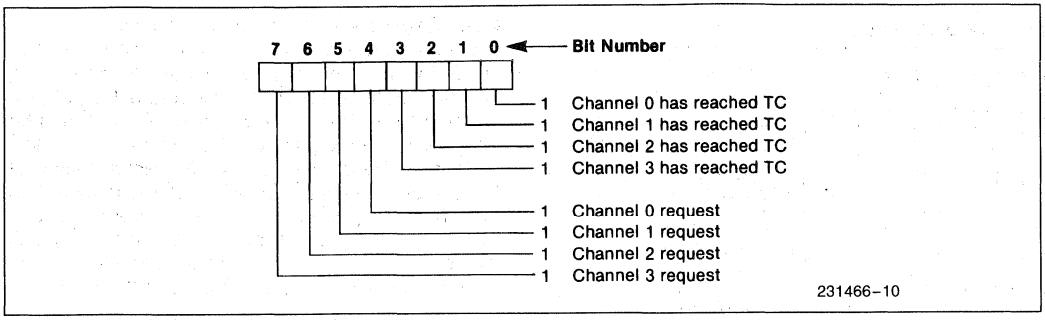
All four bits of the Mask register may also be written with a single command.



Register	Operation	Signals						
		$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0
Command	Write	0	1	0	1	0	0	0
Mode	Write	0	1	0	1	0	1	1
Request	Write	0	1	0	1	0	0	1
Mask	Set/Reset	0	1	0	1	0	1	0
Mask	Write	0	1	0	1	1	1	1
Temporary	Read	0	0	1	1	1	0	1
Status	Read	0	0	1	1	0	0	0

Figure 5. Definition of Register Codes

Status Register—The Status register is available to be read out of the 8237A by the microprocessor. It contains information about the status of the devices at this point. This information includes which channels have reached a terminal count and which chan-



nels have pending DMA requests. Bits 0–3 are set every time a TC is reached by that channel or an external $\overline{EOP}$ is applied. These bits are cleared upon Reset and on each Status Read. Bits 4–7 are set whenever their corresponding channel is requesting service.

Temporary Register—The Temporary register is used to hold data during memory-to-memory transfers. Following the completion of the transfers, the last word moved can be read by the microprocessor in the Program Condition. The Temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a Reset.

Software Commands—These are additional special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The three software commands are:

Clear First/Last Flip-Flop: This command must be executed prior to writing or reading new address or word count information to the 8237A. This initializes the flip-flop to a known state so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.

Master Clear: This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary, and Internal First/Last Flip-Flop registers are cleared and the Mask register is set. The 8237A will enter the Idle cycle.

Clear Mask Register: This command clears the mask bits of all four channels, enabling them to accept DMA requests.

Figure 6 lists the address codes for the software commands.

Signals						Operation
A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip/Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Clear Mask Register
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Figure 6. Software Command Codes

Channel	Register	Operation	Signals							Internal Flip-Flop	Data Bus DB0–DB7
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base and Current Address	Write	0	1	0	0	0	0	0	0	A0–A7
			0	1	0	0	0	0	0	1	A8–A15
	Current Address	Read	0	0	1	0	0	0	0	0	A0–A7
			0	0	1	0	0	0	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	0	0	1	0	W0–W7
			0	1	0	0	0	0	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	0	0	1	0	W0–W7
			0	0	1	0	0	0	1	1	W8–W15
1	Base and Current Address	Write	0	1	0	0	0	1	0	0	A0–A7
			0	1	0	0	0	1	0	1	A8–A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0–A7
			0	0	1	0	0	1	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	0	1	1	0	W0–W7
			0	1	0	0	0	1	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	0	1	1	0	W0–W7
			0	0	1	0	0	1	1	1	W8–W15
2	Base and Current Address	Write	0	1	0	0	1	0	0	0	A0–A7
			0	1	0	0	1	0	0	1	A8–A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0–A7
			0	0	1	0	1	0	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	1	0	1	0	W0–W7
			0	1	0	0	1	0	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	1	0	1	0	W0–W7
			0	0	1	0	1	0	1	1	W8–W15
3	Base and Current Address	Write	0	1	0	0	1	1	0	0	A0–A7
			0	1	0	0	1	1	0	1	A8–A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0–A7
			0	0	1	0	1	1	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	1	1	1	0	W0–W7
			0	1	0	0	1	1	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	1	1	1	0	W0–W7
			0	0	1	0	1	1	1	1	W8–W15

Figure 7. Word Count and Address Register Command Codes

PROGRAMMING

The 8237A will accept programming from the host processor any time that HLDA is inactive; this is true even if HRQ is active. The responsibility of the host is to assure that programming and HLDA are mutually exclusive. Note that a problem can occur if a DMA request occurs, on an unmasked channel while the 8237A is being programmed. For instance, the CPU may be starting to reprogram the two byte Address register of channel 1 when channel 1 receives a DMA request. If the 8237A is enabled (bit 2 in the command register is 0) and channel 1 is unmasked, a DMA service will occur after only one byte of the Address register has been reprogrammed. This can be avoided by disabling the controller (setting bit 2 in the command register) or masking the channel before programming any other registers. Once the programming is complete, the controller can be enabled/unmasked.

After power-up it is suggested that all internal locations, especially the Mode registers, be loaded with some valid value. This should be done even if some channels are unused. An invalid mode may force all control signals to go active at the same time.

APPLICATION INFORMATION (Note 1)

Figure 8 shows a convenient method for configuring a DMA system with the 8237A controller and an 8080A/8085AH microprocessor system. The multimode DMA controller issues a HRQ to the processor whenever there is at least one valid DMA request from a peripheral device. When the processor replies with a HLDA signal, the 8237A takes control of the address bus, the data bus and the control bus. The address for the first transfer operation comes out in two bytes—the least significant 8 bits on the eight address outputs and the most significant 8 bits on the data bus. The contents of the data bus are then latched into an 8-bit latch to complete the full 16 bits of the address bus. The 8282 is a high speed, 8-bit, three-state latch in a 20-pin package. After the initial transfer takes place, the latch is updated only after a carry or borrow is generated in the least significant address byte. Four DMA channels are provided when one 8237A is used.

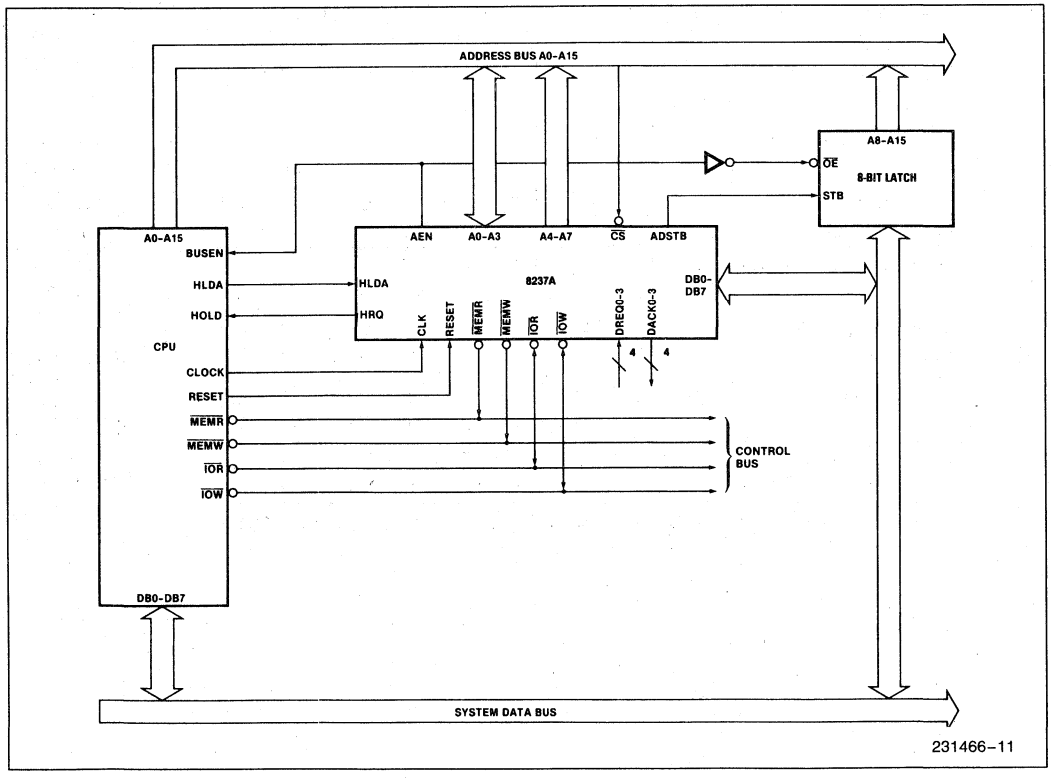


Figure 8. 8237A System Interface

NOTE:
1. See Application Note AP-67 for 8086 design information.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature under Bias .....0°C to 70°C
Case Temperature .....0°C to +75°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin with
Respect to Ground.....−0.5V to +7V
Power Dissipation.....1.5 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, T_{CASE} = 0°C to 75°C, V_{CC} = +5.0V ±5%, GND = 0V

Symbol	Parameter	Min	Typ (Note 1)	Max	Unit	Test Conditions
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −200 μA
		3.3			V	I_{OH} = −100 μA (HRQ Only)
V_{OL}	Output LOW Voltage			0.40	V	I_{OL} = 3.2 mA
V_{IH}	Input HIGH Voltage	2.0		V_{CC} + 0.5	V	
V_{IL}	Input LOW Voltage	−0.5		0.8	V	
I_{LI}	Input Load Current			±10	μA	0V ≤ V_{IN} ≤ V_{CC}
I_{LO}	Output Leakage Current			±10	μA	0.45V ≤ V_{OUT} ≤ V_{CC}
I_{CC}	V_{CC} Supply Current		110	130	mA	T_A = +25°C
			130	150	mA	T_A = 0°C
C_O	Output Capacitance		4	8	pF	fc = 1.0 MHz, Inputs = 0V
C_I	Input Capacitance		8	15	pF	
C_{IO}	I/O Capacitance		10	18	pF	

NOTE:
1. Typical values are for T_A = 25°C, nominal supply voltage and nominal processing parameters.

A.C. CHARACTERISTICS—DMA (MASTER) MODE

T_A = 0°C to 70°C, T_{CASE} = 0°C to 75°C, V_{CC} = +5V ±5%, GND = 0V

Symbol	Parameter	8237A		8237A-4		8237A-5		Unit
		Min	Max	Min	Max	Min	Max	
TAEL	AEN HIGH from CLK LOW (S1) Delay Time		300		225		200	ns
TAET	AEN LOW from CLK HIGH (SI) Delay Time		200		150		130	ns
TAFAB	ADR Active to Float Delay from CLK HIGH		150		120		90	ns
TAFC	$\overline{READ}$ or $\overline{WRITE}$ Float from CLK HIGH		150		120		120	ns
TAFDB	DB Active to Float Delay from CLK HIGH		250		190		170	ns
TAHR	ADR from $\overline{READ}$ HIGH Hold Time	TCY-100		TCY-100		TCY-100		ns
TAHS	DB from ADSTB LOW Hold Time	40		40		30		ns
TAHW	ADR from $\overline{WRITE}$ HIGH Hold Time	TCY-50		TCY-50		TCY-50		ns
TAK	DACK Valid from CLK LOW Delay Time (Note 1)		250		220		170	ns
	$\overline{EOP}$ HIGH from CLK HIGH Delay Time (Note 2)		250		190		170	ns
	$\overline{EOP}$ LOW from CLK HIGH Delay Time		250		190		170	ns
TASM	ADR Stable from CLK HIGH		250		190		170	ns
TASS	DB to ADSTB LOW Setup Time	100		100		100		ns
TCH	Clock High Time (Transitions ≤ 10 ns)	120		100		80		ns
TCL	Clock LOW Time (Transitions ≤ 10 ns)	150		110		68		ns
TCY	CLK Cycle Time	320		250		200		ns
TDCL	CLK HIGH to $\overline{READ}$ or $\overline{WRITE}$ LOW Delay (Note 3)		270		200		190	ns
TDCTR	$\overline{READ}$ HIGH from CLK HIGH (S4) Delay Time (Note 3)		270		210		190	ns
TDCTW	$\overline{WRITE}$ HIGH from CLK HIGH (S4) Delay Time (Note 3)		200		150		130	ns
TDQ1	HRQ Valid from CLK HIGH Delay Time (Note 4)		160		120		120	ns
TDQ2			250		190		120	ns
TEPS	$\overline{EOP}$ LOW from CLK LOW Setup Time	60		45		40		ns
TEPW	$\overline{EOP}$ Pulse Width	300		225		220		ns
TFAAB	ADR Float to Active Delay from CLK HIGH		250		190		170	ns
TFAC	$\overline{READ}$ or $\overline{WRITE}$ Active from CLK HIGH		200		150		150	ns
TFADB	DB Float to Active Delay from CLK HIGH		300		225		200	ns
THS	HLDA Valid to CLK HIGH Setup Time	100		75		75		ns
TIDH	Input Data from $\overline{MEMR}$ HIGH Hold Time	0		0		0		ns
TIDS	Input Data to $\overline{MEMR}$ HIGH Setup Time	250		190		170		ns
TODH	Output Data from $\overline{MEMW}$ HIGH Hold Time	20		20		10		ns
TODV	Output Data Valid to $\overline{MEMW}$ HIGH	200		125		125		ns
TQS	DREQ to CLK LOW (SI, S4) Setup Time (Note 1)	0		0		0		ns
TRH	CLK to READY LOW Hold Time	20		20		20		ns
TRS	READY to CLK LOW Setup Time	100		60		60		ns
TSTL	ADSTB HIGH from CLK HIGH Delay Time		200		150		130	ns
TSTT	ADSTB LOW from CLK HIGH Delay Time		140		110		90	ns

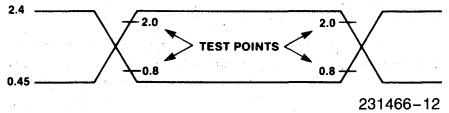
A.C. CHARACTERISTICS—PERIPHERAL (SLAVE) MODE

T_A = 0°C to 70°C, T_{CASE} = 0°C to 75°C, V_{CC} = +5V ±5%, GND = 0V

Symbol	Parameter	8237A		8237A-4		8237A-5		Unit
		Min	Max	Min	Max	Min	Max	
TAR	ADR Valid or $\overline{CS}$ LOW to $\overline{READ}$ LOW	50		50		50		ns
TAW	ADR Valid to $\overline{WRITE}$ HIGH Setup Time	200		150		130		ns
TCW	CS LOW to $\overline{WRITE}$ HIGH Setup Time	200		150		130		ns
TDW	Data Valid to $\overline{WRITE}$ HIGH Setup Time	200		150		130		ns
TRA	ADR or CS Hold from $\overline{READ}$ HIGH	0		0		0		ns
TRDE	Data Access from $\overline{READ}$ LOW (Note 5)		200		200		140	ns
TRDF	DB Float Delay from $\overline{READ}$ HIGH	20	100	20	100	0	70	ns
TRSTD	Power Supply HIGH to RESET LOW Setup Time	500		500		500		ns
TRSTS	RESET to First $\overline{IOWR}$	2TCY		2TCY		2TCY		ns
TRSTW	RESET Pulse Width	300		300		300		ns
TRW	$\overline{READ}$ Width	300		250		200		ns
TWA	ADR from $\overline{WRITE}$ HIGH Hold Time	20		20		20		ns
TWC	CS HIGH from $\overline{WRITE}$ HIGH Hold Time	20		20		20		ns
TWD	Data from $\overline{WRITE}$ HIGH Hold Time	30		30		30		ns
TWWS	Write Width	200		200		160		ns
TWR	End of Write to End of Read in DMA Transfer	0		0		0		ns

NOTES:
1. DREQ and DACK signals may be active high or active low. Timing diagrams assume the active high mode.
2. $\overline{EOP}$ is an open collector output. This parameter assumes the presence of a 2.2K pullup to V_{CC}.
3. The net $\overline{IOW}$ or $\overline{MEMW}$ Pulse width for normal write will be TCY−100 ns and for extended write will be 2TCY−100 ns. The net $\overline{IOR}$ or $\overline{MEMR}$ pulse width for normal read will be 2TCY−50 ns and for compressed read will be TCY−50 ns.
4. TDQ is specified for two different output HIGH levels. TDQ1 is measured at 2.0V. TDQ2 is measured at 3.3V. The value for TDQ2 assumes an external 3.3 KΩ pull-up resistor connected from HRQ to V_{CC}.
5. Output Loading on the Data Bus is 1 TTL Gate plus 100 pF capacitance.

A.C. TESTING INPUT/OUTPUT WAVEFORM

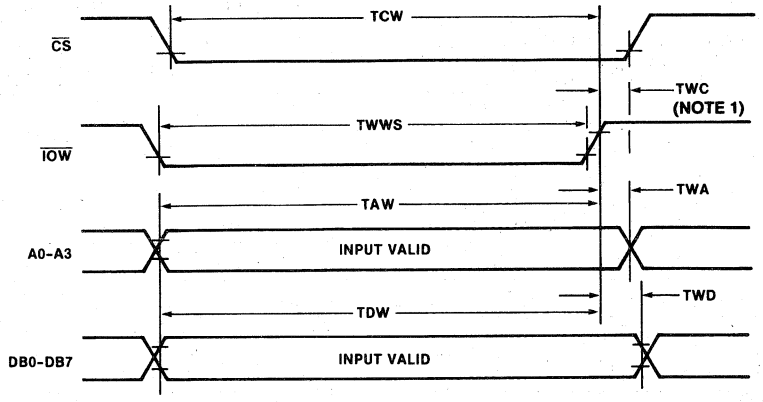


231466-12

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0." Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0." Input timing parameters assume transition times of 20 ns or less. Waveform measurement points for both input and output signals are 2.0V for HIGH and 0.8V for LOW, unless otherwise noted.

WAVEFORMS

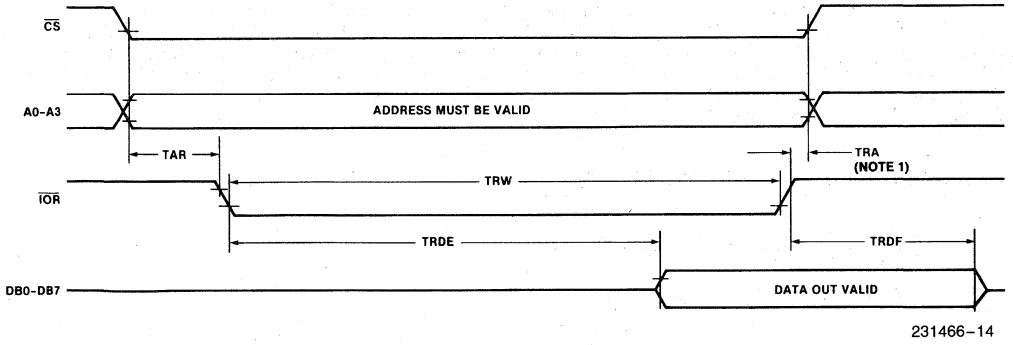
SLAVE MODE WRITE TIMING



NOTE:
1. Successive read and/or write operations by the external processor to program or examine the controller must be timed to allow at least 600 ns for the 8237A, at least 500 ns for the 8237A-4, and at least 400 ns for the 8237A-5 as recovery time between active read or write pulses. The same recovery time is needed between an active read or write pulse followed by a DMA transfer.

Figure 9. Slave Mode Write

SLAVE MODE READ TIMING

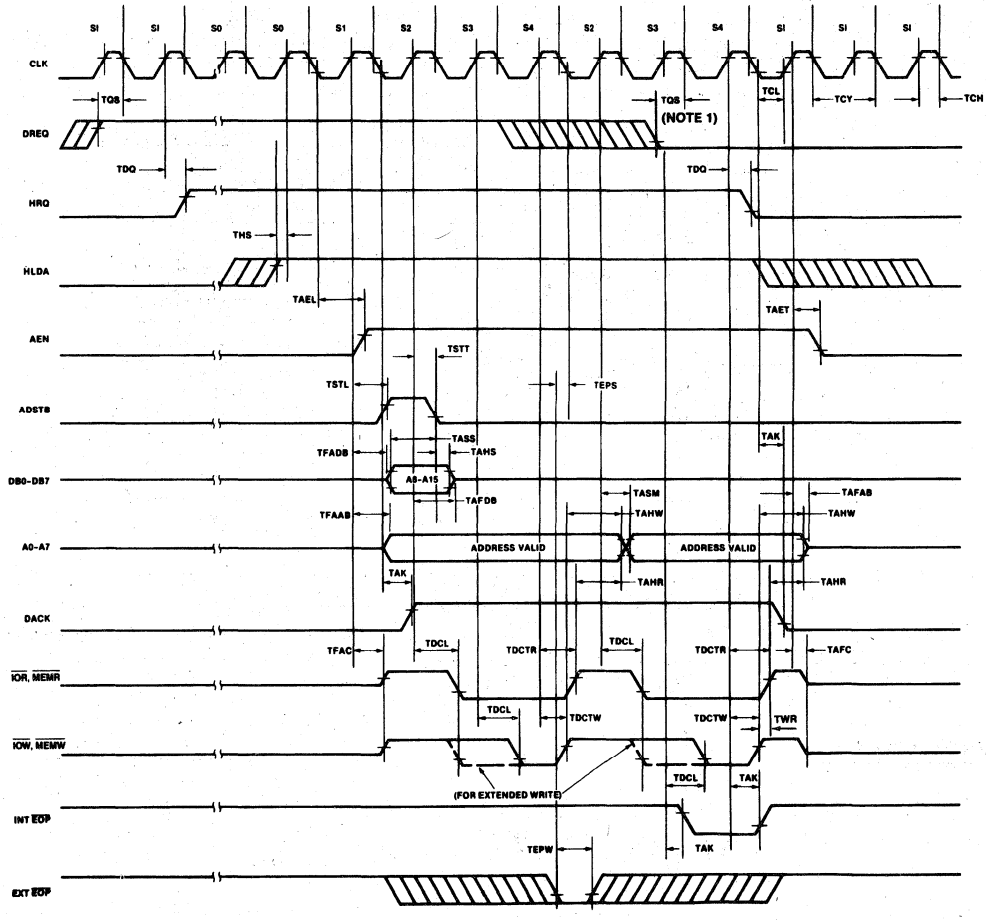


NOTE:
1. Successive read and/or write operations by the external processor to program or examine the controller must be timed to allow at least 600 ns for the 8237A, at least 500 ns for the 8237A-4, and at least 400 ns for the 8237A-5 as recovery time between active read or write pulses. The same recovery time is needed between an active read or write pulse followed by a DMA transfer.

Figure 10. Slave Mode Read

WAVEFORMS (Continued)

DMA TRANSFER TIMING



NOTE:
1. DREQ should be held active until DACK is returned.

Figure 11. DMA Transfer

WAVEFORMS (Continued)

MEMORY-TO-MEMORY TRANSFER TIMING

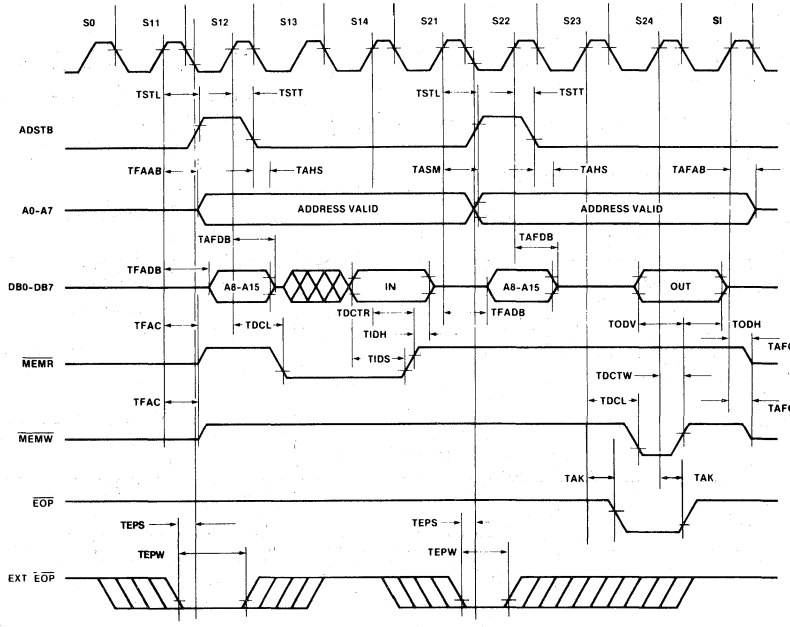


Figure 12. Memory-to-Memory Transfer

READY TIMING

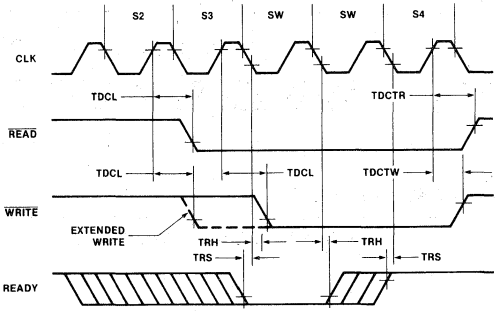


Figure 13. Ready

WAVEFORMS (Continued)

COMPRESSED TRANSFER TIMING

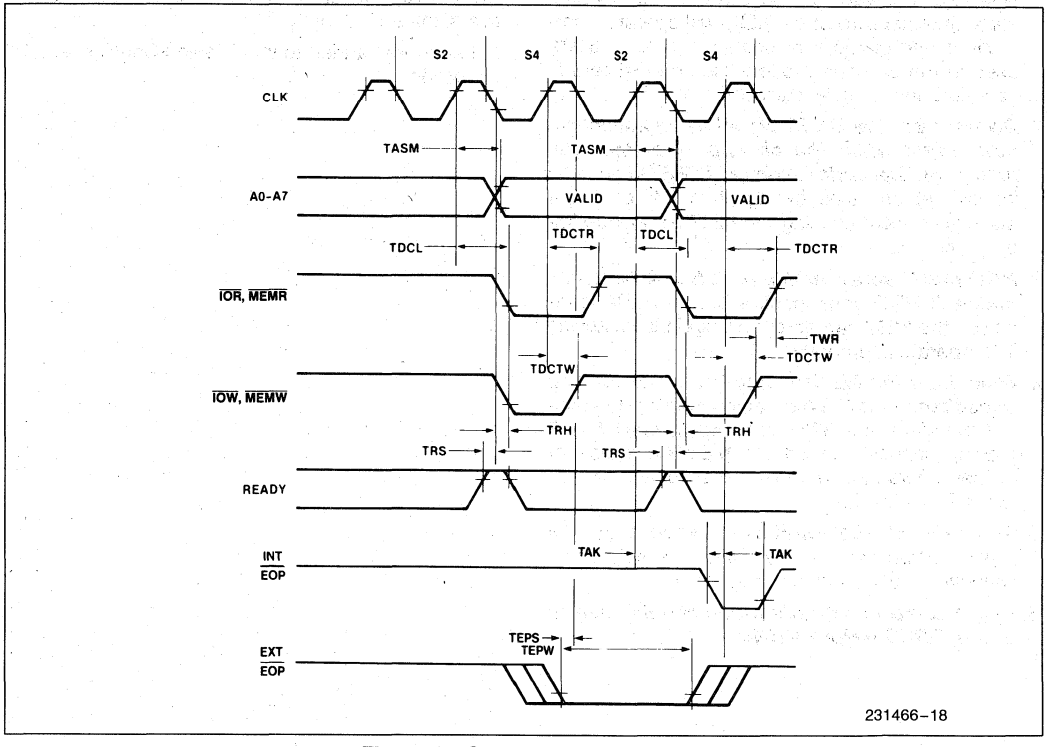


Figure 14. Compressed Transfer

RESET TIMING

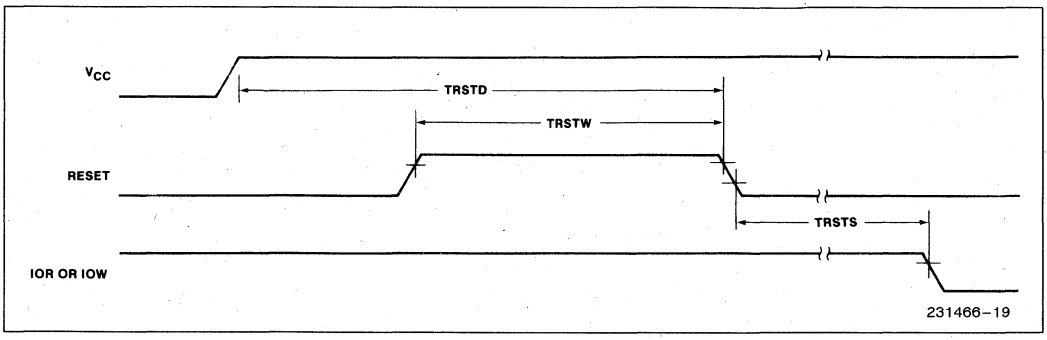


Figure 15. Reset

DESIGN CONSIDERATIONS

1. ***Cascading from channel zero.*** When using multiple 8237s, always start cascading with channel zero. Channel zero of the 8237 will operate incorrectly if one or more of channels 1, 2, or 3 are used in the cascade mode while channel zero is used in a mode other than cascade.
2. ***Do not treat the DREQ signal as an asynchronous input while the channel is in the "demand" or "cascade" modes.*** If DREQ becomes inactive at any time during state S4, an illegal state may occur causing the 8237 to operate improperly.
3. ***HRQ must remain active until HLDA becomes active.*** If HRQ goes inactive before HLDA is received the 8237 can enter an illegal state causing it to operate improperly.
4. ***Make sure the MEMR# line has 50 pF loading capacitance on it.*** When doing memory to memory transfers, the 8237 requires at least 50 pF loading capacitance on the MEMR# signal for proper operation. In most cases board capacitance is sufficient.
5. ***Treat the READY input as a synchronous input.*** If a transition occurs during the setup/hold window, erratic operation may result.
6. ***Any channel in cascade mode should have an active DREQ before a HRQ.***

DATA SHEET REVISION REVIEW

The following list represents key differences between this and the -003 data sheet. Please review this summary carefully.

1. Item 6 was added to the "Design Considerations" section.

82C37A-5
CHMOS HIGH PERFORMANCE PROGRAMMABLE DMA CONTROLLER

- **Pin Compatible with NMOS 8237A-5**
- **Enable/Disable Control of Individual DMA Requests**
- **Fully Static Design with Frequency Range from DC to 5 MHz**
- **Low Power Operation**
- **Four Independent DMA Channels**
- **Independent Autoinitialization of all Channels**
- **Memory-to-Memory Transfers**
- **Memory Block Initialization**
- **Address Increment or Decrement**
- **High performance: 5 MHz Speed Transfers up to 1.6 MBytes/Second**
- **Directly Expandable to any Number of Channels**
- **End of Process Input for Terminating Transfers**
- **Software DMA Requests**
- **Independent Polarity Control for DREQ and DACK Signals**
- **Available in 40-Lead Plastic DIP**

3

The Intel 82C37A-5 Multimode Direct Memory Access (DMA) Controller is a CHMOS peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information from the system memory. Memory-to-memory transfer capability is also provided. The 82C37A-5 offers a wide variety of programmable control features to enhance data throughput and system optimization and to allow dynamic reconfiguration under program control.

The 82C37A-5 is designed to be used in conjunction with an external 8-bit address register. It contains four independent channels and may be expanded to any number of channels by cascading additional controller chips.

The three basic transfer modes allow programmability of the types of DMA service by the user. Each channel can be individually programmed to Autoinitialize to its original condition following an End of Process ($\overline{EOP}$).

Each channel has a full 64K address and word count capability.

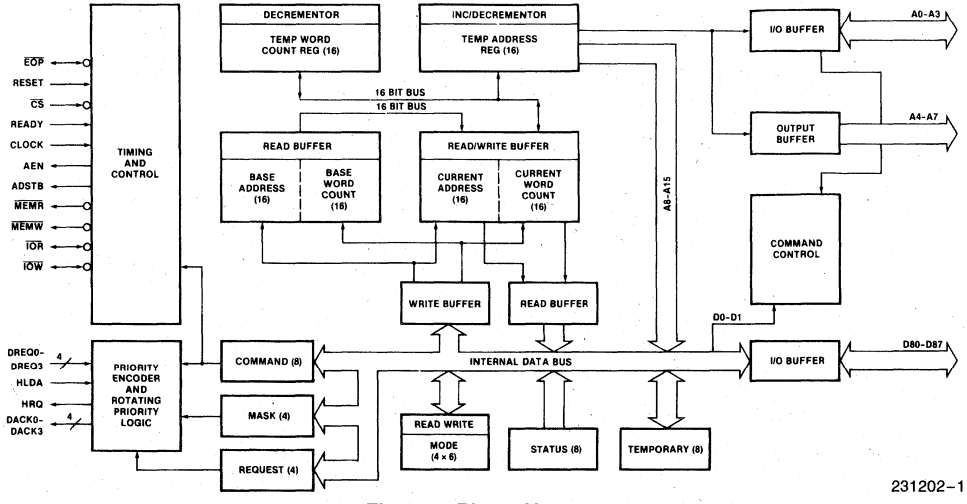


Figure 1. Block Diagram

September 1988
Order Number: 231202-005

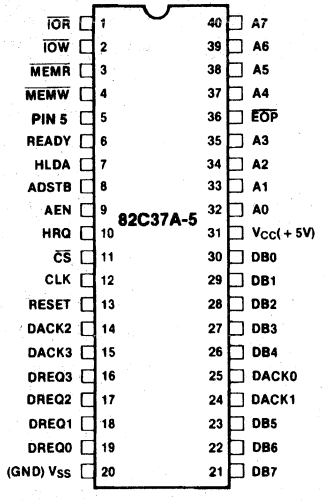


Figure 2. 82C37A-5
40-Lead DIP Configuration

Table 1. Pin Description

Symbol	Type	Name and Function
V_{CC}		**POWER:** +5 volt supply.
V_{SS}		**GROUND:** Ground.
CLK	I	**CLOCK INPUT:** Clock Input controls the internal operations of the 82C37A-5 and its rate of data transfers. The input may be driven at up to 5 MHz for the 82C37A-5.
$\overline{CS}$	I	**CHIP SELECT:** Chip Select is an active low input used to select the 82C37A-5 as an I/O device during the Idle cycle. This allows CPU communication on the data bus.
RESET	I	**RESET:** Reset is an active high input which clears the Command, Status, Request and Temporary registers. It also clears the first/last flip-flop and sets the Mask register. Following a Reset the device is in the Idle cycle.
READY	I	**READY:** Ready is an input used to extend the memory read and write pulses from the 82C37A-5 to accommodate slow memories or I/O peripheral devices. Ready must not make transitions during its specified setup/hold time.
HLDA	I	**HOLD ACKNOWLEDGE:** The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system busses.
DREQ0–DREQ3	I	**DMA REQUEST:** The DMA Request lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In fixed Priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. A request is generated by activating the DREQ line of a channel. DACK will acknowledge the recognition of DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high. DREQ must be maintained until the corresponding DACK goes active.
DB0–DB7	I/O	**DATA BUS:** The Data Bus lines are bidirectional three-state signals connected to the system data bus. The outputs are enabled in the Program condition during the I/O Read to output the contents of an Address register, a Status register, the Temporary register or a Word Count register to the CPU. The outputs are disabled and the inputs are read during an I/O Write cycle when the CPU is programming the 82C37A-5 control registers. During DMA cycles the most significant 8 bits of the address are output onto the data bus to be strobed into an external latch by ADSTB. In memory-to-memory operations, data from the memory comes into the 82C37A-5 on the data bus during the read-from-memory transfer. In the write-to-memory transfer, the data bus outputs place the data into the new memory location.
$\overline{IOR}$	I/O	**I/O READ:** I/O Read is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to read the control registers. In the Active cycle, it is an output control signal used by the 82C37A-5 to access data from a peripheral during a DMA Write transfer.
$\overline{IOW}$	I/O	**I/O WRITE:** I/O Write is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to load information into the 82C37A-5. In the Active cycle, it is an output control signal used by the 82C37A-5 to load data to the peripheral during a DMA Read transfer.

Table 1. Pin Description (Continued)

Symbol	Type	Name and Function
$\overline{EOP}$	I/O	**END OF PROCESS:** End of Process is an active low bidirectional signal. Information concerning the completion of DMA services is available at the bidirectional $\overline{EOP}$ pin. The 82C37A-5 allows an external signal to terminate an active DMA service. This is accomplished by pulling the $\overline{EOP}$ input low with an external $\overline{EOP}$ signal. The 82C37A-5 also generates a pulse when the terminal count (TC) for any channel is reached. This generates an $\overline{EOP}$ signal which is output through the $\overline{EOP}$ Line. The reception of $\overline{EOP}$, either internal or external, will cause the 82C37A-5 to terminate the service, reset the request, and, if Autoinitialize is enabled, to write the base registers to the current registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel by $\overline{EOP}$ unless the channel is programmed for Autoinitialize. In that case, the mask bit remains unchanged. During memory-to-memory transfers, $\overline{EOP}$ will be output when the TC for channel 1 occurs. $\overline{EOP}$ should be tied high with a pull-up resistor if it is not used to prevent erroneous end of process inputs.
A0–A3	I/O	**ADDRESS:** The four least significant address lines are bidirectional three-state signals. In the Idle cycle they are inputs and are used by the CPU to address the register to be loaded or read. In the Active cycle they are outputs and provide the lower 4 bits of the output address.
A4–A7	O	**ADDRESS:** The four most significant address lines are three-state outputs and provide 4 bits of address. These lines are enabled only during the DMA service.
HRQ	O	**HOLD REQUEST:** This is the Hold Request to the CPU and is used to request control of the system bus. If the corresponding mask bit is clear, the presence of any valid DREQ causes 82C37A-5 to issue the HRQ. After HRQ goes active at least one clock cycle (TCY) must occur before HLDA goes active.
DACK0–DACK3	O	**DMA ACKNOWLEDGE:** DMA Acknowledge is used to notify the individual peripherals when one has been granted a DMA cycle. The sense of these lines is programmable. Reset initializes them to active low.
AEN	O	**ADDRESS ENABLE:** Address Enable enables the 8-bit latch containing the upper 8 address bits onto the system address bus. AEN can also be used to disable other system bus drivers during DMA transfers. AEN is active HIGH.
ADSTB	O	**ADDRESS STROBE:** The active high, Address Strobe is used to strobe the upper address byte into an external latch.
$\overline{MEMR}$	O	**MEMORY READ:** The Memory Read signal is an active low three-state output used to access data from the selected memory location during a DMA Read or a memory-to-memory transfer.
$\overline{MEMW}$	O	**MEMORY WRITE:** The Memory Write is an active low three-state output used to write data to the selected memory location during a DMA Write or a memory-to-memory transfer.
PIN5	I	**PIN5:** This pin should always be at a logic HIGH level. An internal pull-up resistor will establish a logic HIGH when the pin is left floating. It is recommended, however, that PIN5 be connected to V_{CC}.

FUNCTIONAL DESCRIPTION

The 82C37A-5 block diagram includes the major logic blocks and all of the internal registers. The data interconnection paths are also shown. Not shown are the various control signals between the blocks. The 82C37A-5 contains 344 bits of internal memory in the form of registers. Figure 3 lists these registers by name and shows the size of each. A detailed description of the registers and their functions can be found under Register Description.

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	6 bits	4
Mask Register	4 bits	1
Request Register	4 bits	1

Figure 3. 82C37A-5 Internal Registers

The 82C37A-5 contains three basic blocks of control logic. The Timing Control block generates internal timing and external control signals for the 82C37A-5. The Program Command Control block decodes the various commands given to the 82C37A-5 by the microprocessor prior to servicing a DMA Request. It also decodes the Mode Control word used to select the type of DMA during the servicing. The Priority Encoder block resolves priority contention between DMA channels requesting service simultaneously.

DMA Operation

The 82C37A-5 is designed to operate in two major cycles. These are called Idle and Active cycles. Each device cycle is made up of a number of states. The 82C37A-5 can assume seven separate states, each composed of one full clock period. State 1 (S1) is the inactive state. It is entered when the 82C37A-5 has no valid DMA requests pending. While in S1, the DMA controller is inactive but may be in the Program Condition, being programmed by the processor. State 0 (S0) is the first state of a DMA service. The 82C37A-5 has requested a hold but the processor has not yet returned an acknowledge. The 82C37A-5 may still be programmed until it receives HLDA from the CPU. An acknowledge from the CPU will signal that DMA transfers may begin. S1, S2, S3 and S4 are the working states of the DMA service. If more time is needed to complete a transfer than is available with normal timing, wait states (SW) can be inserted between S2 or S3 and S4 by the use of the Ready line on the 82C37A-5. Note that the data is transferred directly from the I/O device to memory (or vice versa) with $\overline{IOR}$ and $\overline{MEMW}$ (or $\overline{MEMR}$ and $\overline{IOW}$) being active at the same time. The data is not read into or driven out of the 82C37A-5 in I/O-to-memory or memory-to-I/O DMA transfers.

Memory-to-memory transfers require a read-from and a write-to-memory to complete each transfer. The states, which resemble the normal working states, use two digit numbers for identification. Eight states are required for a single transfer. The first four states (S11, S12, S13, S14) are used for the read-from-memory half and the last four states (S21, S22, S23, S24) for the write-to-memory half of the transfer.

IDLE CYCLE

When no channel is requesting service, the 82C37A-5 will enter the Idle cycle and perform "S1" states. In this cycle the 82C37A-5 will sample the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device will also sample $\overline{CS}$, looking for an attempt by the microprocessor to write or read the internal registers of the 82C37A-5. When $\overline{CS}$ is low and HLDA is low, the 82C37A-5 enters the Program Condition. The CPU can now establish, change or inspect the internal definition of the part by reading from or writing to the internal registers. Address lines A0–A3 are inputs to the device and select which registers will be read or written. The $\overline{IOR}$ and $\overline{IOW}$ lines are used to select and time reads or writes. Due to the number and size of the internal registers, an internal flip-flop is used to generate an additional bit of address. This bit is used to determine the upper or lower byte of the 16-bit Address and Word Count registers. The flip-flop is reset by Master Clear or Reset. A separate software command can also reset this flip-flop.

Special software commands can be executed by the 82C37A-5 in the Program Condition. These commands are decoded as sets of addresses with the $\overline{CS}$ and $\overline{IOW}$. The commands do not make use of the data bus. Instructions include Clear First/Last Flip-Flop and Master Clear.

ACTIVE CYCLE

When the 82C37A-5 is in the Idle cycle and a non-masked channel requests a DMA service, the device

will output an HRQ to the microprocessor and enter the Active cycle. It is in this cycle that the DMA service will take place, in one of four modes:

Single Transfer Mode — In Single Transfer mode the device is programmed to make one transfer only. The word count will be decremented and the address decremented or incremented following each transfer. When the word count "rolls over" from zero to FFFFH, a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so.

DREQ must be held active until DACK becomes active in order to be recognized. If DREQ is held active throughout the single transfer, HRQ will go inactive and release the bus to the system. It will again go active and, upon receipt of a new HLDA, another single transfer will be performed, in 8080A, 8085AH, 80C88, or 80C86 system this will ensure one full machine cycle execution between DMA transfers. Details of timing between the 82C37A-5 and other bus control protocols will depend upon the characteristics of the microprocessor involved.

Block Transfer Mode — In Block Transfer mode the device is activated by DREQ to continue making transfers during the service until a TC, caused by word count going to FFFFH, or an external End of Process (EOP) is encountered. DREQ need only be held active until DACK becomes active. Again, an Autoinitialization will occur at the end of the service if the channel has been programmed for it.

Demand Transfer Mode — In Demand Transfer mode the device is programmed to continue making transfers until a TC or external $\overline{EOP}$ is encountered or until DREQ goes inactive. Thus transfers may continue until the I/O device has exhausted its data capacity. After the I/O device has had a chance to catch up, the DMA service is re-established by means of a DREQ. During the time between services when the microprocessor is allowed to operate, the intermediate values of address and word count are stored in the 82C37A-5 Current Address and Current Word Count registers. Only an $\overline{EOP}$ can cause an Autoinitialize at the end of the service. $\overline{EOP}$ is generated either by TC or by an external signal.

Cascade Mode — This mode is used to cascade more than one 82C37A-5 together for simple system expansion. The HRQ and HLDA signals from the additional 82C37A-5 are connected to the DREQ and DACK signals of a channel of the initial 82C37A-5. This allows the DMA requests of the additional device to propagate through the priority network circuitry of the preceding device. The priority chain is preserved and the new device must wait for its turn to acknowledge requests. Since the cascade channel of the initial 82C37A-5 is used only for prioritizing the additional device, it does not output any address or control signals of its own. These could conflict with the outputs of the active channel in the added device. The 82C37A-5 will respond to DREQ and DACK but all other outputs except HRQ will be disabled. The ready input is ignored.

Figure 4 shows two additional devices cascaded into an initial device using two of the previous channels. This forms a two level DMA system. More 82C37A-5s could be added at the second level by using the remaining channels of the first level. Additional devices can also be added by cascading into the channels of the second level devices, forming a third level.

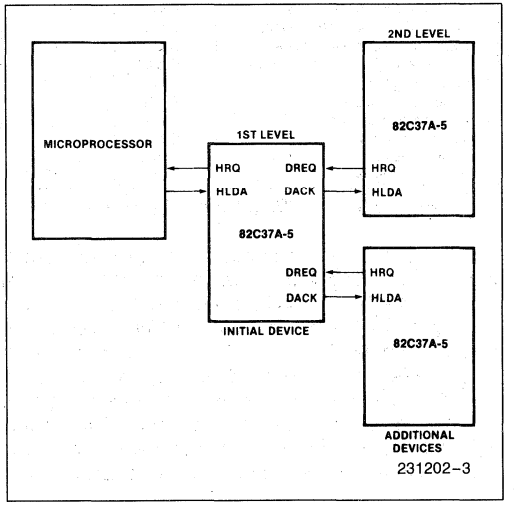


Figure 4. Cascaded 82C37A-5s

TRANSFER TYPES

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write and Verify. Write transfers move data from and I/O device to the memory by activating $\overline{MEMW}$ and $\overline{IOR}$. Read transfers move data from memory to an I/O device by activating $\overline{MEMR}$ and $\overline{IOW}$. Verify transfers are pseudo transfers. The 82C37A-5 operates as in Read or Write transfers generating addresses, and responding to EOP, etc. However, the memory and I/O control lines all remain inactive. The ready input is ignored in verify mode.

Memory-to-Memory — To perform block moves of data from one memory address space to another with a minimum of program effort and time, the 82C37A-5 includes a memory-to-memory transfer feature. Programming a bit in the Command register selects channels 0 to 1 to operate as memory-to-memory transfer channels. The transfer is initiated by setting the software DREQ for channel 0. The

82C37A-5 requests a DMA service in the normal manner. After HLDA is true, the device, using four state transfers in Block Transfer mode, reads data from the memory. The channel 0 Current Address register is the source for the address used and is decremented or incremented in the normal manner. The data byte read from the memory is stored in the 82C37A-5 internal Temporary register. Channel 1 then performs a four-state transfer of the data from the Temporary register to memory using the address in its Current Address register and incrementing or decrementing it in the normal manner. The channel 1 current Word Count is decremented. When the word count of channel 1 goes to FFFFH, a TC is generated causing an $\overline{EOP}$ output terminating the service.

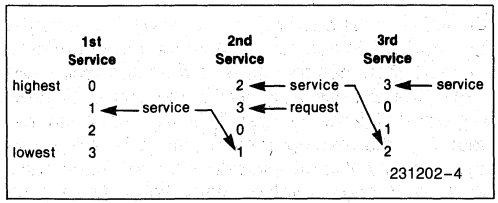
Channel 0 may be programmed to retain the same address for all transfers. This allows a single word to be written to a block of memory.

The 82C37A-5 will respond to external $\overline{EOP}$ signals during memory-to-memory transfers. Data comparators in block search schemes may use this input to terminate the service when a match is found. The timing of memory-to-memory transfers is found in Figure 12. Memory-to-memory operations can be detected as an active AEN with no DACK outputs.

Autoinitialize — By programming a bit in the Mode register, a channel may be set up as an Autoinitialize channel. During Autoinitialize initialization, the original values of the Current Address and Current Word Count registers are automatically restored from the Base Address and Base Word count registers of that channel following $\overline{EOP}$. The base registers are loaded simultaneously with the current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not altered when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to perform another DMA service, without CPU intervention, as soon as a valid DREQ is detected. In order to Autoinitialize both channels in a memory-to-memory transfer, both word counts should be programmed identically. If interrupted externally, $\overline{EOP}$ pulses should be applied in both bus cycles.

Priority — The 82C37A-5 has two types of priority encoding available as software selectable options. The first is Fixed Priority which fixes the channels in priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0. After the recognition of any one channel for service, the other channels are prevented from interfering with that service until it is completed.

The second scheme is Rotating Priority. The last channel to get service becomes the lowest priority channel with the others rotating accordingly.



With Rotating Priority in a single chip DMA system, any device requesting service is guaranteed to be recognized after no more than three higher priority services have occurred. This prevents any one channel from monopolizing the system.

Compressed Timing — In order to achieve even greater throughput where system characteristics permit, the 82C37A-5 can compress the transfer time to two clock cycles. From Figure 11 it can be seen that state S3 is used to extend the access time of the read pulse. By removing state S3, the read pulse width is made equal to the write pulse width and a transfer consists only of state S2 to change the address and state S4 to perform the read/write. S1 states will still occur when A8–A15 need updating (see Address Generation). Timing for compressed transfers is found in Figure 14.

Address Generation — In order to reduce pin count, the 82C37A-5 multiplexes the eight higher order address bits on the data lines. State S1 is used to output the higher order address bits to an external latch from which they may be placed on the address bus. The falling edge of Address Strobe (ADSTB) is used to load these bits from the data lines to the latch. Address Enable (AEN) is used to enable the bits onto the address bus through a three-state enable. The lower order address bits are output by the 82C37A-5 directly. Lines A0–A7 should be connected to the address bus. Figure 11 shows the time relationships between CLK, AEN, ADSTB, DB0–DB7 and A0–A7.

During Block and Demand Transfer mode services, which include multiple transfers, the addresses generated will be sequential. For many transfers the data held in the external address latch will remain the same. This data need only change when a carry or borrow from A7 to A8 takes place in the normal sequence of addresses. To save time and speed transfers, the 82C37A-5 executes S1 states only when updating of A8–A15 in the latch is necessary. This means for long services, S1 states and Address Strobes may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

REGISTER DESCRIPTION

Current Address Register — Each channel has a 16-bit Current Address register. This register holds

the value of the address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written or read by the microprocessor in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialize takes place only after an $\overline{EOP}$.

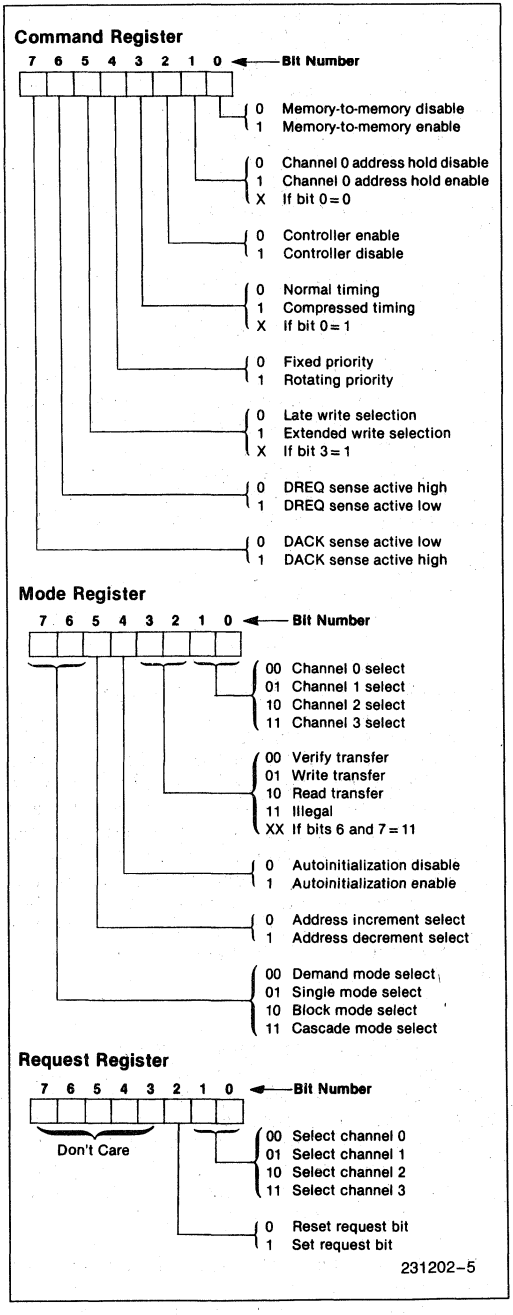
Current Word Register — Each channel has a 16-bit Current Word Count register. This register determines the number of transfers to be performed. The actual number of transfers will be one more than the number programmed in the Current Word Count register (i.e., programming a count of 100 will result in 101 transfers). The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes from zero to FFFFH, a TC will be generated. This register is loaded or read in successive 8-bit bytes by the microprocessor in the Program Condition. Following the end of a DMA service it may also be reinitialized by an Autoinitialization back to its original value. Autoinitialize can occur only when an $\overline{EOP}$ occurs. If it is not Autoinitialized, this register will have a count of FFFFH after TC.

Base Address and Base Word Count Registers — Each channel has a pair of Base Address and Base Word Count registers. These 16-bit registers store the original value of their associated current registers. During Autoinitialize these values are used to restore the current registers to their original values. The base registers are written simultaneously with their corresponding current register in 8-bit bytes in the Program Condition by the microprocessor. These registers cannot be read by the microprocessor.

Command Register — This 8-bit register controls the operation of the 82C37A-5. It is programmed by the microprocessor in the Program Condition and is cleared by Reset or a Master Clear instruction. The following table lists the function of the command bits. See Figure 6 for address coding.

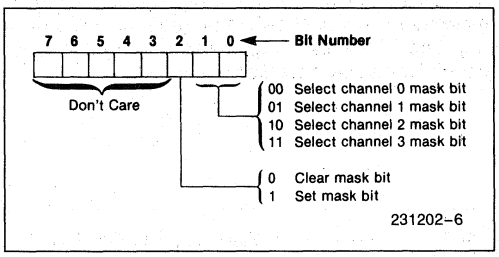
Mode Register — Each channel has a 6-bit Mode register associated with it. When the register is being written to by the microprocessor in the Program Condition, bits 0 and 1 determine which channel Mode register is to be written.

Request Register — The 82C37A-5 can respond to requests for DMA service which are initiated by software as well as by a DREQ. Each channel has a request bit associated with it in the 4-bit Request register. These are non-maskable and subject to prioritization by the Priority Encoder network. Each register bit is set or reset separately under software control or is cleared upon generation of a TC or external $\overline{EOP}$. The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word. See Figure 5 for register ad-

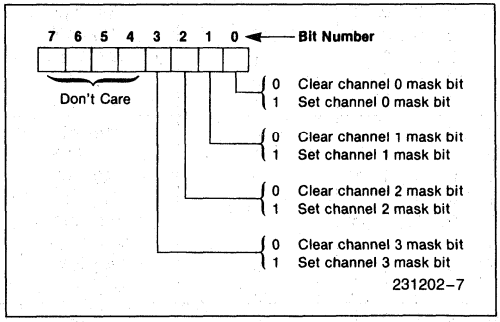


dress coding. In order to make a software request, the channel must be in Block Mode.

Mask Register — Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an $\overline{EOP}$ if the channel is not programmed for Autoinitialize. Each bit of the 4-bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. See Figure 5 for instruction addressing.



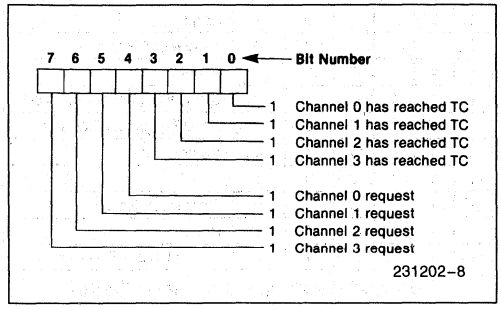
All four bits of the Mask register may also be written with a single command.



Register	Operation	Signals						
		$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0
Command	Write	0	1	0	1	0	0	0
Mode	Write	0	1	0	1	0	1	1
Request	Write	0	1	0	1	0	0	1
Mask	Set/Reset	0	1	0	1	0	1	0
Mask	Write	0	1	0	1	1	1	1
Temporary	Read	0	0	1	1	1	0	1
Status	Read	0	0	1	1	0	0	0

Figure 5. Definition of Register Codes

Status Register — The Status register is available to be read out of the 82C37A-5 by the microprocessor. It contains information about the status of the devices at this point. This information includes which channels have reached a terminal count and which channels have pending DMA requests. Bits 0–3 are set every time a TC is reached by that channel or an external $\overline{EOP}$ is applied. These bits are cleared upon Reset and on each Status Read. Bits 4–7 are set whenever their corresponding channel is requesting service.



Temporary Register — The Temporary register is used to hold data during memory-to-memory transfers. Following the completion of the transfers, the last word moved can be read by the microprocessor in the Program Condition. The Temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a Reset.

Software Commands — These are additional special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The three software commands are:

Clear First/Last Flip-Flop: This command is executed prior to writing or reading new address or word count information to the 82C37A-5. This initializes the flip-flop to a known state so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.

Master Clear: This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary, and Internal First/Last Flip-Flop registers are cleared and the Mask register is set. The 82C37A-5 will enter the Idle cycle.

Clear Mask Register: This command clears the mask bits of all four channels, enabling them to accept DMA requests.

Figure 6 lists the address codes for the software commands:

Signals						Operation
A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip-Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Clear Mask Register
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Figure 6. Software Command Codes

PROGRAMMING

The 82C37A-5 will accept programming from the host processor any time that HLDA is inactive; this is true even if HRQ is active. The responsibility of the host is to assure that programming and HLDA are mutually exclusive. Note that a problem can occur if a DMA request occurs, on an unmasked channel while the 82C37A-5 is being programmed. For instance, the CPU may be starting to reprogram the two byte Address register of channel 1 when channel 1 receives a DMA request. If the 82C37A-5 is enabled (bit 2 in the command register is 0) and channel 1 is unmasked, a DMA service will occur after only one byte of the Address register has been reprogrammed. This can be avoided by disabling the controller (setting bit 2 in the command register) or masking the channel before programming any other registers. Once the programming is complete, the controller can be enabled/unmasked.

Channel	Register	Operation	Signals							Internal Flip-Flop	Data Bus DB0–DB7
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base and Current Address	Write	0	1	0	0	0	0	0	0	A0–A7
			0	1	0	0	0	0	0	1	A8–A15
	Current Address	Read	0	0	1	0	0	0	0	0	A0–A7
			0	0	1	0	0	0	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	0	0	1	0	W0–W7
			0	1	0	0	0	0	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	0	0	1	0	W0–W7
			0	0	1	0	0	0	1	1	W8–W15
1	Base and Current Address	Write	0	1	0	0	0	1	0	0	A0–A7
			0	1	0	0	0	1	0	1	A8–A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0–A7
			0	0	1	0	0	1	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	0	1	1	0	W0–W7
			0	1	0	0	0	1	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	0	1	1	0	W0–W7
			0	0	1	0	0	1	1	1	W8–W15
2	Base and Current Address	Write	0	1	0	0	1	0	0	0	A0–A7
			0	1	0	0	1	0	0	1	A8–A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0–A7
			0	0	1	0	1	0	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	1	0	1	0	W0–W7
			0	1	0	0	1	0	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	1	0	1	0	W0–W7
			0	0	1	0	1	0	1	1	W8–W15
3	Base and Current Address	Write	0	1	0	0	1	1	0	0	A0–A7
			0	1	0	0	1	1	0	1	A8–A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0–A7
			0	0	1	0	1	1	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	1	1	1	0	W0–W7
			0	1	0	0	1	1	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	1	1	1	0	W0–W7
			0	0	1	0	1	1	1	1	W8–W15

Figure 7. Word Count and Address Register Command Codes

After power-up it is suggested that all internal locations, especially the Mode registers, be loaded with some valid value. This should be done even if some channels are unused.

APPLICATION INFORMATION

Figure 8 shows a convenient method for configuring a DMA system with the 82C37A-5 controller and an 8080A/8085AH microprocessor system. The multimode DMA controller issues a HRQ to the processor whenever there is at least one valid DMA request from a peripheral device. When the processor replies with a HLDA signal, the 82C37A-5 takes control of the address bus, the data bus and the control bus. The address for the first transfer operation comes out in two bytes — the least significant 8 bits on the eight address outputs and the most significant 8 bits on the data bus. The contents of the data bus are then latched into the 8-bit latch to complete the full 16 bits of the address bus. After the initial transfer takes place, the latch is updated only after a carry or borrow is generated in the least significant address byte. Four DMA channels are provided when one 82C37A-5 is used.

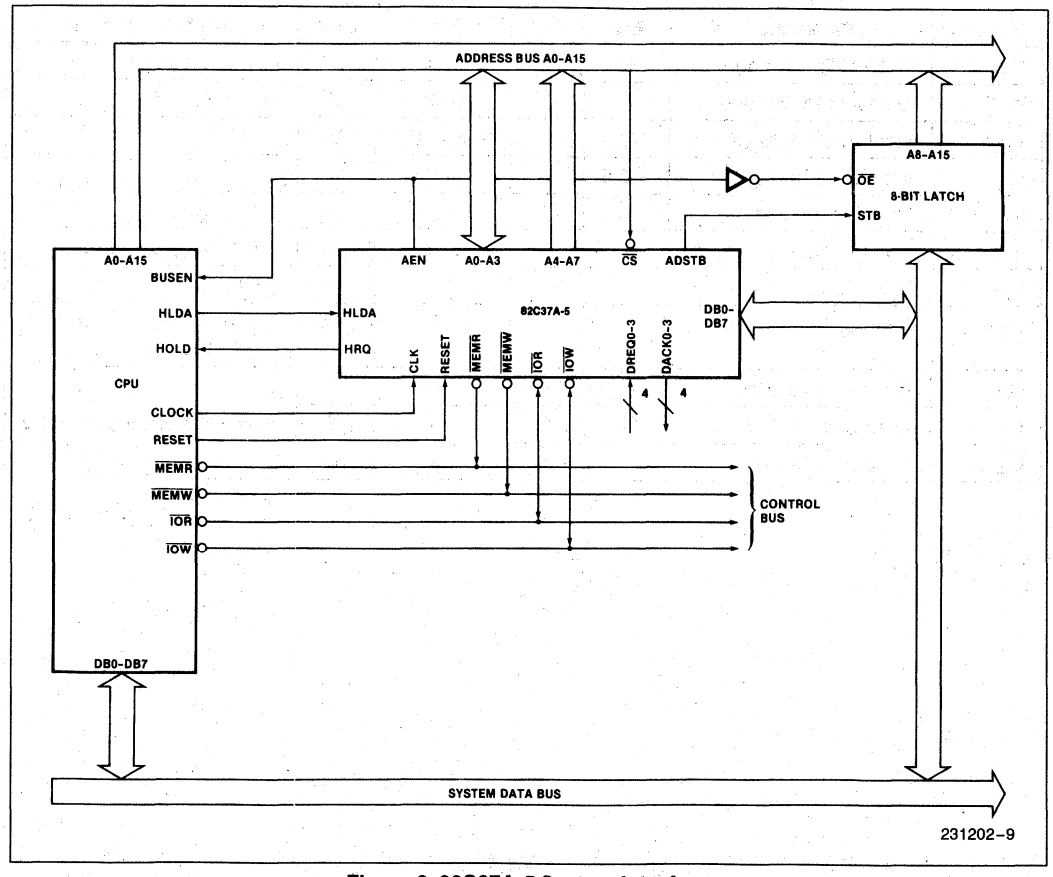


Figure 8. 82C37A-5 System Interface

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature under Bias ..... 0°C to 70°C
Case Temperature ..... 0°C to +75°C
Storage Temperature ..... −55°C to +150°C
Voltage on Any Pin with Respect to Ground ..... −0.5V to +7V
Power Dissipation ..... 1.0 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, T_{CASE} = 0°C to 75°C, V_{CC} = +5.0V ±5%, GND = 0V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{OH}	Output High Voltage	3.7			V	I_{OH} = −1.0 mA
V_{OL}	Output LOW Voltage			0.40	V	I_{OL} = 3.2 mA
V_{IH}	Input HIGH Voltage	2.2		V_{CC} + 0.5	V	
V_{IL}	Input LOW Voltage	−0.5		0.8	V	
I_{LI}	Input Load Current			±10	μA	0V ≤ V_{IN} ≤ V_{CC}
I_{LO}	Output Leakage Current			±10	μA	0V ≤ V_{OUT} ≤ V_{CC}
I_{CC}	V_{CC} Supply Current			10	mA	(Note 1)
I_{CCS}	Standby Supply Current			10	μA	HLDA = 0V, V_{IL} = 0V, V_{IH} = V_{CC}
C_O	Output Capacitance		4	8	pF	fc = 1.0 MHz, Inputs = 0V
C_I	Input Capacitance		8	15	pF	
C_{IO}	I/O Capacitance		10	18	pF	

A.C. CHARACTERISTICS—DMA (MASTER) MODE

T_A = 0°C to 70°C, T_{CASE} = 0°C to 75°C, V_{CC} = +5V ±5%, GND = 0V

Symbol	Parameter	Min	Max	Unit
TAEL	AEN HIGH from CLK LOW (S1) Delay Time		200	ns
TAET	AEN LOW from CLK HIGH (SI) Delay Time		130	ns
TAFAB	ADR Active to Float Delay from CLK HIGH		90	ns
TAFC	$\overline{READ}$ or $\overline{WRITE}$ Float from CLK HIGH		120	ns
TAFDB	DB Active to Float Delay from CLK HIGH		170	ns
TAHR	ADR from $\overline{READ}$ HIGH Hold Time	TCY-100		ns
TAHS	DB from ADSTB LOW Hold Time	30		ns
TAHW	ADR from $\overline{WRITE}$ HIGH Hold Time	TCY-50		ns
TAK	DACK Valid from CLK LOW Delay Time (Note 3)		170	ns
	$\overline{EOP}$ HIGH from CLK HIGH Delay Time (Note 4)		170	ns
	$\overline{EOP}$ LOW from CLK HIGH Delay Time		170	ns
TASM	ADR Stable from CLK HIGH		170	ns
TASS	DB to ADSTB LOW Setup Time	100		ns
TCH	Clock High Time (Transitions ≤ 10 ns)	68		ns
TCL	Clock LOW Time (Transitions ≤ 10 ns)	68		ns
TCY	CLK Cycle Time	200		ns
TDCL	CLK HIGH to $\overline{READ}$ or $\overline{WRITE}$ LOW Delay (Note 2)		190	ns
TDCTR	$\overline{READ}$ HIGH from CLK HIGH (S4) Delay Time (Note 2)		190	ns
TDCTW	$\overline{WRITE}$ HIGH from CLK HIGH (S4) Delay Time (Note 2)		130	ns
TDQ1	HRQ Valid from CLK HIGH Delay Time		120	ns
TEPS	$\overline{EOP}$ LOW from CLK LOW Setup Time	40		ns
TEPW	$\overline{EOP}$ Pulse Width	220		ns
TFAAB	ADR Float to Active Delay from CLK HIGH		170	ns
TFAC	$\overline{READ}$ or $\overline{WRITE}$ Active from CLK HIGH		150	ns
TFADB	DB Float to Active Delay from CLK HIGH		200	ns
THS	HLDA Valid to CLK HIGH Setup Time	75		ns
TIDH	Input Data from $\overline{MEMR}$ HIGH Hold Time	0		ns
TIDS	Input Data to $\overline{MEMR}$ HIGH Setup Time	170		ns
TODH	Output Data from $\overline{MEMW}$ HIGH Hold Time	10		ns
TODV	Output Data Valid to $\overline{MEMW}$ HIGH	125		ns
TQS	DREQ to CLK LOW (SI, S4) Setup Time (Note 3)	0		ns
TRH	CLK to READY LOW Hold Time	20		ns
TRS	READY to CLK LOW Setup Time	60		ns
TSTL	ADSTB HIGH from CLK HIGH Delay Time		130	ns
TSTT	ADSTB LOW from CLK HIGH Delay Time		90	ns

3

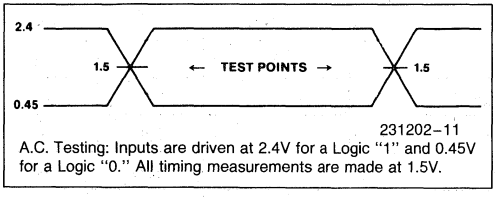
A.C. CHARACTERISTICS—PERIPHERAL (SLAVE) MODE

T_A = 0°C to 70°C, T_{CASE} = 0°C to 75°C, V_{CC} = +5V ±5%, GND = 0V

Symbol	Parameter	Min	Max	Unit
TAR	ADR Valid or $\overline{CS}$ LOW to $\overline{READ}$ LOW	50		ns
TAW	ADR Valid to $\overline{WRITE}$ HIGH Setup Time	130		ns
TCW	CS LOW to $\overline{WRITE}$ HIGH Setup Time	130		ns
TDW	Data Valid to $\overline{WRITE}$ HIGH Setup Time	130		ns
TRA	ADR or CS Hold from $\overline{READ}$ HIGH	0		ns
TRDE	Data Access from $\overline{READ}$ LOW		140	ns
TRDF	DB Float Delay from $\overline{READ}$ HIGH	0	70	ns
TRSTD	Power Supply HIGH to RESET LOW Setup Time	500		ns
TRSTS	RESET to First $\overline{IOWR}$	2TCY		ns
TRSTW	RESET Pulse Width	300		ns
TRW	$\overline{READ}$ Width	200		ns
TWA	ADR from $\overline{WRITE}$ HIGH Hold Time	20		ns
TWC	CS HIGH from $\overline{WRITE}$ HIGH Hold Time	20		ns
TWD	Data from $\overline{WRITE}$ HIGH Hold Time	30		ns
TWWS	Write Width	160		ns

NOTES:
1. Input frequency 5 MHz, when RESET, V_{IN} = 0V/V_{CC}, C_L = 0 pF.
2. The net $\overline{IOW}$ or $\overline{MEMW}$ Pulse width for normal write will be TCY-100 ns and for extended write will be 2TCY-100 ns. The net $\overline{IOR}$ or $\overline{MEMR}$ pulse width for normal read will be 2TCY-50 ns and for compressed read will be TCY-50 ns.
3. DREQ and DACK signals may be active high or active low. Timing diagrams assume the active high mode for DREQ and active low for DACK.
4. $\overline{EOP}$ is an open collector output. This parameter assumes the presence of a 2.2K pullup to V_{CC}.

A.C. TESTING INPUT/OUTPUT WAVEFORM



231202-11

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0." All timing measurements are made at 1.5V.

WAVEFORMS

SLAVE MODE WRITE TIMING

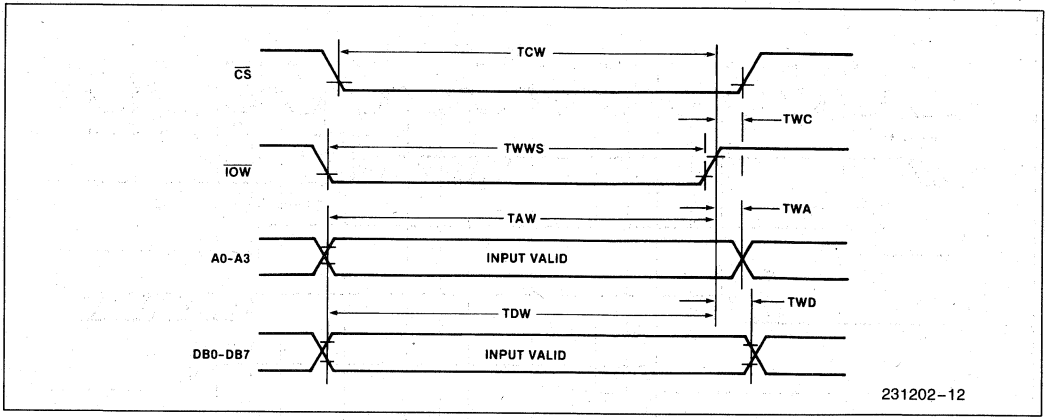


Figure 9. Slave Mode Write

SLAVE MODE READ TIMING

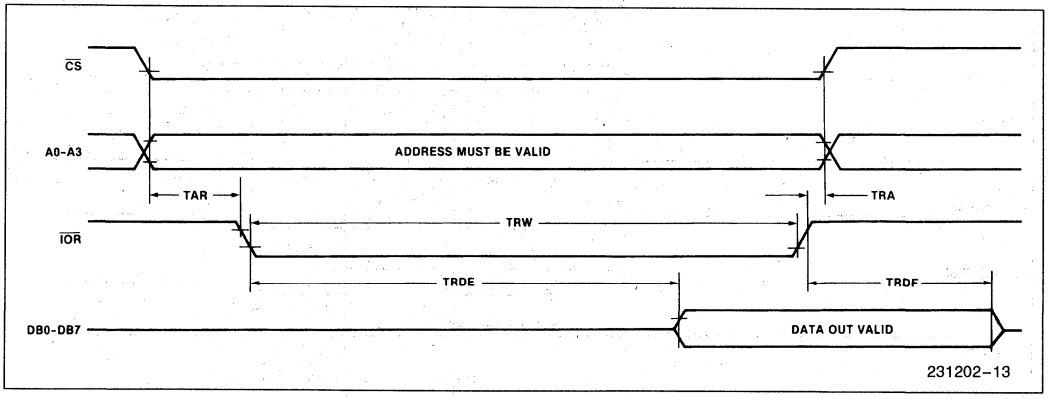


Figure 10. Slave Mode Read

WAVEFORMS (Continued)

DMA TRANSFER TIMING

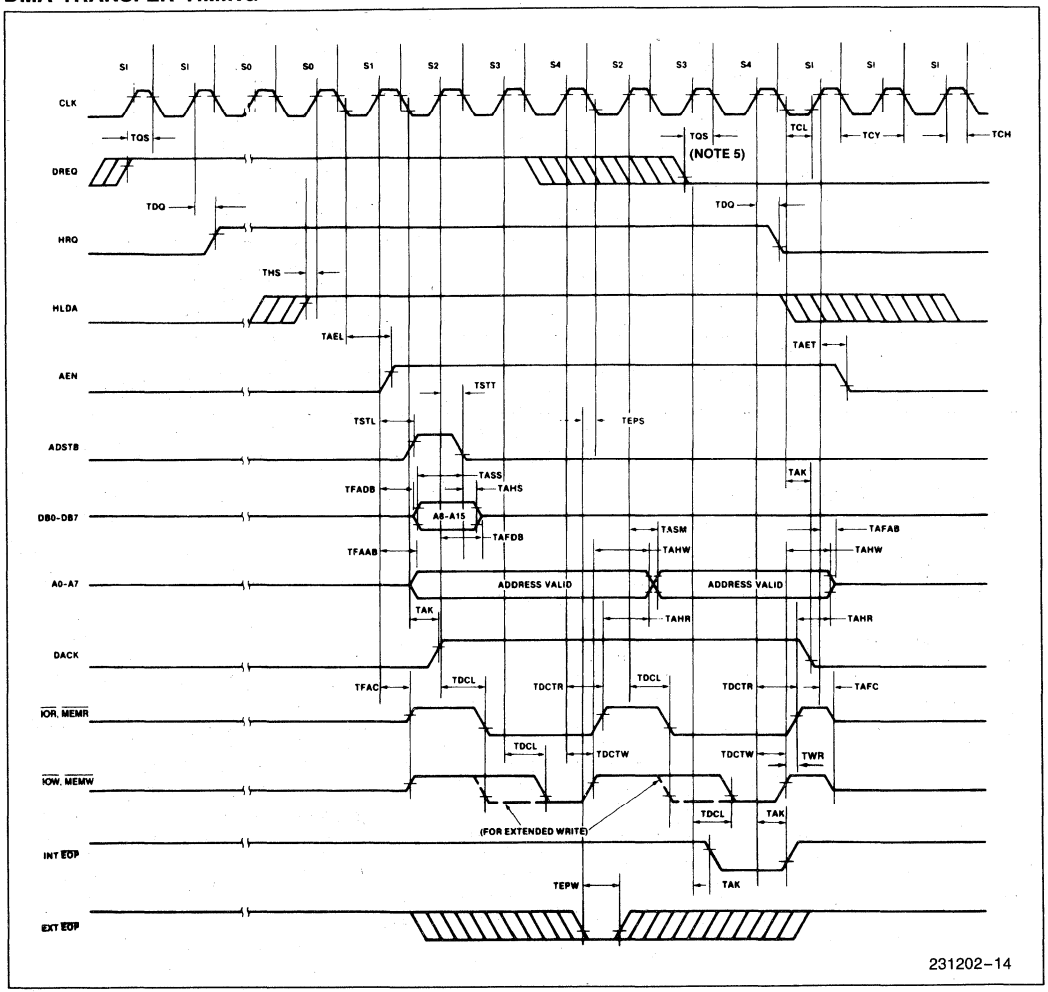


Figure 11. DMA Transfer

WAVEFORMS (Continued)

MEMORY-TO-MEMORY TRANSFER TIMING

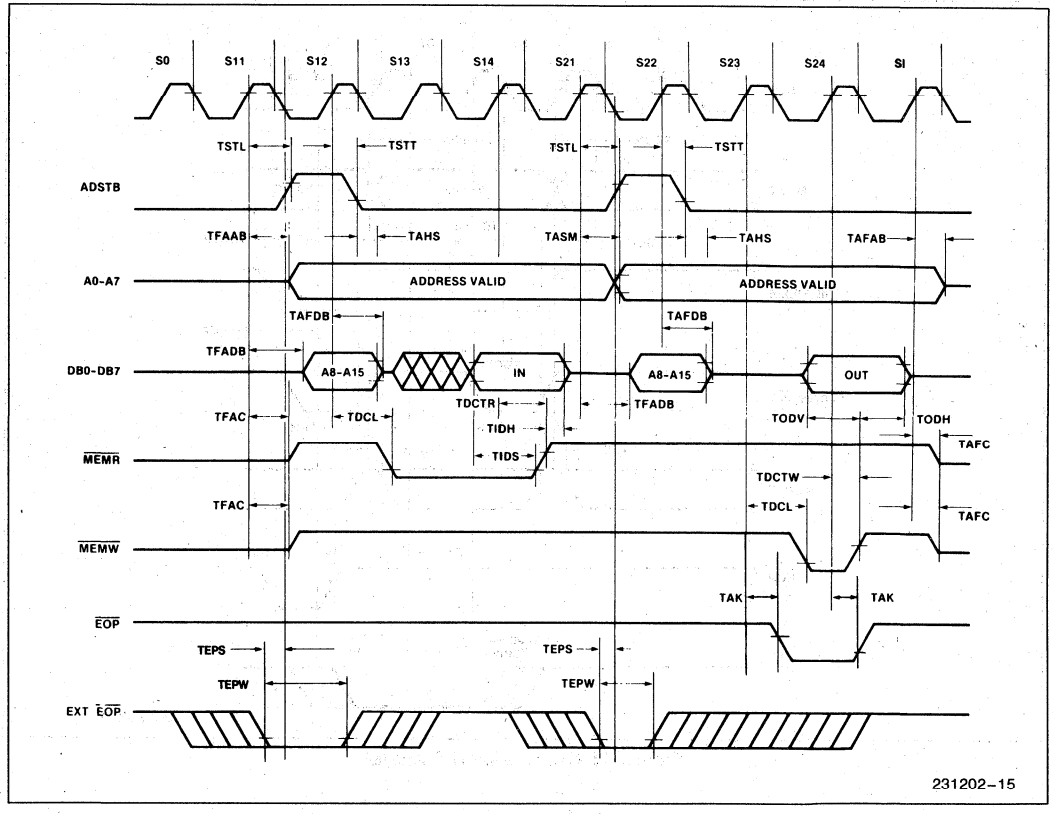


Figure 12. Memory-to-Memory Transfer

READY TIMING

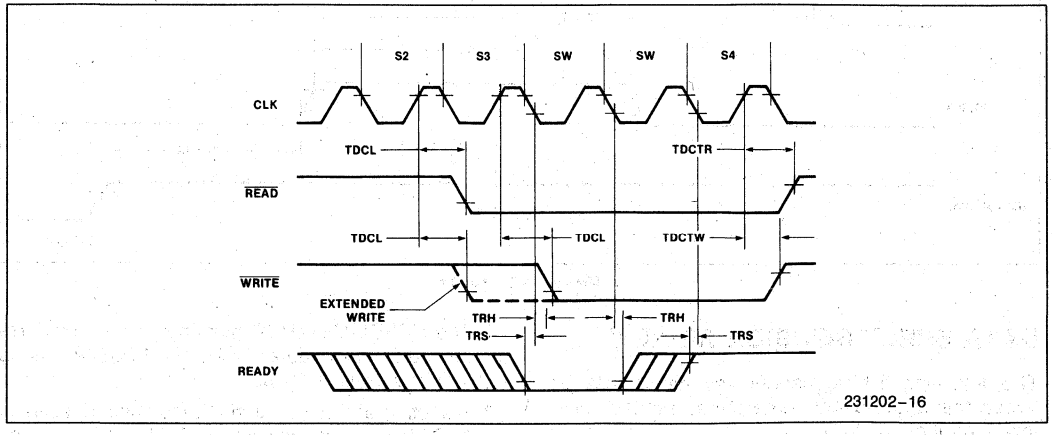


Figure 13. Ready

WAVEFORMS (Continued)

COMPRESSED TRANSFER TIMING

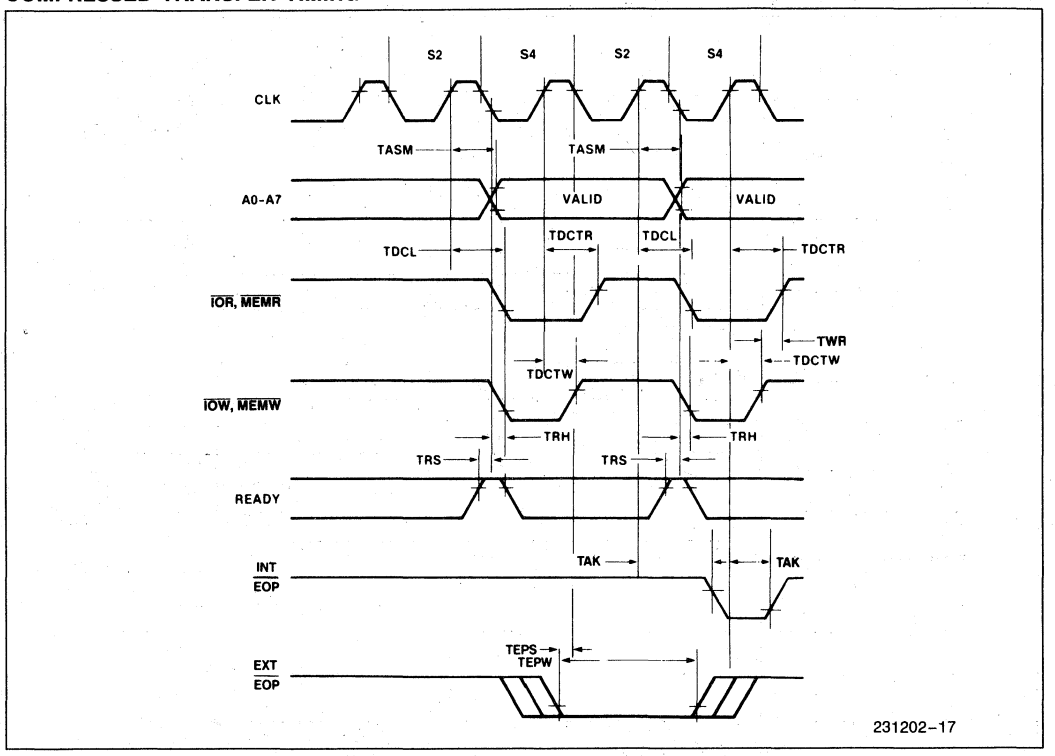


Figure 14. Compressed Transfer

RESET TIMING

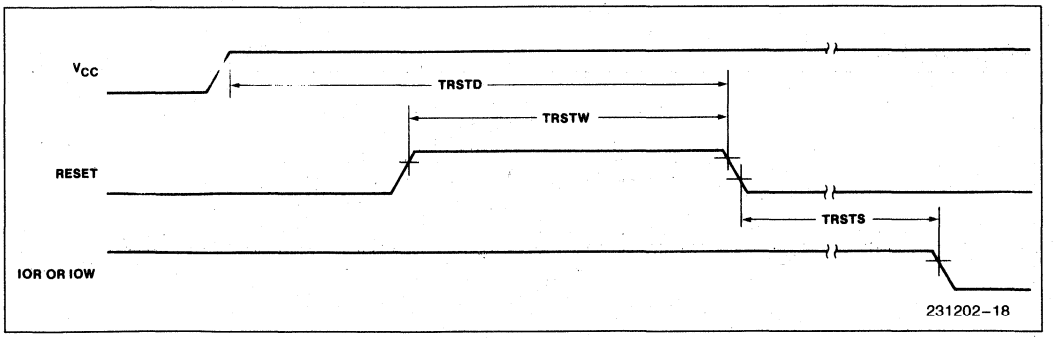


Figure 15. Reset

DATA SHEET REVISION REVIEW

The following list represents key differences between this and the -004 data sheet. Please review this summary carefully.

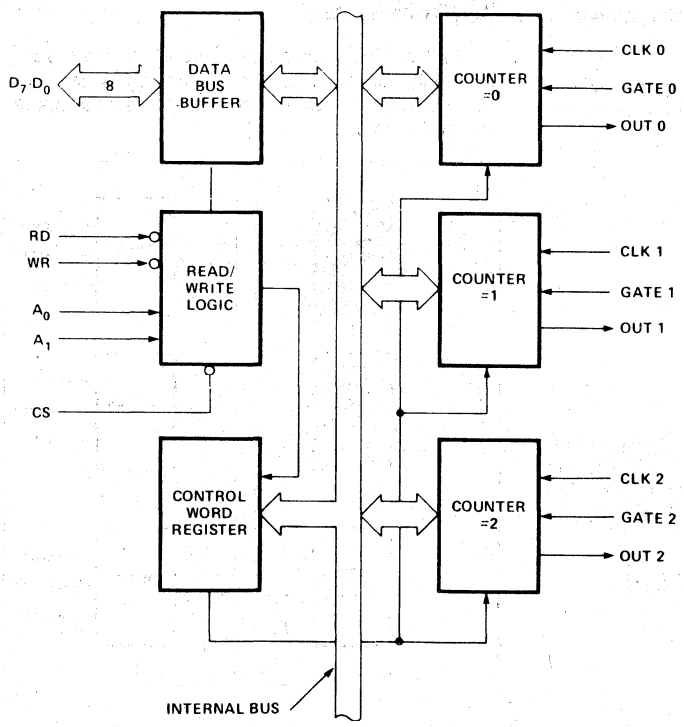
1. The "PRELIMINARY" markings have been removed from the data sheet. The 82C37A-5 is no longer a preliminary part.
2. A section of the Functional Description describing 82C37A-5 operation with the 8085 CPU has been deleted.

8253/8253-5
PROGRAMMABLE INTERVAL TIMER

- **MCS-85™ Compatible 8253-5**
- **3 Independent 16-Bit Counters**
- **DC to 2.6 MHz**
- **Programmable Counter Modes**
- **Count Binary or BCD**
- **Single +5V Supply**
- **Available in EXPRESS**
  - **Standard Temperature Range**
  - **Extended Temperature Range**

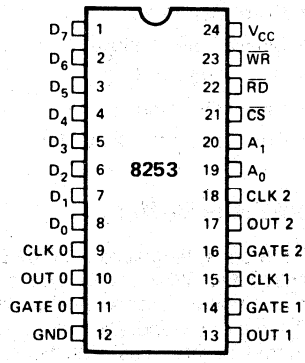
The Intel® 8253 is a programmable counter/timer device designed for use as an Intel microcomputer peripheral. It uses NMOS technology with a single +5V supply and is packaged in a 24-pin plastic DIP.

It is organized as 3 independent 16-bit counters, each with a count rate of up to 2.6 MHz. All modes of operation are software programmable.



231306-1

Figure 1. Block Diagram



231306-2

Figure 2. Pin Configuration

November 1986
Order Number: 231306-001

FUNCTIONAL DESCRIPTION

General

The 8253 is programmable interval timer/counter specifically designed for use with the Intel™ Microcomputer systems. Its function is that of a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 8253 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in systems software, the programmer configures the 8253 to match his requirements, initializes one of the counters of the 8253 with the desired quantity, then upon command the 8253 will count out the delay and interrupt the CPU when it has completed its tasks. It is easy to see that the software overhead is minimal and that multiple delays can easily be maintained by assignment of priority levels.

Other counter/timer functions that are non-delay in nature but also common to most microcomputers can be implemented with the 8253.

- Programmable Rate Generator
- Event Counter
- Binary Rate Multiplier
- Real Time Clock
- Digital One-Shot
- Complex Motor Controller

Data Bus Buffer

The 3-state, bi-directional, 8-bit buffer is used to interface the 8253 to the system data bus. Data is transmitted or received by the buffer upon execution of INput or OUTput CPU instructions. The Data Bus Buffer has three basic functions.

1. Programming the MODES of the 8253.
2. Loading the count registers.
3. Reading the count values.

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and in turn generates control signals for overall device operation. It is enabled or disabled by CS so that no operation can occur to change the function unless the device has been selected by the system logic.

$\overline{RD}$ (Read)

A "low" on this input informs the 8253 that the CPU is inputting data in the form of a counters value.

$\overline{WR}$ (Write)

A "low" on this input informs the 8253 that the CPU is outputting data in the form of mode information or loading counters.

A0, A1

These inputs are normally connected to the address bus. Their function is to select one of the three counters to be operated on and to address the control word register for mode selection.

$\overline{CS}$ (Chip Select)

A "low" on this input enables the 8253. No reading or writing will occur unless the device is selected. The $\overline{CS}$ input has no effect upon the actual operation of the counters.

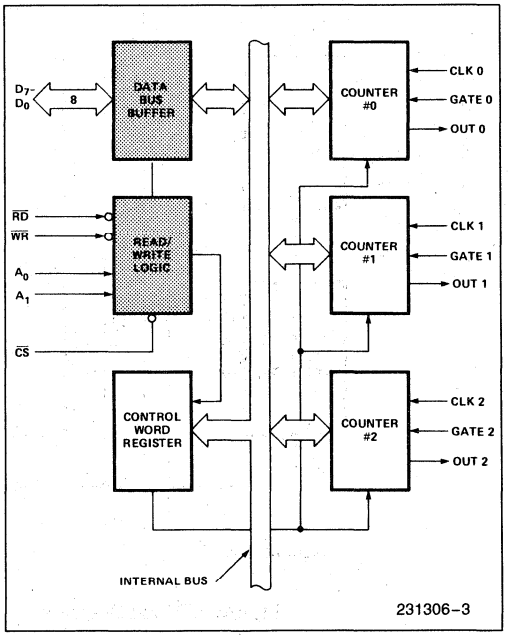


Figure 3. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Load Counter No. 0
0	1	0	0	1	Load Counter No. 1
0	1	0	1	0	Load Counter No. 2
0	1	0	1	1	Write Mode Word
0	0	1	0	0	Read Counter No. 0
0	0	1	0	1	Read Counter No. 1
0	0	1	1	0	Read Counter No. 2
0	0	1	1	1	No-Operation 3-State
1	X	X	X	X	Disable 3-State
0	1	1	X	X	No-Operation 3-State

Control Word Register

The Control Word Register is selected when A0, A1 are 11. It then accepts information from the data bus buffer and stores it in a register. The information stored in this register controls the operation MODE of each counter, selection of binary or BCD counting and the loading of each count register.

The Control Word Register can only be written into; no read operation of its contents is available.

Counter #0, Counter #1, Counter #2

These three functional blocks are identical in operation so only a single counter will be described. Each Counter consists of a single, 16-bit, pre-settable, DOWN counter. The counter can operate in either binary or BCD and its input, gate and output are configured by the selection of MODES stored in the Control Word Register.

The counters are fully independent and each can have separate MODE configuration and counting operation, binary or BCD. Also, there are special features in the control word that handle the loading of the count value so that software overhead can be minimized for these functions.

The reading of the contents of each counter is available to the programmer with simple READ operations for event counting applications and special commands and logic are included in the 8253 so that the contents of each counter can be read "on the fly" without having to inhibit the clock input.

8253 SYSTEM INTERFACE

The 8253 is a component of the Intel™ Microcomputer systems and interfaces in the same manner as all other peripherals of the family. It is treated by the systems software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A0, A1 connect to the A0, A1 address bus signals of the CPU. The $\overline{CS}$ can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an Intel 8205 for larger systems.

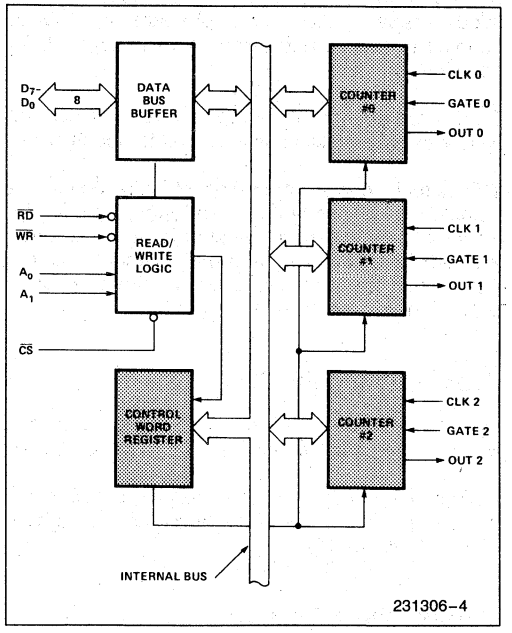


Figure 4. Block Diagram Showing Control Word Register and Counter Functions

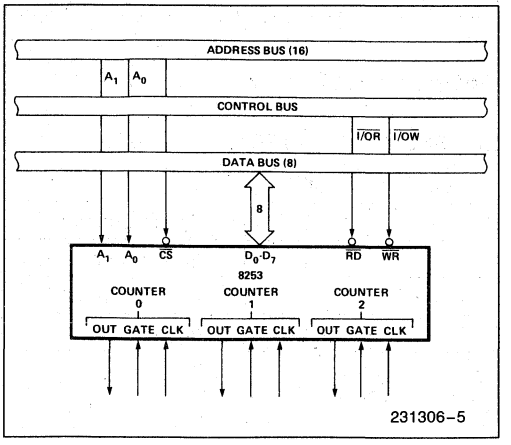


Figure 5. 8253 System Interface

OPERATIONAL DESCRIPTION

General

The complete functional definition of the 8253 is programmed by the systems software. A set of control words *must* be sent out by the CPU to initialize each counter of the 8253 with the desired MODE and quantity information. Prior to initialization, the MODE, count, and output of all counters is undefined. These control words program the MODE, Loading sequence and selection of binary or BCD counting.

Once programmed, the 8253 is ready to perform whatever timing tasks it is assigned to accomplish.

The actual counting operation of each counter is completely independent and additional logic is provided on-chip so that the usual problems associated with efficient monitoring and management of external, asynchronous events or rates to the microcomputer system have been eliminated.

Programming the 8253

All of the MODES for each counter are programmed by the systems software by simple I/O operations.

Each counter of the 8253 is individually programmed by writing a control word into the Control Word Register. (A0, A1 = 11)

Control Word Format

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

Definition Of Control

SC—SELECT COUNTER:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Illegal

RL—READ/LOAD:

RL1	RL0	
0	0	Counter Latching operation (see READ/WRITE Procedure Section).
1	0	Read/Load most significant byte only.
0	1	Read/Load least significant byte only.
1	1	Read/Load least significant byte first, then most significant byte.

M—MODE:

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD:

0	Binary Counter 16-Bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

Counter Loading

The count register is not loaded until the count value is written (one or two bytes, depending on the mode selected by the RL bits), followed by a rising edge and a falling edge of the clock. Any read of the counter prior to that falling clock edge may yield invalid data.

MODE DEFINITION

MODE 0: Interrupt on Terminal Count. The output will be initially low after the mode set operation. After the count is loaded into the selected count register, the output will remain low and the counter will count. When terminal count is reached, the output will go high and remain high until the selected count register is reloaded with the mode or a new count is loaded. The counter continues to decrement after terminal count has been reached.

Rewriting a counter register during counting results in the following:

(1) Write 1st byte stops the current counting.
(2) Write 2nd byte starts the new count.

MODE 1: Programmable One-Shot. The output will go low on the count following the rising edge of the gate input.

The output will go high on the terminal count. If a new count value is loaded while the output is low it will not affect the duration of the one-shot pulse until the succeeding trigger. The current count can be read at any time without affecting the one-shot pulse.

The one-shot is retriggerable, hence the output will remain low for the full count after any rising edge of the gate input.

MODE 2: Rate Generator. Divide by N counter. The output will be low for one period of the input clock. The period from one output pulse to the next equals the number of input counts in the count register. If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value.

The gate input, when low, will force the output high. When the gate input goes high, the counter will start from the initial count. Thus, the gate input can be used to synchronize the counter.

When this mode is set, the output will remain high until after the count register is loaded. The output then can also be synchronized by software.

MODE 3: Square Wave Rate Generator. Similar to MODE 2 except that the output will remain high until one half the count has been completed (or even numbers) and go low for the other half of the count. This is accomplished by decrementing the counter by two on the falling edge of each clock pulse. When the counter reaches terminal count, the state of the output is changed and the counter is reloaded with the full count and the whole process is repeated.

If the count is odd and the output is high, the first clock pulse (after the count is loaded) decrements the count by 1. Subsequent clock pulses decrement the clock by 2. After timeout, the output goes low and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by 3. Subsequent clock pulses decrement the count by 2 until timeout. Then the whole process is repeated. In this way, if the count is odd, the output will be high for $(N + 1)/2$ counts and low for $(N - 1)/2$ counts.

In Modes 2 and 3, if a CLK source other than the system clock is used, GATE should be pulsed immediately following $\overline{WR}$ of a new count value.

MODE 4: Software Triggered Strobe. After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period, then will go high again.

If the count register is reloaded during counting, the new count will be loaded on the next CLK pulse. The count will be inhibited while the GATE input is low.

MODE 5: Hardware Triggered Strobe. The counter will start counting after the rising edge of the trigger input and will go low for one clock period when the terminal count is reached. The counter is retriggerable. The output will not go low until the full count after the rising edge of any trigger.

Signal Status Modes	Low Or Going Low	Rising	High
0	Disables counting	—	Enables counting
1	—	1) Initiates counting 2) Resets output after next clock	—
2	1) Disables counting 2) Sets output immediately high	1) Reloads counter 2) Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	1) Reloads counter 2) Initiates counting	Enables counting
4	Disables counting	—	Enables counting
5	—	Initiates counting	—

Figure 6. Gate Pin Operations Summary

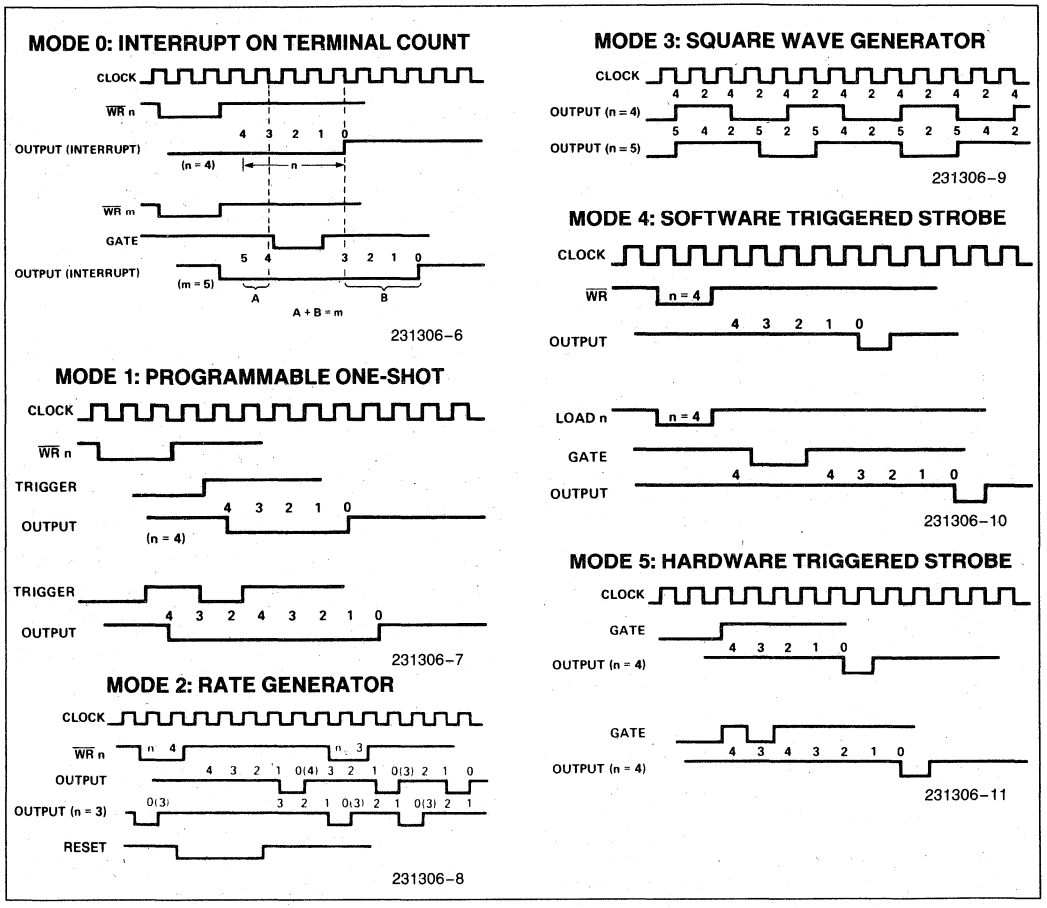


Figure 7. 8253 Timing Diagrams

8253 READ/WRITE PROCEDURE

Write Operations

The systems software must program each counter of the 8253 with the mode and quantity desired. The programmer must write out to the 8253 a MODE control word and the programmed number of count register bytes (1 or 2) prior to actually using the selected counter.

The actual order of the programming is quite flexible. Writing out of the MODE control word can be in any sequence of counter selection, e.g., counter #0 does not have to be first or counter #2 last. Each counter's MODE control word register has a separate address so that its loading is completely sequence independent. (SC0, SC1).

The loading of the Count Register with the actual count value, however, must be done in exactly the sequence programmed in the MODE control word (RL0, RL1). This loading of the counter's count register is still sequence independent like the MODE control word loading, but when a selected count register is to be loaded it *must* be loaded with the number of bytes programmed in the MODE control word (RL0, RL1). The one or two bytes to be loaded in the count register do not have to follow the associated MODE control word. They can be programmed at any time following the MODE control word loading as long as the correct number of bytes is loaded in order.

All counters are down counters. Thus, the value loaded into the count register will actually be decremented. Loading all zeros into a count register will result in the maximum count (2^{16} for Binary or 10^4 for BCD). In MODE 0 the new count will not restart until the load has been completed. It will accept one of two bytes depending on how the MODE control words (RL0, RL1) are programmed. Then proceed with the restart operation.

	MODE Control Word Counter n
LSB	Counter Register byte Counter n
MSB	Counter Register byte Counter n

NOTE:
Format shown is a simple example of loading the 8253 and does not imply that it is the only format that can be used.

Figure 8. Programming Format

			A1	A0
No. 1		MODE Control Word Counter 0	1	1
No. 2		MODE Control Word Counter 1	1	1
No. 3		MODE Control Word Counter 2	1	1
No. 4	LSB	Count Register Byte Counter 1	0	1
No. 5	MSB	Count Register Byte Counter 1	0	1
No. 6	LSB	Count Register Byte Counter 2	1	0
No. 7	MSB	Count Register Byte Counter 2	1	0
No. 8	LSB	Count Register Byte Counter 0	0	0
No. 9	MSB	Count Register Byte Counter 0	0	0

NOTE:
The exclusive addresses of each counter's count register make the task of programming the 8253 a very simple matter, and maximum effective use of the device will result if this feature is fully initilized.

Figure 9. Alternate Programming Formats

Read Operations

In most counter applications it becomes necessary to read the value of the count in progress and make a computational decision based on this quantity. Event counters are probably the most common application that uses this function. The 8253 contains logic that will allow the programmer to easily read the contents of any of the three counters without disturbing the actual count in progress.

There are two methods that the programmer can use to read the value of the counters. The first method involves the use of simple I/O read operations of the selected counter. By controlling the A0, A1 inputs to the 8253 the programmer can select the counter to be read (remember that no read operation of the mode register is allowed A0, A1-11). The only requirement with this method is that in order to assure a stable count reading the actual operation of the selected counter *must be inhibited* either by controlling the Gate input or by external logic that inhibits the clock input. The contents of the counter selected will be available as follows:

First I/O Read contains the least significant byte (LSB).

Second I/O Read contains the most significant byte (MSB).

Due to the internal logic of the 8253 it is absolutely necessary to complete the entire reading procedure. If two bytes are programmed to be read, then two bytes *must* be read before any loading WR command can be sent to the same counter.

Read Operation Chart

A1	A0	RD	
0	0	0	Read Counter No. 0
0	1	0	Read Counter No. 1
1	0	0	Read Counter No. 2
1	1	0	Illegal

Reading While Counting

In order for the programmer to read the contents of any counter without effecting or disturbing the counting operation the 8253 has special internal logic that can be accessed using simple $\overline{WR}$ commands to the MODE register. Basically, when the programmer wishes to read the contents of a selected counter "on the fly" he loads the MODE register with a special code which latches the present count value into a storage register so that its contents contain an accurate, stable quantity. The programmer then issues a normal read command to the selected counter and the contents of the latched register is available.

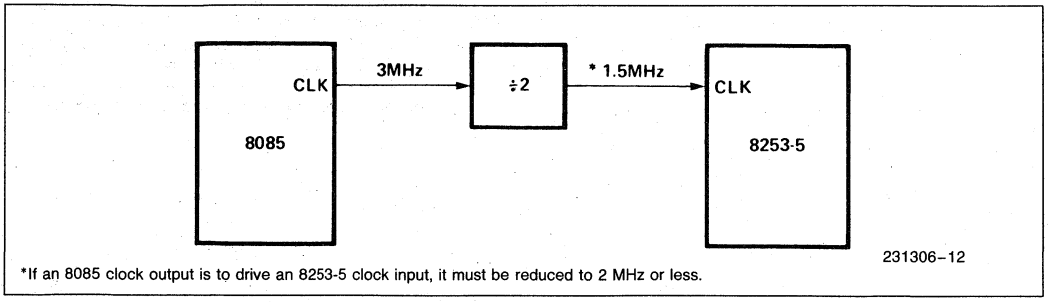
MODE Register for Latching Count

A0, A1 = 11

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	0	0	X	X	X	X

SC1, SC0— specify counter to be latched.
D5, D4 — 00 designates counter latching operation.
X — don't care.

The same limitation applies to this mode of reading the counter as the previous method. That is, it is mandatory to complete the entire read operation as programmed. This command has no effect on the counter's mode.



*If an 8085 clock output is to drive an 8253-5 clock input, it must be reduced to 2 MHz or less.

Figure 10. MCS-85™ Clock Interface*

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . . 0°C to 70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage On Any Pin
with Respect to Ground . . . . . −0.5V to 7V
Power Dissipation . . . . . 1 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%*

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.2	V_{CC} + .5V	V	
V_{OL}	Output Low Voltage		0.45	V	(Note 1)
V_{OH}	Output High Voltage	2.4		V	(Note 2)
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0.45V
I_{CC}	V_{CC} Supply Current		140	mA	

CAPACITANCE T_A = 25°C, V_{CC} = GND = 0V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	fc = 1 MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to V_{SS}

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5.0V ±10%, GND = 0V*

Bus Parameters(3)

READ CYCLE

Symbol	Parameter	8253		8253-5		Unit
		Min	Max	Min	Max	
t_{AR}	Address Stable before $\overline{READ}$	50		30		ns
t_{RA}	Address Hold Time for $\overline{READ}$	5		5		ns
t_{RR}	$\overline{READ}$ Pulse Width	400		300		ns
t_{RD}	Data Delay from $\overline{READ}$(4)		300		200	ns
t_{DF}	$\overline{READ}$ to Data Floating	25	125	25	100	ns
t_{RV}	Recovery Time between $\overline{READ}$ and Any Other Control Signal	1		1		μs

A.C. CHARACTERISTICS (Continued)

WRITE CYCLE

Symbol	Parameter	8253		8253-5		Unit
		Min	Max	Min	Max	
t_{AW}	Address Stable before $\overline{WRITE}$	50		30		ns
t_{WA}	Address Hold Time for $\overline{WRITE}$	30		30		ns
t_{WW}	$\overline{WRITE}$ Pulse Width	400		300		ns
t_{DW}	Data Set Up Time for $\overline{WRITE}$	300		250		ns
t_{WD}	Data Hold Time for $\overline{WRITE}$	40		30		ns
t_{RV}	Recovery Time between $\overline{WRITE}$ and Any Other Control Signal	1		1		μs

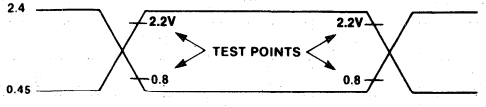
CLOCK AND GATE TIMING

Symbol	Parameter	8253		8253-5		Unit
		Min	Max	Min	Max	
t_{CLK}	Clock Period	380	dc	380	dc	ns
t_{PWH}	High Pulse Width	230		230		ns
t_{PWL}	Low Pulse Width	150		150		ns
t_{GW}	Gate Width High	150		150		ns
t_{GL}	Gate Width Low	100		100		ns
t_{GS}	Gate Set Up Time to CLK ↑	100		100		ns
t_{GH}	Gate Hold Time after CLK ↑	50		50		ns
t_{OD}	Output Delay from CLK ↓ (4)		400		400	ns
t_{ODG}	Output Delay from Gate ↓ (4)		300		300	ns

NOTES:
1. I_{OL} = 2.2 mA.
2. I_{OH} = −400 μA.
3. AC timings measured at V_{OH} 2.2, V_{OL} = 0.8.
4. C_L = 150 pF.

*For Extended Temperature EXPRESS, use M8253 electrical parameters.

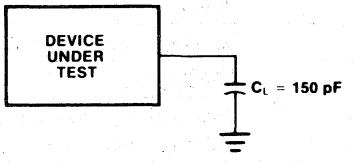
A.C. TESTING INPUT, OUTPUT WAVEFORM



231306-13

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.2V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT

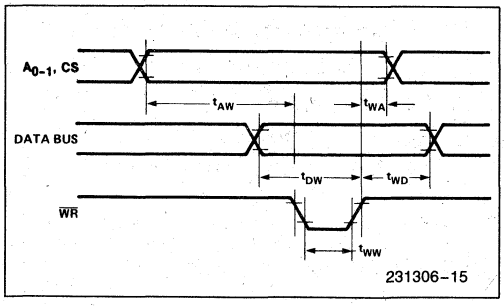


231306-14

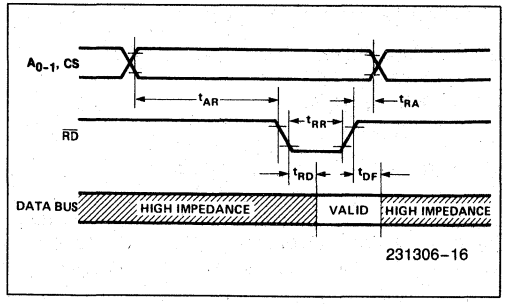
C_L Includes Jig Capacitance

WAVEFORMS

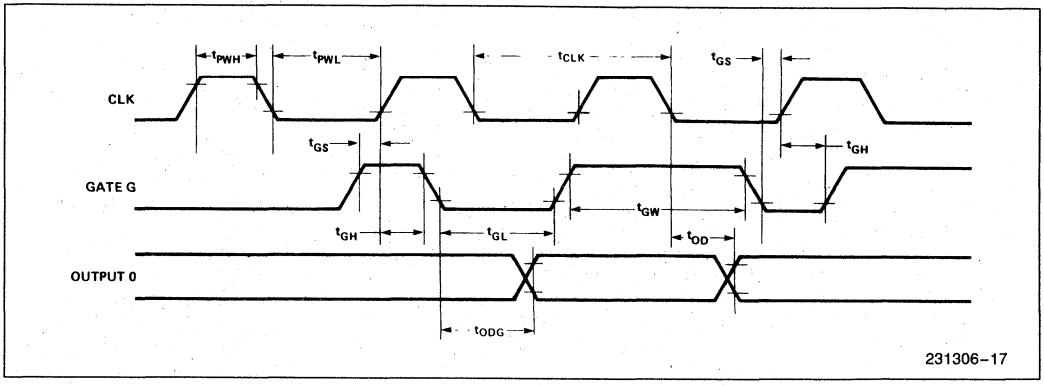
WRITE TIMING



READ TIMING



CLOCK AND GATE TIMING



3

8254
PROGRAMMABLE INTERVAL TIMER

- **Compatible with All Intel and Most Other Microprocessors**
- **Handles Inputs from DC to 10 MHz**
  - **— 5 MHz 8254-5**
  - **— 8 MHz 8254**
  - **— 10 MHz 8254-2**
- **Status Read-Back Command**
- **Six Programmable Counter Modes**
- **Three Independent 16-Bit Counters**
- **Binary or BCD Counting**
- **Single +5V Supply**
- **Available in EXPRESS**
  - **— Standard Temperature Range**

The Intel® 8254 is a counter/timer device designed to solve the common timing control problems in microcomputer system design. It provides three independent 16-bit counters, each capable of handling clock inputs up to 10 MHz. All modes are software programmable. The 8254 is a superset of the 8253.

The 8254 uses HMOS technology and comes in a 24-pin plastic or CERDIP package.

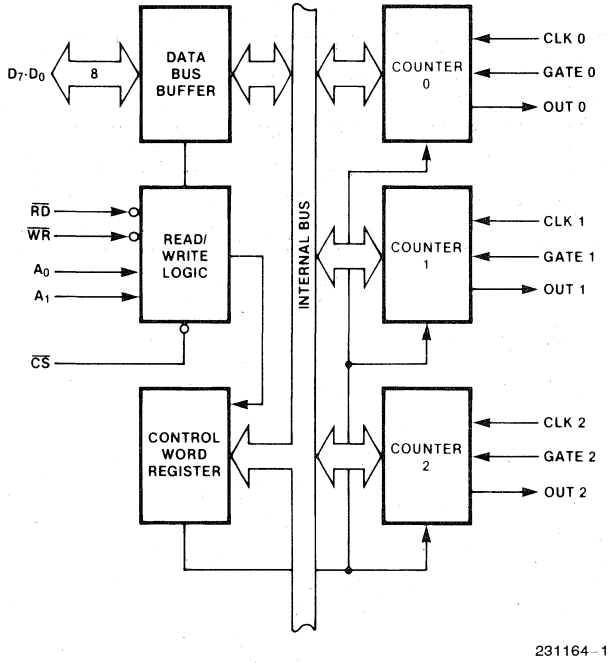


Figure 1. 8254 Block Diagram

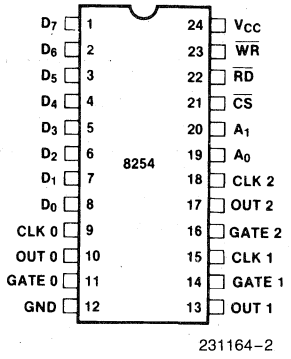


Figure 2. Pin Configuration

August 1987
Order Number: 231164-004

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
D_7-D_0	1-8	I/O	**DATA:** Bi-directional three state data bus lines, connected to system data bus.
CLK 0	9	I	**CLOCK 0:** Clock input of Counter 0.
OUT 0	10	O	**OUTPUT 0:** Output of Counter 0.
GATE 0	11	I	**GATE 0:** Gate input of Counter 0.
GND	12		**GROUND:** Power supply connection.
V_{CC}	24		**POWER:** +5V power supply connection.
$\overline{WR}$	23	I	**WRITE CONTROL:** This input is low during CPU write operations.
$\overline{RD}$	22	I	**READ CONTROL:** This input is low during CPU read operations.
$\overline{CS}$	21	I	**CHIP SELECT:** A low on this input enables the 8254 to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and $\overline{WR}$ are ignored otherwise.
A_1, A_0	20-19	I	**ADDRESS:** Used to select one of the three Counters or the Control Word Register for read or write operations. Normally connected to the system address bus.

A_1	A_0	Selects
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word Register

CLK 2	18	I	**CLOCK 2:** Clock input of Counter 2.
OUT 2	17	O	**OUT 2:** Output of Counter 2.
GATE 2	16	I	**GATE 2:** Gate input of Counter 2.
CLK 1	15	I	**CLOCK 1:** Clock input of Counter 1.
GATE 1	14	I	**GATE 1:** Gate input of Counter 1.
OUT 1	13	O	**OUT 1:** Output of Counter 1.

3

FUNCTIONAL DESCRIPTION

General

The 8254 is a programmable interval timer/counter designed for use with Intel microcomputer systems. It is a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 8254 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in software, the programmer configures the 8254 to match his requirements and programs one of the counters for the desired delay. After the desired delay, the 8254 will interrupt the CPU. Software overhead is minimal and variable length delays can easily be accommodated.

Some of the other counter/timer functions common to microcomputers which can be implemented with the 8254 are:

- Real time clock
- Event-counter
- Digital one-shot
- Programmable rate generator
- Square wave generator
- Binary rate multiplier
- Complex waveform generator
- Complex motor controller

Block Diagram

DATA BUS BUFFER

This 3-state, bi-directional, 8-bit buffer is used to interface the 8254 to the system bus (see Figure 3).

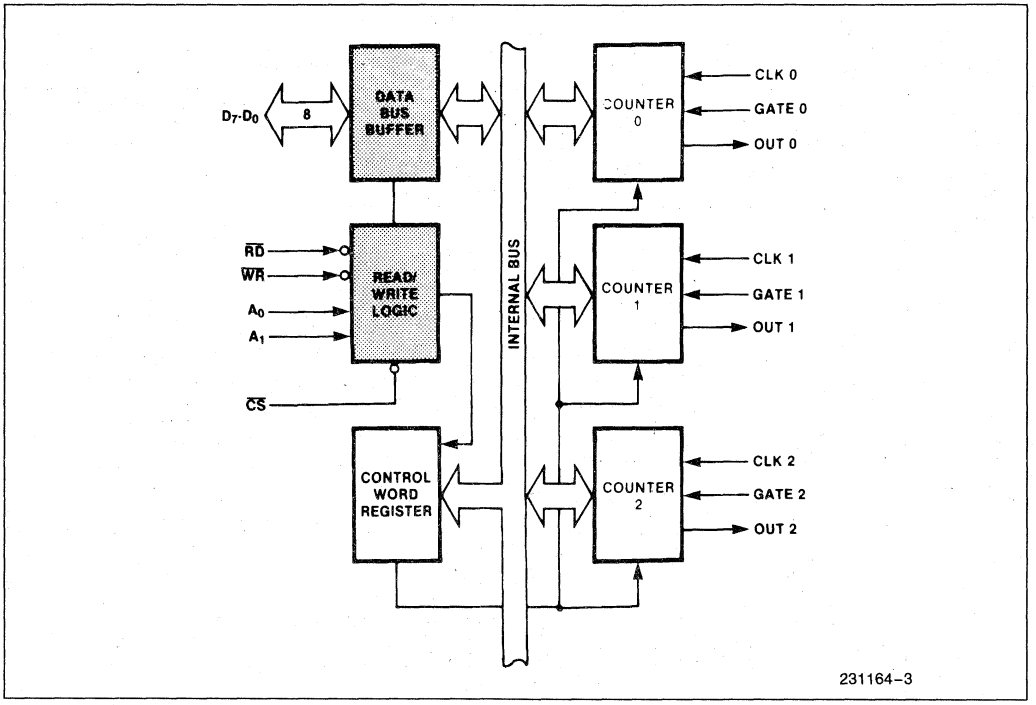


Figure 3. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

READ/WRITE LOGIC

The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the 8254. A_1 and A_0 select one of the three counters or the Control Word Register to be read from/written into. A "low" on the $\overline{RD}$ input tells the 8254 that the CPU is reading one of the counters. A "low" on the $\overline{WR}$ input tells the 8254 that the CPU is writing either a Control Word or an initial count. Both $\overline{RD}$ and $\overline{WR}$ are qualified by $\overline{CS}$; $\overline{RD}$ and $\overline{WR}$ are ignored unless the 8254 has been selected by holding $\overline{CS}$ low.

CONTROL WORD REGISTER

The Control Word Register (see Figure 4) is selected by the Read/Write Logic when $A_1,A_0 = 11$. If the CPU then does a write operation to the 8254, the data is stored in the Control Word Register and is interpreted as a Control Word used to define the operation of the Counters.

The Control Word Register can only be written to; status information is available with the Read-Back Command.

COUNTER 0, COUNTER 1, COUNTER 2

These three functional blocks are identical in operation, so only a single Counter will be described. The internal block diagram of a single counter is shown in Figure 5.

The Counters are fully independent. Each Counter may operate in a different Mode.

The Control Word Register is shown in the figure; it is not part of the Counter itself, but its contents determine how the Counter operates.

The status register, shown in Figure 5, when latched, contains the current contents of the Control Word Register and status of the output and null count flag. (See detailed explanation of the Read-Back command.)

The actual counter is labelled CE (for "Counting Element"). It is a 16-bit presettable synchronous down counter.

OL_M and OL_L are two 8-bit latches. OL stands for "Output Latch"; the subscripts M and L stand for "Most significant byte" and "Least significant byte"

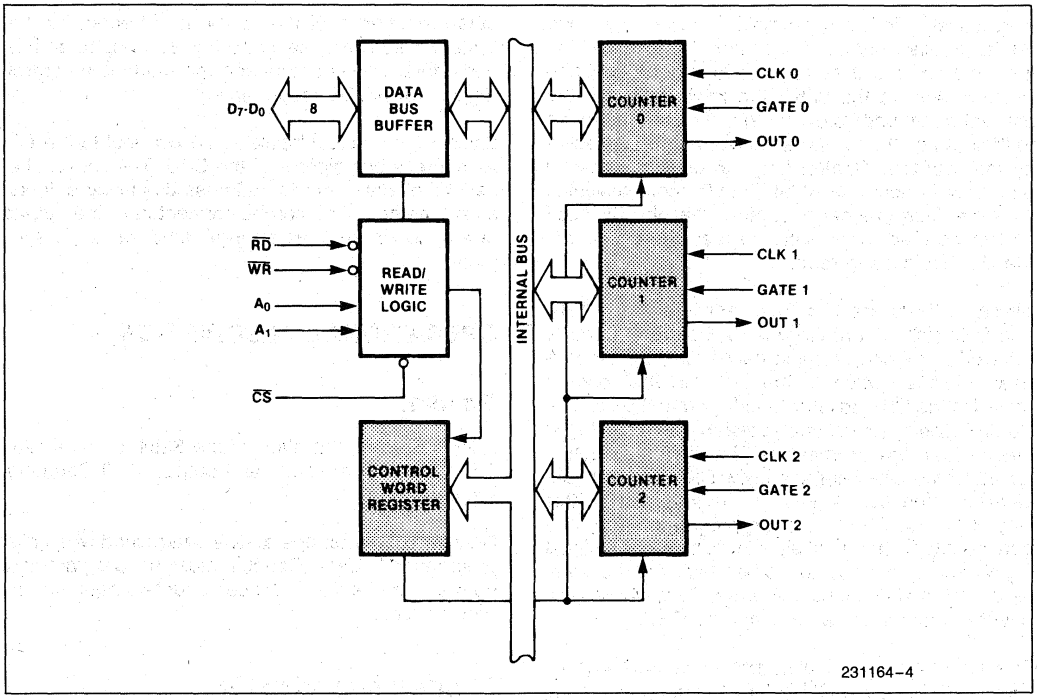


Figure 4. Block Diagram Showing Control Word Register and Counter Functions

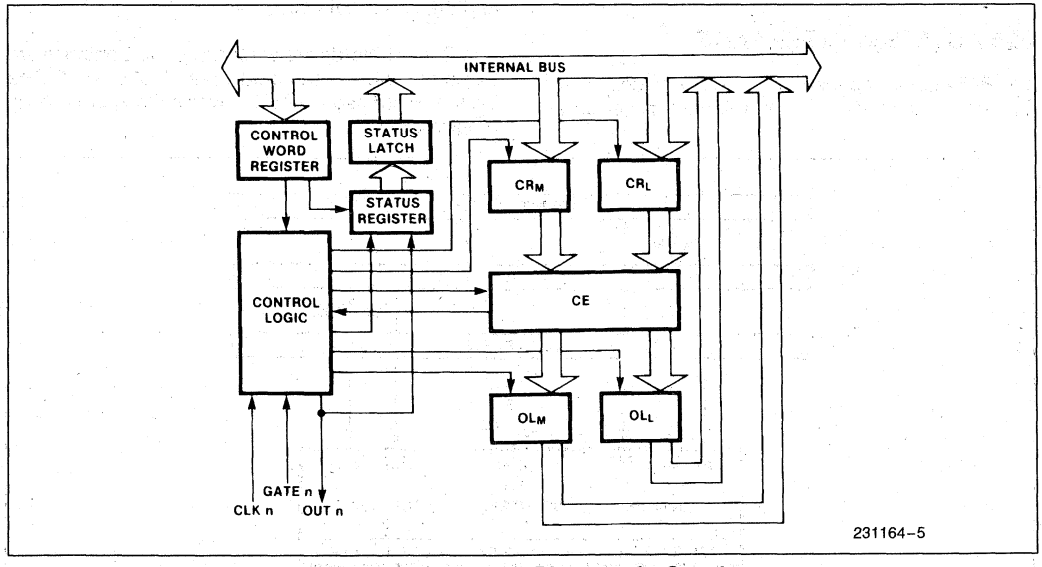


Figure 5. Internal Block Diagram of a Counter

respectively. Both are normally referred to as one unit and called just OL. These latches normally "follow" the CE, but if a suitable Counter Latch Command is sent to the 8254, the latches "latch" the present count until read by the CPU and then return to "following" the CE. One latch at a time is enabled by the counter's Control Logic to drive the internal bus. This is how the 16-bit Counter communicates over the 8-bit internal bus. Note that the CE itself cannot be read; whenever you read the count, it is the OL that is being read.

Similarly, there are two 8-bit registers called CR_M and CR_L (for "Count Register"). Both are normally referred to as one unit and called just CR. When a new count is written to the Counter, the count is stored in the CR and later transferred to the CE. The Control Logic allows one register at a time to be loaded from the internal bus. Both bytes are transferred to the CE simultaneously. CR_M and CR_L are cleared when the Counter is programmed. In this way, if the Counter has been programmed for one byte counts (either most significant byte only or least significant byte only) the other byte will be zero. Note that the CE cannot be written into; whenever a count is written, it is written into the CR.

The Control Logic is also shown in the diagram. CLK n, GATE n, and OUT n are all connected to the outside world through the Control Logic.

8254 SYSTEM INTERFACE

The 8254 is a component of the Intel Microcomputer Systems and interfaces in the same manner as all other peripherals of the family. It is treated by the system's software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A_0,A_1 connect to the A_0, A_1 address bus signals of the CPU. The CS can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an Intel 8205 for larger systems.

OPERATIONAL DESCRIPTION

General

After power-up, the state of the 8254 is undefined. The Mode, count value, and output of all Counters are undefined.

How each Counter operates is determined when it is programmed. Each Counter must be programmed before it can be used. Unused counters need not be programmed.

Programming the 8254

Counters are programmed by writing a Control Word and then an initial count.

The Control Words are written into the Control Word Register, which is selected when A_1,A_0 = 11. The Control Word itself specifies which Counter is being programmed.

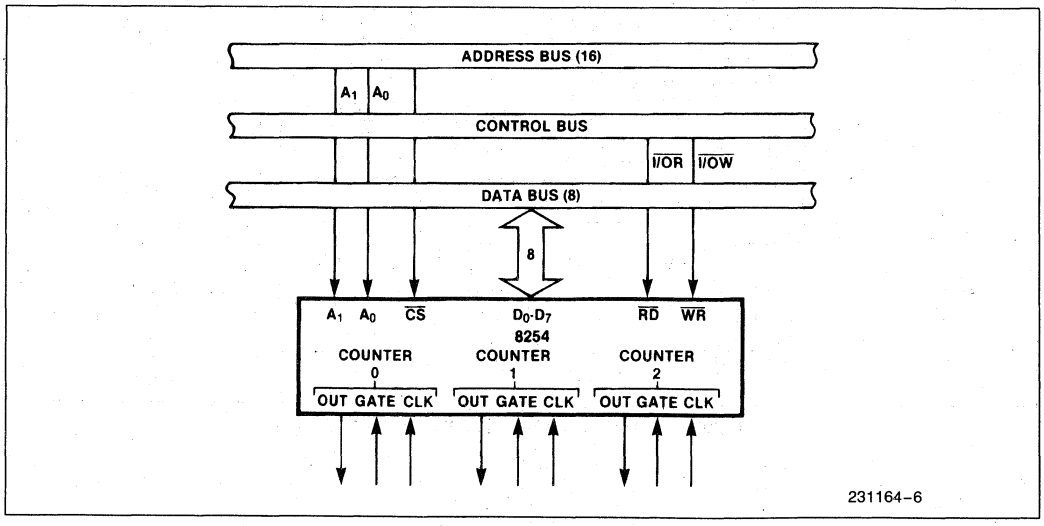


Figure 6. 8254 System Interface

Control Word Format

A_1,A_0 = 11 $\overline{CS}$ = 0 $\overline{RD}$ = 1 $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC—Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (see Read Operations)

M—Mode

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

RW—Read/Write

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only
1	0	Read/Write most significant byte only
1	1	Read/Write least significant byte first, then most significant byte

BCD

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Figure 7. Control Word Format

By contrast, initial counts are written into the Counters, not the Control Word Register. The A_1,A_0 inputs are used to select the Counter to be written into. The format of the initial count is determined by the Control Word used.

Write Operations

The programming procedure for the 8254 is very flexible. Only two conventions need to be remembered:

1) For each Counter, the Control Word must be written before the initial count is written.
2) The initial count must follow the count format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

Since the Control Word Register and the three Counters have separate addresses (selected by the A_1,A_0 inputs), and each Control Word specifies the Counter it applies to (SC0,SC1 bits), no special instruction sequence is required. Any programming sequence that follows the conventions in Figure 7 is acceptable.

A new initial count may be written to a Counter at any time without affecting the Counter's programmed Mode in any way. Counting will be affected as described in the Mode definitions. The new count must follow the programmed count format.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same Counter. Otherwise, the Counter will be loaded with an incorrect count.

	A_1	A_0
Control Word—Counter 0	1	1
LSB of count—Counter 0	0	0
MSB of count—Counter 0	0	0
Control Word—Counter 1	1	1
LSB of count—Counter 1	0	1
MSB of count—Counter 1	0	1
Control Word—Counter 2	1	1
LSB of count—Counter 2	1	0
MSB of count—Counter 2	1	0

	A_1	A_0
Control Word—Counter 2	1	1
Control Word—Counter 1	1	1
Control Word—Counter 0	1	1
LSB of count—Counter 2	1	0
MSB of count—Counter 2	1	0
LSB of count—Counter 1	0	1
MSB of count—Counter 1	0	1
LSB of count—Counter 0	0	0
MSB of count—Counter 0	0	0

	A_1	A_0
Control Word—Counter 0	1	1
Control Word—Counter 1	1	1
Control Word—Counter 2	1	1
LSB of count—Counter 2	1	0
LSB of count—Counter 1	0	1
LSB of count—Counter 0	0	0
MSB of count—Counter 0	0	0
MSB of count—Counter 1	0	1
MSB of count—Counter 2	1	0

	A_1	A_0
Control Word—Counter 1	1	1
Control Word—Counter 0	1	1
LSB of count—Counter 1	0	1
Control Word—Counter 2	1	1
LSB of count—Counter 0	0	0
MSB of count—Counter 1	0	1
LSB of count—Counter 2	1	0
MSB of count—Counter 0	0	0
MSB of count—Counter 2	1	0

NOTE:
In all four examples, all Counters are programmed to read/write two-byte counts. These are only four of many possible programming sequences.

Figure 8. A Few Possible Programming Sequences

Read Operations

It is often desirable to read the value of a Counter without disturbing the count in progress. This is easily done in the 8254.

There are three possible methods for reading the counters: a simple read operation, the Counter Latch Command, and the Read-Back Command. Each is explained below. The first method is to perform a simple read operation. To read the Counter, which is selected with the A1, A0 inputs, the CLK input of the selected Counter must be inhibited by using either the GATE input or external logic. Otherwise, the count may be in the process of changing when it is read, giving an undefined result.

COUNTER LATCH COMMAND

The second method uses the "Counter Latch Command". Like a Control Word, this command is written to the Control Word Register, which is selected when $A_1,A_0 = 11$. Also like a Control Word, the SC0, SC1 bits select one of the three Counters, but two other bits, D5 and D4, distinguish this command from a Control Word.

A_1,A_0 = 11; CS = 0; RD = 1; WR = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	0	0	X	X	X	X

SC1,SC0—specify counter to be latched

SC1	SC0	Counter
0	0	0
0	1	1
1	0	2
1	1	Read-Back Command

D5,D4—00 designates Counter Latch Command

X—don't care

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Figure 9. Counter Latching Command Format

The selected Counter's output latch (OL) latches the count at the time the Counter Latch Command is received. This count is held in the latch until it is read by the CPU (or until the Counter is reprogrammed). The count is then unlatched automatically and the OL returns to "following" the counting element (CE). This allows reading the contents of the Counters "on the fly" without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one Counter. Each latched Counter's OL holds its count until it is read. Counter Latch Commands do not affect the programmed Mode of the Counter in any way.

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued.

With either method, the count must be read according to the programmed format; specifically, if the Counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other; read or write or programming operations of other Counters may be inserted between them.

Another feature of the 8254 is that reads and writes of the same Counter may be interleaved; for example, if the Counter is programmed for two byte counts, the following sequence is valid.

1) Read least significant byte.
2) Write new least significant byte.
3) Read most significant byte.
4) Write new most significant byte.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between reading the first and second byte to another routine which also reads from that same Counter. Otherwise, an incorrect count will be read.

READ-BACK COMMAND

The third method uses the Read-Back Command. This command allows the user to check the count value, programmed Mode, and current states of the OUT pin and Null Count flag of the selected counter(s).

The command is written into the Control Word Register and has the format shown in Figure 10. The command applies to the counters selected by setting their corresponding bits D3, D2, D1 = 1.

A0, A1 = 11 $\overline{CS}$ = 0 $\overline{RD}$ = 1 $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	$\overline{COUNT}$	$\overline{STATUS}$	CNT 2	CNT 1	CNT 0	0

D_5: 0 = Latch count of selected counter(s)
D_4: 0 = Latch status of selected counters(s)
D_3: 1 = Select Counter 2
D_2: 1 = Select Counter 1
D_1: 1 = Select Counter 0
D_0: Reserved for future expansion; Must be 0

Figure 10. Read-Back Command Format

The read-back command may be used to latch multiple counter output latches (OL) by setting the $\overline{COUNT}$ bit D5 = 0 and selecting the desired counter(s). This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read (or the counter is reprogrammed). The counter is automatically unlatched when read, but other counters remain latched until they are read. If multiple count read-back commands are issued to the same counter without reading the count, all but the first are ignored; i.e., the count which will be read is the count at the time the first read-back command was issued.

The read-back command may also be used to latch status information of selected counter(s) by setting $\overline{STATUS}$ bit D4 = 0. Status must be latched to be read; status of a counter is accessed by a read from that counter.

The counter status format is shown in Figure 11. Bits D5 through D0 contain the counter's programmed Mode exactly as written in the last Mode Control Word. OUTPUT bit D7 contains the current state of the OUT pin. This allows the user to monitor the counter's output via software, possibly eliminating some hardware from a system.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Output	Null Count	RW1	RW0	M2	M1	M0	BCD

D_7 1 = OUT Pin is 1
0 = OUT Pin is 0

D_6 1 = Null Count
0 = Count available for reading

D_5–D_0 Counter programmed mode (see Figure 7)

Figure 11. Status Byte

NULL COUNT bit D6 indicates when the last count written to the counter register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the Mode of the counter and is described in the Mode Definitions, but until the count is loaded into the counting element (CE), it can't be read from the counter. If the count is latched or read before this time, the count value will not reflect the new count just written. The operation of Null Count is shown in Figure 12.

This Action	Causes
A. Write to the control word register;(1)	Null Count = 1
B. Write to the count register (CR);(2)	Null Count = 1
C. New Count is loaded into CE (CR → CE);	Null Count = 0

NOTE:
1. Only the counter specified by the control word will have its Null Count set to 1. Null count bits of other counters are unaffected.
2. If the counter is programmed for two-byte counts (least significant byte then most significant byte) Null Count goes to 1 when the second byte is written.

Figure 12. Null Count Operation

If multiple status latch operations of the counter(s) are performed without reading the status, all but the first are ignored; i.e., the status that will be read is the status of the counter at the time the first status read-back command was issued.

Both count and status of the selected counter(s) may be latched simultaneously by setting both $\overline{\text{COUNT}}$ and $\overline{\text{STATUS}}$ bits D5,D4 = 0. This is functionally the same as issuing two separate read-back commands at once, and the above discussions apply here also. Specifically, if multiple count and/or status read-back commands are issued to the same counter(s) without any intervening reads, all but the first are ignored. This is illustrated in Figure 13.

If both count and status of a counter are latched, the first read operation of that counter will return latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two type counts) return latched count. Subsequent reads return unlatched count.

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Write into Counter 0
0	1	0	0	1	Write into Counter 1
0	1	0	1	0	Write into Counter 2
0	1	0	1	1	Write Control Word
0	0	1	0	0	Read from Counter 0
0	0	1	0	1	Read from Counter 1
0	0	1	1	0	Read from Counter 2
0	0	1	1	1	No-Operation (3-State)
1	X	X	X	X	No-Operation (3-State)
0	1	1	X	X	No-Operation (3-State)

Figure 14. Read/Write Operations Summary

Command								Description	Result
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
1	1	0	0	0	0	1	0	Read back count and status of Counter 0	Count and status latched for Counter 0
1	1	1	0	0	1	0	0	Read back status of Counter 1	Status latched for Counter 1
1	1	1	0	1	1	0	0	Read back status of Counters 2, 1	Status latched for Counter 2, but not Counter 1
1	1	0	1	1	0	0	0	Read back count of Counter 2	Count latched for Counter 2
1	1	0	0	0	1	0	0	Read back count and status of Counter 1	Count latched for Counter 1, but not status
1	1	1	0	0	0	1	0	Read back status of Counter 1	Command ignored, status already latched for Counter 1

Figure 13. Read-Back Command Example

Mode Definitions

The following are defined for use in describing the operation of the 8254.

CLK Pulse: a rising edge, then a falling edge, in that order, of a Counter's CLK input.

Trigger: a rising edge of a Counter's GATE input.

Counter loading: the transfer of a count from the CR to the CE (refer to the "Functional Description")

MODE 0: INTERRUPT ON TERMINAL COUNT

Mode 0 is typically used for event counting. After the Control Word is written, OUT is initially low, and will remain low until the Counter reaches zero. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written into the Counter.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After the Control Word and initial count are written to a Counter, the initial count will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not go high until N + 1 CLK pulses after the initial count is written.

If a new count is written to the Counter, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte disables counting. OUT is set low immediately (no clock pulse required)
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the counting sequence to be synchronized by software. Again, OUT does not go high until N+1 CLK pulses after the new count of N is written.

If an initial count is written while GATE = 0, it will still be loaded on the next CLK pulse. When GATE goes high, OUT will go high N CLK pulses later; no CLK pulse is needed to load the Counter as this has already been done.

MODE 1: HARDWARE RETRIGGERABLE ONE-SHOT

OUT will be initially high. OUT will go low on the CLK pulse following a trigger to begin the one-shot pulse, and will remain low until the Counter reaches zero. OUT will then go high and remain high until the CLK pulse after the next trigger.

After writing the Control Word and initial count, the Counter is armed. A trigger results in loading the Counter and setting OUT low on the next CLK pulse, thus starting the one-shot pulse. An initial count of N will result in a one-shot pulse N CLK cycles in duration. The one-shot is retriggerable, hence OUT will remain low for N CLK pulses after any trigger. The one-shot pulse can be repeated without rewriting the same count into the counter. GATE has no effect on OUT.

If a new count is written to the Counter during a one-shot pulse, the current one-shot is not affected unless the counter is retriggered. In that case, the Counter is loaded with the new count and the one-shot pulse continues until the new count expires.

MODE 2: RATE GENERATOR

This Mode functions like a divide-by-N counter. It is typically used to generate a Real Time Clock interrupt. OUT will initially be high. When the initial count has decremented to 1, OUT goes low for one CLK pulse. OUT then goes high again, the Counter reloads the initial count and the process is repeated. Mode 2 is periodic; the same sequence is repeated indefinitely. For an initial count of N, the sequence repeats every N CLK cycles.

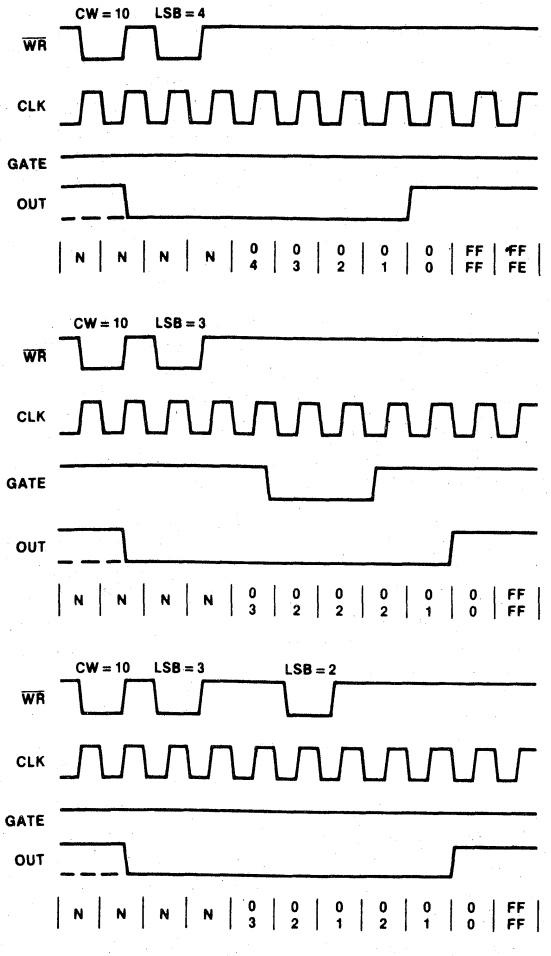
GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low during an output pulse, OUT is set high immediately. A trigger reloads the Counter with the initial count on the next CLK pulse; OUT goes low N CLK pulses after the trigger. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. OUT goes low N CLK Pulses after the initial count is written. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current period, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current counting cycle. In mode 2, a COUNT of 1 is illegal.

MODE 3: SQUARE WAVE MODE

Mode 3 is typically used for Baud rate generation. Mode 3 is similar to Mode 2 except for the duty cycle of OUT. OUT will initially be high. When half the



231164-7

NOTE:
The following conventions apply to all mode timing diagrams:
1. Counters are programmed for binary (not BCD) counting and for reading/writing least significant byte (LSB) only.
2. The counter is always selected ($\overline{CS}$ always low).
3. CW stands for "Control Word"; CW = 10 means a control word of 10 HEX is written to the counter.
4. LSB stands for "Least Significant Byte" of count.
5. Numbers below diagrams are count values. The lower number is the least significant byte. The upper number is the most significant byte. Since the counter is programmed to read/write LSB only, the most significant byte cannot be read.
   N stands for an undefined count.
   Vertical lines show transitions between count values.

Figure 15. Mode 0

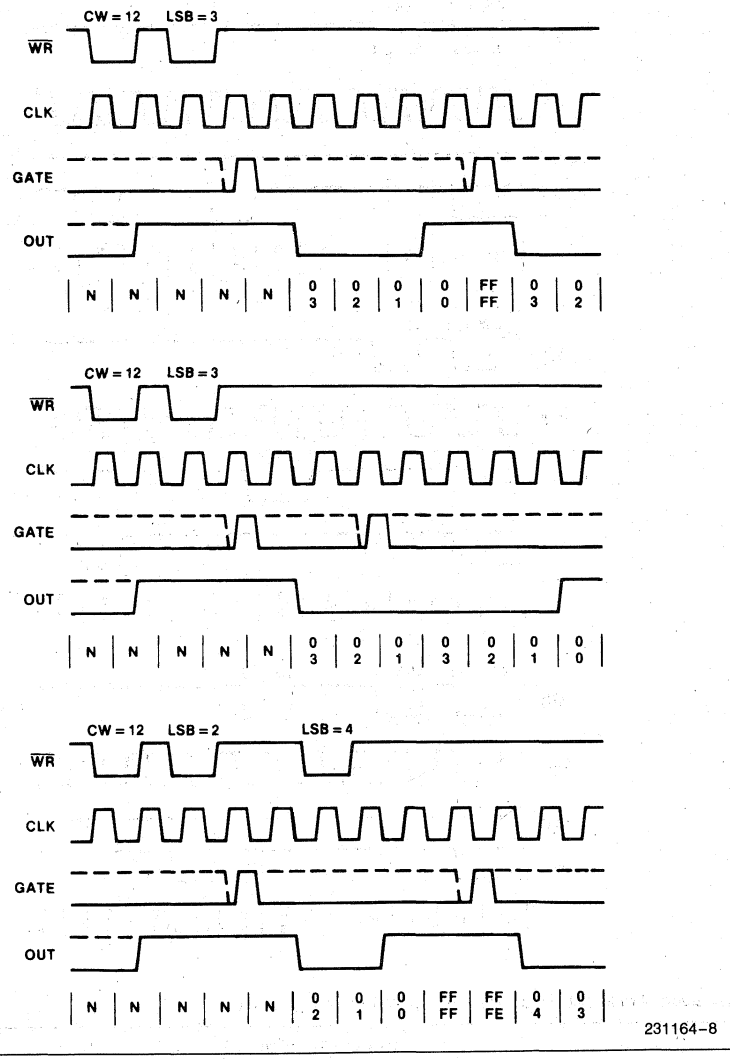


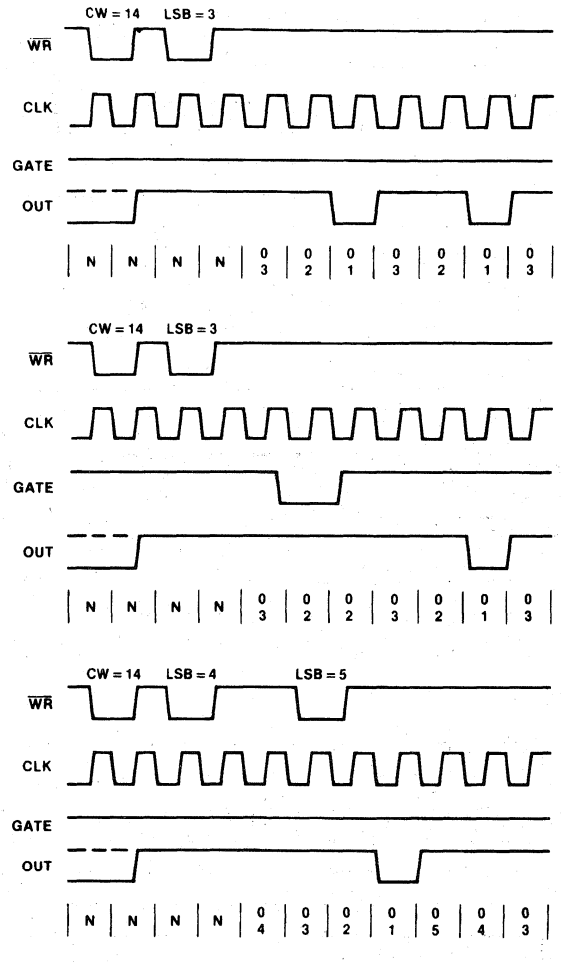
Figure 16. Mode 1

initial count has expired, OUT goes low for the remainder of the count. Mode 3 is periodic; the sequence above is repeated indefinitely. An initial count of N results in a square wave with a period of N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low while OUT is low, OUT is set high immediately; no CLK pulse is required. A trigger reloads the Counter with the initial count on the next CLK pulse. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current half-cycle of the square wave, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the



NOTE:
A GATE transition should not occur one clock prior to terminal count.

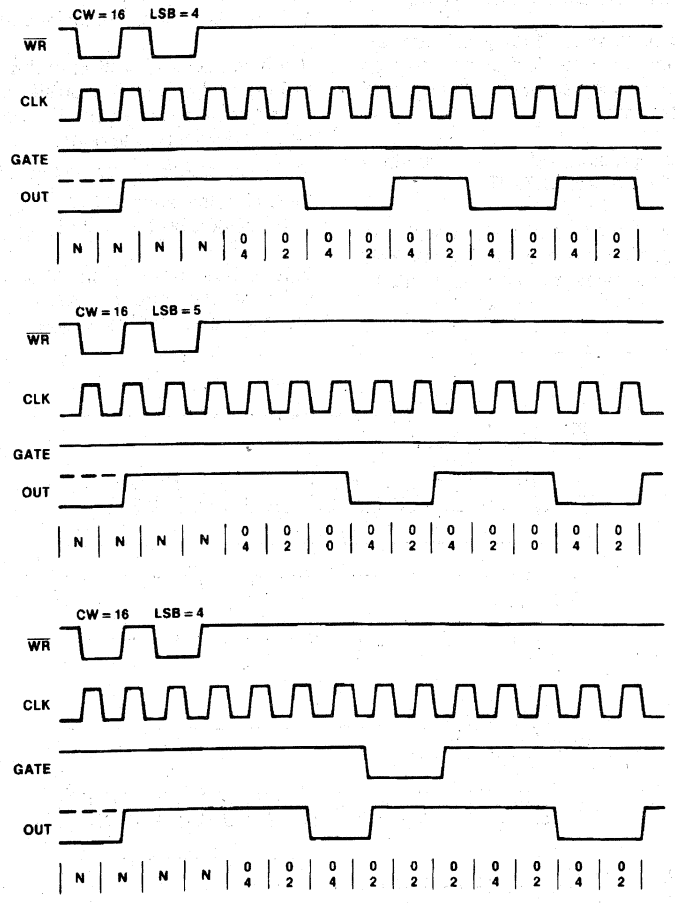
Figure 17. Mode 2

new count. Otherwise, the new count will be loaded at the end of the current half-cycle.

Mode 3 is implemented as follows:

Even counts: OUT is initially high. The initial count is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. When the count expires OUT changes value and the Counter is reloaded with the initial count. The above process is repeated indefinitely.

Odd counts: OUT is initially high. The initial count minus one (an even number) is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. One CLK pulse *after* the count expires, OUT goes low and the Counter is reloaded with the initial count minus one. Succeeding CLK pulses decrement the count by two. When the count expires, OUT goes high again and the Counter is reloaded with the initial count minus one. The above process is repeated indefinitely. So for odd counts, OUT will be high for $(N + 1)/2$ counts and low for $(N - 1)/2$ counts.



NOTE:
A GATE transition should not occur one clock prior to terminal count.

Figure 18. Mode 3

MODE 4: SOFTWARE TRIGGERED STROBE

OUT will be initially high. When the initial count expires, OUT will go low for one CLK pulse and then go high again. The counting sequence is "triggered" by writing the initial count.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N + 1 CLK pulses after the initial count is written.

If a new count is written during counting, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte has no effect on counting.
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the sequence to be "retriggered" by software. OUT strobes low N + 1 CLK pulses after the new count of N is written.

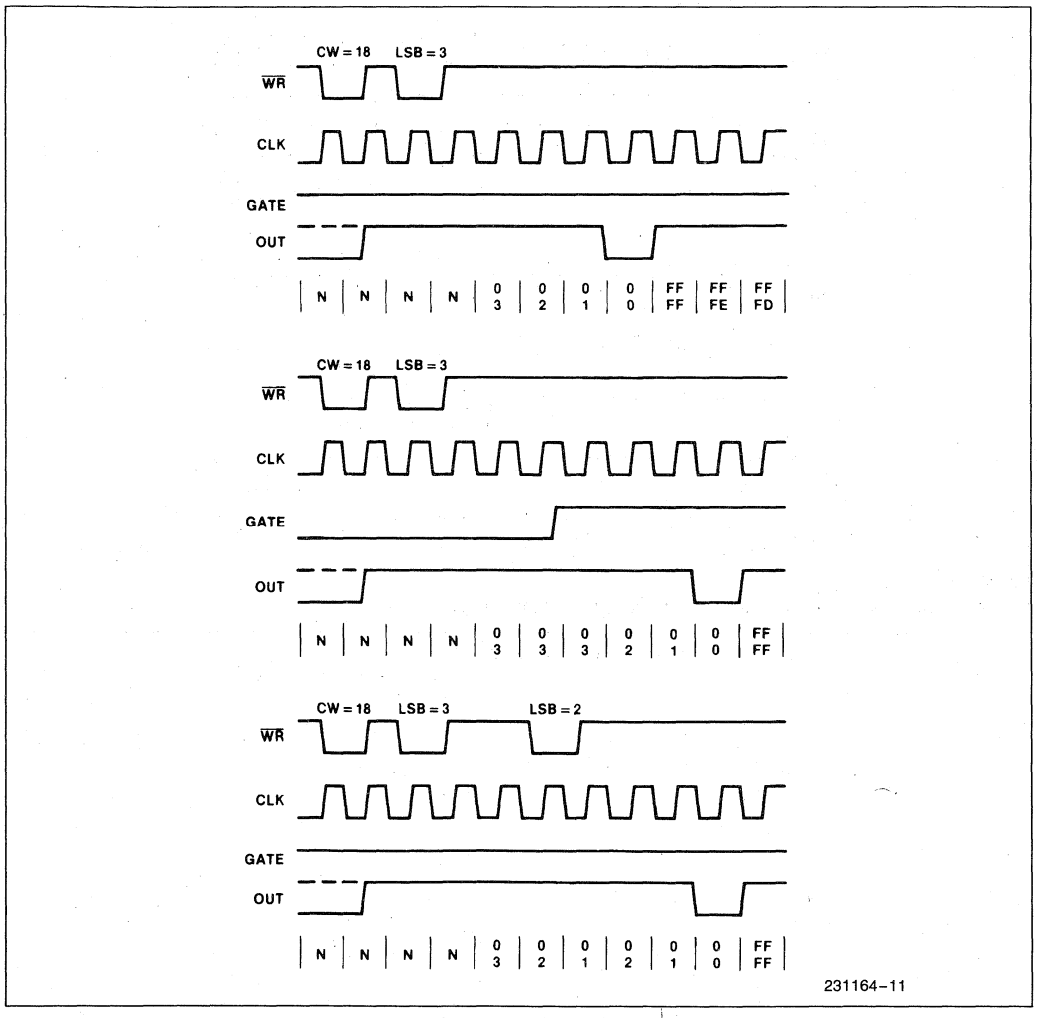


Figure 19. Mode 4

MODE 5: HARDWARE TRIGGERED STROBE (RETRIGGERABLE)

OUT will initially be high. Counting is triggered by a rising edge of GATE. When the initial count has expired, OUT will go low for one CLK pulse and then go high again.

After writing the Control Word and initial count, the counter will not be loaded until the CLK pulse after a trigger. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N + 1 CLK pulses after a trigger.

A trigger results in the Counter being loaded with the initial count on the next CLK pulse. The counting sequence is retriggerable. OUT will not strobe low for N + 1 CLK pulses after any trigger. GATE has no effect on OUT.

If a new count is written during counting, the current counting sequence will not be affected. If a trigger occurs after the new count is written but before the current count expires, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from there.

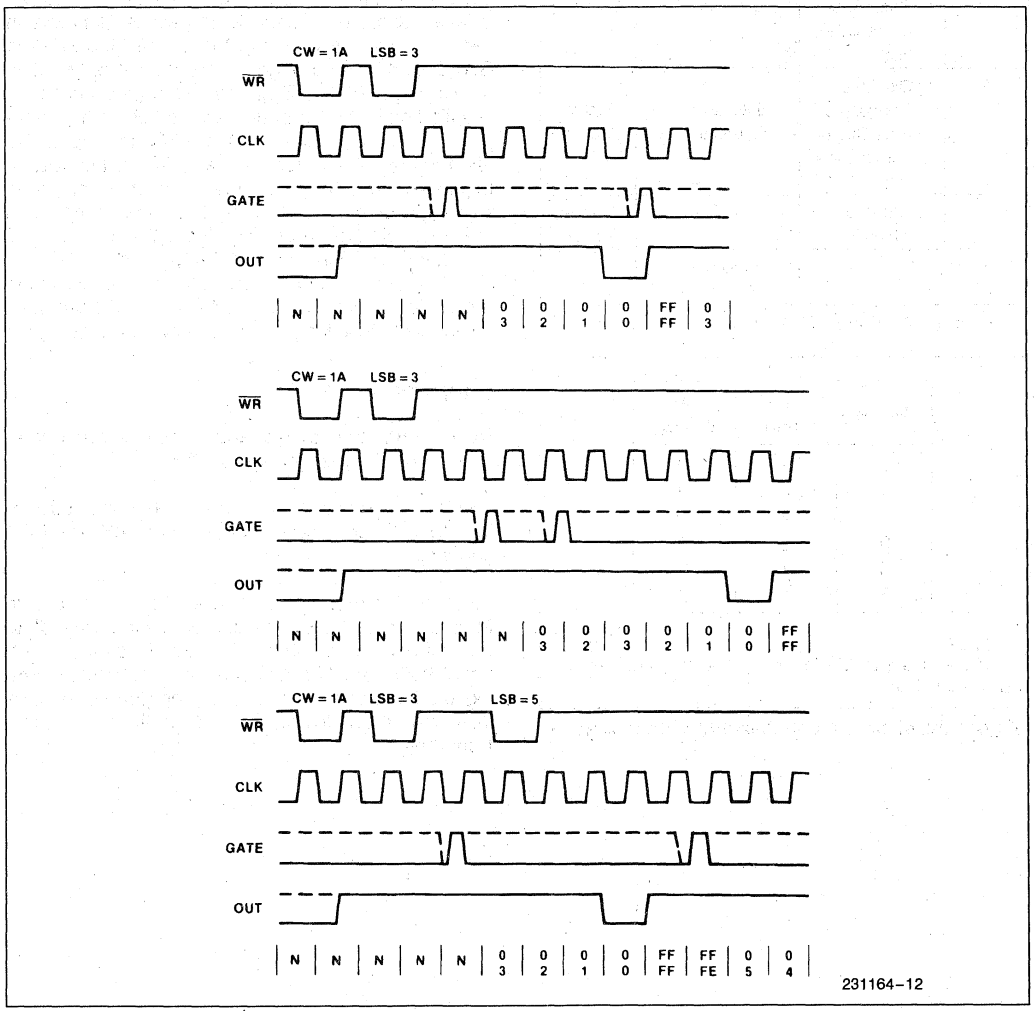


Figure 20. Mode 5

Signal Status Modes	Low Or Going Low	Rising	High
0	Disables Counting	— —	Enables Counting
1	— —	1) Initiates Counting 2) Resets Output after Next Clock	— —
2	1) Disables Counting 2) Sets Output Immediately High	Initiates Counting	Enables Counting
3	1) Disables Counting 2) Sets Output Immediately High	Initiates Counting	Enables Counting
4	Disables Counting	— —	Enables Counting
5	— —	Initiates Counting	— —

Figure 21. Gate Pin Operations Summary

Mode	Min Count	Max Count
0	1	0
1	1	0
2	2	0
3	2	0
4	1	0
5	1	0

NOTE:
0 is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

Figure 22. Minimum and Maximum Initial Counts

Operation Common to All Modes

PROGRAMMING

When a Control Word is written to a Counter, all Control Logic is immediately reset and OUT goes to a known initial state; no CLK pulses are required for this.

GATE

The GATE input is always sampled on the rising edge of CLK. In Modes 0, 2, 3, and 4 the GATE input is level sensitive, and the logic level is sampled on the rising edge of CLK. In Modes 1, 2, 3, and 5 the GATE input is rising-edge sensitive. In these Modes, a rising edge of GATE (trigger) sets an edge-sensitive flip-flop in the Counter. This flip-flop is then sampled on the next rising edge of CLK; the flip-flop is reset immediately after it is sampled. In this way, a trigger will be detected no matter when it occurs—a high logic level does not have to be maintained until the next rising edge of CLK. Note that in Modes 2 and 3, the GATE input is both edge- and level-sensitive. In Modes 2 and 3, if a CLK source other than the system clock is used, GATE should be pulsed immediately following $\overline{WR}$ of a new count value.

COUNTER

New counts are loaded and Counters are decremented on the falling edge of CLK.

The largest possible initial count is 0; this is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

The Counter does not stop when it reaches zero. In Modes 0, 1, 4, and 5 the Counter "wraps around" to the highest count, either FFFF hex for binary counting or 9999 for BCD counting, and continues counting. Modes 2 and 3 are periodic; the Counter reloads itself with the initial count and continues counting from there.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with Respect to Ground ..... −0.5V to +7V
Power Dissipation 1W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} +0.5V	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0.45V
I_{CC}	V_{CC} Supply Current		170	mA	
C_{IN}	Input Capacitance		10	pF	f_c = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to V_{SS}(4)

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%, GND = 0V

Bus Parameters(1)

READ CYCLE

Symbol	Parameter	8254-5		8254		8254-2		Unit
		Min	Max	Min	Max	Min	Max	
t_{AR}	Address Stable Before $\overline{RD}$ ↓	45		45		30		ns
t_{SR}	$\overline{CS}$ Stable Before $\overline{RD}$ ↓	0		0		0		ns
t_{RA}	Address Hold Time After $\overline{RD}$ ↑	0		0		0		ns
t_{RR}	$\overline{RD}$ Pulse Width	150		150		95		ns
t_{RD}	Data Delay from $\overline{RD}$ ↓		120		120		85	ns
t_{AD}	Data Delay from Address		220		220		185	ns
t_{DF}	$\overline{RD}$ ↑ to Data Floating	5	90	5	90	5	65	ns
t_{RV}	Command Recovery Time	200		200		165		ns

NOTE:
1. AC timings measured at V_{OH} = 2.0V, V_{OL} = 0.8V.

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%, GND = 0V (Continued)

WRITE CYCLE

Symbol	Parameter	8254-5		8254		8254-2		Unit
		Min	Max	Min	Max	Min	Max	
t_{AW}	Address Stable Before $\overline{WR}$ ↓	0		0		0		ns
t_{SW}	$\overline{CS}$ Stable Before $\overline{WR}$ ↓	0		0		0		ns
$t_{\overline{WA}}$	Address Hold Time After $\overline{WR}$ ↓	0		0		0		ns
t_{WW}	$\overline{WR}$ Pulse Width	150		150		95		ns
t_{DW}	Data Setup Time Before $\overline{WR}$ ↑	120		120		95		ns
t_{WD}	Data Hold Time After $\overline{WR}$ ↑	0		0		0		ns
t_{RV}	Command Recovery Time	200		200		165		ns

CLOCK AND GATE

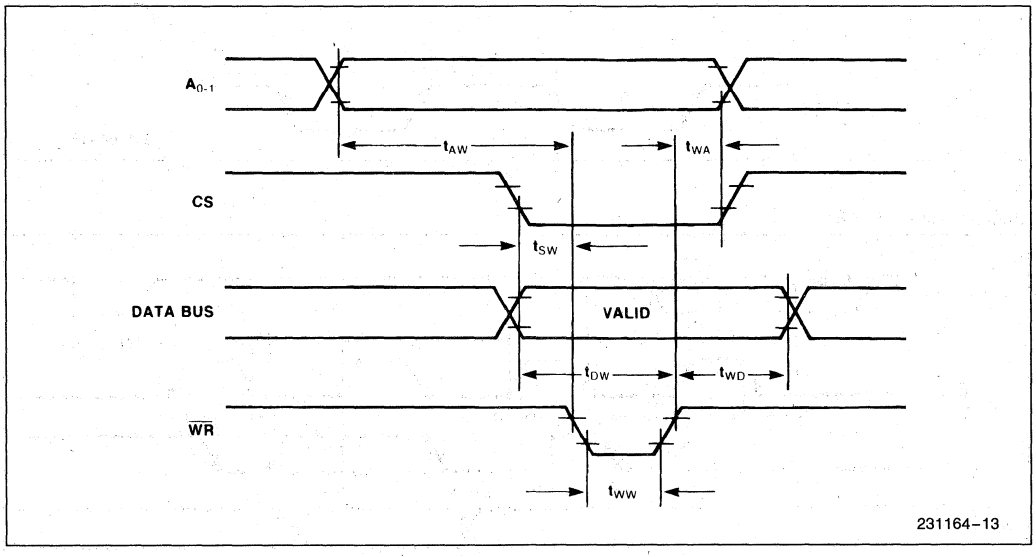
Symbol	Parameter	8254-5		8254		8254-2		Unit
		Min	Max	Min	Max	Min	Max	
t_{CLK}	Clock Period	200	DC	125	DC	100	DC	ns
t_{PWH}	High Pulse Width	60(3)		60(3)		30(3)		ns
t_{PWL}	Low Pulse Width	60(3)		60(3)		50(3)		ns
t_R	Clock Rise Time		25		25		25	ns
t_F	Clock Fall Time		25		25		25	ns
t_{GW}	Gate Width High	50		50		50		ns
t_{GL}	Gate Width Low	50		50		50		ns
t_{GS}	Gate Setup Time to CLK ↑	50		50		40		ns
t_{GH}	Gate Setup Time After CLK ↑	50(2)		50(2)		50(2)		ns
t_{OD}	Output Delay from CLK ↓		150		150		100	ns
t_{ODG}	Output Delay from Gate ↓		120		120		100	ns
t_{WC}	CLK Delay for Loading ↓	0	55	0	55	0	55	ns
t_{WG}	Gate Delay for Sampling	−5	50	−5	50	−5	40	ns
t_{WO}	OUT Delay from Mode Write		260		260		240	ns
t_{CL}	CLK Set Up for Count Latch	−40	45	−40	45	−40	40	ns

NOTES:

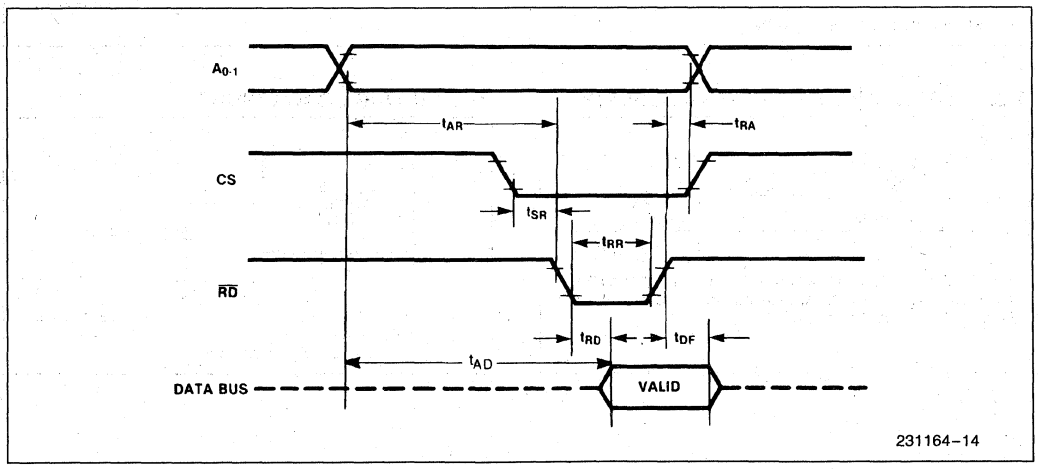
2. In Modes 1 and 5 triggers are sampled on each rising clock edge. A second trigger within 120 ns (70 ns for the 8254-2) of the rising clock edge may not be detected.
3. Low-going glitches that violate t_{PWH}, t_{PWL} may cause errors requiring counter reprogramming.
4. Sampled, not 100% tested. T_A = 25°C.
5. If CLK present at TWC min then Count equals N+2 CLK pulses, TWC max equals Count N+1 CLK pulse. TWC min to TWC max, count will be either N+1 or N+2 CLK pulses.
6. In Modes 1 and 5, if GATE is present when writing a new Count value, at TWG min Counter will not be triggered, at TWG max Counter will be triggered.
7. If CLK present when writing a Counter Latch or ReadBack Command, at TCL min CLK will be reflected in count value latched, at TCL max CLK will not be reflected in the count value latched.

WAVEFORMS

WRITE

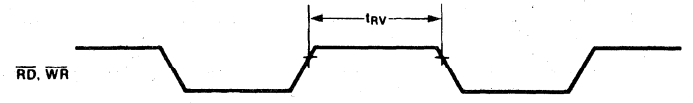


READ



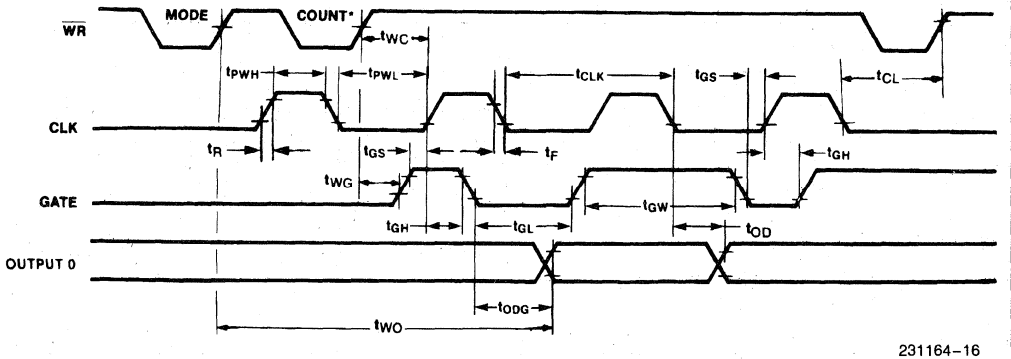
WAVEFORMS (Continued)

RECOVERY



231164-15

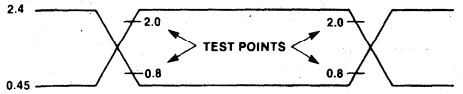
CLOCK AND GATE



231164-16

*Last byte of count being written.

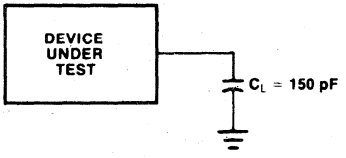
A.C. TESTING INPUT, OUTPUT WAVEFORM



231164-17

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0." Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT



231164-18

C_L = 150 pF
C_L Includes Jig Capacitance

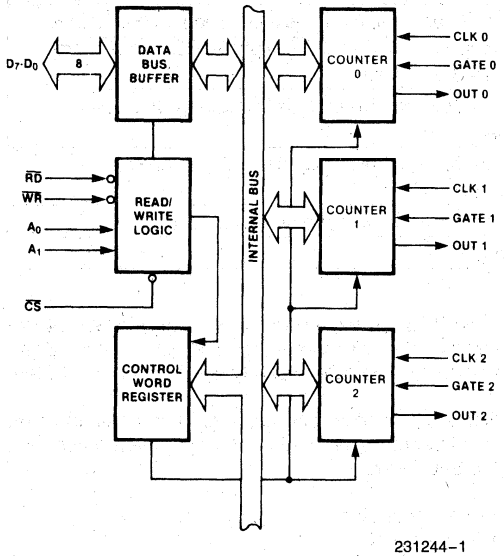
82C54
CHMOS PROGRAMMABLE INTERVAL TIMER

- **Compatible with all Intel and most other microprocessors**
- **High Speed, "Zero Wait State" Operation with 8 MHz 8086/88 and 80186/188**
- **Handles Inputs from DC to 8 MHz — 10 MHz for 82C54-2**
- **Available in EXPRESS — Standard Temperature Range — Extended Temperature Range**
- **Three independent 16-bit counters**
- **Low Power CHMOS — I_{CC} = 10 mA @ 8 MHz Count frequency**
- **Completely TTL Compatible**
- **Six Programmable Counter Modes**
- **Binary or BCD counting**
- **Status Read Back Command**
- **Available in 24-Pin DIP and 28-Pin PLCC**

The Intel 82C54 is a high-performance, CHMOS version of the industry standard 8254 counter/timer which is designed to solve the timing control problems common in microcomputer system design. It provides three independent 16-bit counters, each capable of handling clock inputs up to 10 MHz. All modes are software programmable. The 82C54 is pin compatible with the HMOS 8254, and is a superset of the 8253.

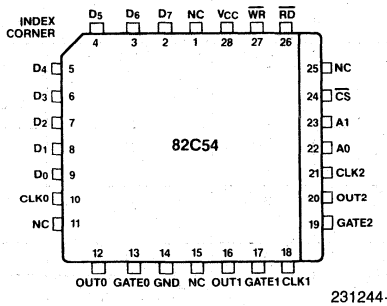
Six programmable timer modes allow the 82C54 to be used as an event counter, elapsed time indicator, programmable one-shot, and in many other applications.

The 82C54 is fabricated on Intel's advanced CHMOS III technology which provides low power consumption with performance equal to or greater than the equivalent HMOS product. The 82C54 is available in 24-pin DIP and 28-pin plastic leaded chip carrier (PLCC) packages.



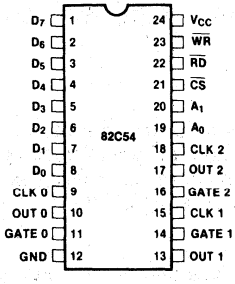
231244-1

Figure 1. 82C54 Block Diagram



231244-3

PLASTIC LEADED CHIP CARRIER



231244-2

Diagrams are for pin reference only.
Package sizes are not to scale.

Figure 2. 82C54 Pinout

Table 1. Pin Description

Symbol	Pin Number		Type	Function
	DIP	PLCC		
D_7-D_0	1-8	2-9	I/O	Data: Bidirectional tri-state data bus lines, connected to system data bus.
CLK 0	9	10	I	Clock 0: Clock input of Counter 0.
OUT 0	10	12	O	Output 0: Output of Counter 0.
GATE 0	11	13	I	Gate 0: Gate input of Counter 0.
GND	12	14		Ground: Power supply connection.
OUT 1	13	16	O	Out 1: Output of Counter 1.
GATE 1	14	17	I	Gate 1: Gate input of Counter 1.
CLK 1	15	18	I	Clock 1: Clock input of Counter 1.
GATE 2	16	19	I	Gate 2: Gate input of Counter 2.
OUT 2	17	20	O	Out 2: Output of Counter 2.
CLK 2	18	21	I	Clock 2: Clock input of Counter 2.
A_1, A_0	20-19	23-22	I	Address: Used to select one of the three Counters or the Control Word Register for read or write operations. Normally connected to the system address bus.
				A_1 / A_0 / Selects: 0 / 0 / Counter 0; 0 / 1 / Counter 1; 1 / 0 / Counter 2; 1 / 1 / Control Word Register
$\overline{CS}$	21	24	I	Chip Select: A low on this input enables the 82C54 to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and $\overline{WR}$ are ignored otherwise.
$\overline{RD}$	22	26	I	Read Control: This input is low during CPU read operations.
$\overline{WR}$	23	27	I	Write Control: This input is low during CPU write operations.
V_{CC}	24	28		Power: +5V power supply connection.
NC		1, 11, 15, 25		No Connect

FUNCTIONAL DESCRIPTION

General

The 82C54 is a programmable interval timer/counter designed for use with Intel microcomputer systems. It is a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 82C54 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in software, the programmer configures the 82C54 to match his requirements and programs one of the counters for the desired delay. After the desired delay, the 82C54 will interrupt the CPU. Software overhead is minimal and variable length delays can easily be accommodated.

Some of the other counter/timer functions common to microcomputers which can be implemented with the 82C54 are:

- Real time clock
- Even counter
- Digital one-shot
- Programmable rate generator
- Square wave generator
- Binary rate multiplier
- Complex waveform generator
- Complex motor controller

Block Diagram

DATA BUS BUFFER

This 3-state, bi-directional, 8-bit buffer is used to interface the 82C54 to the system bus (see Figure 3).

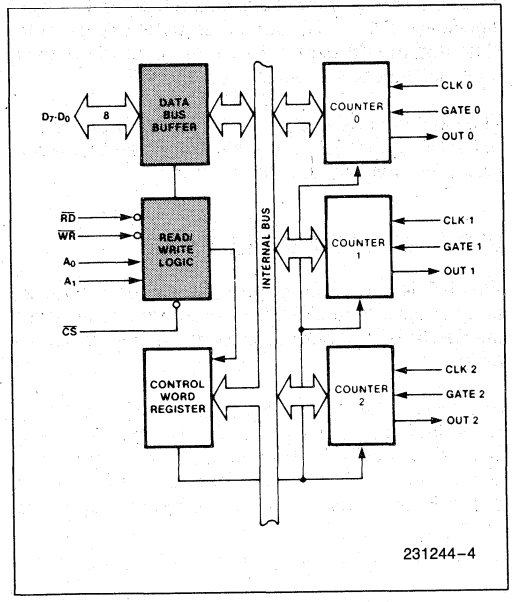


Figure 3. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

READ/WRITE LOGIC

The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the 82C54. A_1 and A_0 select one of the three counters or the Control Word Register to be read from/written into. A "low" on the $\overline{RD}$ input tells the 82C54 that the CPU is reading one of the counters. A "low" on the $\overline{WR}$ input tells the 82C54 that the CPU is writing either a Control Word or an initial count. Both $\overline{RD}$ and $\overline{WR}$ are qualified by $\overline{CS}$; $\overline{RD}$ and $\overline{WR}$ are ignored unless the 82C54 has been selected by holding $\overline{CS}$ low.

CONTROL WORD REGISTER

The Control Word Register (see Figure 4) is selected by the Read/Write Logic when A_1, A_0 = 11. If the CPU then does a write operation to the 82C54, the data is stored in the Control Word Register and is interpreted as a Control Word used to define the operation of the Counters.

The Control Word Register can only be written to; status information is available with the Read-Back Command.

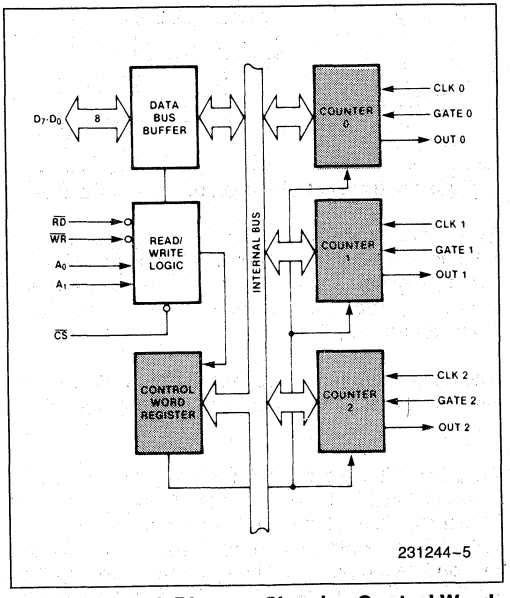


Figure 4. Block Diagram Showing Control Word Register and Counter Functions

COUNTER 0, COUNTER 1, COUNTER 2

These three functional blocks are identical in operation, so only a single Counter will be described. The internal block diagram of a single counter is shown in Figure 5.

The Counters are fully independent. Each Counter may operate in a different Mode.

The Control Word Register is shown in the figure; it is not part of the Counter itself, but its contents determine how the Counter operates.

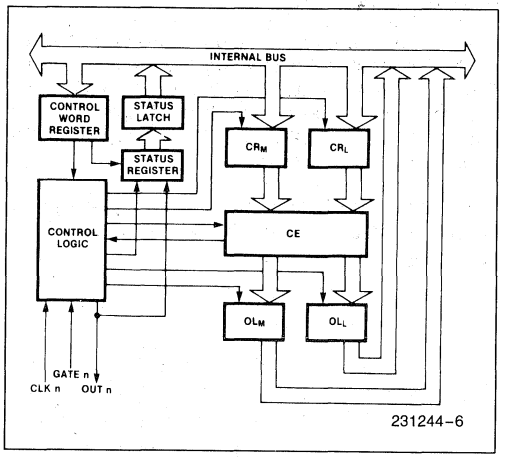


Figure 5. Internal Block Diagram of a Counter

The status register, shown in the Figure, when latched, contains the current contents of the Control Word Register and status of the output and null count flag. (See detailed explanation of the Read-Back command.)

The actual counter is labelled CE (for "Counting Element"). It is a 16-bit presettable synchronous down counter.

OL_M and OL_L are two 8-bit latches. OL stands for "Output Latch"; the subscripts M and L stand for "Most significant byte" and "Least significant byte" respectively. Both are normally referred to as one unit and called just OL. These latches normally "follow" the CE, but if a suitable Counter Latch Command is sent to the 82C54, the latches "latch" the present count until read by the CPU and then return to "following" the CE. One latch at a time is enabled by the counter's Control Logic to drive the internal bus. This is how the 16-bit Counter communicates over the 8-bit internal bus. Note that the CE itself cannot be read; whenever you read the count, it is the OL that is being read.

Similarly, there are two 8-bit registers called CR_M and CR_L (for "Count Register"). Both are normally referred to as one unit and called just CR. When a new count is written to the Counter, the count is stored in the CR and later transferred to the CE. The Control Logic allows one register at a time to be loaded from the internal bus. Both bytes are transferred to the CE simultaneously. CR_M and CR_L are cleared when the Counter is programmed. In this way, if the Counter has been programmed for one byte counts (either most significant byte only or least significant byte only) the other byte will be zero. Note that the CE cannot be written into; whenever a count is written, it is written into the CR.

The Control Logic is also shown in the diagram. CLK n, GATE n, and OUT n are all connected to the outside world through the Control Logic.

82C54 SYSTEM INTERFACE

The 82C54 is treated by the systems software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A_0, A_1 connect to the A_0, A_1 address bus signals of the CPU. The $\overline{CS}$ can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an Intel 8205 for larger systems.

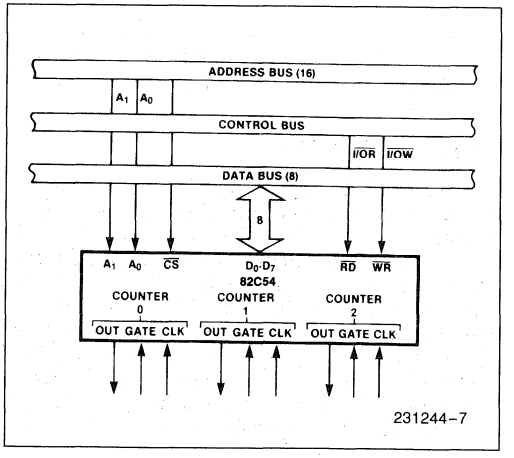


Figure 6. 82C54 System Interface

OPERATIONAL DESCRIPTION

General

After power-up, the state of the 82C54 is undefined. The Mode, count value, and output of all Counters are undefined.

How each Counter operates is determined when it is programmed. Each Counter must be programmed before it can be used. Unused counters need not be programmed.

Programming the 82C54

Counters are programmed by writing a Control Word and then an initial count. The control word format is shown in Figure 7.

All Control Words are written into the Control Word Register, which is selected when A_1, A_0 = 11. The Control Word itself specifies which Counter is being programmed.

By contrast, initial counts are written into the Counters, not the Control Word Register. The A_1, A_0 inputs are used to select the Counter to be written into. The format of the initial count is determined by the Control Word used.

Control Word Format

A_1, A_0 = 11 $\overline{CS}$ = 0 $\overline{RD}$ = 1 $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC — Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (See Read Operations)

RW — Read/Write:

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only.
1	0	Read/Write most significant byte only.
1	1	Read/Write least significant byte first, then most significant byte.

M — MODE:

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD:

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE: Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Figure 7. Control Word Format

Write Operations

The programming procedure for the 82C54 is very flexible. Only two conventions need to be remembered:

1) For each Counter, the Control Word must be written before the initial count is written.
2) The initial count must follow the count format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

Since the Control Word Register and the three Counters have separate addresses (selected by the A_1, A_0 inputs), and each Control Word specifies the Counter it applies to (SC0, SC1 bits), no special instruction sequence is required. Any programming sequence that follows the conventions above is acceptable.

A new initial count may be written to a Counter at any time without affecting the Counter's programmed Mode in any way. Counting will be affected as described in the Mode definitions. The new count must follow the programmed count format.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same Counter. Otherwise, the Counter will be loaded with an incorrect count.

		A_1	A_0
Control Word —	Counter 0	1	1
LSB of count —	Counter 0	0	0
MSB of count —	Counter 0	0	0
Control Word —	Counter 1	1	1
LSB of count —	Counter 1	0	1
MSB of count —	Counter 1	0	1
Control Word —	Counter 2	1	1
LSB of count —	Counter 2	1	0
MSB of count —	Counter 2	1	0

		A_1	A_0
Control Word —	Counter 2	1	1
Control Word —	Counter 1	1	1
Control Word —	Counter 0	1	1
LSB of count —	Counter 2	1	0
MSB of count —	Counter 2	1	0
LSB of count —	Counter 1	0	1
MSB of count —	Counter 1	0	1
LSB of count —	Counter 0	0	0
MSB of count —	Counter 0	0	0

		A_1	A_0
Control Word —	Counter 0	1	1
Counter Word —	Counter 1	1	1
Control Word —	Counter 2	1	1
LSB of count —	Counter 2	1	0
LSB of count —	Counter 1	0	1
LSB of count —	Counter 0	0	0
MSB of count —	Counter 0	0	0
MSB of count —	Counter 1	0	1
MSB of count —	Counter 2	1	0

		A_1	A_0
Control Word —	Counter 1	1	1
Control Word —	Counter 0	1	1
LSB of count —	Counter 1	0	1
Control Word —	Counter 2	1	1
LSB of count —	Counter 0	0	0
MSB of count —	Counter 1	0	1
LSB of count —	Counter 2	1	0
MSB of count —	Counter 0	0	0
MSB of count —	Counter 2	1	0

NOTE:
In all four examples, all counters are programmed to read/write two-byte counts.
These are only four of many possible programming sequences.

Figure 8. A Few Possible Programming Sequences

Read Operations

It is often desirable to read the value of a Counter without disturbing the count in progress. This is easily done in the 82C54.

There are three possible methods for reading the counters: a simple read operation, the Counter Latch Command, and the Read-Back Command. Each is explained below. The first method is to perform a simple read operation. To read the Counter, which is selected with the A1, A0 inputs, the CLK input of the selected Counter must be inhibited by using either the GATE input or external logic. Otherwise, the count may be in the process of changing when it is read, giving an undefined result.

COUNTER LATCH COMMAND

The second method uses the "Counter Latch Command". Like a Control Word, this command is written to the Control Word Register, which is selected when A_1, A_0 = 11. Also like a Control Word, the SC0, SC1 bits select one of the three Counters, but two other bits, D5 and D4, distinguish this command from a Control Word.

$A_1, A_0 = 11; \overline{CS} = 0; \overline{RD} = 1; \overline{WR} = 0$

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	0	0	X	X	X	X

SC1, SC0 - specify counter to be latched

SC1	SC0	Counter
0	0	0
0	1	1
1	0	2
1	1	Read-Back Command

D5,D4 - 00 designates Counter Latch Command

X - don't care

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Figure 9. Counter Latching Command Format

The selected Counter's output latch (OL) latches the count at the time the Counter Latch Command is received. This count is held in the latch until it is read by the CPU (or until the Counter is reprogrammed). The count is then unlatched automatically and the OL returns to "following" the counting element (CE). This allows reading the contents of the Counters "on the fly" without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one Counter. Each latched Counter's OL holds its count until it is read. Counter Latch Commands do not affect the programmed Mode of the Counter in any way.

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued.

With either method, the count must be read according to the programmed format; specifically, if the Counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other; read or write or programming operations of other Counters may be inserted between them.

Another feature of the 82C54 is that reads and writes of the same Counter may be interleaved; for example, if the Counter is programmed for two byte counts, the following sequence is valid.

1. Read least significant byte.
2. Write new least significant byte.
3. Read most significant byte.
4. Write new most significant byte.

If a Counter is programmed to read/write two-byte counts, the following precaution applies; A program must not transfer control between reading the first and second byte to another routine which also reads from that same Counter. Otherwise, an incorrect count will be read.

READ-BACK COMMAND

The third method uses the Read-Back command. This command allows the user to check the count value, programmed Mode, and current state of the OUT pin and Null Count flag of the selected counter(s).

The command is written into the Control Word Register and has the format shown in Figure 10. The command applies to the counters selected by setting their corresponding bits D3,D2,D1 = 1.

A0, A1 = 11 $\overline{CS}$ = 0 $\overline{RD}$ = 1 $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	$\overline{COUNT}$	$\overline{STATUS}$	CNT 2	CNT 1	CNT 0	0

D_5: 0 = Latch count of selected counter(s)
D_4: 0 = Latch status of selected counter(s)
D_3: 1 = Select counter 2
D_2: 1 = Select counter 1
D_1: 1 = Select counter 0
D_0: Reserved for future expansion; must be 0

Figure 10. Read-Back Command Format

The read-back command may be used to latch multiple counter output latches (OL) by setting the $\overline{COUNT}$ bit D5 = 0 and selecting the desired counter(s). This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read (or the counter is reprogrammed). That counter is automatically unlatched when read, but other counters remain latched until they are read. If multiple count read-back commands are issued to the same counter without reading the

count, all but the first are ignored; i.e., the count which will be read is the count at the time the first read-back command was issued.

The read-back command may also be used to latch status information of selected counter(s) by setting $\overline{\text{STATUS}}$ bit D4 = 0. Status must be latched to be read; status of a counter is accessed by a read from that counter.

The counter status format is shown in Figure 11. Bits D5 through D0 contain the counter's programmed Mode exactly as written in the last Mode Control Word. OUTPUT bit D7 contains the current state of the OUT pin. This allows the user to monitor the counter's output via software, possibly eliminating some hardware from a system.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
OUTPUT	NULL COUNT	RW1	RW0	M2	M1	M0	BCD

D_7 1 = Out Pin is 1
0 = Out Pin is 0
D_6 1 = Null count
0 = Count available for reading
D_5-D_0 Counter Programmed Mode (See Figure 7)

Figure 11. Status Byte

NULL COUNT bit D6 indicates when the last count written to the counter register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the Mode of the counter and is described in the Mode Definitions, but until the count is loaded into the counting element (CE), it can't be read from the counter. If the count is latched or read before this time, the count value will not reflect the new count just written. The operation of Null Count is shown in Figure 12.

THIS ACTION:	CAUSES:
A. Write to the control word register:[1]	Null count = 1
B. Write to the count register (CR);[2]	Null count = 1
C. New count is loaded into CE (CR → CE);	Null count = 0

[1] Only the counter specified by the control word will have its null count set to 1. Null count bits of other counters are unaffected.

[2] If the counter is programmed for two-byte counts (least significant byte then most significant byte) null count goes to 1 when the second byte is written.

Figure 12. Null Count Operation

If multiple status latch operations of the counter(s) are performed without reading the status, all but the first are ignored; i.e., the status that will be read is the status of the counter at the time the first status read-back command was issued.

Both count and status of the selected counter(s) may be latched simultaneously by setting both $\overline{\text{COUNT}}$ and $\overline{\text{STATUS}}$ bits D5,D4 = 0. This is functionally the same as issuing two separate read-back commands at once, and the above discussions apply here also. Specifically, if multiple count and/or status read-back commands are issued to the same counter(s) without any intervening reads, all but the first are ignored. This is illustrated in Figure 13.

If both count and status of a counter are latched, the first read operation of that counter will return latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two type counts) return latched count. Subsequent reads return unlatched count.

Command								Description	Results
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
1	1	0	0	0	0	1	0	Read back count and status of Counter 0	Count and status latched for Counter 0
1	1	1	0	0	1	0	0	Read back status of Counter 1	Status latched for Counter 1
1	1	1	0	1	1	0	0	Read back status of Counters 2, 1	Status latched for Counter 2, but not Counter 1
1	1	0	1	1	0	0	0	Read back count of Counter 2	Count latched for Counter 2
1	1	0	0	0	1	0	0	Read back count and status of Counter 1	Count latched for Counter 1, but not status
1	1	1	0	0	0	1	0	Read back status of Counter 1	Command ignored, status already latched for Counter 1

Figure 13. Read-Back Command Example

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Write into Counter 0
0	1	0	0	1	Write into Counter 1
0	1	0	1	0	Write into Counter 2
0	1	0	1	1	Write Control Word
0	0	1	0	0	Read from Counter 0
0	0	1	0	1	Read from Counter 1
0	0	1	1	0	Read from Counter 2
0	0	1	1	1	No-Operation (3-State)
1	X	X	X	X	No-Operation (3-State)
0	1	1	X	X	No-Operation (3-State)

Figure 14. Read/Write Operations Summary

Mode Definitions

The following are defined for use in describing the operation of the 82C54.

CLK PULSE: a rising edge, then a falling edge, in that order, of a Counter's CLK input.

TRIGGER: a rising edge of a Counter's GATE input.

COUNTER LOADING: the transfer of a count from the CR to the CE (refer to the "Functional Description")

MODE 0: INTERRUPT ON TERMINAL COUNT

Mode 0 is typically used for event counting. After the Control Word is written, OUT is initially low, and will remain low until the Counter reaches zero. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written into the Counter.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

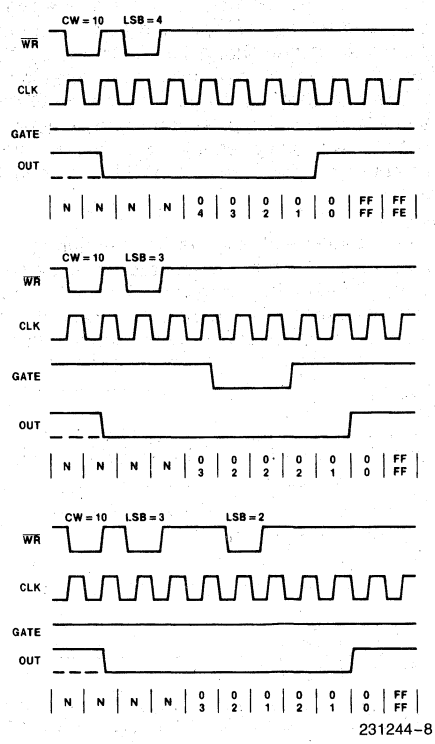
After the Control Word and initial count are written to a Counter, the initial count will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not go high until N + 1 CLK pulses after the initial count is written.

If a new count is written to the Counter, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte disables counting. OUT is set low immediately (no clock pulse required).
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the counting sequence to be synchronized by software. Again, OUT does not go high until N + 1 CLK pulses after the new count of N is written.

If an initial count is written while GATE = 0, it will still be loaded on the next CLK pulse. When GATE goes high, OUT will go high N CLK pulses later; no CLK pulse is needed to load the Counter as this has already been done.



NOTE:
The Following Conventions Apply To All Mode Timing Diagrams:
1. Counters are programmed for binary (not BCD) counting and for Reading/Writing least significant byte (LSB) only.
2. The counter is always selected ($\overline{CS}$ always low).
3. CW stands for "Control Word"; CW = 10 means a control word of 10, hex is written to the counter.
4. LSB stands for "Least Significant Byte" of count.
5. Numbers below diagrams are count values.
The lower number is the least significant byte.
The upper number is the most significant byte. Since the counter is programmed to Read/Write LSB only, the most significant byte cannot be read.
N stands for an undefined count.
Vertical lines show transitions between count values.

Figure 15. Mode 0

MODE 1: HARDWARE RETRIGGERABLE ONE-SHOT

OUT will be initially high. OUT will go low on the CLK pulse following a trigger to begin the one-shot pulse, and will remain low until the Counter reaches zero. OUT will then go high and remain high until the CLK pulse after the next trigger.

After writing the Control Word and initial count, the Counter is armed. A trigger results in loading the Counter and setting OUT low on the next CLK pulse, thus starting the one-shot pulse. An initial count of N will result in a one-shot pulse N CLK cycles in duration. The one-shot is retriggerable, hence OUT will remain low for N CLK pulses after any trigger. The one-shot pulse can be repeated without rewriting the same count into the counter. GATE has no effect on OUT.

If a new count is written to the Counter during a one-shot pulse, the current one-shot is not affected unless the Counter is retriggered. In that case, the Counter is loaded with the new count and the one-shot pulse continues until the new count expires.

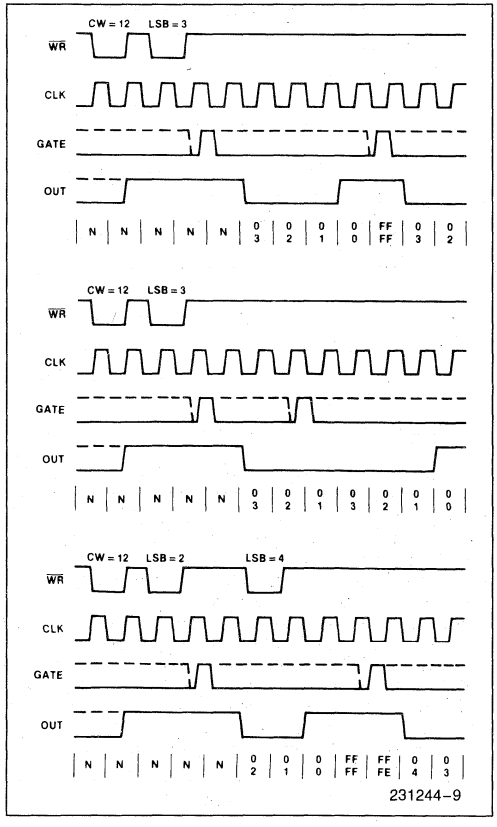


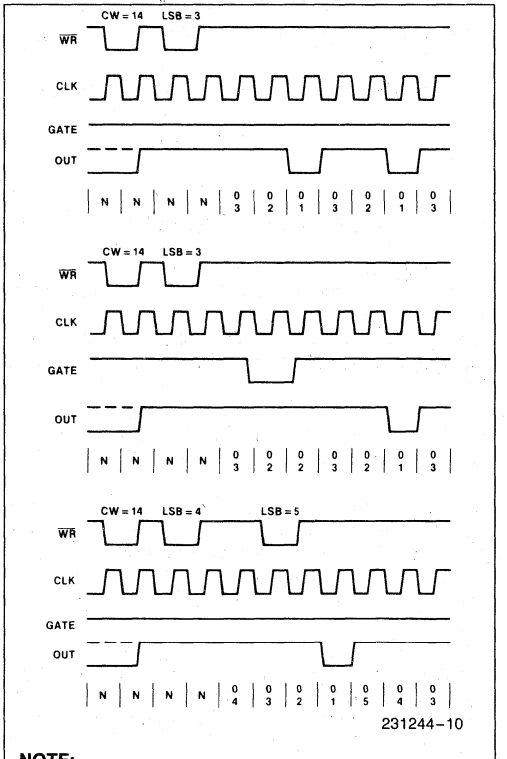
Figure 16. Mode 1

MODE 2: RATE GENERATOR

This Mode functions like a divide-by-N counter. It is typically used to generate a Real Time Clock interrupt. OUT will initially be high. When the initial count has decremented to 1, OUT goes low for one CLK pulse. OUT then goes high again, the Counter reloads the initial count and the process is repeated. Mode 2 is periodic; the same sequence is repeated indefinitely. For an initial count of N, the sequence repeats every N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low during an output pulse, OUT is set high immediately. A trigger reloads the Counter with the initial count on the next CLK pulse; OUT goes low N CLK pulses after the trigger. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. OUT goes low N CLK Pulses after the initial count is written. This allows the Counter to be synchronized by software also.



NOTE:
A GATE transition should not occur one clock prior to terminal count.

Figure 17. Mode 2

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current period, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current counting cycle. In mode 2, a COUNT of 1 is illegal.

MODE 3: SQUARE WAVE MODE

Mode 3 is typically used for Baud rate generation. Mode 3 is similar to Mode 2 except for the duty cycle of OUT. OUT will initially be high. When half the initial count has expired, OUT goes low for the remainder of the count. Mode 3 is periodic; the sequence above is repeated indefinitely. An initial count of N results in a square wave with a period of N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low while OUT is low, OUT is set high immediately; no CLK pulse is required. A trigger reloads the Counter with the initial count on the next CLK pulse. Thus the GATE input can be used to synchronize the Counter.

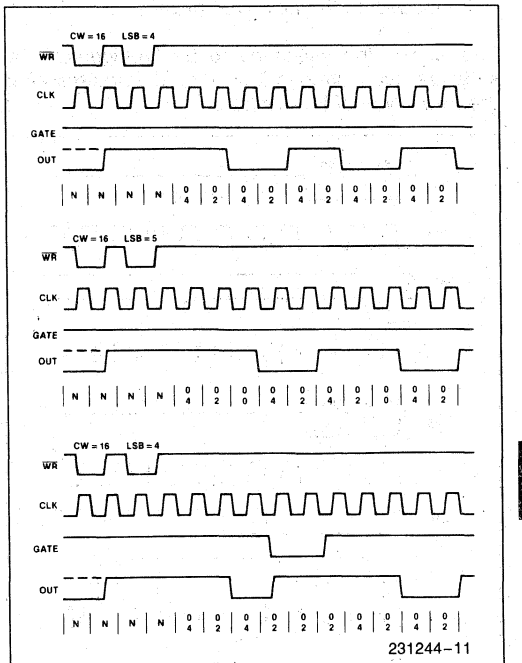
After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current half-cycle of the square wave, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current half-cycle.

Mode 3 is implemented as follows:

Even counts: OUT is initially high. The initial count is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. When the count expires OUT changes value and the Counter is reloaded with the initial count. The above process is repeated indefinitely.

Odd counts: OUT is initially high. The initial count minus one (an even number) is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. One CLK pulse *after* the count expires, OUT goes low and the Counter is reloaded with the initial count minus one. Succeeding CLK pulses decrement the count by two. When the count expires, OUT goes high again and the Counter is reloaded with the initial count minus one. The above process is repeated indefinitely. So for odd counts, OUT will be high for (N + 1)/2 counts and low for (N −1)/2 counts.



NOTE:
A GATE transition should not occur one clock prior to terminal count.

Figure 18. Mode 3

MODE 4: SOFTWARE TRIGGERED STROBE

OUT will be initially high. When the initial count expires, OUT will go low for one CLK pulse and then go high again. The counting sequence is "triggered" by writing the initial count.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N + 1 CLK pulses after the initial count is written.

If a new count is written during counting, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte has no effect on counting.
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the sequence to be "retriggered" by software. OUT strobes low N+1 CLK pulses after the new count of N is written.

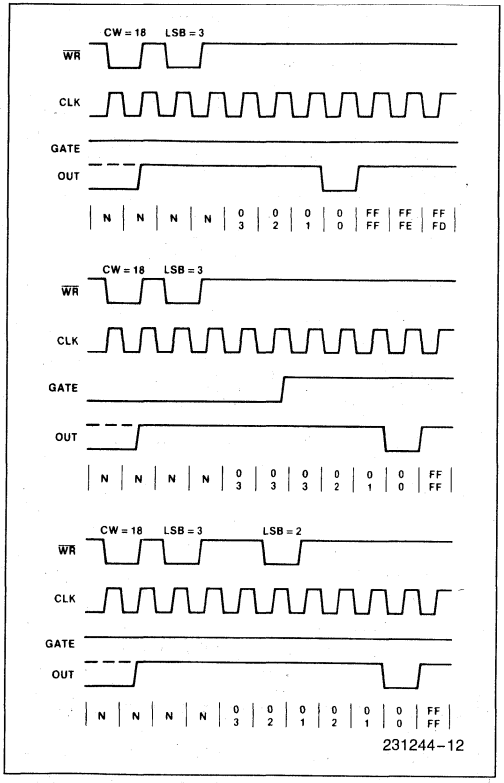


Figure 19. Mode 4

MODE 5: HARDWARE TRIGGERED STROBE (RETRIGGERABLE)

OUT will initially be high. Counting is triggered by a rising edge of GATE. When the initial count has expired, OUT will go low for one CLK pulse and then go high again.

After writing the Control Word and initial count, the counter will not be loaded until the CLK pulse after a trigger. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N+1 CLK pulses after a trigger.

A trigger results in the Counter being loaded with the initial count on the next CLK pulse. The counting sequence is retriggerable. OUT will not strobe low for N + 1 CLK pulses after any trigger. GATE has no effect on OUT.

If a new count is written during counting, the current counting sequence will not be affected. If a trigger occurs after the new count is written but before the current count expires, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from there.

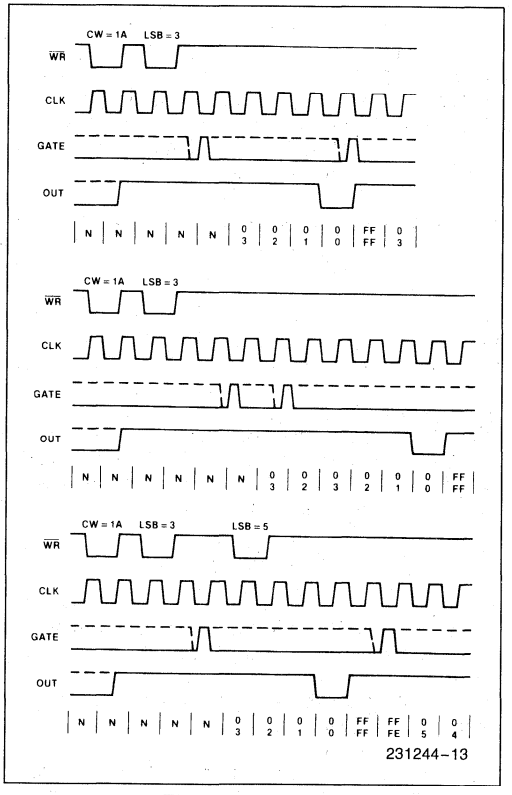


Figure 20. Mode 5

Signal Status Modes	Low Or Going Low	Rising	High
0	Disables counting	—	Enables counting
1	—	1) Initiates counting 2) Resets output after next clock	—
2	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	—	Enables counting
5	—	Initiates counting	—

Figure 21. Gate Pin Operations Summary

MODE	MIN COUNT	MAX COUNT
0	1	0
1	1	0
2	2	0
3	2	0
4	1	0

NOTE:
0 is equivalent to 2^{16} for binary counting and 10^4 for BCD counting

Figure 22. Minimum and Maximum initial Counts

Operation Common to All Modes

Programming

When a Control Word is written to a Counter, all Control Logic is immediately reset and OUT goes to a known initial state; no CLK pulses are required for this.

GATE

The GATE input is always sampled on the rising edge of CLK. In Modes 0, 2, 3, and 4 the GATE input is level sensitive, and the logic level is sampled on the rising edge of CLK. In Modes 1, 2, 3, and 5 the GATE input is rising-edge sensitive. In these Modes, a rising edge of GATE (trigger) sets an edge-sensitive flip-flop in the Counter. This flip-flop is then sampled on the next rising edge of CLK; the flip-flop is reset immediately after it is sampled. In this way, a trigger will be detected no matter when it occurs—a high logic level does not have to be maintained until the next rising edge of CLK. Note that in Modes 2 and 3, the GATE input is both edge- and level-sensitive. In Modes 2 and 3, if a CLK source other than the system clock is used, GATE should be pulsed immediately following $\overline{WR}$ of a new count value.

COUNTER

New counts are loaded and Counters are decremented on the falling edge of CLK.

The largest possible initial count is 0; this is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

The Counter does not stop when it reaches zero. In Modes 0, 1, 4, and 5 the Counter "wraps around" to the highest count, either FFFF hex for binary counting or 9999 for BCD counting, and continues counting. Modes 2 and 3 are periodic; the Counter reloads itself with the initial count and continues counting from there.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias.....0°C to 70°C
Storage Temperature .....−65° to +150°C
Supply Voltage .....−0.5 to +8.0V
Operating Voltage .....+4V to +7V
Voltage on any Input.....GND −2V to +6.5V
Voltage on any Output ..GND−0.5V to V_{CC} + 0.5V
Power Dissipation .....1 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS

(T_A=0°C to 70°C, V_{CC}=5V± 10%, GND=0V) (T_A = −40°C to +85°C for Extended Temperature)

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.5 mA
V_{OH}	Output High Voltage	3.0 V_{CC} − 0.4		V V	I_{OH} = −2.5 mA I_{OH} = −100 μA
I_{IL}	Input Load Current		±2.0	μA	V_{IN}=V_{CC} to 0V
I_{OFL}	Output Float Leakage Current		±10	μA	V_{OUT}=V_{CC} to 0.0V
I_{CC}	V_{CC} Supply Current		20	mA	Clk Freq= 8MHz 82C54 10MHz 82C54-2
I_{CCSB}	V_{CC} Supply Current-Standby		10	μA	CLK Freq = DC $\overline{CS}$ = V_{CC}. All Inputs/Data Bus V_{CC} All Outputs Floating
I_{CCSB1}	V_{CC} Supply Current-Standby		150	μA	CLK Freq = DC $\overline{CS}$ = V_{CC}. All Other Inputs, I/O Pins = V_{GND}, Outputs Open
C_{IN}	Input Capacitance		10	pF	f_c = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to GND(5)
C_{OUT}	Output Capacitance		20	pF	

A.C. CHARACTERISTICS

(T_A = 0°C to 70°C, V_{CC} = 5V ±10%, GND =0V) (T_A = −40°C to +85°C for Extended Temperature)

BUS PARAMETERS (Note 1)

READ CYCLE

Symbol	Parameter	82C54		82C54-2		Units
		Min	Max	Min	Max	
t_{AR}	Address Stable Before $\overline{RD}$ ↓	45		30		ns
t_{SR}	$\overline{CS}$ Stable Before $\overline{RD}$ ↓	0		0		ns
t_{RA}	Address Hold Time After $\overline{RD}$ ↑	0		0		ns
t_{RR}	$\overline{RD}$ Pulse Width	150		95		ns
t_{RD}	Data Delay from $\overline{RD}$ ↓		120		85	ns
t_{AD}	Data Delay from Address		220		185	ns
t_{DF}	$\overline{RD}$ ↑ to Data Floating	5	90	5	65	ns
t_{RV}	Command Recovery Time	200		165		ns

NOTE:
1. AC timings measured at V_{OH} = 2.0V, V_{OL} = 0.8V.

A.C. CHARACTERISTICS (Continued)

WRITE CYCLE

Symbol	Parameter	82C54		82C54-2		Units
		Min	Max	Min	Max	
t_{AW}	Address Stable Before $\overline{WR}\downarrow$	0		0		ns
t_{SW}	$\overline{CS}$ Stable Before $\overline{WR}\downarrow$	0		0		ns
t_{WA}	Address Hold Time After $\overline{WR}\uparrow$	0		0		ns
t_{WW}	$\overline{WR}$ Pulse Width	150		95		ns
t_{DW}	Data Setup Time Before $\overline{WR}\uparrow$	120		95		ns
t_{WD}	Data Hold Time After $\overline{WR}\uparrow$	0		0		ns
t_{RV}	Command Recovery Time	200		165		ns

CLOCK AND GATE

Symbol	Parameter	82C54		82C54-2		Units
		Min	Max	Min	Max	
t_{CLK}	Clock Period	125	DC	100	DC	ns
t_{PWH}	High Pulse Width	60(3)		30(3)		ns
t_{PWL}	Low Pulse Width	60(3)		50(3)		ns
T_R	Clock Rise Time		25		25	ns
t_F	Clock Fall Time		25		25	ns
t_{GW}	Gate Width High	50		50		ns
t_{GL}	Gate Width Low	50		50		ns
t_{GS}	Gate Setup Time to CLK $\uparrow$	50		40		ns
t_{GH}	Gate Hold Time After CLK $\uparrow$	50(2)		50(2)		ns
T_{OD}	Output Delay from CLK $\downarrow$		150		100	ns
t_{ODG}	Output Delay from Gate $\downarrow$		120		100	ns
t_{WC}	CLK Delay for Loading(4)	0	55	0	55	ns
t_{WG}	Gate Delay for Sampling(4)	−5	50	−5	40	ns
t_{WO}	OUT Delay from Mode Write		260		240	ns
t_{CL}	CLK Set Up for Count Latch	−40	45	−40	40	ns

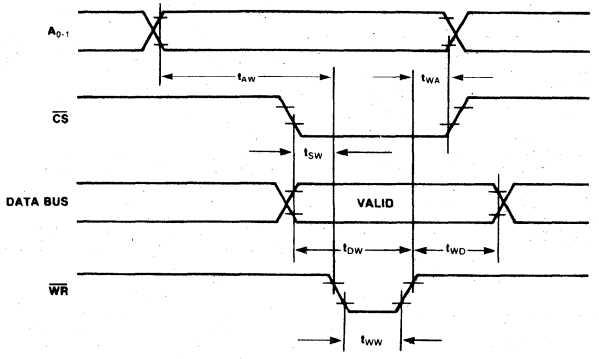
NOTES:
2. In Modes 1 and 5 triggers are sampled on each rising clock edge. A second trigger within 120 ns (70 ns for the 82C54-2) of the rising clock edge may not be detected.
3. Low-going glitches that violate t_{PWH}, t_{PWL} may cause errors requiring counter reprogramming.
4. Except for Extended Temp., See Extended Temp. A.C. Characteristics below.
5. Sampled not 100% tested. T_A = 25°C.
6. If CLK present at T_{WC} min then Count equals N+2 CLK pulses, T_{WC} max equals Count N+1 CLK pulse. T_{WC} min to T_{WC} max, count will be either N+1 or N+2 CLK pulses.
7. In Modes 1 and 5, if GATE is present when writing a new Count value, at T_{WG} min Counter will not be triggered, at T_{WG} max Counter will be triggered.
8. If CLK present when writing a Counter Latch or ReadBack Command, at T_{CL} min CLK will be reflected in count value latched, at T_{CL} max CLK will not be reflected in the count value latched. Writing a Counter Latch or ReadBack Command between T_{CL} min and T_{WL} max will result in a latched count vallue which is ± one least significant bit.

EXTENDED TEMPERATURE (T_A = −40°C to +85°C for Extended Temperature)

Symbol	Parameter	82C54		82C54-2		Units
		Min	Max	Min	Max	
t_{WC}	CLK Delay for Loading	−25	25	−25	25	ns
t_{WG}	Gate Delay for Sampling	−25	25	−25	25	ns

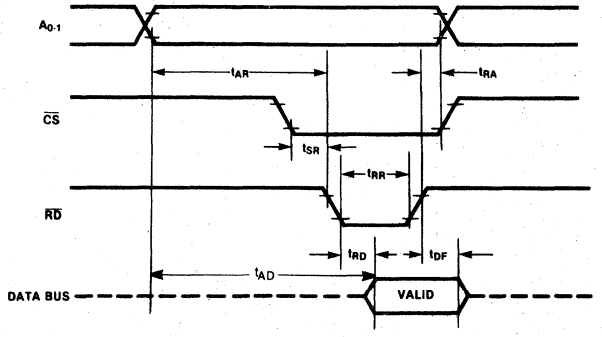
WAVEFORMS

WRITE



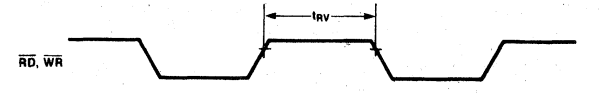
231244-14

READ



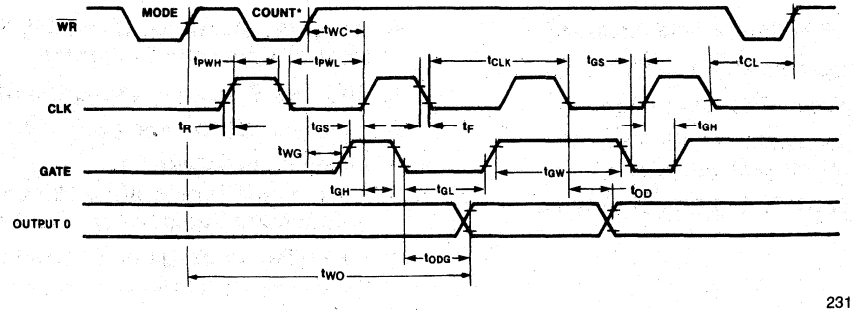
231244-15

RECOVERY



231244-16

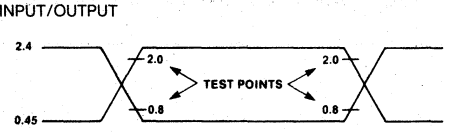
CLOCK AND GATE



231244-17

* Last byte of count being written

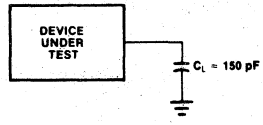
A.C. TESTING INPUT, OUTPUT WAVEFORM



231244-18

A.C. Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0." Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0."

A.C. TESTING LOAD CIRCUIT



231244-19

C_L = 150 pF

C_L includes jig capacitance

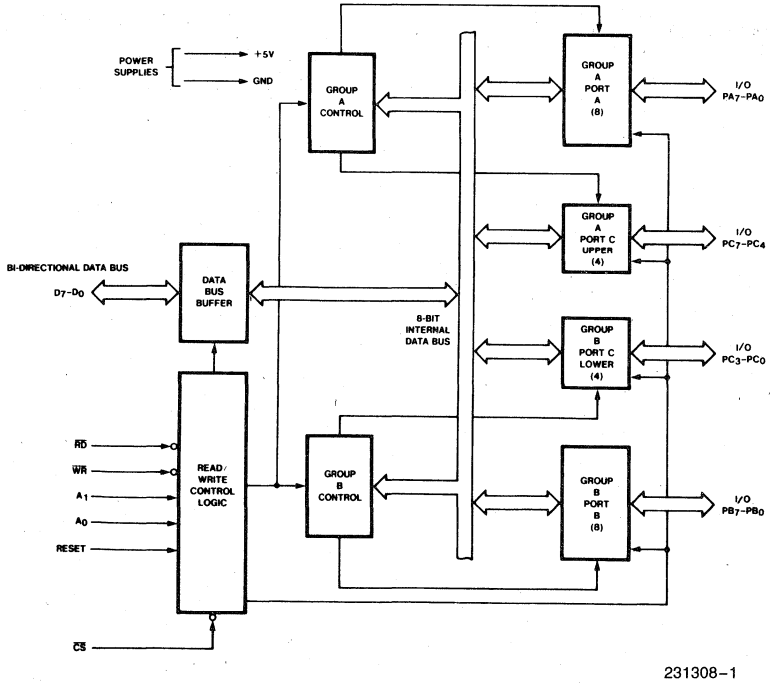
3

8255A/8255A-5
PROGRAMMABLE PERIPHERAL INTERFACE

- **MCS-85™ Compatible 8255A-5**
- **24 Programmable I/O Pins**
- **Completely TTL Compatible**
- **Fully Compatible with Intel Microprocessor Families**
- **Improved Timing Characteristics**
- **Direct Bit Set/Reset Capability Easing Control Application Interface**
- **Reduces System Package Count**
- **Improved DC Driving Capability**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**
- **40 Pin DIP Package or 44 Lead PLCC**

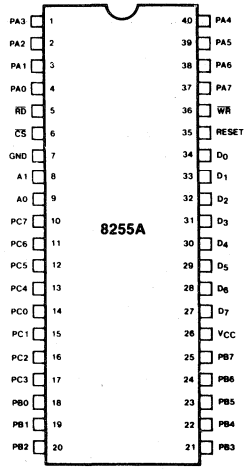
(See Intel Packaging: Order Number: 231369)

The Intel 8255A is a general purpose programmable I/O device designed for use with Intel microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.



231308-1

Figure 1. 8255A Block Diagram



231308-2

Figure 2. Pin Configuration

November 1987
Order Number: 231308-002

8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

($\overline{CS}$)

Chip Select. A "low" on this input pin enables the communication between the 8255A and the CPU.

($\overline{RD}$)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

($\overline{WR}$)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

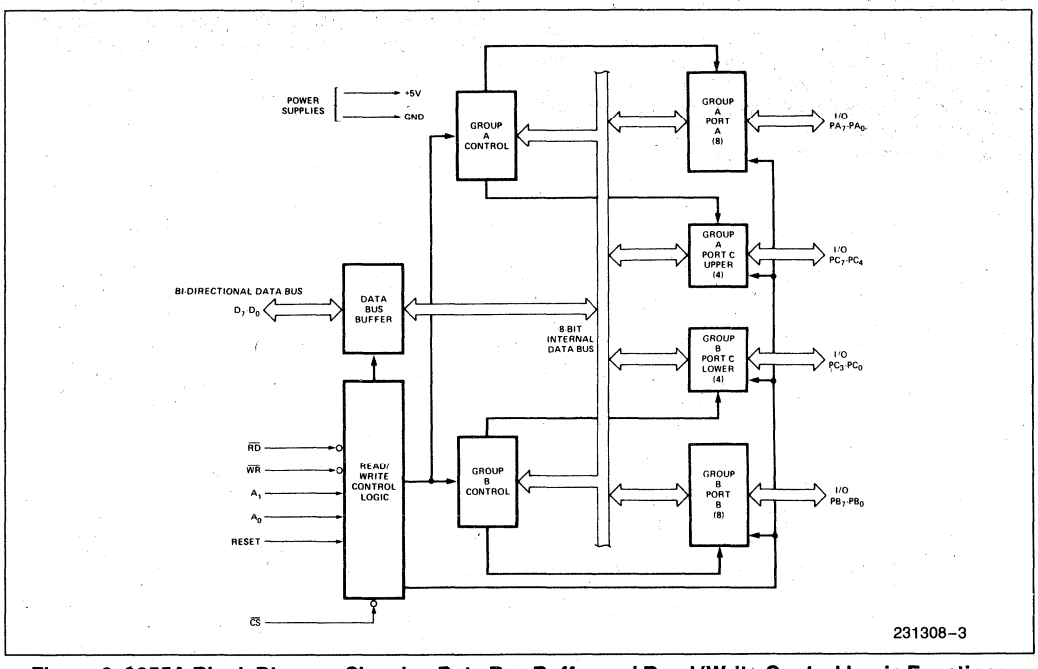


Figure 3. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

8255A BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Input Operation (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
					Output Operation (WRITE)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					Disable Function
X	X	X	X	1	Data Bus → 3-State
1	1	0	1	0	Illegal Condition
X	X	1	1	0	Data Bus → 3-State

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A—Port A and Port C upper (C7–C4)

Control Group B—Port B and Port C lower (C3–C0)

The Control Word Register can **Only** be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

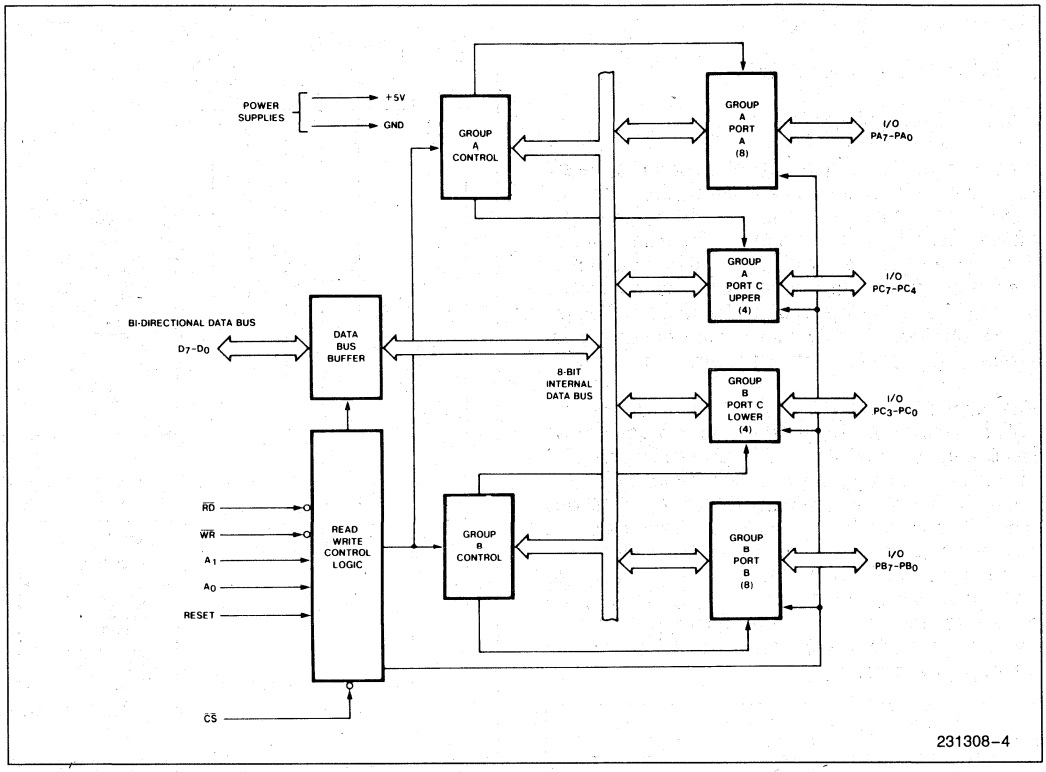
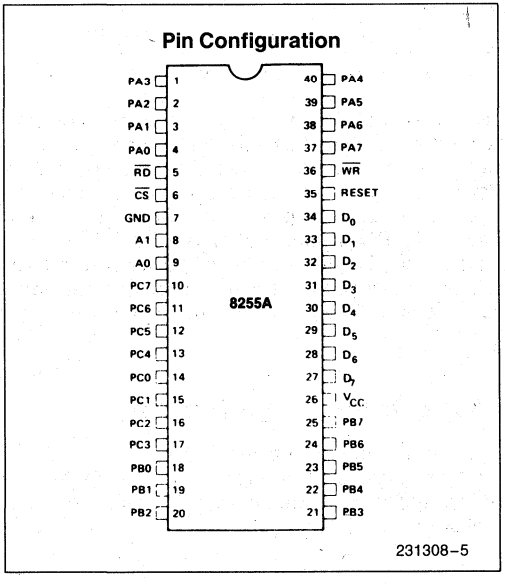


Figure 4. 8225A Block Diagram Showing Group A and Group B Control Functions



Pin Names

D_7-D_0	Data Bus (Bi-Directional)
RESET	Reset Input
$\overline{CS}$	Chip Select
$\overline{RD}$	Read Input
$\overline{WR}$	Write Input
A0, A1	Port Address
PA7–PA0	Port A (BIT)
PB7–PB0	Port B (BIT)
PC7–PC0	Port C (BIT)
V_{CC}	+5 Volts
GND	0 Volts

8255A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0—Basic Input/Output

Mode 1—Strobed Input/Output

Mode 2—Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

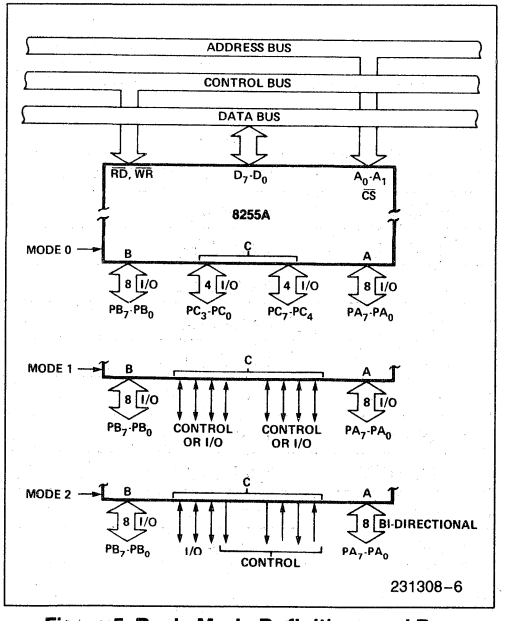


Figure 5. Basic Mode Definitions and Bus Interface

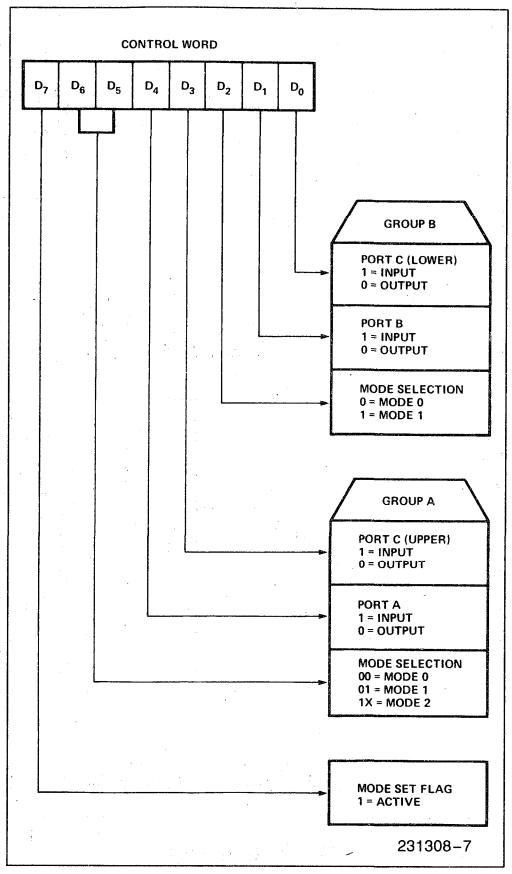


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

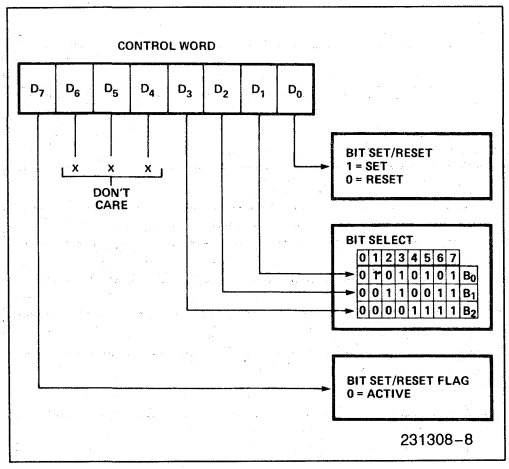


Figure 7. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET)—INTE is set—Interrupt enable

(BIT-RESET)—INTE is RESET—Interrupt disable

NOTE:
All Mask flip-flops are automatically reset during mode selection and device Reset.

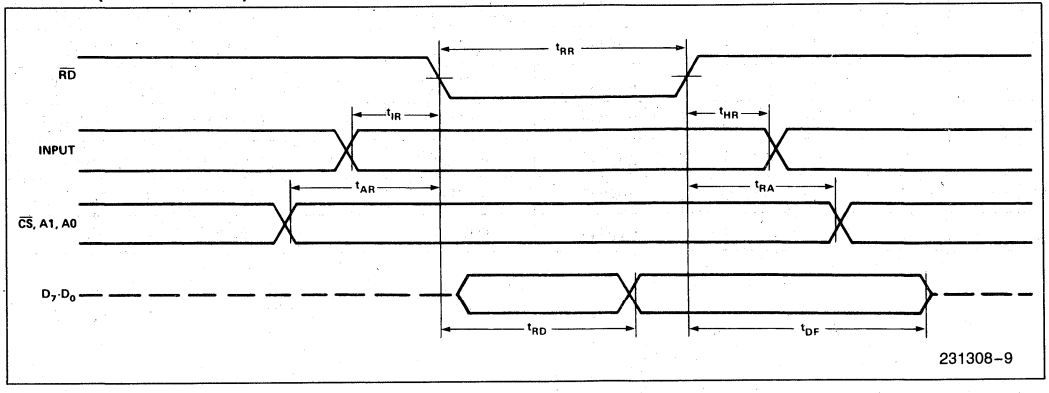
Operating Modes

MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

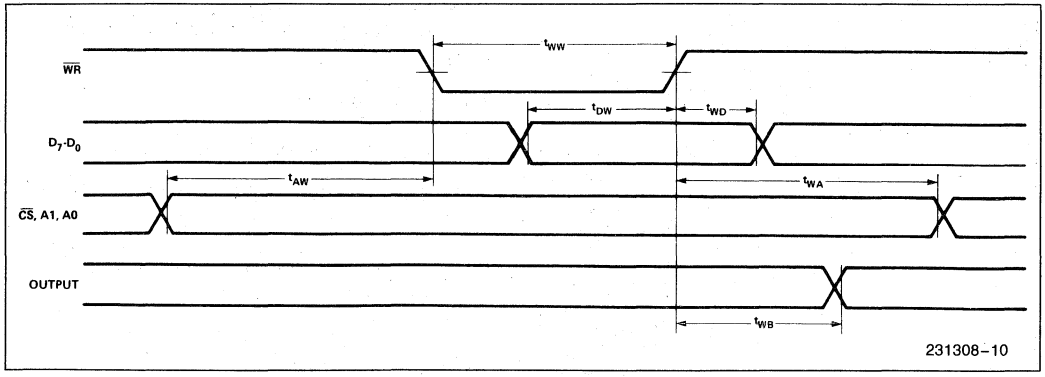
Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.

MODE 0 (BASIC INPUT)



MODE 0 (BASIC OUTPUT)



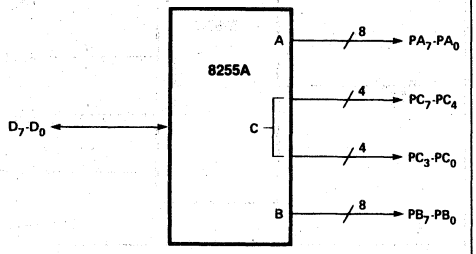
MODE 0 PORT DEFINITION

A		B		Group A			Group B	
D_4	D_3	D_1	D_0	Port A	Port C (Upper)	#	Port B	Port C (Lower)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE CONFIGURATIONS

CONTROL WORD #0

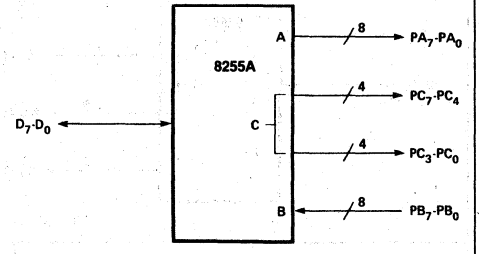
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	0



231308-11

CONTROL WORD #2

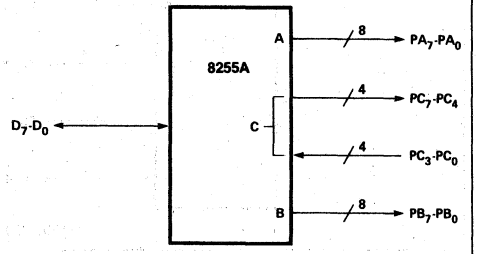
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	0



231308-12

CONTROL WORD #1

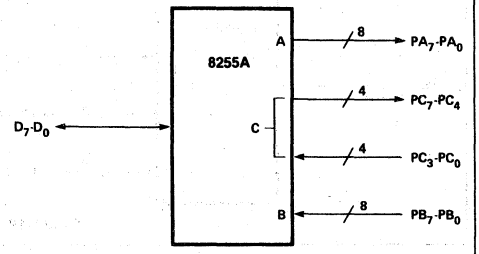
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	1



231308-13

CONTROL WORD #3

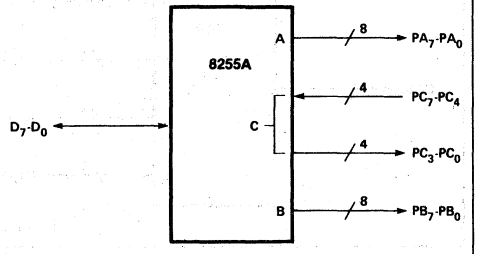
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	1



231308-14

CONTROL WORD #4

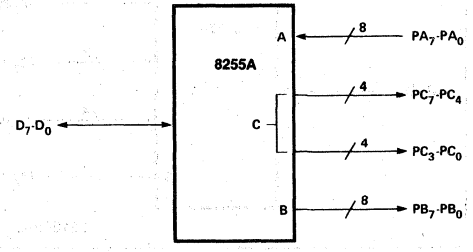
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



231308-15

CONTROL WORD #8

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0

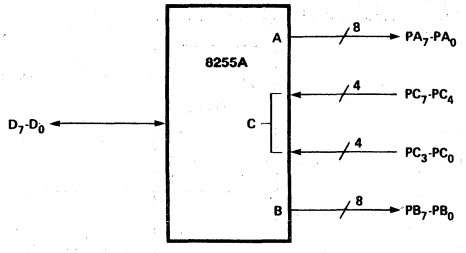


231308-16

3

CONTROL WORD #5

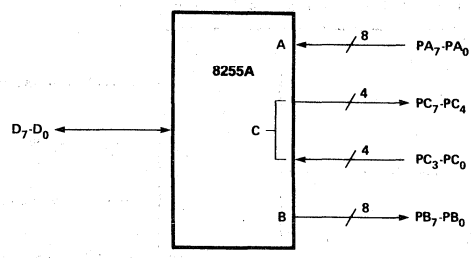
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



231308-17

CONTROL WORD #9

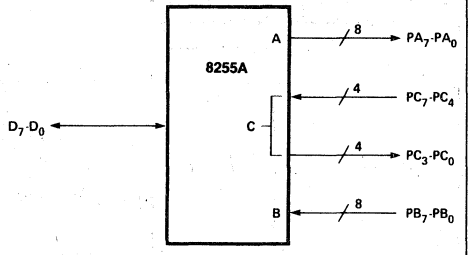
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



231308-18

CONTROL WORD #6

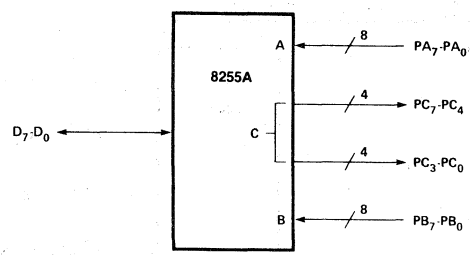
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



231308-19

CONTROL WORD #10

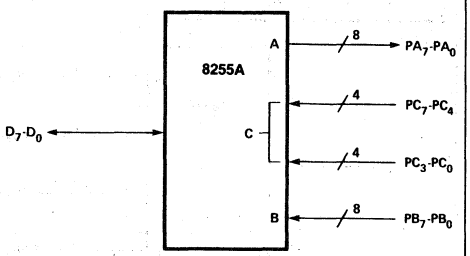
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0



231308-20

CONTROL WORD #7

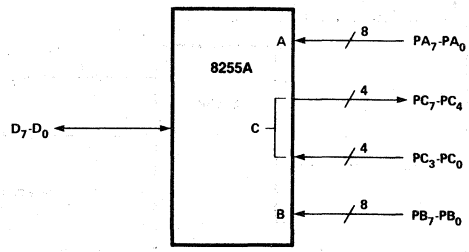
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



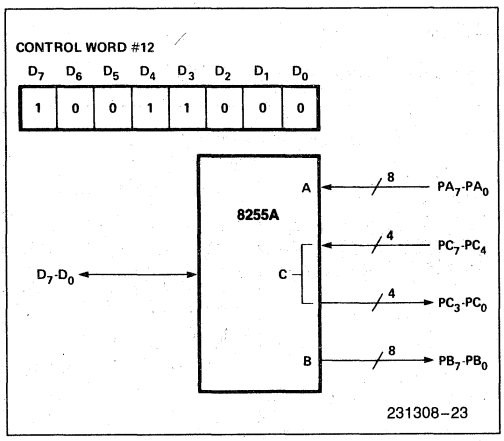
231308-21

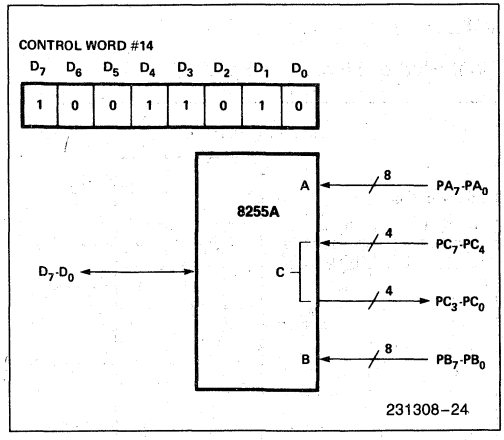
CONTROL WORD #11

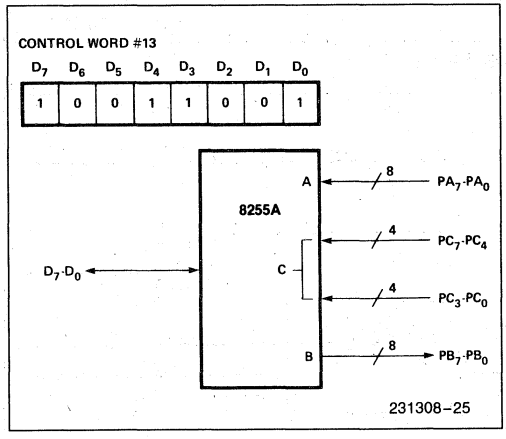
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1

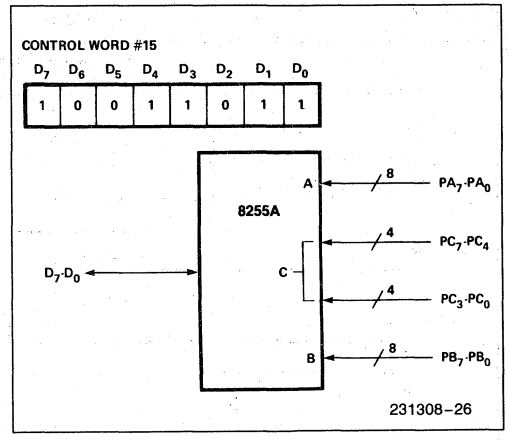


231308-22









Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

$\overline{\text{STB}}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the $\overline{\text{STB}}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{\text{RD}}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

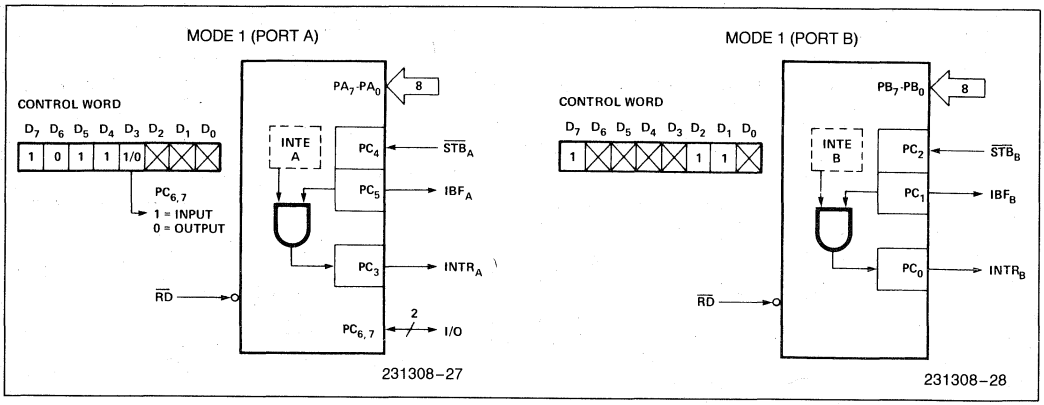


Figure 8. MODE 1 Input

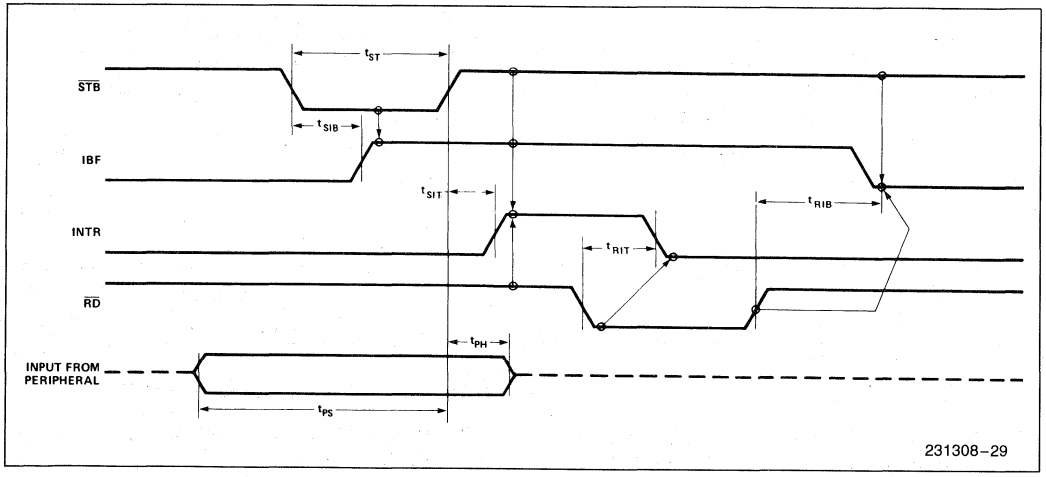


Figure 9. MODE 1 (Strobed Input)

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to the specified port. The $\overline{OBF}$ F/F will be set by the rising edge of the $\overline{WR}$ input and reset by $\overline{ACK}$ input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", $\overline{OBF}$ is a "one", and INTE is a "one". It is reset by the falling edge of $\overline{WR}$.

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

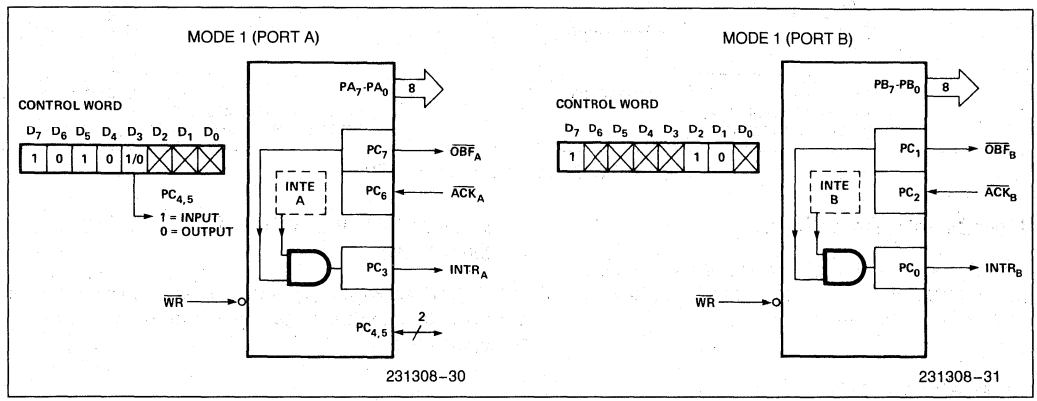


Figure 10. MODE 1 Output

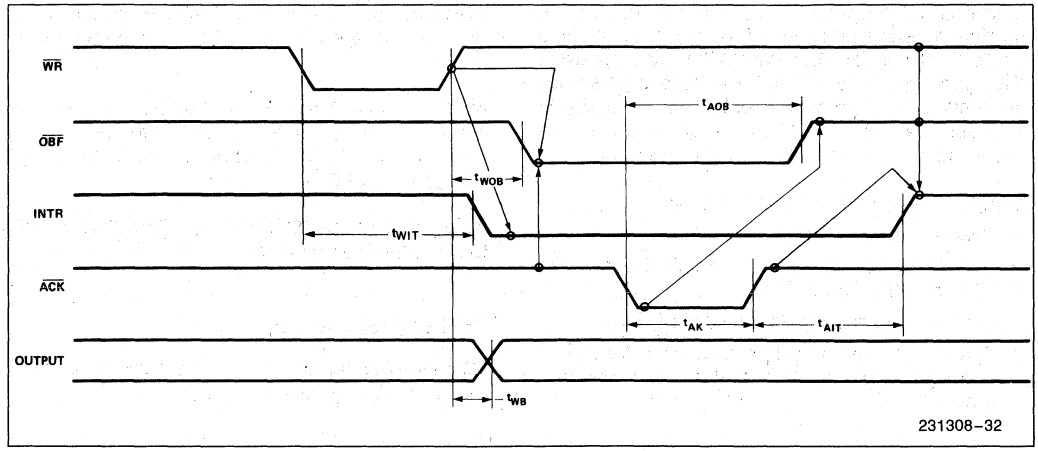


Figure 11. MODE 1 (Strobed Output)

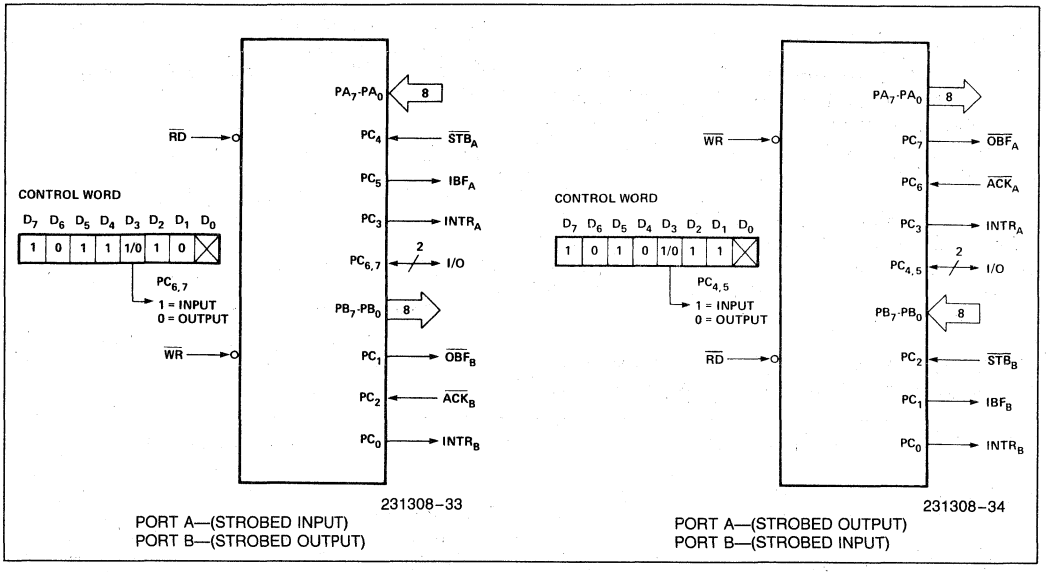


Figure 12. Combinations of MODE 1

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in MODE 1 to support a wide variety of strobed I/O applications.

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A **only**.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Full). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to port A.

$\overline{ACK}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with OBF). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

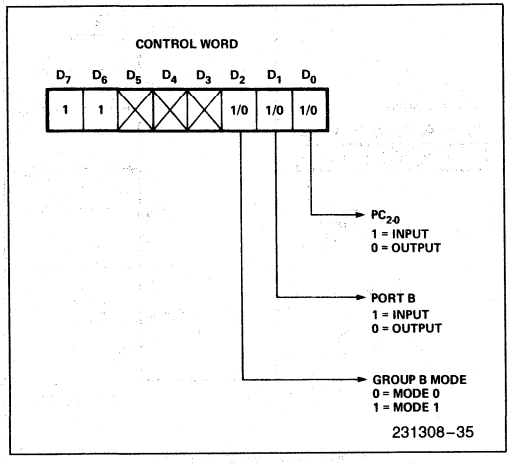


Figure 13. MODE Control Word

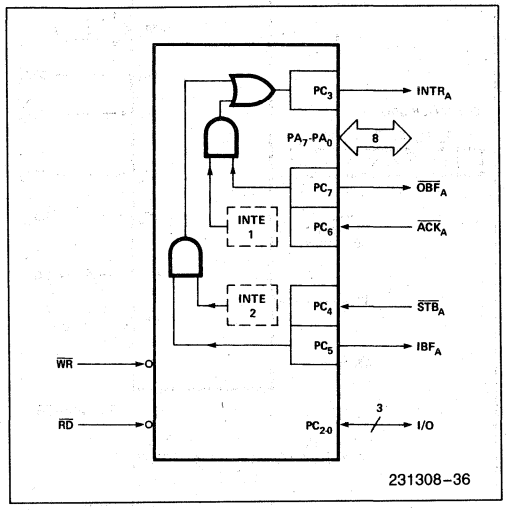
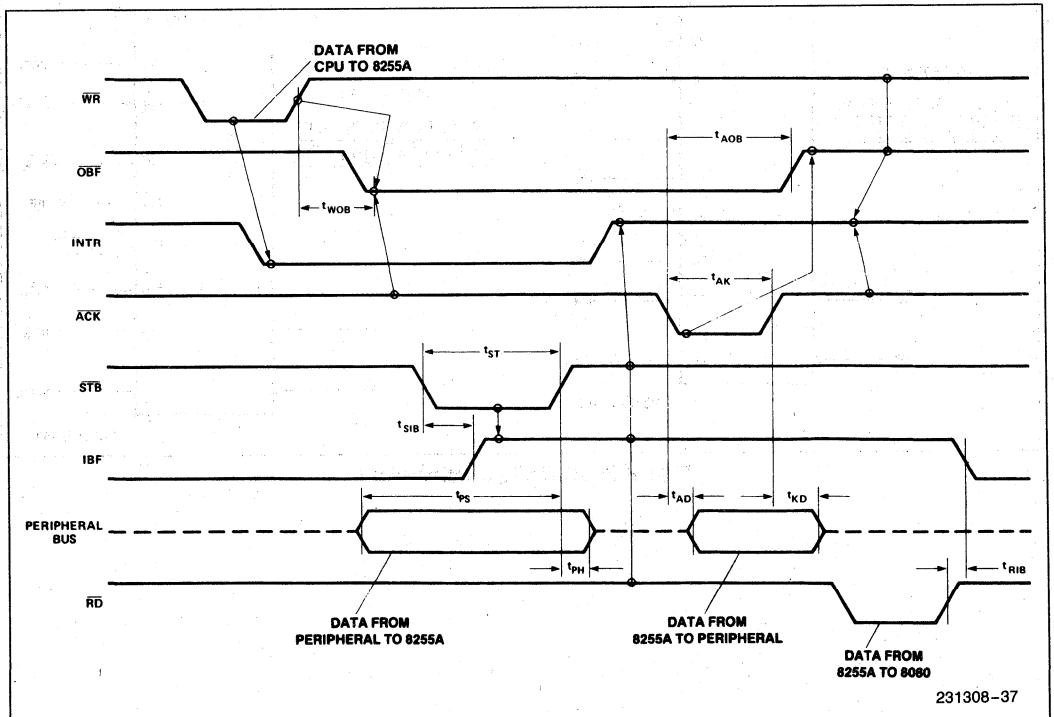


Figure 14. MODE 2



NOTE:
Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

Figure 15. MODE 2 (Bidirectional)

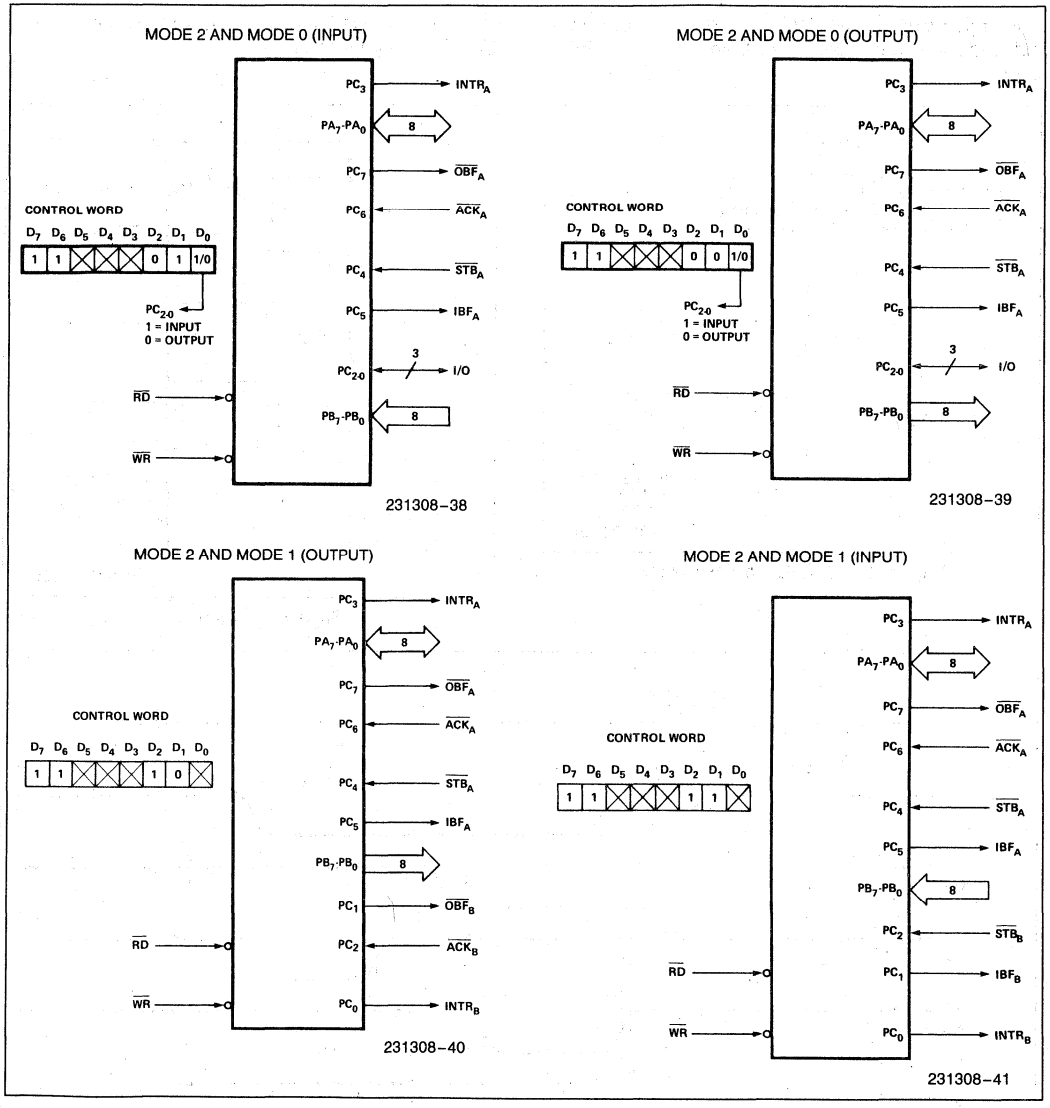


Figure 16. MODE ¼ Combinations

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	↔	
PA_1	IN	OUT	IN	OUT	↔	
PA_2	IN	OUT	IN	OUT	↔	
PA_3	IN	OUT	IN	OUT	↔	
PA_4	IN	OUT	IN	OUT	↔	
PA_5	IN	OUT	IN	OUT	↔	
PA_6	IN	OUT	IN	OUT	↔	
PA_7	IN	OUT	IN	OUT	↔	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs—

All input lines can be accessed during a normal Port C read.

If Programmed as Outputs—

Bits in C upper (PC_7–PC_4) must be individually accessed using the bit set/reset function.

Bits in C lower (PC_3–PC_0) can be accessed using the bit set/reset function or accessed as a threesome by writing into Port C.

Source Current Capability on Port B and Port C

Any set of **eight** output buffers, selected randomly from Ports B and C can source 1 mA at 1.5 volts. This feature allows the 8255 to directly drive Darlington type drivers and high-voltage displays that require such source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 8255 is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

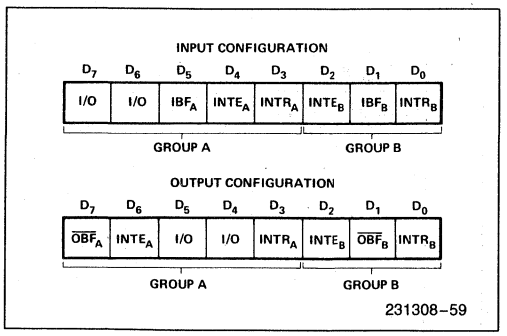


Figure 17. MODE 1 Status Word Format

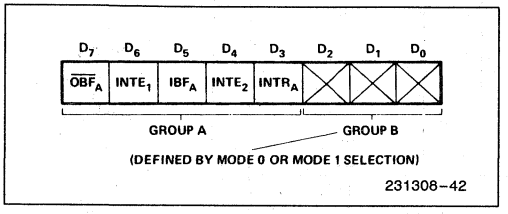


Figure 18. MODE 2 Status Word Format

APPLICATIONS OF THE 8255A

The 8255A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 8255A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the 8255A to exactly "fit" the application. Figures 19 through 25 represent a few examples of typical applications of the 8255A.

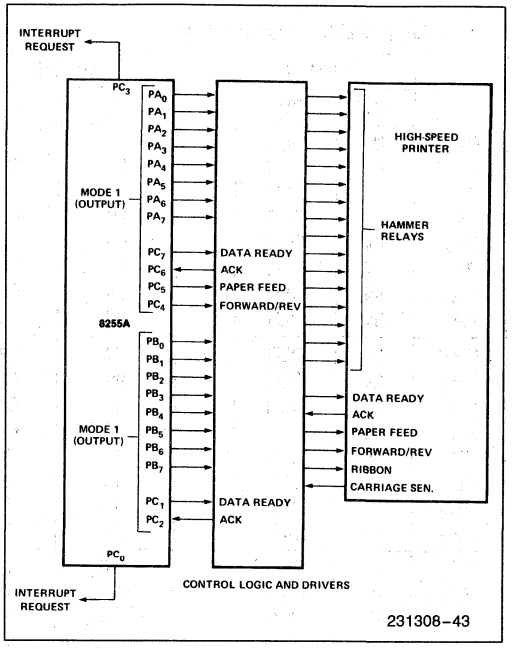


Figure 19. Printer Interface

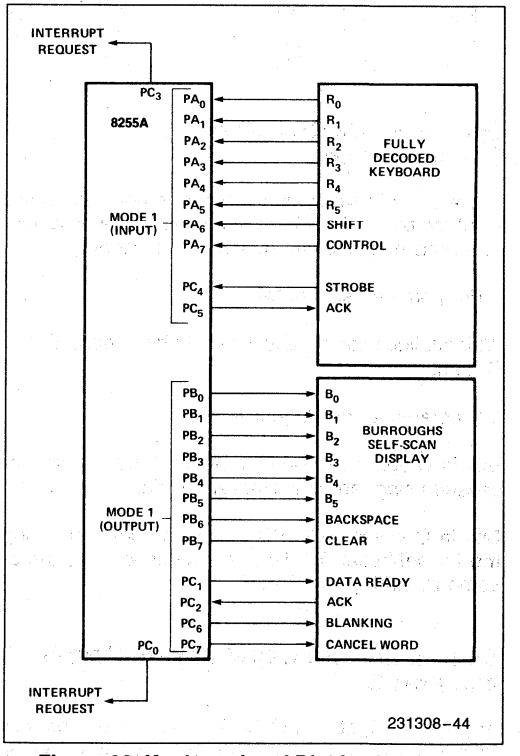


Figure 20. Keyboard and Display Interface

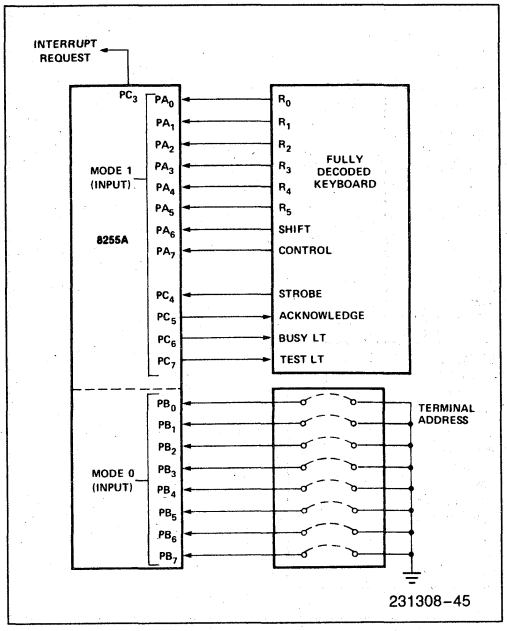


Figure 21. Keyboard and Terminal Address Interface

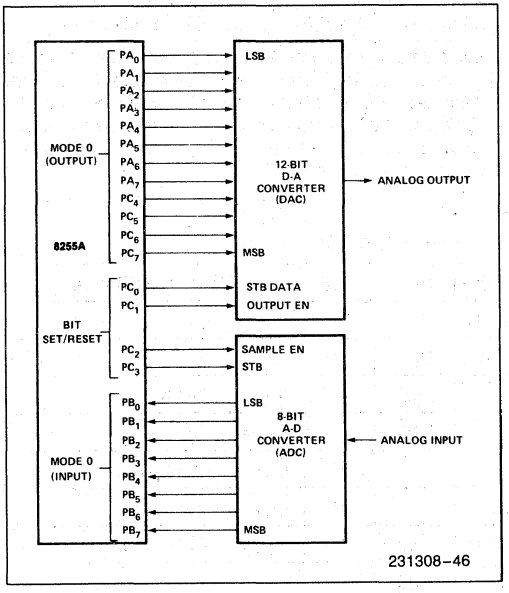


Figure 22. Digital to Analog, Analog to Digital

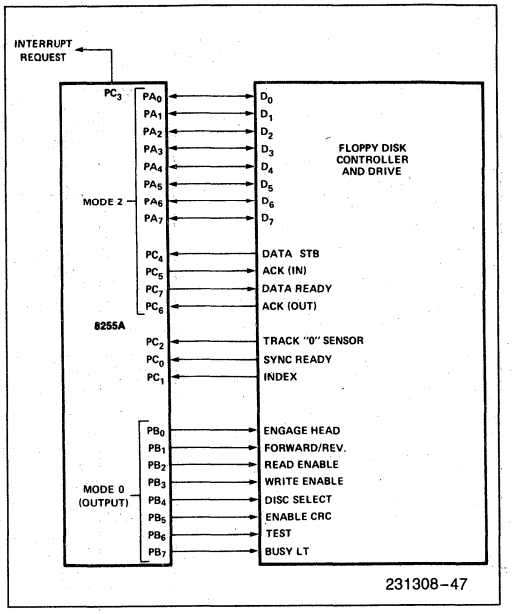


Figure 23. Basic Floppy Disk Interface

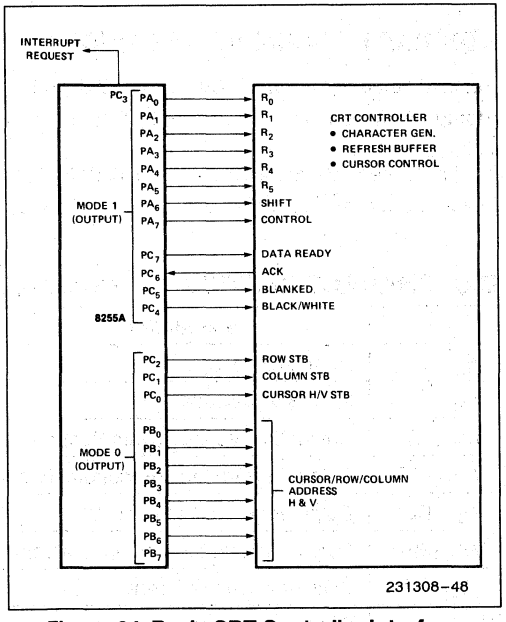


Figure 24. Basic CRT Controller Interface

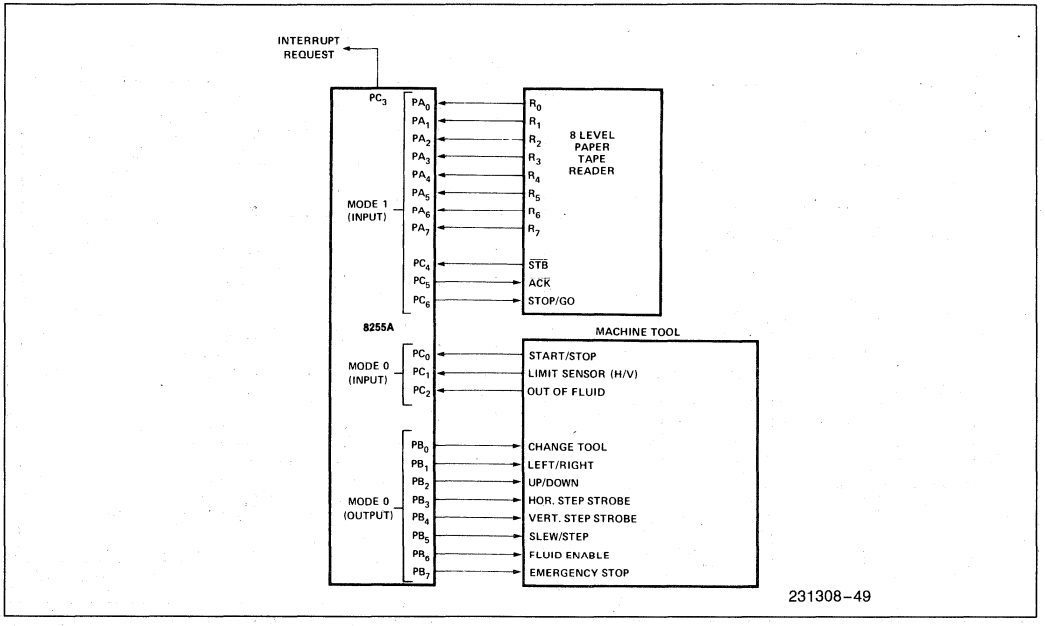


Figure 25. Machine Tool Controller Interface

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin with Respect to Ground.....−0.5V to +7V
Power Dissipation .....1 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V*

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL} (DB)	Output Low Voltage (Data Bus)		0.45*	V	I_{OL} = 2.5 mA
V_{OL} (PER)	Output Low Voltage (Peripheral Port)		0.45*	V	I_{OL} = 1.7 mA
V_{OH} (DB)	Output High Voltage (Data Bus)	2.4		V	I_{OH} = −400 μA
V_{OH} (PER)	Output High Voltage (Peripheral Port)	2.4		V	I_{OH} = −200 μA
I_{DAR}(1)	Darlington Drive Current	−1.0	−4.0	mA	R_{EXT} = 750Ω; V_{EXT} = 1.5V
I_{CC}	Power Supply Current		120	mA	
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0.45V

NOTE:
1. Available on any 8 pins from Port B and C.

CAPACITANCE T_A = 25°C, V_{CC} = GND = 0V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	f_c = 1 MHz(4)
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to GND(4)

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V*

Bus Parameters

READ

Symbol	Parameter	8255A		8255A-5		Unit
		Min	Max	Min	Max	
t_{AR}	Address Stable before READ	0		0		ns
t_{RA}	Address Stable after READ	0		0		ns
t_{RR}	READ Pulse Width	300		300		ns
t_{RD}	Data Valid from READ(1)		250		200	ns
t_{DF}	Data Float after READ	10	150	10	100	ns
t_{RV}	Time between READs and/or WRITEs	850		850		ns

WRITE

Symbol	Parameter	8255A		8255A-5		Unit
		Min	Max	Min	Max	
t_{AW}	Address Stable before WRITE	0		0		ns
t_{WA}	Address Stable after WRITE	20		20		ns
t_{WW}	WRITE Pulse Width	400		300		ns
t_{DW}	Data Valid to WRITE (T.E.)	100		100		ns
t_{WD}	Data Valid after WRITE	30		30		ns

OTHER TIMINGS

Symbol	Parameter	8255A		8255A-5		Unit
		Min	Max	Min	Max	
t_{WB}	WR = 1 to Output(1)		350		350	ns
t_{IR}	Peripheral Data before RD	0		0		ns
t_{HR}	Peripheral Data after RD	0		0		ns
t_{AK}	ACK Pulse Width	300		300		ns
t_{ST}	STB Pulse Width	500		500		ns
t_{PS}	Per. Data before T.E. of STB	0		0		ns
t_{PH}	Per. Data after T.E. of STB	180		180		ns
t_{AD}	ACK = 0 to Output(1)		300		300	ns
t_{KD}	ACK = 1 to Output Float	20	250	20	250	ns

3

A.C. CHARACTERISTICS (Continued)

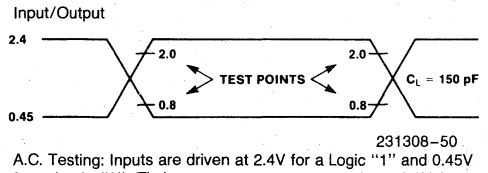
OTHER TIMINGS (Continued)

Symbol	Parameter	8255A		8255A-5		Unit
		Min	Max	Min	Max	
t_{WOB}	WR = 1 to OBF = 0(1)		650		650	ns
t_{AOB}	ACK = 0 to OBF = 1(1)		350		350	ns
t_{SIB}	STB = 0 to IBF = 1(1)		300		300	ns
t_{RIB}	RD = 1 to IBF = 0(1)		300		300	ns
t_{RIT}	RD = 0 to INTR = 0(1)		400		400	ns
t_{SIT}	STB = 1 to INTR = 1(1)		300		300	ns
t_{AIT}	ACK = 1 to INTR = 1(1)		350		350	ns
t_{WIT}	WR = 0 to INTR = 0(1, 3)		850		850	ns

NOTES:
1. Test Conditions: C_L = 150 pF.
2. Period of Reset pulse must be at least 50 μs during or after power on. Subsequent Reset pulse can be 500 ns min.
3. INTR ↑ may occur as early as $\overline{WR}$ ↓.
4. Sampled, not 100% tested.

*For Extended Temperature EXPRESS, use M8255A electrical parameters.

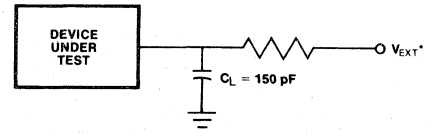
A.C. TESTING INPUT, OUTPUT WAVEFORM



231308-50

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT

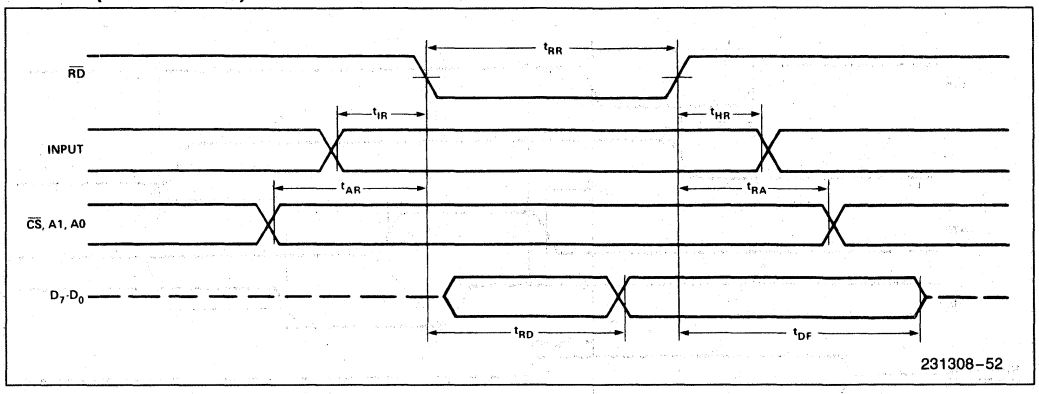


231308-51

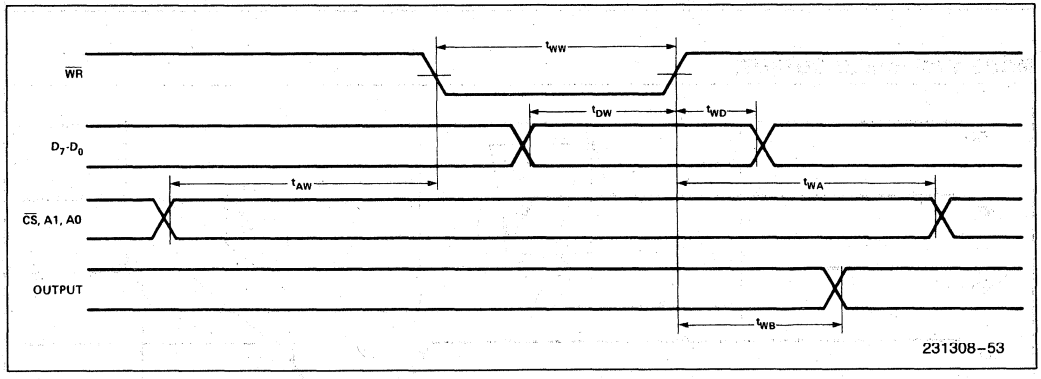
*V_{EXT} is set at various voltages during testing to guarantee the specification. C_L includes jig capacitance.

WAVEFORMS

MODE 0 (BASIC INPUT)



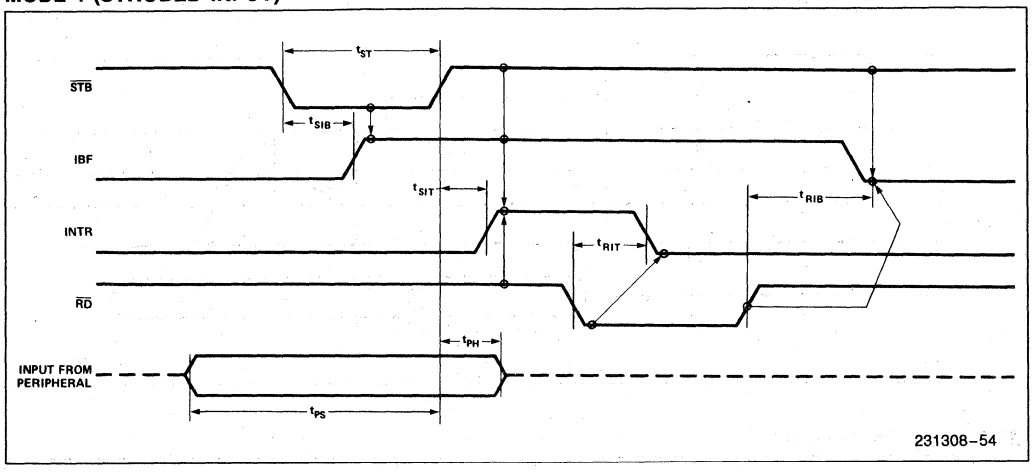
MODE 0 (BASIC OUTPUT)



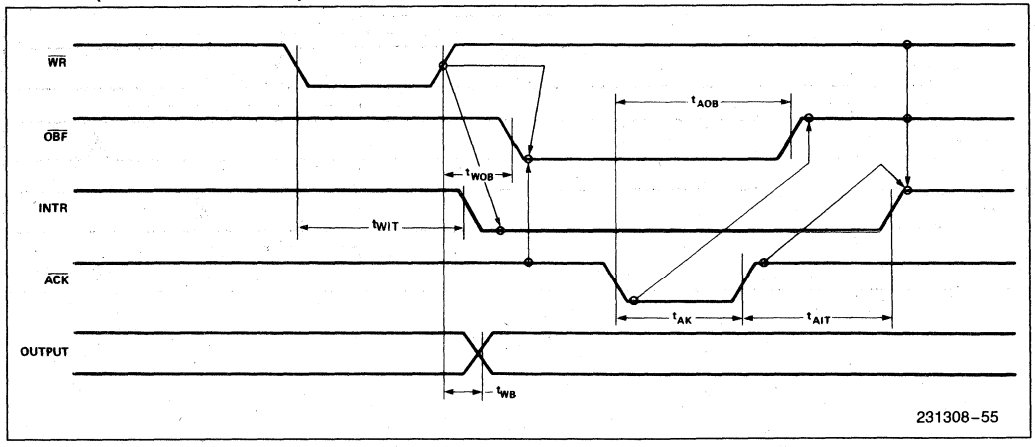
3

WAVEFORMS (Continued)

MODE 1 (STROBED INPUT)

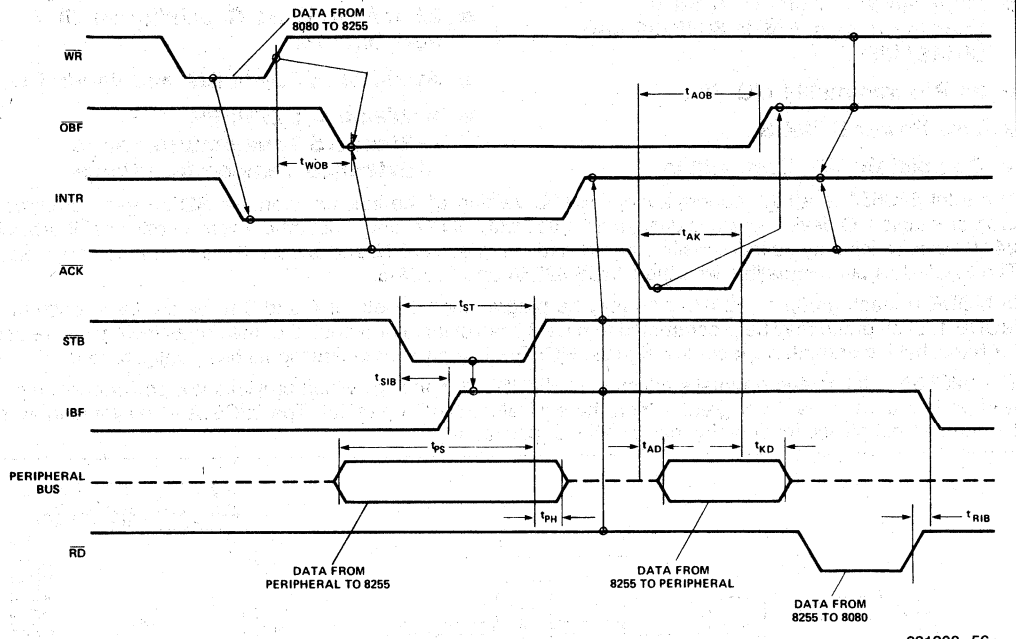


MODE 1 (STROBED OUTPUT)



WAVEFORMS (Continued)

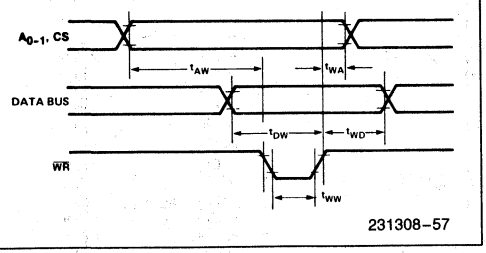
MODE 2 (BIDIRECTIONAL)



231308-56

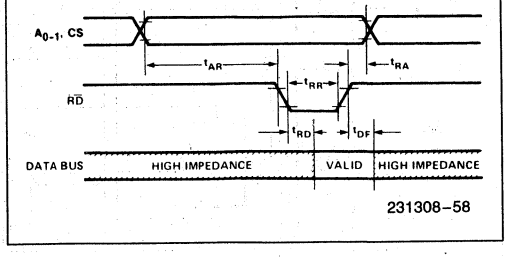
NOTE:
Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
($INTR = IBF \bullet \overline{MASK} \bullet \overline{STB} \bullet \overline{RD} + \overline{OBF} \bullet \overline{MASK} \bullet \overline{ACK} \bullet \overline{WR}$)

WRITE TIMING



231308-57

READ TIMING



231308-58

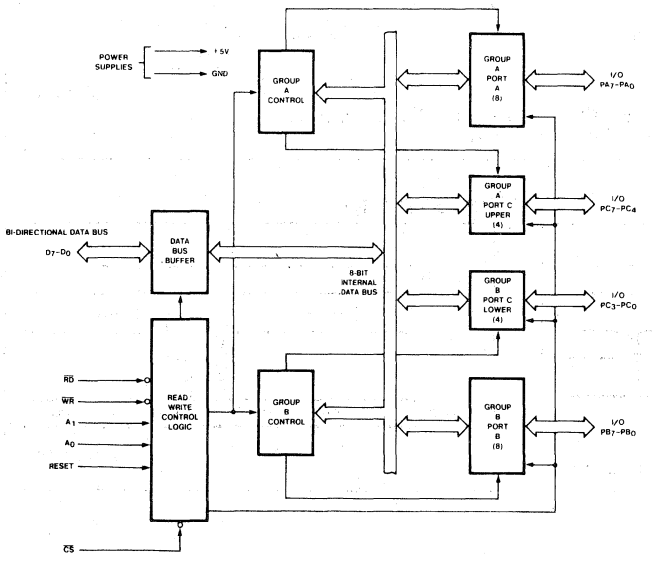
82C55A
CHMOS PROGRAMMABLE PERIPHERAL INTERFACE

- **Compatible with all Intel and Most Other Microprocessors**
- **High Speed, "Zero Wait State" Operation with 8 MHz 8086/88 and 80186/188**
- **24 Programmable I/O Pins**
- **Low Power CHMOS**
- **Completely TTL Compatible**
- **Control Word Read-Back Capability**
- **Direct Bit Set/Reset Capability**
- **2.5 mA DC Drive Capability on all I/O Port Outputs**
- **Available in 40-Pin DIP and 44-Pin PLCC**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**

The Intel 82C55A is a high-performance, CHMOS version of the industry standard 8255A general purpose programmable I/O device which is designed for use with all Intel and most other microprocessors. It provides 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. The 82C55A is pin compatible with the NMOS 8255A and 8255A-5.

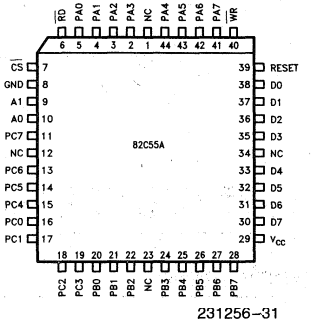
In MODE 0, each group of 12 I/O pins may be programmed in sets of 4 and 8 to be inputs or outputs. In MODE 1, each group may be programmed to have 8 lines of input or output. 3 of the remaining 4 pins are used for handshaking and interrupt control signals. MODE 2 is a strobed bi-directional bus configuration.

The 82C55A is fabricated on Intel's advanced CHMOS III technology which provides low power consumption with performance equal to or greater than the equivalent NMOS product. The 82C55A is available in 40-pin DIP and 44-pin plastic leaded chip carrier (PLCC) packages.

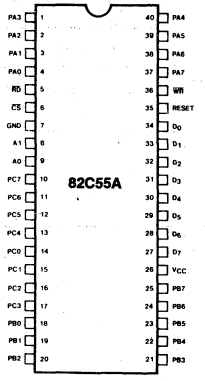


231256-1

Figure 1. 82C55A Block Diagram



231256-31



231256-2

Figure 2. 82C55A Pinout

Diagrams are for pin reference only. Package sizes are not to scale.

Table 1. Pin Description

Symbol	Pin Number Dip	Pin Number PLCC	Type	Name and Function
PA_{3-0}	1–4	2–5	I/O	**PORT A, PINS 0–3:** Lower nibble of an 8-bit data output latch/buffer and an 8-bit data input latch.
$\overline{RD}$	5	6	I	**READ CONTROL:** This input is low during CPU read operations.
$\overline{CS}$	6	7	I	**CHIP SELECT:** A low on this input enables the 82C55A to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and WR are ignored otherwise.
GND	7	8		**System Ground**
A_{1-0}	8–9	9–10	I	**ADDRESS:** These input signals, in conjunction $\overline{RD}$ and $\overline{WR}$, control the selection of one of the three ports or the control word registers.

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Input Operation (Read)
0	0	0	1	0	Port A - Data Bus
0	1	0	1	0	Port B - Data Bus
1	0	0	1	0	Port C - Data Bus
1	1	0	1	0	Control Word - Data Bus
					Output Operation (Write)
0	0	1	0	0	Data Bus - Port A
0	1	1	0	0	Data Bus - Port B
1	0	1	0	0	Data Bus - Port C
1	1	1	0	0	Data Bus - Control
					Disable Function
X	X	X	X	1	Data Bus - 3 - State
X	X	1	1	0	Data Bus - 3 - State

PC_{7-4}	10–13	11,13–15	I/O	**PORT C, PINS 4–7:** Upper nibble of an 8-bit data output latch/buffer and an 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.
PC_{0-3}	14–17	16–19	I/O	**PORT C, PINS 0–3:** Lower nibble of Port C.
PB_{0-7}	18–25	20–22, 24–28	I/O	**PORT B, PINS 0–7:** An 8-bit data output latch/buffer and an 8-bit data input buffer.
V_{CC}	26	29		**SYSTEM POWER:** + 5V Power Supply.
D_{7-0}	27–34	30–33, 35–38	I/O	**DATA BUS:** Bi-directional, tri-state data bus lines, connected to system data bus.
RESET	35	39	I	**RESET:** A high on this input clears the control register and all ports are set to the input mode.
$\overline{WR}$	36	40	I	**WRITE CONTROL:** This input is low during CPU write operations.
PA_{7-4}	37–40	41–44	I/O	**PORT A, PINS 4–7:** Upper nibble of an 8-bit data output latch/buffer and an 8-bit data input latch.
NC		1, 12, 23, 34		No Connect

3

82C55A FUNCTIONAL DESCRIPTION

General

The 82C55A is a programmable peripheral interface device designed for use in Intel microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 82C55A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 82C55A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 82C55A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 82C55A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A - Port A and Port C upper (C7–C4)
Control Group B - Port B and Port C lower (C3–C0)

The control word register can be both written and read as shown in the address decode table in the pin descriptions. Figure 6 shows the control word format for both Read and Write operations. When the control word is read, bit D7 will always be a logic "1", as this implies control word mode information.

Ports A, B, and C

The 82C55A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 82C55A.

Port A. One 8-bit data output latch/buffer and one 8-bit input latch buffer. Both "pull-up" and "pull-down" bus hold devices are present on Port A.

Port B. One 8-bit data input/output latch/buffer. Only "pull-up" bus hold devices are present on Port B.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B. Only "pull-up" bus hold devices are present on Port C.

See Figure 4 for the bus-hold circuit configuration for Port A, B, and C.

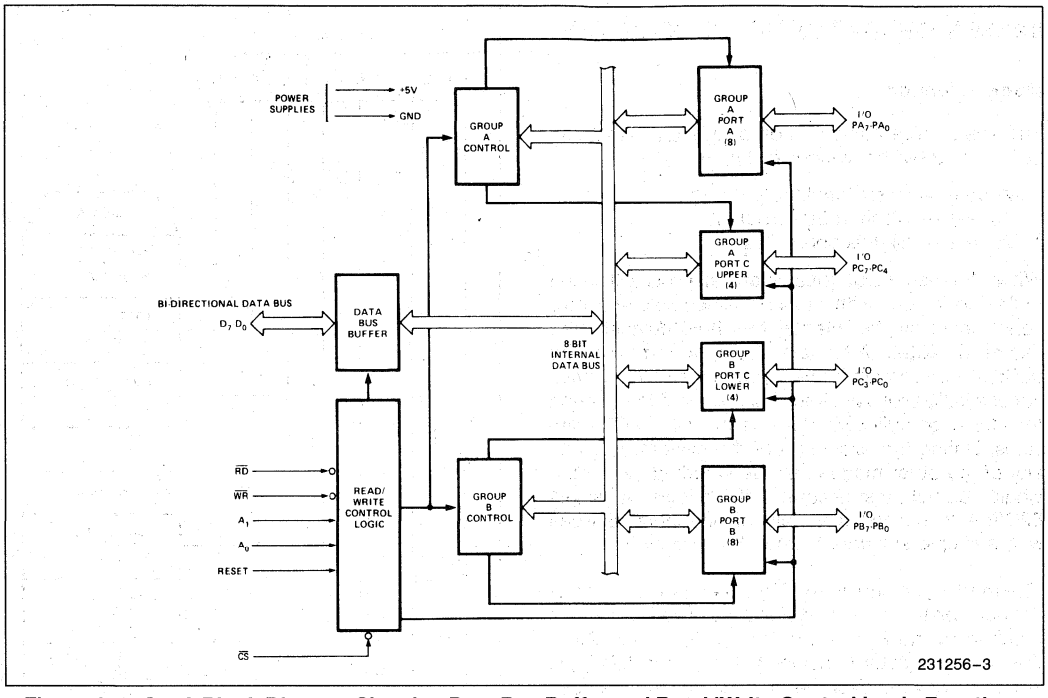
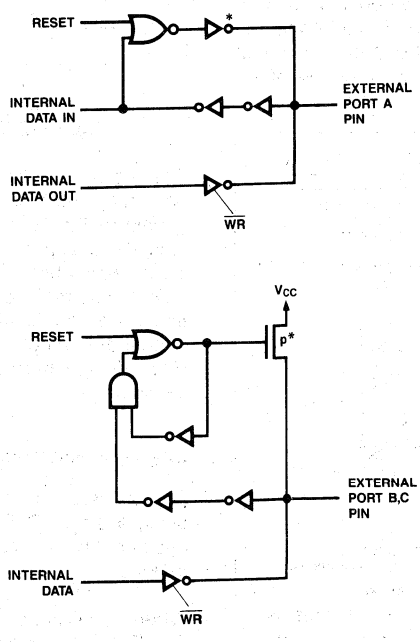


Figure 3. 82C55A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions



***NOTE:**
Port pins loaded with more than 20 pF capacitance may not have their logic level guaranteed following a hardware reset.

Figure 4. Port A, B, C, Bus-hold Configuration

82C55A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0 — Basic input/output
Mode 1 — Strobed Input/output
Mode 2 — Bi-directional Bus

When the reset input goes "high" all ports will be set to the input mode with all 24 port lines held at a logic "one" level by the internal bus hold devices (see Figure 4 Note). After the reset is removed the 82C55A can remain in the input mode with no additional initialization required. This eliminates the need for pullup or pulldown devices in "all CMOS" designs. During the execution of the system program, any of the other modes may be selected by using a single output instruction. This allows a single 82C55A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

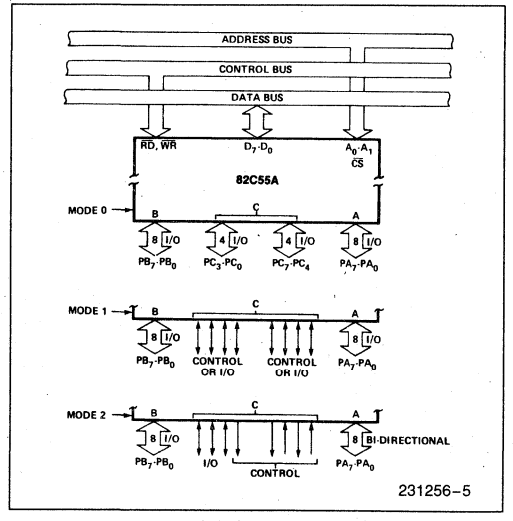


Figure 5. Basic Mode Definitions and Bus Interface

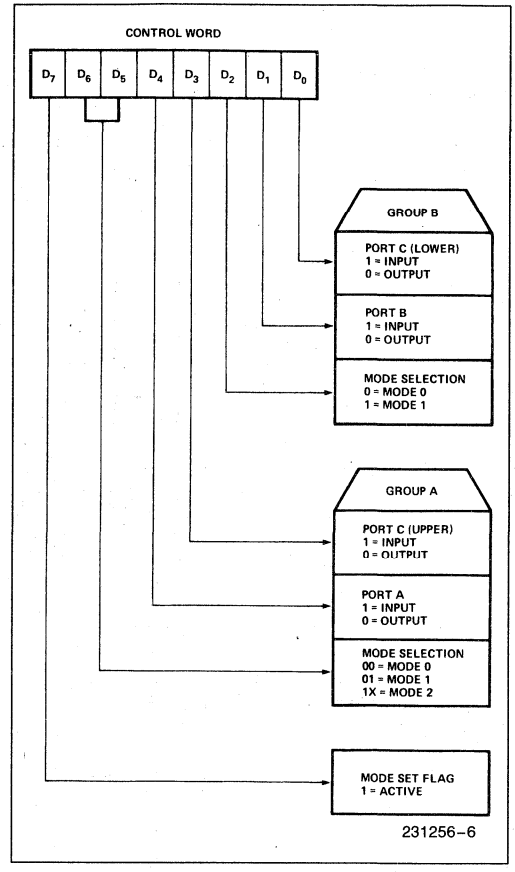


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 82C55A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

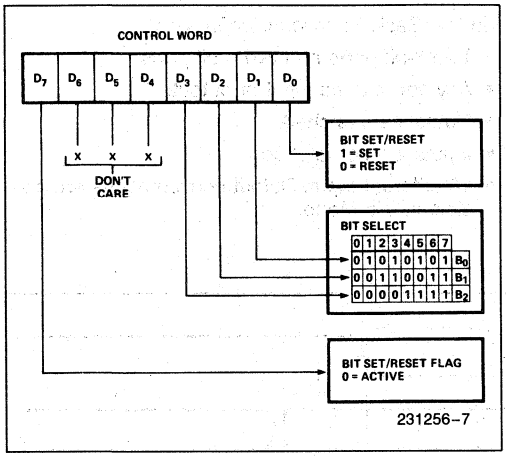


Figure 7. Bit Set/Reset Format

Interrupt Control Functions

When the 82C55A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET)—INTE is SET—Interrupt enable
(BIT-RESET)—INTE is RESET—Interrupt disable

Note:
All Mask flip-flops are automatically reset during mode selection and device Reset.

3

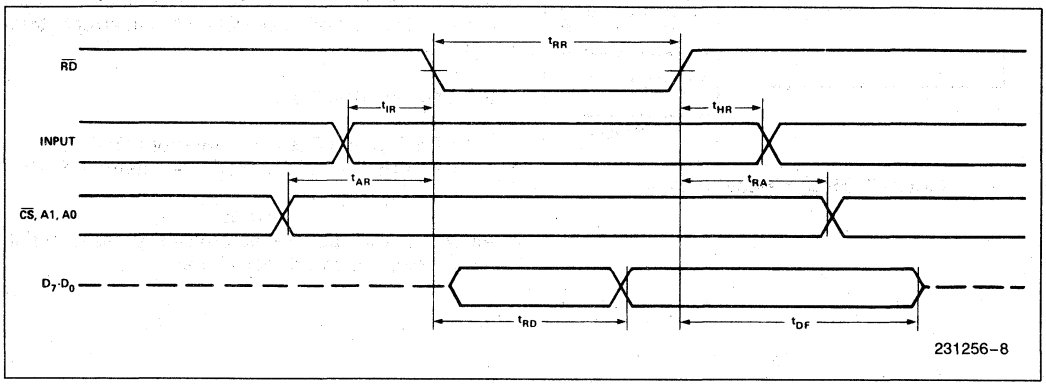
Operating Modes

Mode 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

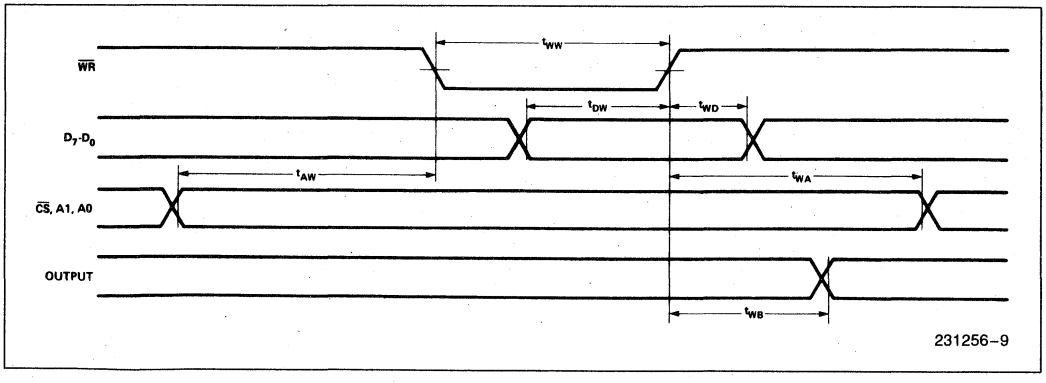
Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.

MODE 0 (BASIC INPUT)



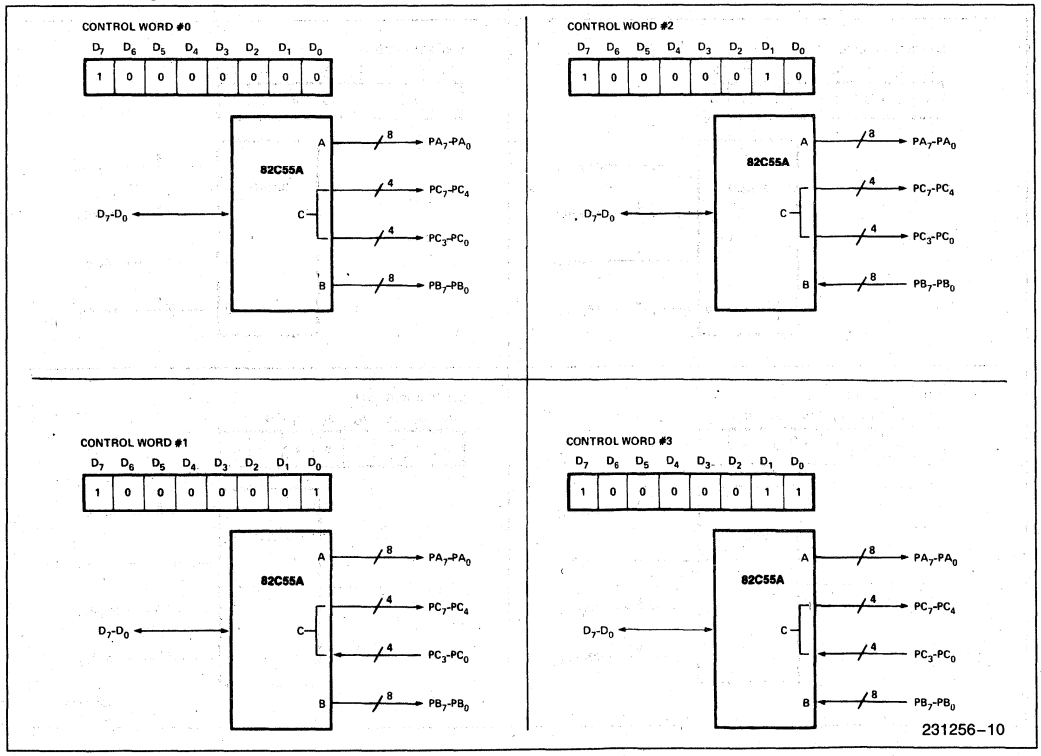
MODE 0 (BASIC OUTPUT)



MODE 0 Port Definition

A		B		GROUP A			GROUP B	
D_4	D_3	D_1	D_0	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

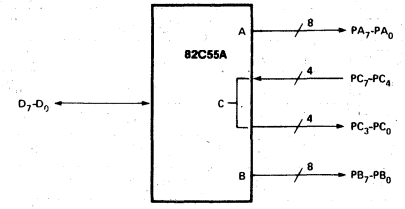
MODE 0 Configurations



MODE 0 Configurations (Continued)

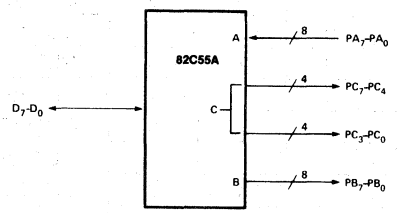
CONTROL WORD #4

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



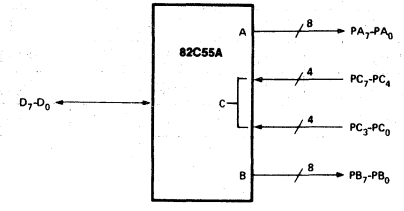
CONTROL WORD #8

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0



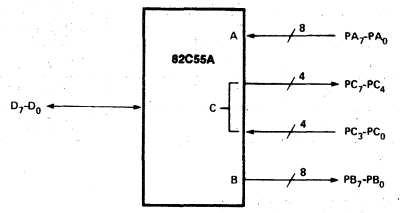
CONTROL WORD #5

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



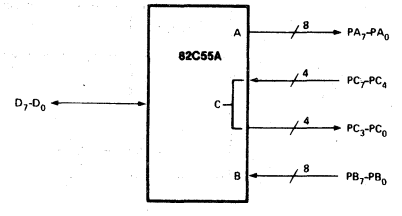
CONTROL WORD #9

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



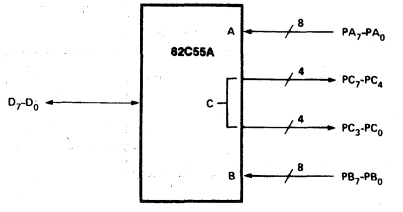
CONTROL WORD #6

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



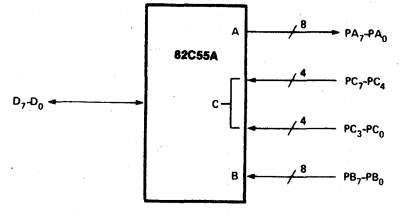
CONTROL WORD #10

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0



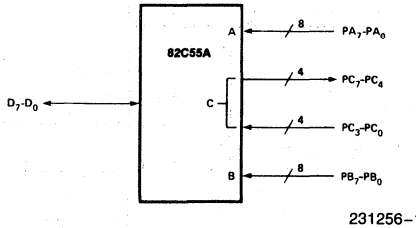
CONTROL WORD #7

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



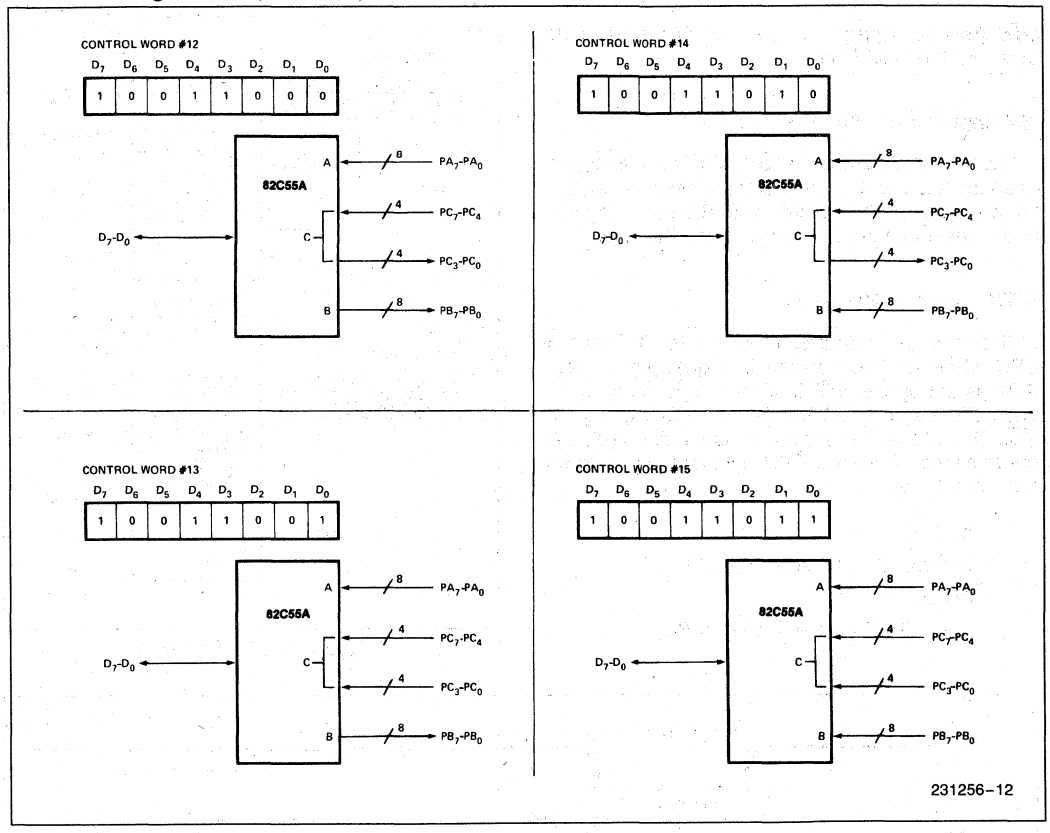
CONTROL WORD #11

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1



231256-11

MODE 0 Configurations (Continued)



Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, Port A and Port B use the lines on Port C to generate or accept these "handshaking" signals.

Mode 1 Basic functional Definitions:

- Two Groups (Group A and Group B).
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by $\overline{STB}$ input being low and is reset by the rising edge of the $\overline{RD}$ input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the $\overline{STB}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

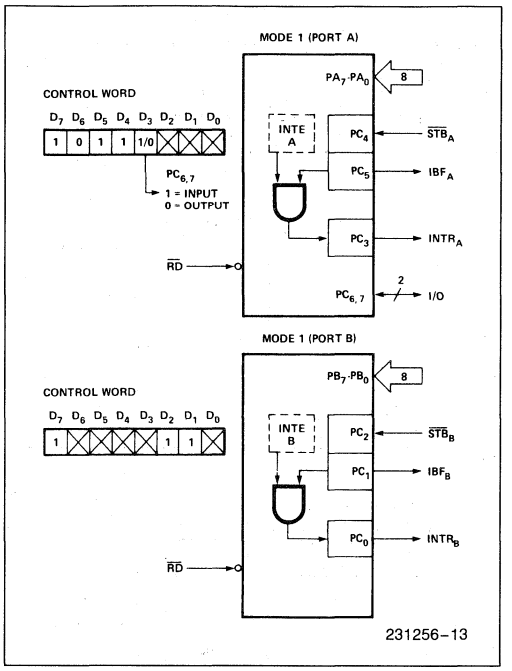


Figure 8. MODE 1 Input

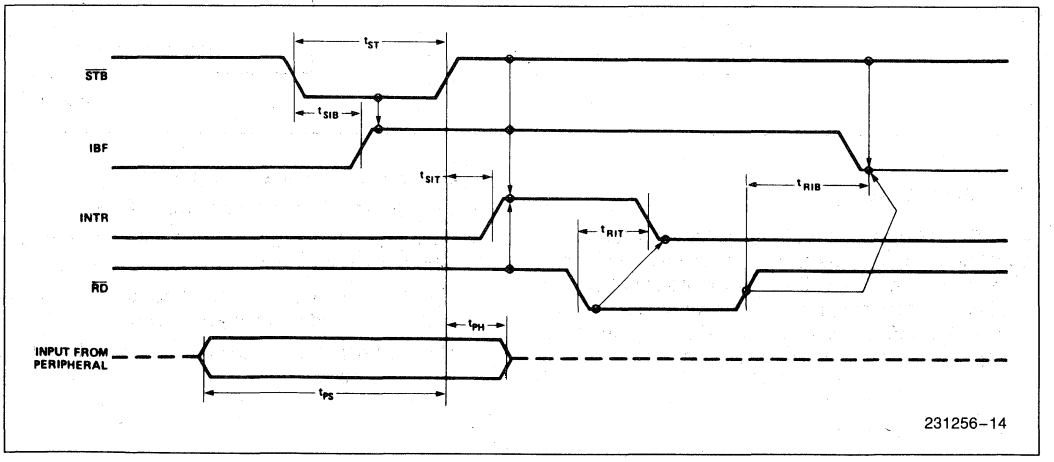


Figure 9. MODE 1 (Strobed Input)

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to the specified port. The $\overline{OBF}$ F/F will be set by the rising edge of the $\overline{WR}$ input and reset by $\overline{ACK}$ Input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the 82C55A that the data from Port A or Port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", $\overline{OBF}$ is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{WR}$.

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

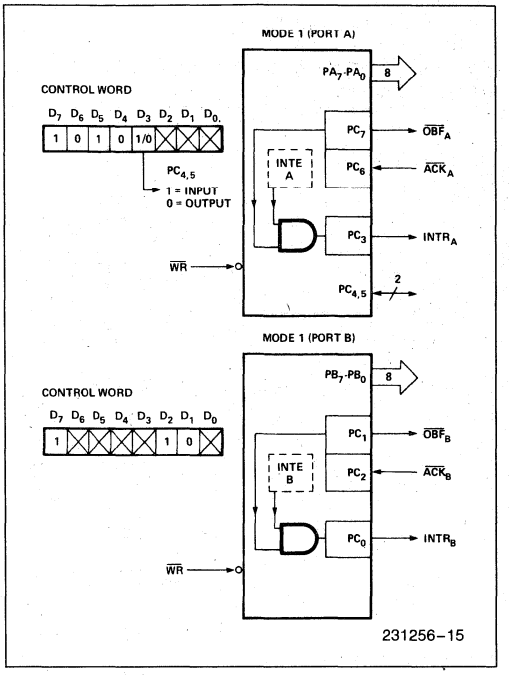


Figure 10. MODE 1 Output

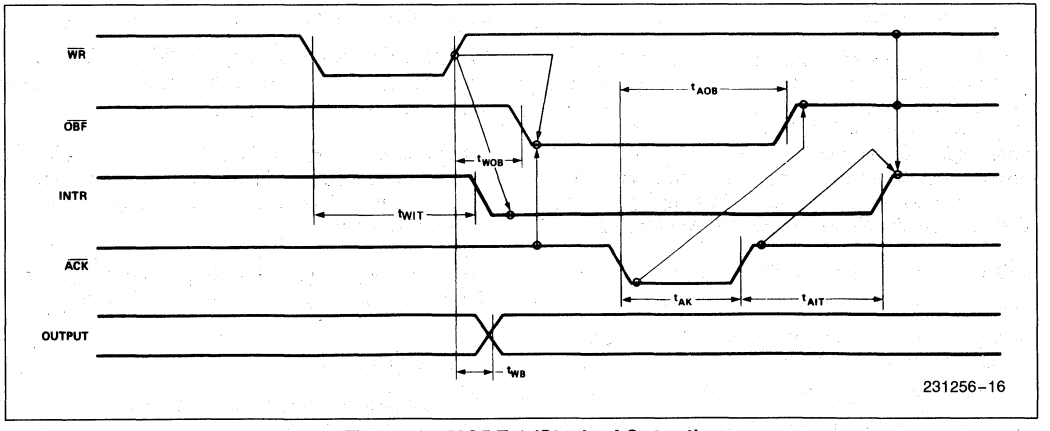


Figure 11. MODE 1 (Strobed Output)

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

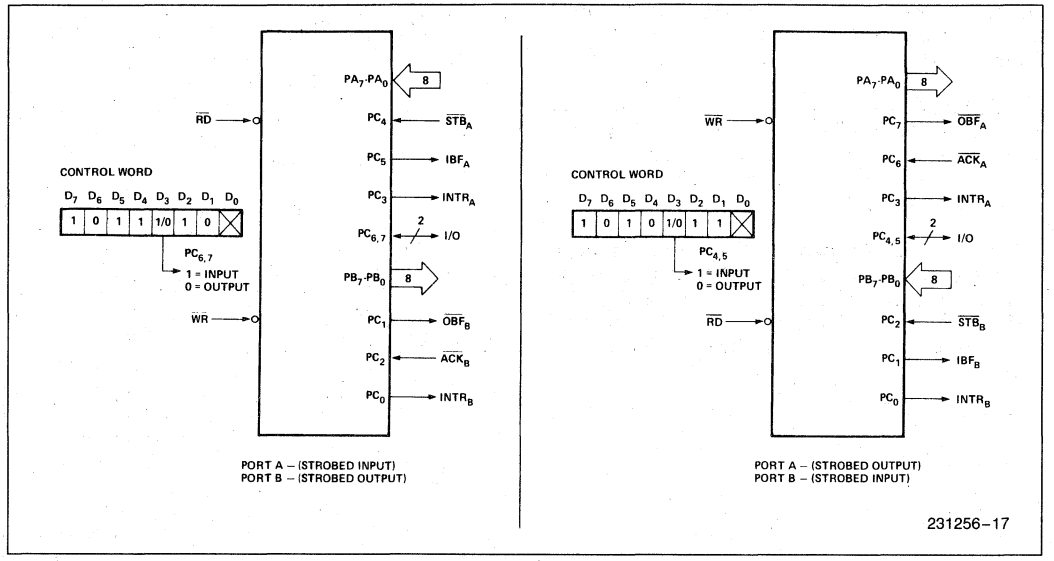


Figure 12. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A **only**.
- One 8-bit, bi-directional bus port (Port A) and a 5-bit control port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Full). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to port A.

$\overline{ACK}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of Port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with $\overline{OBF}$). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

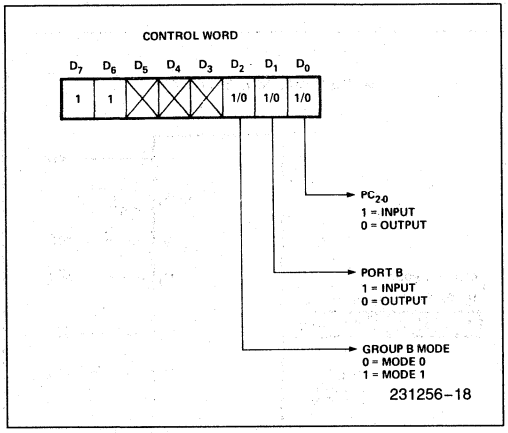


Figure 13. MODE Control Word

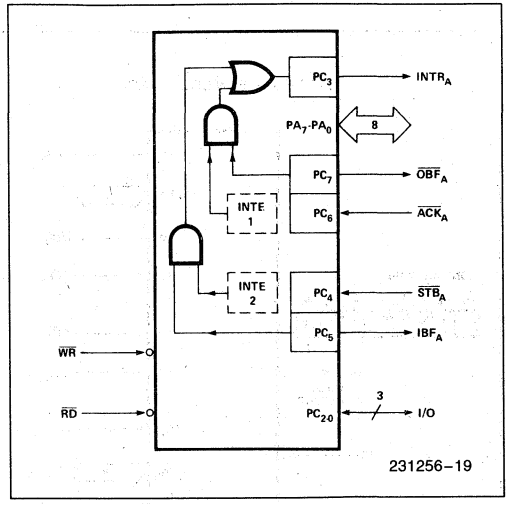


Figure 14. MODE 2

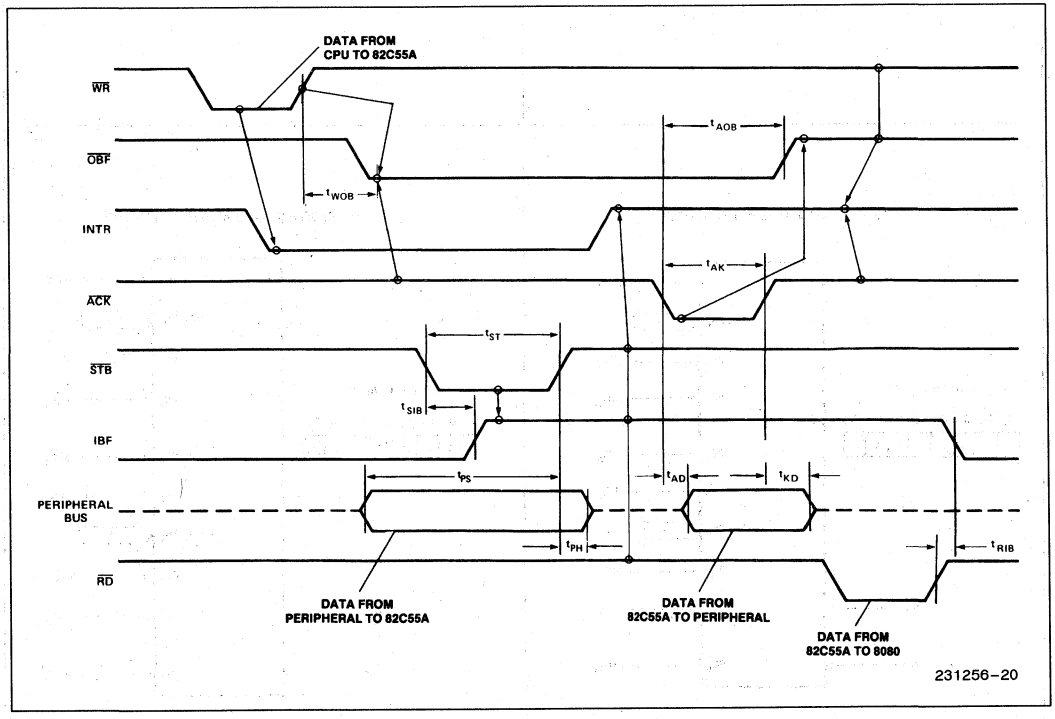


Figure 15. MODE 2 (Bidirectional)

NOTE:
Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$, and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

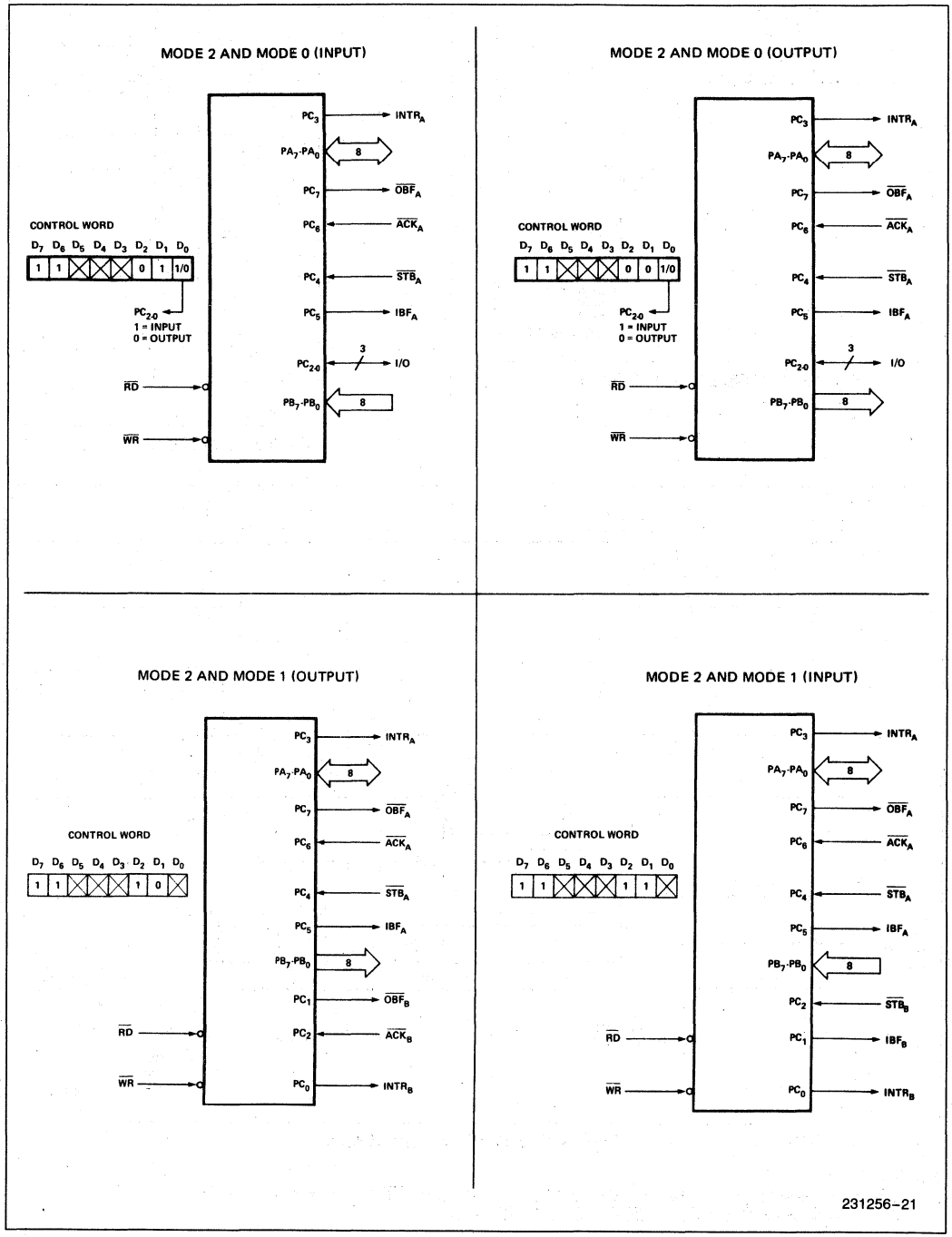


Figure 16. MODE ¼ Combinations

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	⟷	
PA_1	IN	OUT	IN	OUT	⟷	
PA_2	IN	OUT	IN	OUT	⟷	
PA_3	IN	OUT	IN	OUT	⟷	
PA_4	IN	OUT	IN	OUT	⟷	
PA_5	IN	OUT	IN	OUT	⟷	
PA_6	IN	OUT	IN	OUT	⟷	
PA_7	IN	OUT	IN	OUT	⟷	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

3

Special Mode Combination Considerations

There are several combinations of modes possible. For any combination, some or all of the Port C lines are used for control or status. The remaining bits are either inputs or outputs as defined by a "Set Mode" command.

During a read of Port C, the state of all the Port C lines, except the $\overline{ACK}$ and $\overline{STB}$ lines, will be placed on the data bus. In place of the $\overline{ACK}$ and $\overline{STB}$ line states, flag status will appear on the data bus in the PC2, PC4, and PC6 bit positions as illustrated by Figure 18.

Through a "Write Port C" command, only the Port C pins programmed as outputs in a Mode 0 group can be written. No other pins can be affected by a "Write Port C" command, nor can the interrupt enable flags be accessed. To write to any Port C output programmed as an output in a Mode 1 group or to change an interrupt enable flag, the "Set/Reset Port C Bit" command must be used.

With a "Set/Reset Port C Bit" command, any Port C line programmed as an output (including INTR, IBF and $\overline{OBF}$) can be written, or an interrupt enable flag can be either set or reset. Port C lines programmed as inputs, including $\overline{ACK}$ and $\overline{STB}$ lines, associated with Port C are not affected by a "Set/Reset Port C Bit" command. Writing to the corresponding Port C bit positions of the $\overline{ACK}$ and $\overline{STB}$ lines with the "Set/Reset Port C Bit" command will affect the Group A and Group B interrupt enable flags, as illustrated in Figure 18.

Current Drive Capability

Any output on Port A, B or C can sink or source 2.5 mA. This feature allows the 82C55A to directly drive Darlington type drivers and high-voltage displays that require such sink or source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 82C55A is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

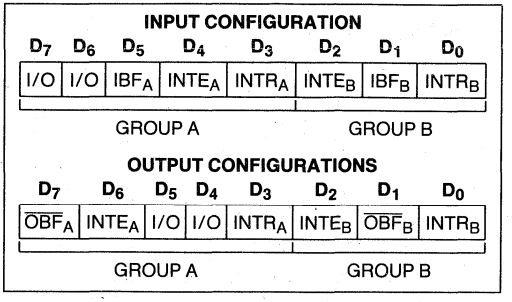


Figure 17a. MODE 1 Status Word Format

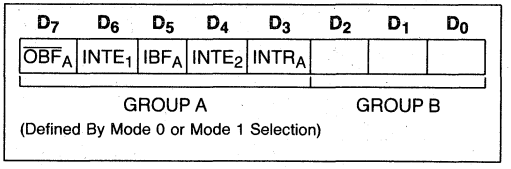


Figure 17b. MODE 2 Status Word Format

Interrupt Enable Flag	Position	Alternate Port C Pin Signal (Mode)
INTE B	PC2	$\overline{ACK}_B$ (Output Mode 1) or $\overline{STB}_B$ (Input Mode 1)
INTE A2	PC4	$\overline{STB}_A$ (Input Mode 1 or Mode 2)
INTE A1	PC6	$\overline{ACK}_A$ (Output Mode 1 or Mode 2

Figure 18. Interrupt Enable Flags in Modes 1 and 2

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . 0°C to + 70°C
Storage Temperature . . . . . − 65°C to + 150°C
Supply Voltage . . . . . − 0.5 to + 8.0V
Operating Voltage . . . . . + 4V to + 7V
Voltage on any Input . . . . . GND − 2V to + 6.5V
Voltage on any Output . . GND − 0.5V to V_{CC} + 0.5V
Power Dissipation . . . . . 1 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V (T_A = −40°C to +85°C for Extended Temperture)

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.5 mA
V_{OH}	Output High Voltage	3.0 V_{CC} − 0.4		V V	I_{OH} = −2.5 mA I_{OH} = −100 μA
I_{IL}	Input Leakage Current		±1	μA	V_{IN} = V_{CC} to 0V (Note 1)
I_{OFL}	Output Float Leakage Current		±10	μA	V_{IN} = V_{CC} to 0V (Note 2)
I_{DAR}	Darlington Drive Current	±2.5	(Note 4)	mA	Ports A, B, C R_{ext} = 500Ω V_{ext} = 1.7V
I_{PHL}	Port Hold Low Leakage Current	+50	+300	μA	V_{OUT} = 1.0V Port A only
I_{PHH}	Port Hold High Leakage Current	−50	−300	μA	V_{OUT} = 3.0V Ports A, B, C
I_{PHLO}	Port Hold Low Overdrive Current	−350		μA	V_{OUT} = 0.8V
I_{PHHO}	Port Hold High Overdrive Current	+350		μA	V_{OUT} = 3.0V
I_{CC}	V_{CC} Supply Current		10	mA	(Note 3)
I_{CCSB}	V_{CC} Supply Current-Standby		10	μA	V_{CC} = 5.5V V_{IN} = V_{CC} or GND Port Conditions If I/P = Open/High O/P = Open Only With Data Bus = High/Low $\overline{CS}$ = High Reset = Low Pure Inputs = Low/High

NOTES:
1. Pins A_1, A_0, $\overline{CS}$, $\overline{WR}$, $\overline{RD}$, Reset.
2. Data Bus; Ports B, C.
3. Outputs open.
4. Limit output current to 4.0 mA.

CAPACITANCE

T_A = 25°C, V_{CC} = GND = 0V

Symbol	Parameter	Min	Max	Units	Test Conditions
C_{IN}	Input Capacitance		10	pF	Unmeasured pins returned to GND f_c = 1 MHz(5)
$C_{I/O}$	I/O Capacitance		20	pF	

NOTE:
5. Sampled not 100% tested.

A.C. CHARACTERISTICS

T_A = 0° to 70°C, V_{CC} = +5V ±10%, GND = 0V

T_A = −40°C to +85°C for Extended Temperature

BUS PARAMETERS

READ CYCLE

Symbol	Parameter	82C55A-2		Units	Test Conditions
		Min	Max		
t_{AR}	Address Stable Before $\overline{RD}$ ↓	0		ns	
t_{RA}	Address Hold Time After $\overline{RD}$ ↑	0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	150		ns	
t_{RD}	Data Delay from $\overline{RD}$ ↓		120	ns	
t_{DF}	$\overline{RD}$ ↑ to Data Floating	10	75	ns	
t_{RV}	Recovery Time between $\overline{RD}/\overline{WR}$	200		ns	

WRITE CYCLE

Symbol	Parameter	82C55A-2		Units	Test Conditions
		Min	Max		
t_{AW}	Address Stable Before $\overline{WR}$ ↓	0		ns	
t_{WA}	Address Hold Time After $\overline{WR}$ ↑	20		ns	Ports A & B
		20		ns	Port C
t_{WW}	$\overline{WR}$ Pulse Width	100		ns	
t_{DW}	Data Setup Time Before $\overline{WR}$ ↑	100		ns	
t_{WD}	Data Hold Time After $\overline{WR}$ ↑	30		ns	Ports A & B
		30		ns	Port C

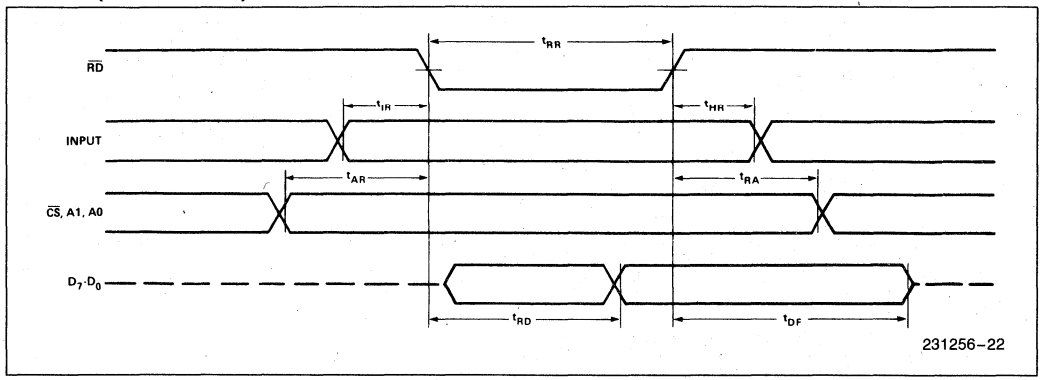
OTHER TIMINGS

Symbol	Parameter	82C55A-2		Units Conditions	Test
		Min	Max		
t_{WB}	$\overline{WR}$ = 1 to Output		350	ns	
t_{IR}	Peripheral Data Before $\overline{RD}$	0		ns	
t_{HR}	Peripheral Data After $\overline{RD}$	0		ns	
t_{AK}	$\overline{ACK}$ Pulse Width	200		ns	
t_{ST}	$\overline{STB}$ Pulse Width	100		ns	
t_{PS}	Per. Data Before $\overline{STB}$ High	20		ns	
t_{PH}	Per. Data After $\overline{STB}$High	50		ns	
t_{AD}	$\overline{ACK}$ = 0 to Output		175	ns	
t_{KD}	$\overline{ACK}$ = 1 to Output Float	20	250	ns	
t_{WOB}	$\overline{WR}$ = 1 to $\overline{OBF}$ = O		150	ns	
t_{AOB}	$\overline{ACK}$ = 0 to $\overline{OBF}$ = 1		150	ns	
t_{SIB}	$\overline{STB}$ = 0 to IBF = 1		150	ns	
t_{RIB}	$\overline{RD}$ = 1 to IBF = 0		150	ns	
t_{RIT}	$\overline{RD}$ = 0 to INTR = 0		200	ns	
t_{SIT}	$\overline{STB}$ = 1 to INTR = 1		150	ns	
t_{AIT}	$\overline{ACK}$ = 1 to INTR = 1		150	ns	
t_{WIT}	$\overline{WR}$ = 0 to INTR = 0		200	ns	see note 1
t_{RES}	Reset Pulse Width	500		ns	see note 2

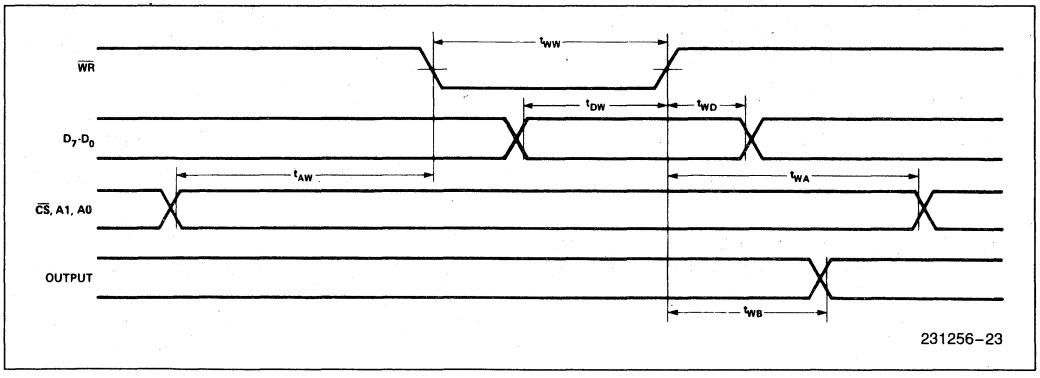
NOTE:
1. INTR ↑ may occur as early as $\overline{WR}$ ↓.
2. Pulse width of initial Reset pulse after power on must be at least 50 μSec. Subsequent Reset pulses may be 500 ns minimum.

WAVEFORMS

MODE 0 (BASIC INPUT)

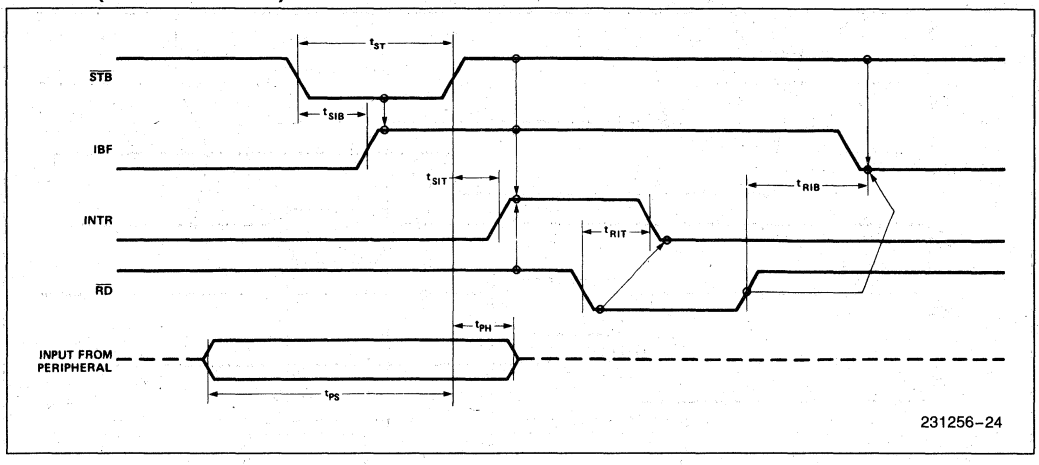


MODE 0 (BASIC OUTPUT)

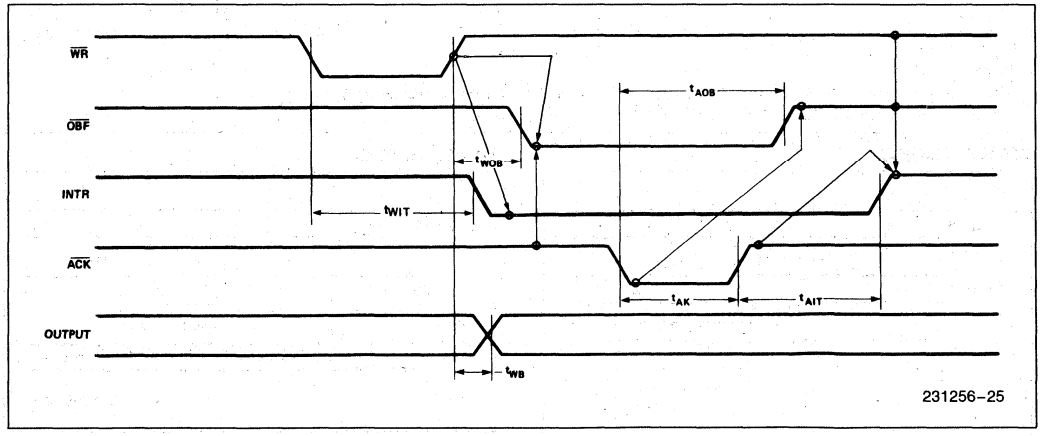


WAVEFORMS (Continued)

MODE 1 (STROBED INPUT)

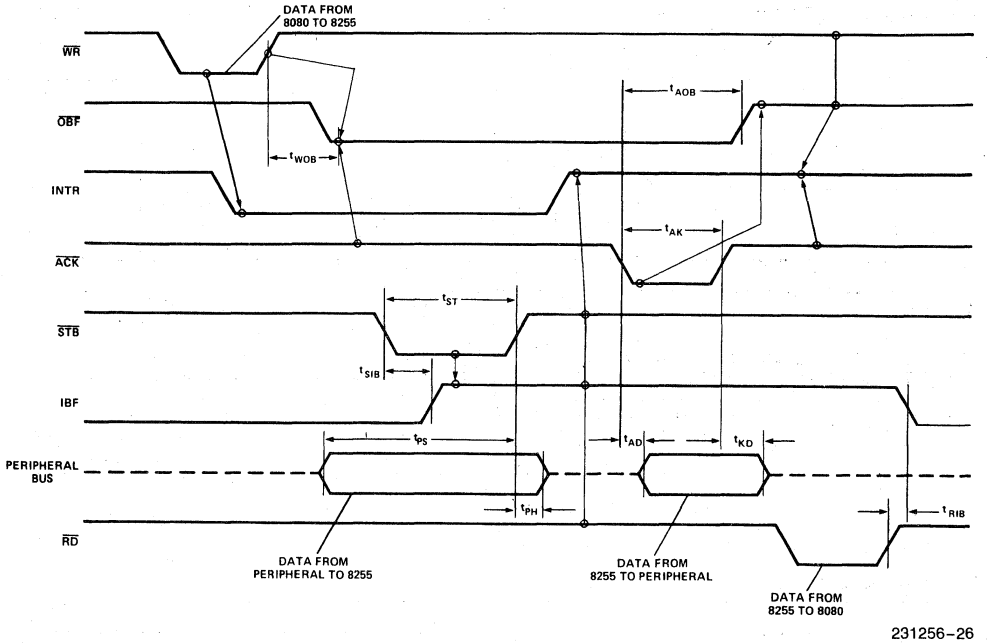


MODE 1 (STROBED OUTPUT)



WAVEFORMS (Continued)

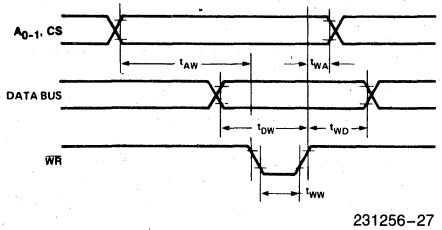
MODE 2 (BIDIRECTIONAL)



231256-26

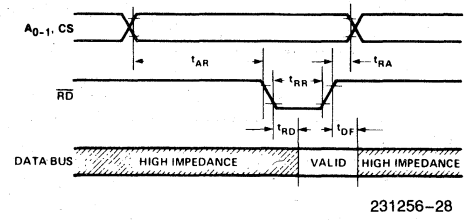
Note:
Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ AND $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

WRITE TIMING



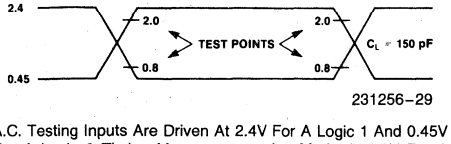
231256-27

READ TIMING



231256-28

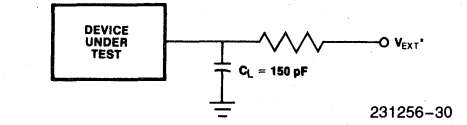
A.C. TESTING INPUT, OUTPUT WAVEFORM



231256-29

A.C. Testing Inputs Are Driven At 2.4V For A Logic 1 And 0.45V For A Logic 0 Timing Measurements Are Made At 2.0V For A Logic 1 And 0.8 For A Logic 0.

A.C. TESTING LOAD CIRCUIT



231256-30

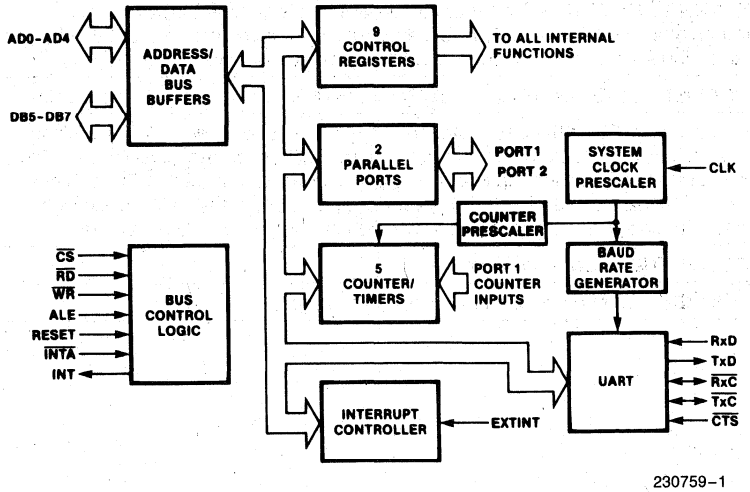
*V_{EXT} Is Set At Various Voltages During Testing To Guarantee The Specification. C_L Includes Jig Capacitance.

8256AH
MULTIFUNCTION MICROPROCESSOR SUPPORT CONTROLLER

- **Programmable Serial Asynchronous Communications Interface for 5-, 6-, 7-, or 8-Bit Characters, 1, 1½, or 2 Stop Bits, and Parity Generation**
- **On-Board Baud Rate Generator Programmable for 13 Common Baud Rates up to 19.2 KBits/Second, or an External Baud Clock Maximum of 1M Bit/Second**
- **Five 8-Bit Programmable Timer/Counters; Four Can Be Cascaded to Two 16-Bit Timer/Counters**
- **Two 8-Bit Programmable Parallel I/O Ports; Port 1 Can Be Programmed for Port 2 Handshake Controls and Event Counter Inputs**
- **Eight-Level Priority Interrupt Controller Programmable for 8085 or iAPX 86, iAPX 88 Systems and for Fully Nested Interrupt Capability**
- **Programmable System Clock to 1 ×, 2 ×, 3 ×, or 5 × 1.024 MHz**

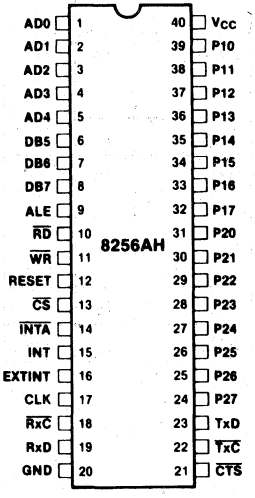
The Intel® 8256AH Multifunction Universal Asynchronous Receiver-Transmitter (MUART) combines five commonly used functions into a single 40-pin device. It is designed to interface to the 8086/88, iAPX 186/188, and 8051 to perform serial communications, parallel I/O, timing, event counting, and priority interrupt functions. All of these functions are fully programmable through nine internal registers. In addition, the five timer/counters and two parallel I/O ports can be accessed directly by the microprocessor.

3



230759-1

Figure 1. MUART Block Diagram



230759-2

Figure 2. MUART Pin Configuration

Table 1. Pin Description

Symbol	Pin	Type	Name and Function
AD0–AD4 DB5–DB7	1–5 6–8	I/O	**ADDRESS/DATA:** Three-state address/data lines which interface to the lower 8 bits of the microprocessor's multiplexed address/data bus. The 5-bit address is latched on the falling edge of ALE. In the 8-bit mode, AD0–AD3 are used to select the proper register, while AD1–AD4 are used in the 16-bit mode. AD4 in the 8-bit mode is ignored as an address, while AD0 in the 16-bit mode is used as a second chip select, active low.
ALE	9	I	**ADDRESS LATCH ENABLE:** Latches the 5 address lines on AD0–AD4 and $\overline{CS}$ on the falling edge.
$\overline{RD}$	10	I	**READ CONTROL:** When this signal is low, the selected register is gated onto the data bus.
$\overline{WR}$	11	I	**WRITE CONTROL:** When this signal is low, the value on the data bus is written into the selected register.
RESET	12	I	**RESET:** An active high pulse on this pin forces the chip into its initial state. The chip remains in this state until control information is written.
$\overline{CS}$	13	I	**CHIP SELECT:** A low on this signal enables the MUART. It is latched with the address on the falling edge of ALE, and $\overline{RD}$ and $\overline{WR}$ have no effect unless $\overline{CS}$ was latched low during the ALE cycle.
$\overline{INTA}$	14	I	**INTERRUPT ACKNOWLEDGE:** If the MUART has been enabled to respond to interrupts, this signal informs the MUART that its interrupt request is being acknowledged by the microprocessor. During this acknowledgement the MUART puts an RSTn instruction on the data bus for the 8-bit mode or a vector for the 16-bit mode.
INT	15	O	**INTERRUPT REQUEST:** A high signals the microprocessor that the MUART needs service.
EXTINT	16	I	**EXTERNAL INTERRUPT:** An external device can request interrupt service through this input. The input is level sensitive (high), therefore it must be held high until an $\overline{INTA}$ occurs or the interrupt address register is read.
CLK	17	I	**SYSTEM CLOCK:** The reference clock for the baud rate generator and the timers.
RxC	18	I/O	**RECEIVE CLOCK:** If the baud rate bits in the Command Register 2 are all 0, this pin is an input which clocks serial data into the RxD pin on the rising edge of RxC. If baud rate bits in Command Register 2 are programmed from 1–0FH, this pin outputs a square wave whose rising edge indicates when the data on RxD is being sampled. This output remains high during start, stop, and parity bits.
RxD	19	I	**RECEIVE DATA:** Serial data input.
GND	20	PS	**GROUND:** Power supply and logic ground reference.

Table 1. Pin Description (Continued)

Symbol	Pin	Type	Name and Function
$\overline{CTS}$	21	I	**CLEAR TO SEND:** This input enables the serial transmitter. If 1, 1.5, or 2 stop bits are selected $\overline{CTS}$ is level sensitive. As long as $\overline{CTS}$ is low, any character loaded into the transmitter buffer register will be transmitted serially. A single negative going pulse causes the transmission of a single character previously loaded into the transmitter buffer register. If a baud rate from 1–0FH is selected, $\overline{CTS}$ must be low for at least $1/32$ of a bit, or it will be ignored. If the transmitter buffer is empty, this pulse will be ignored. If this pulse occurs during the transmission of a character up to the time where $1/2$ the first (or only) stop bit is sent out, it will be ignored. If it occurs afterwards, but before the end of the stop bits, the next character will be transmitted immediately following the current one. If $\overline{CTS}$ is still high when the transmitter register is sending the last stop bit, the transmitter will enter its idle state until the next high-to-low transition on $\overline{CTS}$ occurs. If 0.75 stop bits is chosen, the $\overline{CTS}$ input is edge sensitive. A negative edge on $\overline{CTS}$ results in the immediate transmission of the next character. The length of the stop bits is determined by the time interval between the beginning of the first stop bit and the next negative edge on $\overline{CTS}$. A high-to-low transition has no effect if the transmitter buffer is empty or if the time interval between the beginning of the stop bit and next negative edge is less than 0.75 bits. A high or a low level or a low-to-high transition has no effect on the transmitter for the 0.75 stop bit mode.
TxC	22	I/O	**TRANSMIT CLOCK:** If the baud rate bits in command register 2 are all set to 0, this input clocks data out of the transmitter on the falling edge. If baud rate bits are programmed for 1 or 2, this input permits the user to provide a 32× or 64× clock which is used for the receiver and transmitter. If the baud rate bits are programmed for 3–0FH, the internal transmitter clock is output. As an output it delivers the transmitter clock at the selected bit rate. If $1\frac{1}{2}$ or 0.75 stop bits are selected, the transmitter divider will be asynchronously reset at the beginning of each start bit, immediately causing a high-to-low transition on TxC. TxC makes a high-to-low transition at the beginning of each serial bit, and a low-to-high transition at the center of each bit.
TxD	23	O	**TRANSMIT DATA:** Serial data output.
P27–P20	24–31	I/O	**PARALLEL I/O PORT 2:** Eight bit general purpose I/O port. Each nibble (4 bits) of this port can be either an input or an output. The outputs are latched whereas the input signals are not. Also, this port can be used as an 8-bit input or output port when using the two-wire handshake. In the handshake mode both inputs and outputs are latched.
P17–P10	32–39	I/O	**PARALLEL I/O PORT 1:** Each pin can be programmed as an input or an output to perform general purpose I/O. All outputs are latched whereas inputs are not. Alternatively these pins can serve as control pins which extend the functional spectrum of the chip.
V_{CC}	40	PS	**POWER:** +5V power supply.

FUNCTIONAL DESCRIPTION

The 8256AH Multi-Function Universal Asynchronous Receiver-Transmitter (MUART) combines five commonly used functions into a single 40-pin device. The MUART performs asynchronous serial communications, parallel I/O, timing, event counting, and interrupt control. For detailed application information, see Intel AP Note #153, Designing with the 8256.

Serial Communications

The serial communications portion of the MUART contains a full-duplex asynchronous receiver-transmitter (UART). A programmable baud rate generator is included on the MUART to permit a variety of operating speeds without external components. The UART can be programmed by the CPU for a variety of character sizes, parity generation and detection, error detection, and start/stop bit handling. The receiver checks the start and stop bits in the center of the bit, and a break halts the reception of data. The transmitter can send breaks and can be controlled by an external enable pin.

Parallel I/O

The MUART includes 16 bits of general purpose parallel I/O. Eight bits (Port 1) can be individually changed from input to output or used for special I/O functions. The other eight bits (Port 2) can be used as nibbles (4 bits) or as bytes. These eight bits also include a handshaking capability using two pins on Port 1.

Counter/Timers

There are five 8-bit counter/timers on the MUART. The timers can be programmed to use either a 1 kHz or 16 kHz clock generated from the system clock. Four of the 8-bit counter/timers can be cascaded to two 16-bit counter/timers, and one of the 8-bit counter/timers can be reset to its initial value by an external signal.

Interrupts

An eight-level priority interrupt controller can be configured for fully nested or normal interrupt priority. Seven of the eight interrupts service functions on the MUART (counter/timers, UART), and one external interrupt is provided which can be used for a particular function or for chaining interrupt controllers or more MUARTs. The MUART will support 8085 and 8086/88 systems with direct interrupt vectoring, or the MUART can be polled to determine the cause of the interrupt. If additional interrupt control capability is needed, the MUART's interrupt controller can be cascaded into another MUART, into an Intel 8259A Programmable Interrupt Controller, or into the interrupt controller of the iAPX 186/188 High-Integration Microprocessor.

INITIALIZATION

In general the MUART's functions are independent of each other and only the registers and bits associated with a particular function need to be initialized, not the entire chip. The command sequence is arbitrary since every register is directly addressable; however, **Command Byte 1** must be loaded first. To put the device into a fully operational condition, it is necessary to write the following commands:

Command byte 1
Command byte 2
Command byte 3
Mode byte
Port 1 control
Set Interrupts

The modification register may be loaded if required for special applications; normally this operation is not necessary. The MUART should be reset before initialization. (Either a hardware or a software reset will do.)

INTERFACING

This section describes the hardware interface between the 8256 MUART and the 80186 microprocessor. Figure 3 displays the block diagram for this interface. The MUART can be interfaced to many other microprocessors using these basic principles.

In all cases the 8256 will be connected directly to the CPU's multiplexed address/data bus. If latches or data bus buffers are used in a system, the MUART should be on the microprocessor side of the address/data bus. The MUART latches the address internally on the falling edge of ALE. The address consists of Chip Select (CS) and four address lines. For 8-bit microprocessors, AD0–AD3 are the address lines. For 16-bit microprocessors, AD1–AD4 are the address lines; AD0 is used as a second chip select which is active low. Since chip select is internally latched along with the address, it does not have to remain active during the entire instruction cycle. As long as the chip select setup and hold times are met, it can be derived from multiplexed ad-

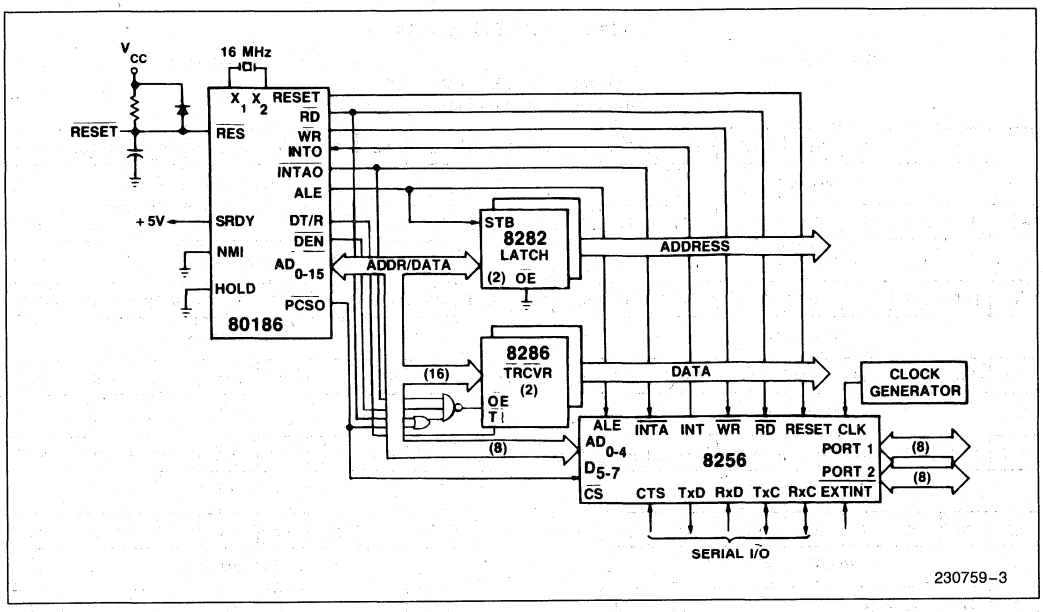


Figure 3. 80186/8256 Interface

dress/data lines or multiplexed address/status lines. When the 8256 is in the 16-bit mode, A0 serves as a second chip select. As a result the MUART's internal registers will all have even addresses since A0 must be zero to select the device. Normally the MUART will be placed on the lower data byte. If the MUART is placed on the upper data byte, the internal registers will be 512 address locations apart and the chip would occupy an 8k word address space.

DESCRIPTION OF THE REGISTERS

The following section will provide a description of the registers and define the bits within the registers where appropriate. Table 2 lists the registers and their addresses.

Command Register 1

L1	L0	S1	S0	BRKI	BITI	8086	FRQ
(0R)					(0W)		

FRQ—TIMER FREQUENCY SELECT

This bit selects between two frequencies for the five timers. If FRQ = 0, the timer input frequency is 16 kHz (62.5 μs). If FRQ = 1, the timer input frequency is 1 kHz (1 ms). The selected clock frequency is shared by all the counter/timers enabled for timing; thus, all timers must run with the same time base.

8086—8086 MODE ENABLE

This bit selects between 8085 mode and 8086/8088 mode. In 8085 mode (8086 = 0), A0 to A3 are used to address the internal registers, and an RSTn instruction is generated in response to the first $\overline{\text{INTA}}$. In 8086 mode (8086 = 1), A1 to A4 are used to address the internal registers, and A0 is used as an extra chip select (A0 must equal zero to be enabled). The response to $\overline{\text{INTA}}$ is for 8086 interrupts where the first $\overline{\text{INTA}}$ is ignored, and an interrupt vector (40H to 47H) is placed on the bus in response to the second $\overline{\text{INTA}}$.

BITI—INTERRUPT ON BIT CHANGE

This bit selects between one of two interrupt sources on Priority Level 1, either Counter/Timer 2 or Port 1 P17 interrupt. When this bit equals 0, Counter/Timer 2 will be mapped into Priority Level 1. If BITI equals 0 and Level 1 interrupt is enabled, a transition from 1 to 0 in Counter/Timer 2 will generate an interrupt request on Level 1. When BITI equals 1, Port 1 P17 external edge triggered interrupt source is mapped into Priority Level 1. In this case if Level 1 is enabled, a low-to-high transition on P17 generates an interrupt request on Level 1.

Table 2. MUART Registers

Read Registers								8085 Mode: AD3	AD2	AD1	AD0	Write Registers							
								8086 Mode: AD4	AD3	AD2	AD1								
L1	L0	S1	S0	BRKI	BITI	8086	FRQ	0	0	0	0	L1	L0	S1	S0	BRKI	BITI	8086	FRQ
Command 1												Command 1							
PEN	EP	C1	C0	B3	B2	B1	B0	0	0	0	1	PEN	EP	C1	C0	B3	B2	B1	B0
Command 2												Command 2							
0	RxE	IAE	NIE	0	SBRK	TBRK	0	0	0	1	0	SET	RxE	IAE	NIE	END	SBRK	TBRK	RST
Command 3												Command 3							
T35	T24	T5C	CT3	CT2	P2C2	P2C1	P2C0	0	0	1	1	T35	T24	T5C	CT3	CT2	P2C2	P2C1	P2C0
Mode												Mode							
P17	P16	P15	P14	P13	P12	P11	P10	0	1	0	0	P17	P16	P15	P14	P13	P12	P11	P10
Port 1 Control												Port 1 Control							
L7	L6	L5	L4	L3	L2	L1	L0	0	1	0	1	L7	L6	L5	L4	L3	L2	L1	L0
Interrupt Enable												Set Interrupts							
D7	D6	D5	D4	D3	D2	D1	D0	0	1	1	0	L7	L6	L5	L4	L3	L2	L1	L0
Interrupt Address												Reset Interrupts							
D7	D6	D5	D4	D3	D2	D1	D0	0	1	1	1	D7	D6	D5	D4	D3	D2	D1	D0
Receiver Buffer												Transmitter Buffer							
D7	D6	D5	D4	D3	D2	D1	D0	1	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0
Port 1												Port 1							
D7	D6	D5	D4	D3	D2	D1	D0	1	0	0	1	D7	D6	D5	D4	D3	D2	D1	D0
Port 2												Port 2							
D7	D6	D5	D4	D3	D2	D1	D0	1	0	1	0	D7	D6	D5	D4	D3	D2	D1	D0
Timer 1												Timer 1							
D7	D6	D5	D4	D3	D2	D1	D0	1	0	1	1	D7	D6	D5	D4	D3	D2	D1	D0
Timer 2												Timer 2							
D7	D6	D5	D4	D3	D2	D1	D0	1	1	0	0	D7	D6	D5	D4	D3	D2	D1	D0
Timer 3												Timer 3							
D7	D6	D5	D4	D3	D2	D1	D0	1	1	0	1	D7	D6	D5	D4	D3	D2	D1	D0
Timer 4												Timer 4							
D7	D6	D5	D4	D3	D2	D1	D0	1	1	1	0	D7	D6	D5	D4	D3	D2	D1	D0
Timer 5												Timer 5							
INT	RBF	TBE	TRE	BD	PE	OE	FE	1	1	1	1	0	RS4	RS3	RS2	RS1	RS0	TME	DSC
Status												Modification							

BRKI—BREAK-IN DETECT ENABLE

If this bit equals 0, Port 1 P16 is a general purpose I/O port. When BRKI equals 1, the Break-In Detect feature is enabled on Port 1 P16. A Break-In condition is present on the transmission line when it is forced to the start bit voltage level by the receiving station. Port 1 P16 must be connected externally to the transmission line in order to detect a Break-In. A Break-In is polled by the MUART during the transmission of the last or only stop bit of a character.

A Break-In Detect is OR-ed with Break Detect in Bit 3 of the Status Register. The distinction can be made through the interrupt controller. If the transmit and receive interrupts are enabled, a Break-In will generate an interrupt on Level 5, the transmit interrupt, while Break will generate an interrupt on Level 4, the receive interrupt.

S0, S1—STOP BIT LENGTH

S1	S0	Stop Bit Length
0	0	1
0	1	1.5
1	0	2
1	1	0.75

The relationship of the number of stop bits and the function of input $\overline{CTS}$ is discussed in the Pin Description section under "$\overline{CTS}$".

L0, L1—CHARACTER LENGTH

L1	L0	Character Length
0	0	8
0	1	7
1	0	6
1	1	5

Command Register 2

PEN	EP	C1	C0	B3	B2	B1	B0
(1R)				(1W)			

Programming bits 0 . . . 3 with values from 3H to FH enables the internal baud rate generator as a common clock source for the transmitter and receiver and determines its divider ratio.

Programming bits 0 . . . 3 with values of 1H or 2H enables input TxC as a common clock source for the transmitter and receiver. The external clock must provide a frequency of either 32× or 64× the baud rate. The data transmission rates range from 0. . .32 Kbaud.

If bits 0 . . . 3 are set to 0, separate clocks must be input to pin RxC for the receiver and pin TxC for the transmitter. Thus, different baud rates can be used for transmission and reception. In this case, prescalers are disabled and the input serial clock frequency must match the baud rate. The input serial clock frequency can range from 0 MHz to 1.024 MHz.

B0, B1, B2, B3—BAUD RATE SELECT

These four bits select the bit clock's source, sampling rate, and serial rate for the internal baud rate generator.

B3	B2	B1	B0	Baud Rate	Sampling Rate
0	0	0	0	$\overline{TxC}$, $\overline{RxC}$	1
0	0	0	1	$\overline{TxC}$/64	64
0	0	1	0	$\overline{TxC}$/32	32
0	0	1	1	19200	32
0	1	0	0	9600	64
0	1	0	1	4800	64
0	1	1	0	2400	64
0	1	1	1	1200	64
1	0	0	0	600	64
1	0	0	1	300	64
1	0	1	0	200	64
1	0	1	1	150	64
1	1	0	0	110	64
1	1	0	1	100	64
1	1	1	0	75	64
1	1	1	1	50	64

The following table gives an overview of the function of pins TxC and RxC:

Bits 3 to 0 (Hex.)	TxC	RxC
0	Input: 1 × baud rate clock for the transmitter	Input: 1 × baud rate clock for the receiver
1, 2	Input: 32 × or 64 × baud rate for transmitter and receiver	Output: receiver bit clock with a low-to-high transition at data bit sampling time. Otherwise: high level
3 to F	Output: baud rate clock of the transmitter	Output: as above

As an output, RxC outputs a low-to-high transition at sampling time of every data bit of a character. Thus, data can be loaded, e.g., into a shift register externally. The transition occurs only if data bits of a character are present. It does not occur for start, parity, and stop bits (RxC = high).

As an output, TxC outputs the internal baud rate clock of the transmitter. There will be a high-to-low transition at every beginning of a bit.

C0, C1—SYSTEM CLOCK PRESCALER (BITS 4, 5)

Bits 4 and 5 define the system clock prescaler divider ratio. The internal operating frequency of 1.024 MHz is derived from the system clock.

C1	C0	Divider Ratio	Clock at Pin CLK
0	0	5	5.12 MHz
0	1	3	3.072 MHz
1	0	2	2.048 MHz
1	1	1	1.024 MHz

EP—EVEN PARITY (BIT 6)

EP = 0: Odd parity
EP = 1: Even parity

PEN—PARITY ENABLE (BIT 7)

Bit 7 enables parity generation and checking.

PEN = 0: No parity bit
PEN = 1: Even parity bit

The parity bit according to Command Register 2 bit 6 (see above) is inserted between the last data bit of a character and the first or only stop bit. The parity bit is checked during reception. A false parity bit generates an error indication in the Status Register and an Interrupt Request on Level 4.

Command Register 3

SET	RxE	IAE	NIW	END	SBRK	TBRK	RST
	(2R)				(2W)		

Command Register 3 is different from the first two registers because it has a bit set/reset capability. Writing a byte with Bit 7 high sets any bits which were also high. Writing a byte with Bit 7 low resets any bits which were high. If any bit 0–6 is low, no change occurs to that bit. When command Register 3 is read, bits 0, 3, and 7 will always be zero.

RST—RESET

If RST is set, the following events occur:

1. All bits in the Status Register except bits 4 and 5 are cleared, and bits 4 and 5 are set.
2. The Interrupt Enable, Interrupt Request, and Interrupt Service Registers are cleared. Pending requests and indications for interrupts in service will be cancelled. Interrupt signal INT will go low.
3. The receiver and transmitter are reset. The transmitter goes idle (TxD is high), and the receiver enters start bit search mode.
4. If Port 2 is programmed for handshake mode, IBF and OBF are reset high.

RST does *not* alter ports, data registers or command registers, but it halts any operation in progress. RST is automatically cleared.

RST = 0 has no effect. The reset operation triggered by Command Register 3 is a subset of the hardware reset.

TBRK—TRANSMIT BREAK

The transmission data output TxD will be set low as soon as the transmission of the previous character has been finished. It stays low until TBRK is cleared. The state of $\overline{CTS}$ is of no significance for this operation. As long as break is active, data transfer from the Transmitter Buffer to the Transmitter Register will be inhibited. As soon as TBRK is reset, the break condition will be deactivated and the transmitter will be re-enabled.

SBRK—SINGLE CHARACTER BREAK

This causes the transmitter data to be set low for one character including start bit, data bits, parity bit, and stop bits. SBRK is automatically cleared when time for the last data bit has passed. It will start after the character in progress completes, and will delay the next data transfer from the Transmitter Buffer to the Transmitter Register until TxD returns to an idle (marking) state. If both TBRK and SBRK are set, break will be set as long as TBRK is set, but SBRK will be cleared after one character time of break. If SBRK is set again, it remains set for another character. The user can send a definite number of break characters in this manner by clearing TBRK after setting SBRK for the last character time.

END—END OF INTERRUPT

If fully nested interrupt mode is selected, this bit resets the currently served interrupt level in the Interrupt Service Register. *This command must occur at the end of each interrupt service routine during fully nested interrupt mode.* END is automatically cleared when the Interrupt Service Register (internal) is cleared. END is ignored if nested interrupts are not enabled.

NIE—NESTED INTERRUPT ENABLE

When NIE equals 1, the interrupt controller will operate in the nested interrupt mode. When NIE equals 0, the interrupt controller will operate in the normal interrupt mode. Refer to the "Interrupt controller" section of AP-153 under "Normal Mode" and "Nested Mode" for a detailed description of these operations.

IAE—INTERRUPT ACKNOWLEDGE ENABLE

This bit enables an automatic response to $\overline{\text{INTA}}$. The particular response is determined by the 8086 bit in Command Register 1.

RxE—RECEIVE ENABLE

This bit enables the serial receiver and its associated status bits in the status register. If this bit is reset, the serial receiver will be disabled and the receive status bits will not be updated.

Note that the detection of break characters remains enabled while the receiver is disabled; i.e., Status Register Bit 3 (BD) will be set while the receiver is disabled whenever a break character has been recognized at the receive data input RxD.

SET—BIT SET/RESET

If this bit is high during a write to Command Register 3, then any bit marked by a high will set. If this bit is low, then any bit marked by a high will be cleared.

MODE REGISTER

T35	T24	T5C	CT3	CT2	P2C2	P2C1	P2C0
(3R)					(3W)		

If test mode is selected, the output from the internal baud rate generator is placed on bit 4 of Port 1 (pin 35).

To achieve this, it is necessary to program bit 4 of Port 1 as an output (Port 1 Control Register Bit P14 = 1), and to program Command Register 2 bits B3–B0 with a value ≥ 3H.

P2C2, P2C1, P2C0—PORT 2 CONTROL

P2C2	P2C1	P2C0	Mode	Direction	
				Upper	Lower
0	0	0	Nibble	Input	Input
0	0	1	Nibble	Input	Output
0	1	0	Nibble	Output	Input
0	1	1	Nibble	Output	Output
1	0	0	Byte Handshake	Input	
1	0	1	Byte Handshake	Output	
1	1	0	*DO NOT USE*		
1	1	1	Test		

NOTE:
If Port 2 is operating in handshake mode, Interrupt Level 7 is not available for Timer 5. Instead it is assigned to Port 2 handshaking.

CT2, CT3—COUNTER/TIMER MODE

Bit 3 and 4 defines the mode of operation of event counter/timers 2 and 3 regardless of its use as a single unit or as a cascaded one.

If CT2 or CT3 are high, then counter/timer 2 or 3 respectively is configured as an event counter on bit 2 or 3 respectively of Port 1 (pins 37 or 36). The event counter decrements the count by one on each low-to-high transition of the external input. If CT2 or CT3 is low, then the respective counter/timer is configured as a timer and the Port 1 pins are used for parallel I/O.

T5C—TIMER 5 CONTROL

If T5C is set, then Timer 5 can be preset and started by an external signal. Writing to the Timer 5 register loads the Timer 5 save register and stops the timer. A high-to-low transition on bit 5 of Port 1 (pin 34) loads the timer with the saved value and starts the timer. The next high-to-low transition on pin 34 retriggers the timer by reloading it with the initial value and continues timing.

Following a hardware reset, the save register is reset to 00H and both clock and trigger inputs are disabled. Transferring an instruction with T5C = 1 enables the trigger input; the save register can now be loaded with an initial value. The first trigger pulse causes the initial value to be loaded from the save register and enables the counter to count down to zero.

When the timer reaches zero it issues an interrupt request, disables its interrupt level and continues

counting. A subsequent high-to-low transition on pin 5 resets Timer 5 to its initial value. For another timer interrupt, the Timer 5 interrupt enable bit must be set again.

T35, T24—CASCADE TIMERS

These two bits cascade Timers 3 and 5 or 2 and 4.

Timers 2 and 3 are the lower bytes, while Timers 4 and 5 are the upper bytes. It T5C is set, then both Timers 3 and 5 can be preset and started by an external pulse.

When a high-to-low transition occurs, Timer 5 is preset to its saved value, but Timer 3 is always preset to all ones. If either CT2 or CT3 is set, then the corresponding timer pair is a 16-bit event counter.

A summary of the counter/timer control bits is given in Table 3.

NOTE:
Interrupt levels assigned to single counters are partly not occupied if event counters/timers are cascaded. Level 2 will be vacated if event counters/timers 2 and 4 are cascaded. Likewise, Level 7 will be vacated if event counters/timers 3 and 5 are cascaded.

Single event counters/timers generate an interrupt request on the transition from 01H to 00H, while cascaded ones generate it on the transition from 0001H to 0000H.

Port 1 Control Register

P17	P16	P15	P14	P13	P12	P11	P10
	(4R)				(4W)		

Each bit in the Port 1 Control Register configures the direction of the corresponding pin. If the bit is high, the pin is an output, and if it is low the pin is an input. Every Port 1 pin has another function which is controlled by other registers. If that special function is disabled, the pin functions as a general I/O pin as specified by this register. The special functions for each pin are described below.

Port 10, 11—HANDSHAKE CONTROL

If byte handshake control is enabled for Port 2 by the Mode Register, then Port 10 is programmed as $\overline{STB}/\overline{ACK}$ handshake control input, and Port 11 is programmed as $\overline{IBF}/\overline{OBF}$ handshake control output.

If byte handshake mode is enabled for output on Port 2 $\overline{OBF}$ indicates that a character has been loaded into the Port 2 output buffer. When an external

Table 3. Event Counters/Timers Mode of Operation

Event Counter/ Timer	Function	Programming (Mode Word)	Clock Source
1	8-bit timer	—	Internal clock
2	8-bit timer	T24 = 0, CT2 = 0	Internal clock
	8-bit event counter	T24 = 0, CT2 = 1	P12 pin 37
2	8-bit timer	T35 = 0, CT3 = 0	Internal clock
	8-bit event counter	T35 = 0, CT3 = 1	P13 pin 36
4	8-bit timer	T24 = 0	Internal clock
5	8-bit timer, normal mode	T35 = 0, T5C = 0	Internal clock
	8-bit timer, retriggerable mode	T35 = 0, T5C = 1	Internal clock
2 and 4 cascaded	16-bit timer	T24 = 1, CT2 = 0	Internal clock
	16-bit event counter	T24 = 1, CT2 = 1	P12 pin 37
3 and 5 cascaded	16-bit timer, normal mode	T35 = 1, T5C = 0, CT3 = 0	Internal clock
	16-bit event counter, normal mode	T35 = 1, T5C = 0, CT3 = 1	P13 pin 36
	16-bit timer, retriggerable mode	T35 = 1, T5C = 1, CT3 = 0	Internal clock
	16-bit event counter, retriggerable mode	T35 = 1, T5C = 1, CT3 = 1	P13 pin 36

device reads the data, it acknowledges this operation by driving $\overline{ACK}$ low. $\overline{OBF}$ is set low by writing to Port 2 and is reset by $\overline{ACK}$.

If byte handshake mode is enabled for input on Port 2, $\overline{STB}$ is an input. $\overline{IBF}$ is driven low after $\overline{STB}$ goes low. On the rising edge of $\overline{STB}$ the data from Port 2 is latched.

$\overline{IBF}$ is reset high when Port 2 is read.

PORT 12, 13—COUNTER 2, 3 INPUT

If Timer 2 or Timer 3 is programmed as an event counter by the Mode Register, then Port 12 or Port 13 is the counter input for Event Counter 2 or 3, respectively.

PORT 14—BAUD RATE GENERATOR OUTPUT CLOCK

If test mode is enabled by the Mode Register and Command Register 2 baud rate select is greater than 2, then Port 14 is an output from the internal baud rate generator.

P14 in Port 1 control register must be set to 1 for the baud rate generator clock to be output. The baud rate generator clock is 64 × the serial bit rate excet at 19.2 Kbps when it is 32 × the bit rate.

PORT 15—TIMER 5 TRIGGER

If T5C is set in the Mode Register enabling a retriggerable timer, then Port 15 is the input which starts and reloads Timer 5.

A high-to-low transition on P15 (Pin 34) loads the timer with the slave register and starts the timer.

PORT 16—BREAK-IN DETECT

If Break-In Detect is enabled by BRKI in Command Register 1, then this input is used to sense a Break-In. If Port 16 is low while the serial transmitter is sending the last stop bit, then a Break-In condition is signaled.

PORT 17—PORT INTERRUPT SOURCE

If BITI in Command Register 1 is set, then a low-to-high transition on Port 17 generates an interrupt request on Priority Level 1.

Port 17 is edge triggered.

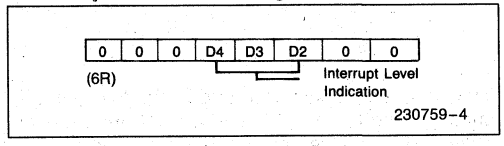
Interrupt Enable Register

L7	L6	L5	L4	L3	L2	L1	L0
(5R)					(5W = enable, 6W = disable)		

Interrupts are enabled by writing to the Set Interrupts Register (5W). Interrupts are disabled by writing to the Reset Interrupts Register (6W). Each bit set by the Set Interrupts Register (5W) will enable that level interrupt, and each bit set in the Reset Interrupts Register (6W) will disable that level interrupt. The user can determine which interrupts are enabled by reading the Interrupt Enable Register (5R).

Priority		Source
Highest	L0	Timer 1
	L1	Timer 2 or Port Interrupt
	L2	External Interrupt (EXTINT)
	L3	Timer 3 or Timers 3 & 5
	L4	Receive Interrupt
	L5	Transmitter Interrupt
	L6	Timer 4 or Timers 2 & 4
Lowest	L7	Timer 5 or Port 2 Handshaking

Interrupt Address Register



Reading the interrupt address register transfers an identifier for the currently requested interrupt level on the system data bus. This identifier is the number of the interrupt level multiplied by 4. It can be used by the CPU as an offset address for interrupt handling. Reading the interrupt address register has the same effect as a hardware interrupt acknowledge $\overline{INTA}$; it clears the interrupt request pin (INT) and indicates an interrupt acknowledgement to the interrupt controller.

Receiver and Transmitter Buffer

D7	D6	D5	D4	D3	D2	D1	D0
(7R)					(7W)		

Both the receiver and transmitter in the MUART are double buffered. This means that the transmitter and receiver have a shift register and a buffer register. The buffer registers are directly addressable by reading or writing to register seven. After the receiver buffer is full, the RBF bit in the status register is set.

Reading the receive buffer clears the RBF status bit. The transmit buffer should be written to only if the TBE bit in the status register is set. Bytes written to the transmit buffer are held there until the transmit shift register is empty, assuming $\overline{CTS}$ is low. If the transmit buffer and shift register are empty, writing to the transmit buffer immediately transfers the byte to the transmit shift register. If a serial character length is less than 8 bits, the unused most significant bits are set to zero when reading the receive buffer, and are ignored when writing to the transmit buffer.

Port 1

D7	D6	D5	D4	D3	D2	D1	D0
(8R)					(8W)		

Writing to Port 1 sets the data in the Port 1 output latch. Writing to an input pin does not affect the pin, but the data is stored and will be output if the direction of the pin is changed later. If the pin is used as a control signal, the pin will not be affected, but the data is stored. Reading Port 1 transfers the data in Port 1 onto the data bus.

Port 2

D7	D6	D5	D4	D3	D2	D1	D0
(9R)					(9W)		

Writing to Port 2 sets the data in the Port 2 output latch. Writing to an input pin does not affect the pin, but it does store the data in the latch. Reading Port 2 puts the input pins onto the bus or the contents of the output latch for output pins.

Timer 1-5

D7	D6	D5	D4	D3	D2	D1	D0
($0A_{16}$-OE_{16}R)					($0A_{16}$-OE_{16}W)		

Reading Timer N puts the contents of the timer onto the data bus. If the counter changes while $\overline{RD}$ is low, the value on the data bus will not change. If two timers are cascaded, reading the high-order byte will cause the low-order byte to be latched. Reading the low-order byte will unlatch them both. Writing to either timer or decascading them also clears the latch condition. Writing to a timer sets the starting value of that timer. If two timers are cascaded, writing to the high-order byte presets the low-order byte to all ones. Loading only the high-order byte with a value of X leads to a count of X *256 + 255. Timers count down continuously. If the interrupt is enabled, it occurs when the counter changes from 1 to 0.

The timer/counter interrupts are automatically disabled when the interrupt request is generated.

Status Register

INT	RBF	TBE	TRE	BD	PE	OE	FE
($0F_{16}$R)							

Reading the status register gates its contents onto the data bus. It holds the operational status of the serial interface as well as the status of the interrupt in INT. The status register can be read at any time. The flags are stable and well defined at all instants.

FE—FRAMING ERROR, TRANSMISSION MODE

Bit 0 can be used in two modes. Normally, FE indicates framing error which can be changed to transmission mode indication by setting the TME bit in the modification register.

If transmission mode is disabled (in Modification Register), then FE indicates a framing error. A framing error is detected during the *first* stop bit. The error is reset by reading the Status Register or by a chip reset. A framing error does not inhibit the loading of the Receiver Buffer. If RxD remains low, the receiver will assemble the next character. The false stop bit is treated as the next start bit, and no high-to-low transition on RxD is required to synchronize the receiver.

When the TME bit in the Modification Register is set, FE is used to indicate that the transmitter was active during the reception of a character, thus indicating that the character received was transmitted by its own transmitter. FE is reset when the transmitter is not active during the reception of character. Reading the status register will not reset the FE bit in the transmission mode.

OE—OVERRUN ERROR

If the user does not read the character in the Receiver Buffer before the next character is received and transferred to this register, then the OE bit is set. The OE flag is set during the reception of the first stop bit and is cleared when the Status Register is read or when a hardware or software reset occurs. The first character received in this case will be lost.

PE—PARITY ERROR

This bit indicates a parity error has occurred during the reception of a character. A parity error is present

if value of the parity bit in the received character is different from the one expected according to command word 2 bits 6 EP. The parity bit is expected and checked only if it is enabled by command word 2 bit 7 PEN.

A parity error is set during the first stop bit and is reset by reading the Status Register or by a chip reset.

BD—BREAK/BREAK-IN

The BD bit flags whether a break character has been received, or a Break-In condition exists on the transmission line. Command Register 1 Bit 3 (BRKI) enables the Break-In Detect function.

Whenever a break character has been received, Status Register Bit 3 will be set and in addition an interrupt request on Level 4 is generated. The receiver will be idled. It will be started again with the next high-to-low transition at pin RxD.

The break character received will not be loaded into the receiver buffer register.

If Break-In Detection is enabled and a Break-In condition occurs, Status Register Bit 3 will be set and in addition an interrupt request on Level 5 is generated.

The BD status bit will be reset on reading the status register or on a hardware or software reset. For more information on Break/Break-In, refer to the "Serial Asynchronous Communication" section of AP-153 under "Receive Break Detect" and "Break-In Detect."

TRE—TRANSMIT REGISTER EMPTY

When TRE is set the transmit register is empty and an interrupt request is generated on Level 5 if enabled. When TRE equals 0 the transmit register is in the process of sending data. TRE is set by a chip reset and when the last stop bit has left the transmitter. It is reset when a character is loaded into the Transmitter Register. If $\overline{CTS}$ is low, the Transmitter Register will be loaded during the transmission of the start bit. If $\overline{CTS}$ is high at the end of a character, TRE will remain high and no character will be loaded into the Transmitter Register until $\overline{CTS}$ goes low. If the transmitter was inactive before a character is loaded into the Transmitter Buffer, the Transmitter Register will be empty temporarily while the buffer is full. However, the data in the buffer will be transferred to the transmitter register immediately and TRE will be cleared while TBE is set.

TBE—TRANSMITTER BUFFER EMPTY

TBE indicates the Transmitter Buffer is empty and is ready to accept a character. TBE is set by a chip reset or the transfer of data to the Transmitter Register, and is cleared when a character is written to the transmitter buffer. When TBE is set, an interrupt request is generated on Level 5 if enabled.

RBF—RECEIVER BUFFER FULL

RBF is set when the Receiver Buffer has been loaded with a new character during the sampling of the first stop bit. RBF is cleared by reading the receiver buffer or by a chip reset.

INT—INTERRUPT PENDING

The INT bit reflects the state of the INT Pin (Pin 15) and indicates an interrupt is pending. It is reset by $\overline{INTA}$ or by reading the Interrupt Address Register if only one interrupt is pending and by a chip reset.

FE, OE, PE, RBF, and Break Detect all generate a Level 4 interrupt when the receiver samples the first stop bit. TRE, TBE, and Break-In Detect generate a Level 5 interrupt. TRE generates an interrupt when TBE is set and the Transmitter Register finished transmitting. The Break-In Detect interrupt is issued at the same time as TBE or TRE.

MODIFICATION REGISTER

0	RS4	RS3	RS2	RS1	RS0	TME	DSC

($0F_{16}W$)

DSC—DISABLE START BIT CHECK

DSC disables the receivers start bit check. In this state the receiver will not be reset if RxD is not low at the center of the start bit.

TME—TRANSMISSION MODE ENABLE

TME enables transmission mode and disables framing error detection. For information on transmission mode see the description of the framing error bit in the status register.

RS0, RS1, RS2, RS3, RS4—RECEIVER SAMPLE TIME

The number in RSn alters when the receiver samples RxD. The receiver sample time can be modified only if the receiver is *not* clocked by RxC.

NOTE:
The modification register cannot be read. Reading from address 0FH, 8086: 1EH gates the contents of the status register onto the data bus.

A hardware reset (reset, Pin 12) resets all modification register bits to 0, i.e.:

- The start bit check is enabled.
- Status Register Bit 0 (FE) indicates framing error.
- The sampling time of the serial receiver is the bit center.

A software reset (Command Word 3, RST) does not affect the modification register.

Hardware Reset

A reset signal on pin RESET (HIGH level) forces the device 8256 into a well-defined initial state. This state is characterized as follows:

1) Command registers 1, 2 and 3, mode register, Port 1 control register, and modification register are reset. Thus, all bits of the parallel interface are set to be intputs and event counters/timers are configured as independent 8-bit timers.
2) Status register bits are reset with the exception of bits 4 and 5. Bits 4 and 5 are set indicating that both transmitter register and transmitter buffer register are empty.
3) The interrupt mask, interrupt request, and interrupt service register bits are reset and disable all requests. As a consequence, interrupt signal INT IS INACTIVE (LOW).
4) The transmit data output is set to the marking state (HIGH) and the receiver section is disabled until it is enabled by Command Register 3 Bit 6.
5) The start bit will be checked at sampling time. The receiver will return to start bit search mode if input RxD is not LOW at this time.
6) Status Register Bit 0 implies framing error.
7) The receiver samples input RxD at bit center.

Reset has no effect on the contents of receiver buffer register, transmitter buffer register, the intermediate latches of parallel ports, and event counters/timers, respectively.

RS4	RS3	RS2	RS1	RS0	Point of Time Between Start of Bit and End of Bit Measured in Steps of 1/32 Bit Length
0	1	1	1	1	1 (Start of Bit)
0	1	1	1	0	2
0	1	1	0	1	3
0	1	1	0	0	4
0	1	0	1	1	5
0	1	0	1	0	6
0	1	0	0	1	7
0	1	0	0	0	8
0	0	1	1	1	9
0	0	1	1	0	10
0	0	1	0	1	11
0	0	1	0	0	12
0	0	0	1	1	13
0	0	0	1	0	14
0	0	0	0	1	15
0	0	0	0	0	16 (Bit center)
1	1	1	1	1	17
1	1	1	1	0	18
1	1	1	0	1	19
1	1	1	0	0	20
1	1	0	1	1	21
1	1	0	1	0	22
1	1	0	0	1	23
1	1	0	0	0	24
1	0	1	1	1	25
1	0	1	1	0	26
1	0	1	0	1	27
1	0	1	0	0	28
1	0	0	1	1	29
1	0	0	1	0	30
1	0	0	0	1	31
1	0	0	0	0	32 (End of Bit)

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage On Any Pin
with Respect to Ground.....−0.5V to +7V
Power Dissipation 1W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = +5.0V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.5 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
I_{IL}	Input Leakage		10 −10	μA μA	V_{IN} = V_{CC} V_{IN} = 0V
I_{LO}	Output Leakage		10 −10	μA μA	V_{OUT} = V_{CC} V_{OUT} = 0.45V
I_{CC}	V_{CC} Supply Current		160	mA	
C_{IN}	Input Capacitance		10	pF	f_c = 1 MHz(1)
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured Pins Returned to V_{SS}(1)

NOTE:
1. Sampled, not 100% tested. T_A = 25°C.

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = +5.0V ±10%, GND = 0V

Symbol	Parameter	8256AH		Units
		Min	Max	
BUS PARAMETERS				
t_{LL}	ALE Pulse Width	50		ns
t_{CSL}	$\overline{CS}$ to ALE Setup Time	0		ns
t_{AL}	Address to ALE Setup Time	20		ns
t_{LA}	Address Hold Time after ALE	25		ns
t_{LC}	ALE to $\overline{RD}/\overline{WR}$	20		ns
t_{CC}	$\overline{RD}$, $\overline{WR}$, $\overline{INTA}$ Pulse Width	200		ns
t_{RD}	Data Valid from $\overline{RD}$(1)		120	ns

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = +5.0V ±10%, GND = 0V (Continued)

Symbol	Parameter	8256AH		Units
		Min	Max	
BUS PARAMETERS (Continued)				
t_{DF}	Data Float after $\overline{RD}$ (2)		50	ns
t_{DW}	Data Valid to $\overline{WR}$	150		ns
t_{WD}	Data Valid after $\overline{WR}$	50		ns
t_{CL}	$\overline{RD}/\overline{WR}$ Control to Latch Enable	25		ns
t_{LDR}	ALE to Data Valid		150	ns
t_{RST}	Reset Pulse Width	300		ns
t_{RV}	Recovery Time between $\overline{RD}/\overline{WR}$	500		ns
TIMER/COUNTER PARAMETERS				
t_{CPI}	Counter Input Cycle Time (P12, P13)	2.2		μs
t_{CPWH}	Counter Input Pulse Width High	1.1		μs
t_{CPWL}	Counter Input Pulse Width Low	1.1		μs
t_{TPI}	Counter Input ↑ to INT ↑ at Terminal Count		2.75	μs
t_{TIH}	LOAD Pulse High Time Counter 5	1.1		μs
t_{TIL}	LOAD Pulse Low Time Counter 5	1.1		μs
t_{PP}	Counter 5 Load before Next Clock Pulse on P13	1.1		μs
t_{CR}	External Count Clock ↑ to $\overline{RD}$ ↓ to Ensure Clock is Reflected in Count	2.2		μs
t_{RC}	$\overline{RD}$ ↑ to External Count Clock ↑ to Ensure Clock is not Reflected in Count	0		ns
t_{CW}	External Count Clock ↑ to $\overline{WR}$ ↑ to Ensure Count Written is Not Decremented	2.2		μs
t_{WC}	$\overline{WR}$ ↑ to External Count Clock to Ensure Count Written is Decremented	0		ns
INTERRUPT PARAMETERS				
t_{DEX}	EXTINT ↑ to INT ↑		200	ns
t_{DPI}	Interrupt Request on P17 ↑ to INT ↑		$2t_{CY}+500$	ns
t_{PI}	Pulse Width of Interrupt Request on P17	$t_{CY}+100$		ns
t_{HEA}	$\overline{INTA}$ ↑ or $\overline{RD}$ ↑ to EXTINT ↓	30		ns
t_{HIA}	$\overline{INTA}$ ↑ or $\overline{RD}$ ↑ to INT ↓		300	μs
SERIAL INTERFACE AND CLOCK PARAMETERS				
t_{CY}	Clock Period	195	1000	ns
t_{CLKH}	Clock High Pulse Width	65		ns
t_{CLKL}	Clock Low Pulse Width	65		ns
t_R	Clock Rise Time		20	ns
t_F	Clock Fall Time		20	ns

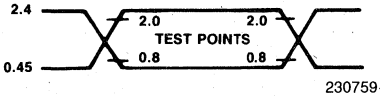
A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = +5.0V ±10%, GND = 0V (Continued)

Symbol	Parameter	8256AH		Units
		Min	Max	
SERIAL INTERFACE AND CLOCK PARAMETERS (Continued)				
t_{SCY}	Serial Clock Period (4)	975		ns
t_{SPD}	Serial Clock High (4)	350		ns
t_{SPW}	Serial Clock Low (4)	350		ns
t_{STD}	Internal Status Update Delay from Center of Stop Bit (5)		300	ns
t_{DTX}	$\overline{TxC}$ to TxD Data Valid		300	ns
t_{IRBF}	INT Delay from Center of First Stop Bit		$2t_{CY}+500$	ns
t_{ITBE}	INT Delay from Falling Edge of Transmit Clock at End of Start Bit		$2t_{CY}+500$	ns
t_{CTS}	Pulse Width for Single Character Transmission	(6)		
PARALLEL I/O PORT PARAMETERS				
t_{WP}	$\overline{WR}\uparrow$ to P1/P2 Data Valid		0	ns
t_{PR}	P1/P2 Data Stable before $\overline{RD}\downarrow$ (7)	300		ns
t_{RP}	P1/P2 Data Hold Time	50		ns
t_{AK}	$\overline{ACK}$ Pulse Width	150		ns
t_{ST}	Strobe Pulse Width	t_{SIB}		ns
t_{PS}	Data Setup to $\overline{STB}\uparrow$	50		ns
t_{PH}	Data Hold after $\overline{STB}\uparrow$	50		ns
t_{WOB}	$\overline{WR}\uparrow$ to $\overline{OBF}\uparrow$		250	ns
t_{AOB}	$\overline{ACK}\downarrow$ to $\overline{OBF}\downarrow$		250	ns
t_{SIB}	$\overline{STB}\downarrow$ to $\overline{IBF}\downarrow$		250	ns
t_{RI}	$\overline{RD}\uparrow$ to $\overline{IBF}\uparrow$		250	ns
t_{SIT}	$\overline{STB}\uparrow$ to $\overline{INT}\uparrow$		$2t_{CY}$ +500	ns
t_{AIT}	$\overline{ACK}\uparrow$ to $\overline{INT}\uparrow$		$2t_{CY}$ +500	ns
t_{AED}	$\overline{OBF}\downarrow$ to $\overline{ACK}\downarrow$ Delay	0		ns

NOTES:
1. C_L = pF all outputs.
2. Measured from logic "one" or "zero" to 1.5V at C_L = 150 pF.
3. P12, P13 are external clock inputs.
4. Note that RxC may be used as an input only in 1× mode, otherwise it will be an output.
5. The center of the Stop Bit will be the receiver sample time, as programmed by the modification register.
6. 1/16th bit length for 32×, 64×; 100 ns for 1×.
7. To ensure t_{RD} spec is met.

WAVEFORMS

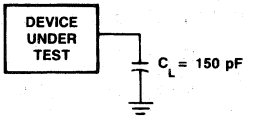
A.C. TESTING INPUT, OUTPUT WAVEFORM



230759-5

A.C. testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

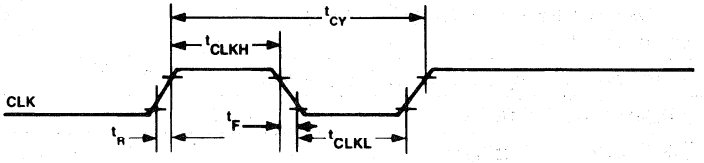
A.C. TESTING LOAD CIRCUIT



230759-6

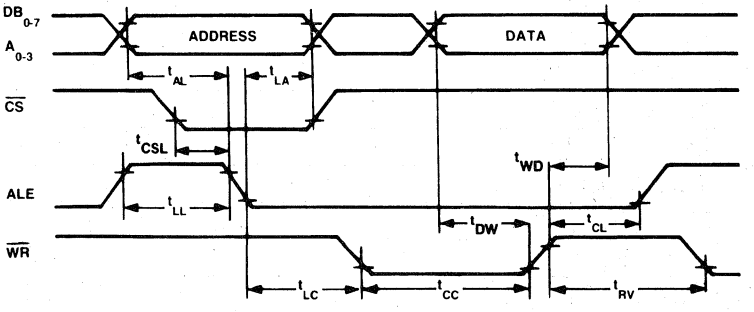
C_L = 150 pF
C_L Includes Jig Capacitance

SYSTEM CLOCK



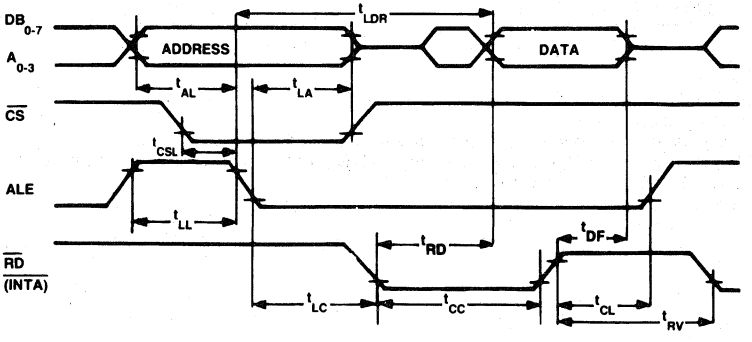
230759-7

WRITE CYCLE



230759-8

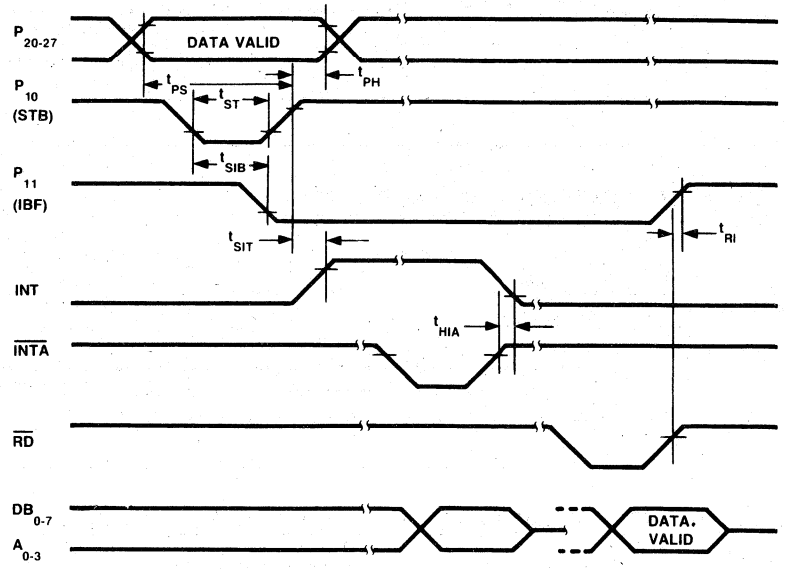
READ CYCLE



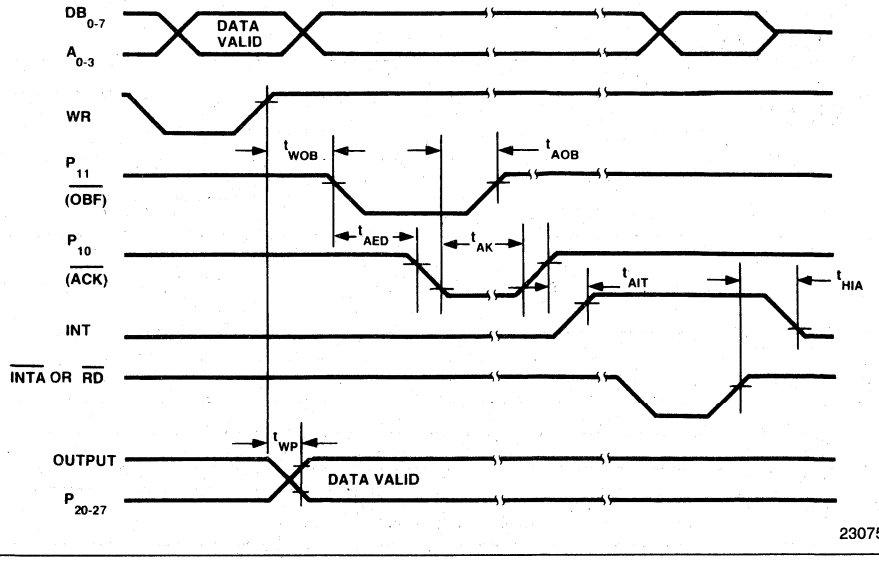
230759-9

WAVEFORMS (Continued)

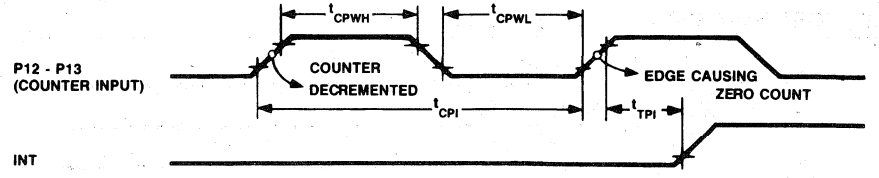
PARALLEL PORT HANDSHAKING—INPUT MODE



PARALLEL PORT HANDSHAKING—OUTPUT MODE

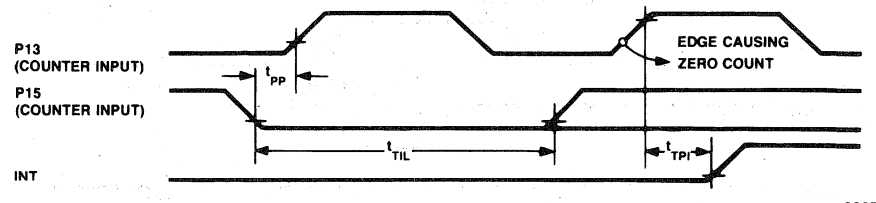


COUNT PULSE TIMINGS



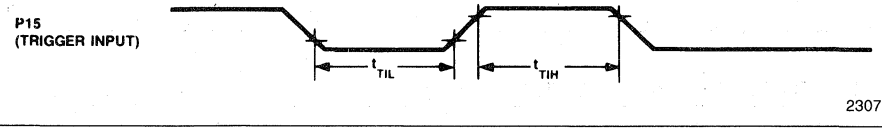
230759-12

LOADING TIMER (OR CASCADED COUNTER/TIMER 3 AND 5)



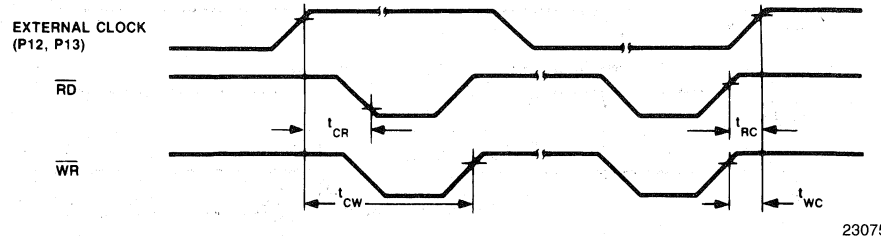
230759-13

TRIGGER PULSE FOR TIMER 5 (CASCADED EVENT COUNTER/TIMER 3 AND 5)



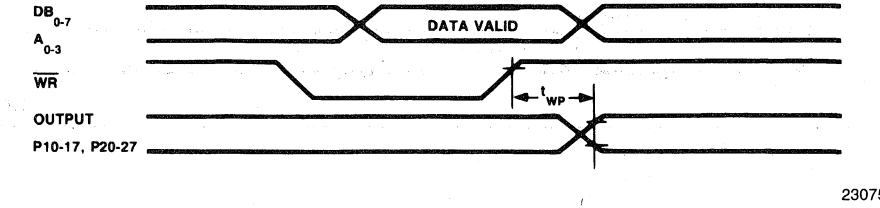
230759-14

COUNTER TIMER TIMING



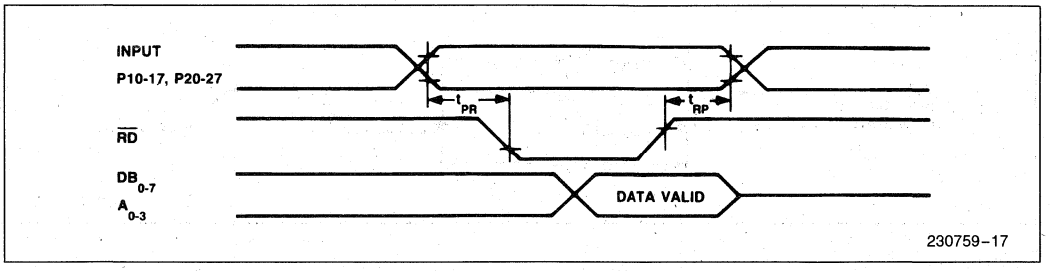
230759-15

OUTPUT FROM PORT 1 AND PORT 2

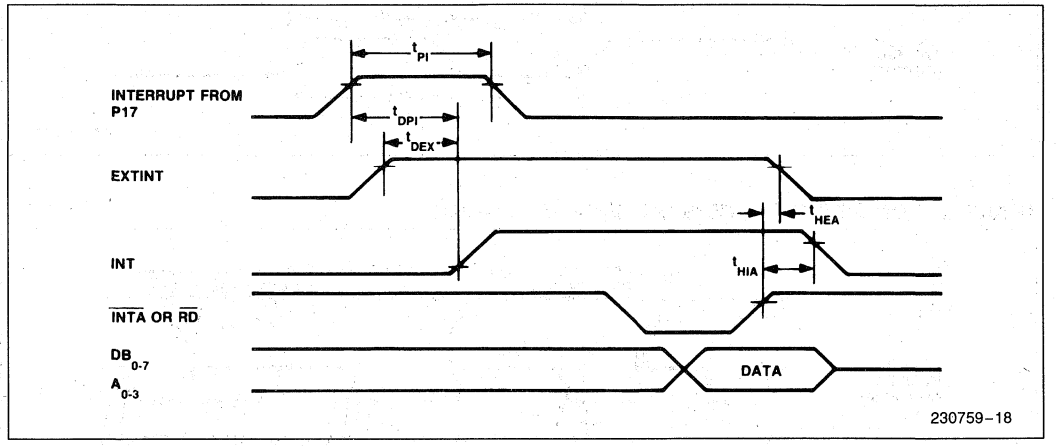


230759-16

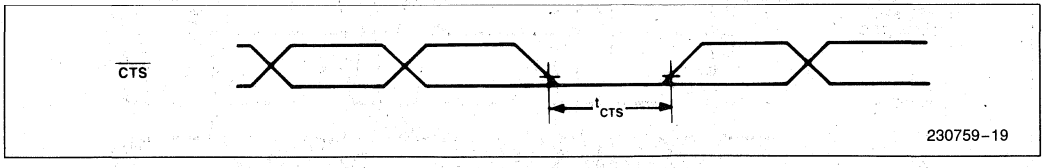
INPUT FROM PORT 1 AND PORT 2



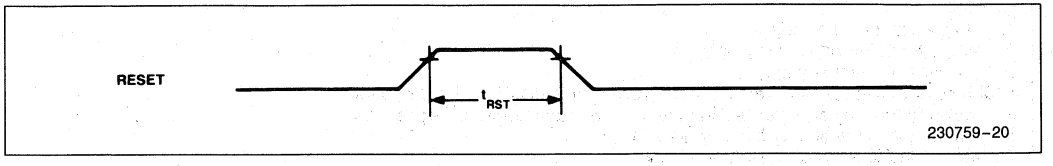
INTERRUPT TIMING



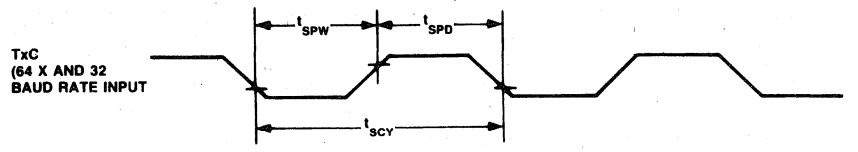
CTS FOR SINGLE CHARACTER TRANSMISSION



RESET TIMING

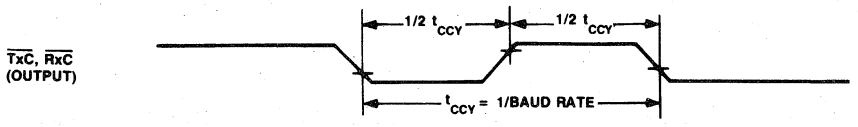


EXTERNAL BAUD RATE CLOCK FOR SERIAL INTERFACE



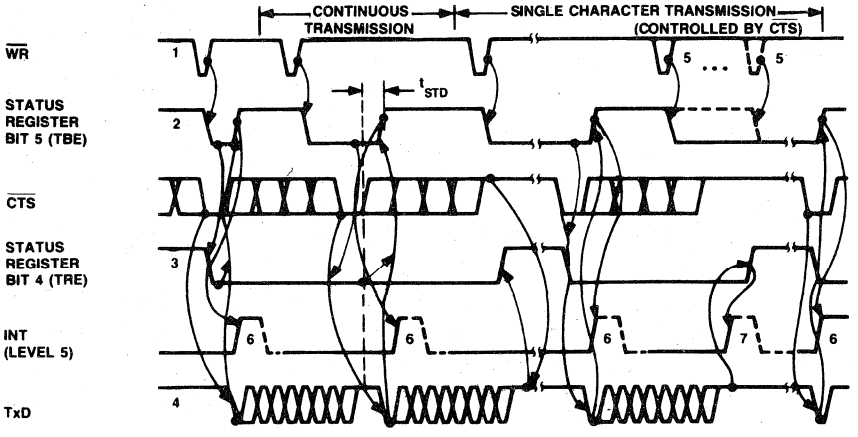
230759-21

TRANSMITTER AND RECEIVER CLOCK FROM INTERNAL CLOCK SOURCE



230759-22

TRANSMISSION OF CHARACTERS ON SERIAL INTERFACE

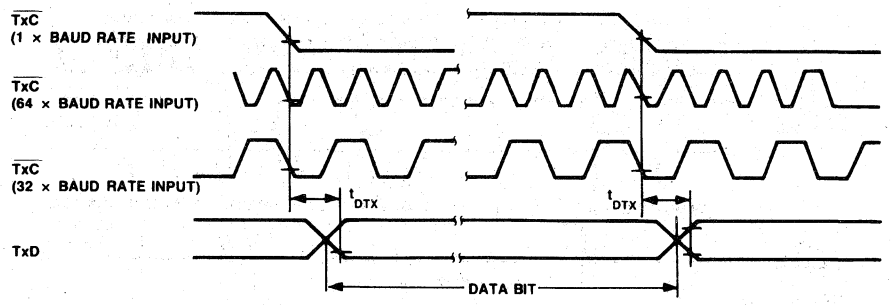


230759-23

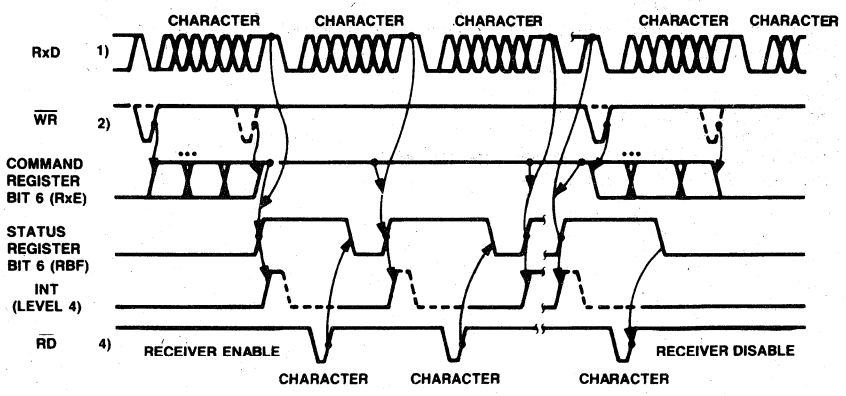
NOTES:
1. Load transmitter buffer register.
2. Transmitter buffer register is empty.
3. Transmitter register is empty.
4. Character format for this example: 7 Data Bits with Parity Bit and 2 Stop Bits.
5. Loading of transmitter buffer register must be complete before $\overline{CTS}$ goes low.
6. Interrupt due to transmitter buffer register empty.
7. Interrupt due to transmitter register empty.

No status bits are altered when $\overline{RD}$ is active.

DATA BIT OUTPUT ON SERIAL INTERFACE

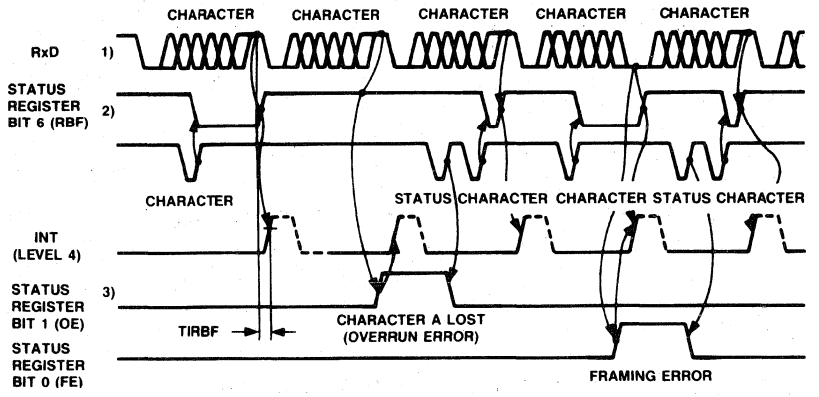


CONTINUOUS RECEPTION OF CHARACTERS ON SERIAL INTERFACE WITHOUT ERROR CONDITION



NOTES:
1. Character format for this example: 6 data bits with parity bit and one stop bit.
2. Set or reset bit 6 of command register 3 (enable receiver).
3. Receiver buffer located.
4. Read receiver buffer register.

ERROR CONDITIONS DURING RECEPTION OF CHARACTERS ON THE SERIAL INTERFACE



230759-26

NOTES:
1. Character format for this example: 6 data bits without parity and one stop bit.
2. Receiver buffer register loaded.
3. Overrun error.
4. Framing error.
5. Interrupt from receiver buffer register loading.
6. Interrupt from overrun error.
7. Interrupt from framing error and loading receiver buffer register.

No status bits are altered when $\overline{RD}$ is active.

8259A
PROGRAMMABLE INTERRUPT CONTROLLER
(8259A/8259A-2)

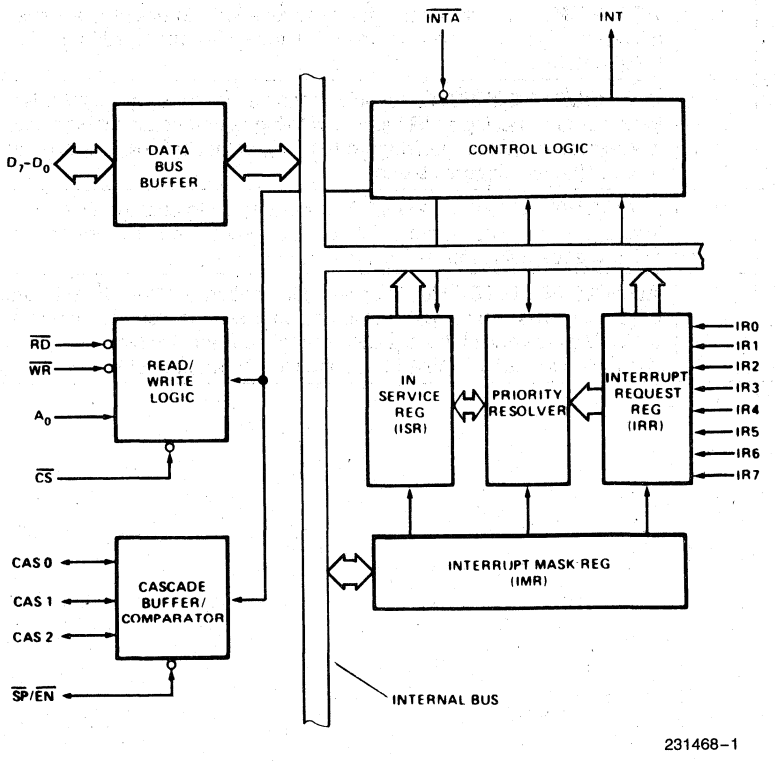
- **8086, 8088 Compatible**
- **MCS-80®, MCS-85® Compatible**
- **Eight-Level Priority Controller**
- **Expandable to 64 Levels**
- **Programmable Interrupt Modes**
- **Individual Request Mask Capability**
- **Single +5V Supply (No Clocks)**
- **Available in 28-Pin DIP and 28-Lead PLCC Package**
  (See Packaging Spec., Order #231369)
- **Available in EXPRESS**
  — Standard Temperature Range
  — Extended Temperature Range

The Intel 8259A Programmable Interrupt Controller handles up to eight vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts without additional circuitry. It is packaged in a 28-pin DIP, uses NMOS technology and requires a single +5V supply. Circuitry is static, requiring no clock input.

The 8259A is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirements.

The 8259A is fully upward compatible with the Intel 8259. Software originally written for the 8259 will operate the 8259A in all 8259 equivalent modes (MCS-80/85, Non-Buffered, Edge Triggered).

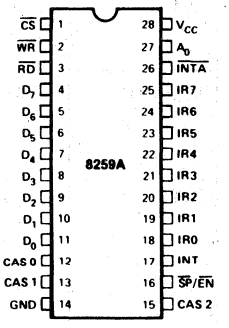
3



231468-1

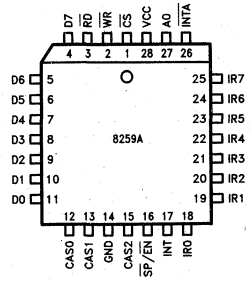
Figure 1. Block Diagram

DIP



231468-2

PLCC



231468-31

Figure 2. Pin Configurations

December 1988
Order Number: 231468-003

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
V_{CC}	28	I	**SUPPLY:** +5V Supply.
GND	14	I	**GROUND**
$\overline{CS}$	1	I	**CHIP SELECT:** A low on this pin enables $\overline{RD}$ and $\overline{WR}$ communication between the CPU and the 8259A. INTA functions are independent of CS.
$\overline{WR}$	2	I	**WRITE:** A low on this pin when CS is low enables the 8259A to accept command words from the CPU.
$\overline{RD}$	3	I	**READ:** A low on this pin when CS is low enables the 8259A to release status onto the data bus for the CPU.
D_7-D_0	4–11	I/O	**BIDIRECTIONAL DATA BUS:** Control, status and interrupt-vector information is transferred via this bus.
CAS_0-CAS_2	12, 13, 15	I/O	**CASCADE LINES:** The CAS lines form a private 8259A bus to control a multiple 8259A structure. These pins are outputs for a master 8259A and inputs for a slave 8259A.
$\overline{SP}/\overline{EN}$	16	I/O	**SLAVE PROGRAM/ENABLE BUFFER:** This is a dual function pin. When in the Buffered Mode it can be used as an output to control buffer transceivers (EN). When not in the buffered mode it is used as an input to designate a master (SP = 1) or slave (SP = 0).
INT	17	O	**INTERRUPT:** This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, thus it is connected to the CPU's interrupt pin.
IR_0-IR_7	18–25	I	**INTERRUPT REQUESTS:** Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode).
$\overline{INTA}$	26	I	**INTERRUPT ACKNOWLEDGE:** This pin is used to enable 8259A interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
A_0	27	I	**A0 ADDRESS LINE:** This pin acts in conjunction with the $\overline{CS}$, $\overline{WR}$, and $\overline{RD}$ pins. It is used by the 8259A to decipher various Command Words the CPU writes and status the CPU wishes to read. It is typically connected to the CPU A0 address line (A1 for 8086, 8088).

FUNCTIONAL DESCRIPTION

Interrupts in Microcomputer Systems

Microcomputer system design requires that I.O devices such as keyboards, displays, sensors and other components receive servicing in a an efficient manner so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

The most common method of servicing such devices is the *Polled* approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuous polling cycle and that such a method would have a serious detrimental effect on system throughput, thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desirable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, however, the processor would resume exactly where it left off.

This method is called *Interrupt*. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PIC, after issuing an Interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. This "pointer" is an address in a vectoring table and will often be referred to, in this document, as vectoring data.

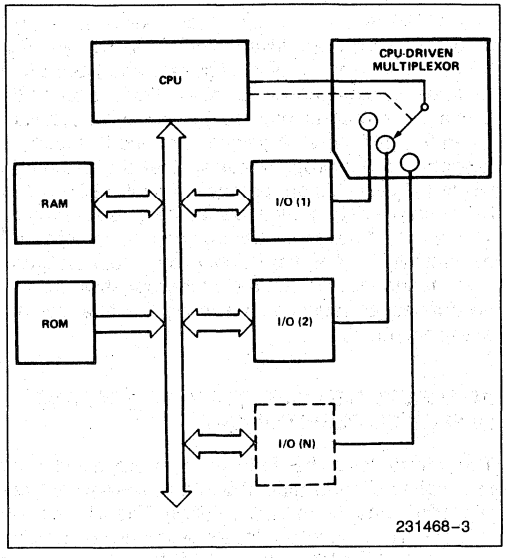


Figure 3a. Polled Method

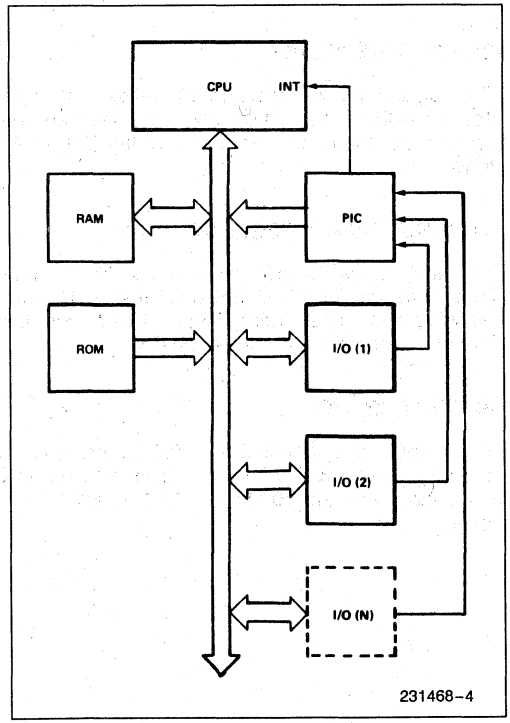


Figure 3b. Interrupt Method

The 8259A is a device specifically designed for use in real time, interrupt driven microcomputer systems. It manages eight levels or requests and has built-in features for expandability to other 8259A's (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 8259A can be configured to match his system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined as required, based on the total system environment.

INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service (ISR). The IRR is used to store all the interrupt levels which are requesting service; and the ISR is used to store all the interrupt levels which are being serviced.

PRIORITY RESOLVER

This logic block determines the priorites of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during $\overline{INTA}$ pulse.

INTERRUPT MASK REGISTER (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower quality.

INT (INTERRUPT)

This output goes directly to the CPU interrupt input. The V_{OH} level on this line is designed to be fully compatible with the 8080A, 8085A and 8086 input levels.

$\overline{INTA}$ (INTERRUPT ACKNOWLEDGE)

$\overline{INTA}$ pulses will cause the 8259A to release vectoring information onto the data bus. The format of this data depends on the system mode (μPM) of the 8259A.

DATA BUS BUFFER

This 3-state, bidirectional 8-bit buffer is used to interface the 8259A to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

READ/WRITE CONTROL LOGIC

The function of this block is to accept OUTput commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 8259A to be transferred onto the Data Bus.

$\overline{CS}$ (CHIP SELECT)

A LOW on this input enables the 8259A. No reading or writing of the chip will occur unless the device is selected.

$\overline{WR}$ (WRITE)

A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 8259A.

$\overline{RD}$ (READ)

A LOW on this input enables the 8259A to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.

A_0

This input signal is used in conjunction with $\overline{WR}$ and $\overline{RD}$ signals to write commands into the various command registers, as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

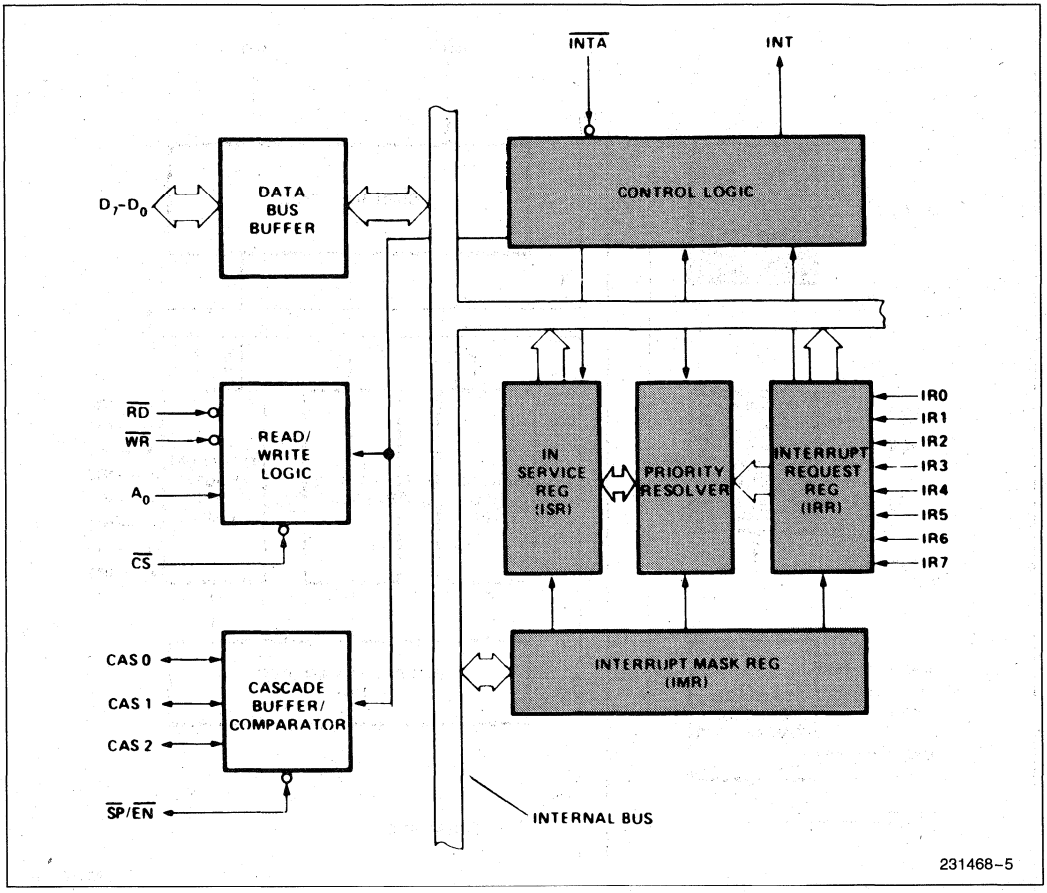


Figure 4a. 8259A Block Diagram

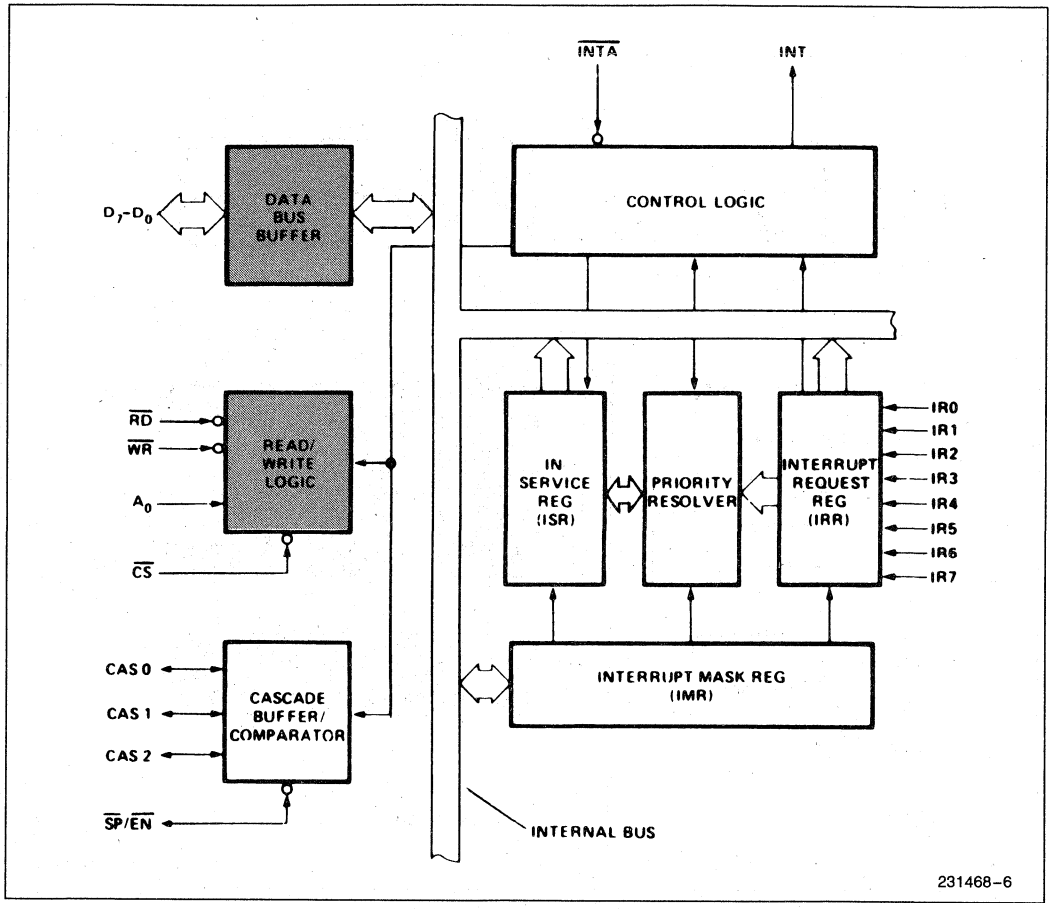


Figure 4b. 8259A Block Diagram

THE CASCADE BUFFER/COMPARATOR

This function block stores and compares the IDs of all 8259A's used in the system. The associated three I/O pins (CAS0-2) are outputs when the 8259A is used as a master and are inputs when the 8259A is used as a slave. As a master, the 8259A sends the ID of the interrupting slave device onto the CAS0–2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive $\overline{\text{INTA}}$ pulses. (See section "Cascading the 8259A".)

INTERRUPT SEQUENCE

The powerful features of the 8259A in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt depends on the type of CPU being used.

The events occur as follows in an MCS-80/85 system:

1. One or more of the INTERRUPT REQUEST lines (IR7–0) are raised high, setting the corresponding IRR bit(s).
2. The 8259A evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an $\overline{\text{INTA}}$ pulse.
4. Upon receiving an $\overline{\text{INTA}}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 8259A will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7–0 pins.
5. This CALL instruction will initiate two more $\overline{\text{INTA}}$ pulses to be sent to the 8259A from the CPU group.
6. These two $\overline{\text{INTA}}$ pulses allow the 8259A to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{\text{INTA}}$ pulse and the higher 8-bit address is released at the second $\overline{\text{INTA}}$ pulse.
7. This completes the 3-byte CALL instruction released by the 8259A. In the AEOI mode the ISR bit is reset at the end of the third $\overline{\text{INTA}}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

The events occuring in an 8086 system are the same until step 4.

4. Upon receiving an $\overline{\text{INTA}}$ from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive the Data Bus during this cycle.
5. The 8086 will initiate a second $\overline{\text{INTA}}$ pulse. During this pulse, the 8259A releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second $\overline{\text{INTA}}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present at step 4 of either sequence (i.e., the request was too short in duration) the 8259A will issue an interrupt level 7. Both the vectoring bytes and the CAS lines will look like an interrupt level 7 was requested.

When the 8259A PIC receives an interrupt, INT becomes active and an interrupt acknowledge cycle is started. If a higher priority interrupt occurs between the two INTA pulses, the INT line goes inactive immediately after the second INTA pulse. After an unspecified amount of time the INT line is activated again to signify the higher priority interrupt waiting for service. This inactive time is not specified and can vary between parts. The designer should be aware of this consideration when designing a system which uses the 8259A. It is recommended that proper asynchronous design techniques be followed.

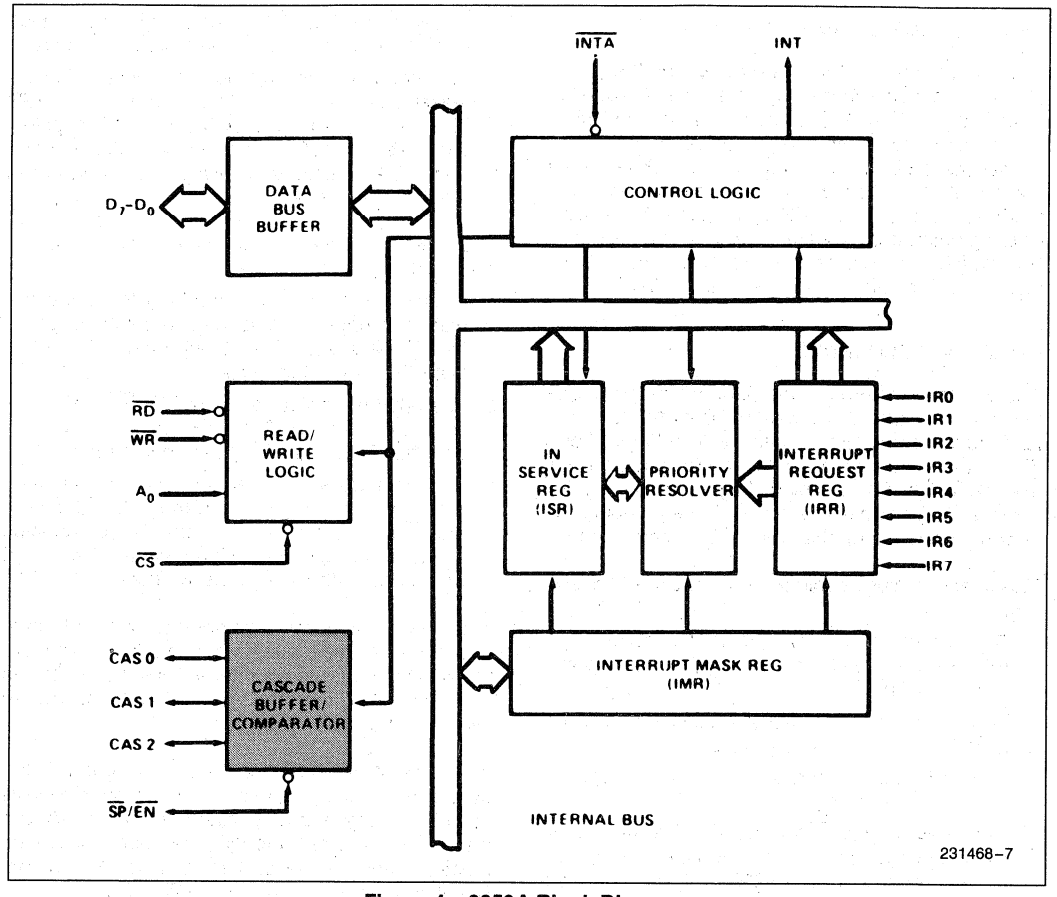


Figure 4c. 8259A Block Diagram

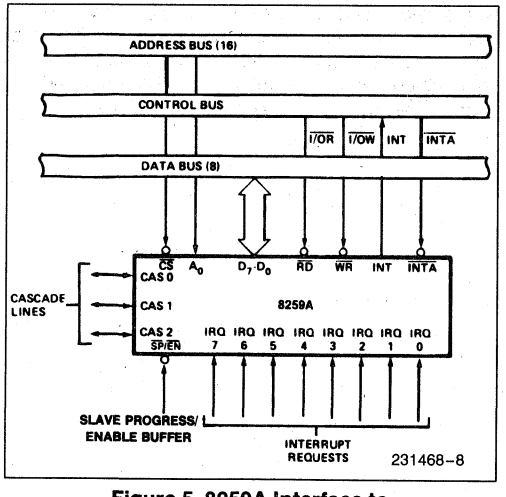


Figure 5. 8259A Interface to Standard System Bus

INTERRUPT SEQUENCE OUTPUTS

MCS-80®, MCS-85®

This sequence is timed by three $\overline{INTA}$ pulses. During the first $\overline{INTA}$ pulse the CALL opcode is enabled onto the data bus.

Content of First Interrupt Vector Byte

	D7	D6	D5	D4	D3	D2	D1	D0
CALL CODE	1	1	0	0	1	1	0	1

During the second $\overline{INTA}$ pulse the lower address of the appropriate service routine is enabled onto the data bus. When Interval = 4 bits A_5–A_7 are programmed, while A_0–A_4 are automatically inserted by the 8259A. When Interval = 8 only A_6 and A_7 are programmed, while A_0–A_5 are automatically inserted.

Content of Second Interrupt Vector Byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

During the third $\overline{INTA}$ pulse the higher address of the appropriate service routine, which was programmed as byte 2 of the initialization sequence (A_8–A_{15}), is enabled onto the bus.

Content of Third Interrupt Vector Byte

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

8086, 8088

8086 mode is similar to MCS-80 mode except that only two Interrupt Acknowledge cycles are issued by the processor and no CALL opcode is sent to the processor. The first interrupt acknowledge cycle is similar to that of MCS-80, 85 systems in that the 8259A uses it to internally freeze the state of the interrupts for priority resolution and as a master it issues the interrupt code on the cascade lines at the end of the $\overline{INTA}$ pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle in 8086 mode the master (or slave if so programmed) will send a byte of data to the processor with the acknowledged interrupt code composed as follows (note the state of the ADI mode control is ignored and A_5–A_{11} are unused in 8086 mode):

Content of Interrupt Vector Byte for 8086 System Mode

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

PROGRAMMING THE 8259A

3

The 8259A accepts two types of command words generated by the CPU:

1. *Initialization Command Words (ICWs):* Before normal operation can begin, each 8259A in the system must be brought to a starting point—by a sequence of 2 to 4 bytes timed by $\overline{WR}$ pulses.
2. *Operation Command Words (OCWs):* These are the command words which command the 8259A to operate in various interrupt modes. These modes are:
   a. Fully nested mode
   b. Rotating priority mode
   c. Special mask mode
   d. Polled mode

The OCWs can be written into the 8259A anytime after initialization.

INITIALIZATION COMMAND WORDS (ICWS)

General

Whenever a command is issued with A0 = 0 and D4 = 1, this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the intiitalization sequence during which the following automatically occur.

a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IR) input must make a low-to-high transistion to generate an interrupt.

b. The Interrupt Mask Register is cleared.

c. IR7 input is assigned priority 7.

d. The slave mode address is set to 7.

e. Special Mask Mode is cleared and Status Read is set to IRR.

f. If IC4 = 0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no Auto-EOI, MCS-80, 85 system).

***NOTE:**
Master/Slave in ICW4 is only used in the buffered mode.

Initialization Command Words 1 and 2 (ICW1, ICW2)

A_5–A_{15}: *Page starting address of service routines.* In an MCS 80/85 system, the 8 request levels will generate CALLs to 8 locations equally spaced in memory. These can be programmed to be spaced at intervals of 4 or 8 memory locations, thus the 8 routines will occupy a page of 32 or 64 bytes, respectively.

The address format is 2 bytes long (A_0–A_{15}). When the routine interval is 4, A_0–A_4 are automatically inserted by the 8259A, while A_5–A_{15} are programmed externally. When the routine interval is 8, A_0–A_5 are automatically inserted by the 8259A, while A_6–A_{15} are programmed externally.

The 8-byte interval will maintain compatibility with current software, while the 4-byte interval is best for a compact jump table.

In an 8086 system A_{15}–A_{11} are inserted in the five most significant bits of the vectoring byte and the 8259A sets the three least significant bits according to the interrupt level. A_{10}–A_5 are ignored and ADI (Address interval) has no effect.

LTIM: If LTIM = 1, then the 8259A will operate in the level interrupt mode. Edge detect logic on the interrupt inputs will be disabled.

ADI: CALL address interval. ADI = 1 then interval = 4; ADI = 0 then interval = 8.

SNGL: Single. Means that this is the only 8259A in the system. If SNGL = 1 no ICW3 will be issued.

IC4: If this bit is set—ICW4 has to be read. If ICW4 is not needed, set IC4 = 0.

Initialization Command Word 3 (ICW3)

This word is read only when there is more than one 8259A in the system and cascading is used, in which case SNGL = 0. It will load the 8-bit slave register. The functions of this register are:

a. In the master mode (either when SP = 1, or in buffered mode when M/S = 1 in ICW4) a "1" is set for each slave in the system. The master then will release byte 1 of the call sequence (for MCS-80/85 system) and will enable the corresponding slave to release bytes 2 and 3 (for 8086 only byte 2) through the cascade lines.

b. In the slave mode (either when $\overline{SP}$ = 0, or if BUF = 1 and M/S = 0 in ICW4) bits 2–0 identify the slave. The slave compares its cascade input with these bits and, if they are equal, bytes 2 and 3 of the call sequence (or just byte 2 for 8086) are released by it on the Data Bus.

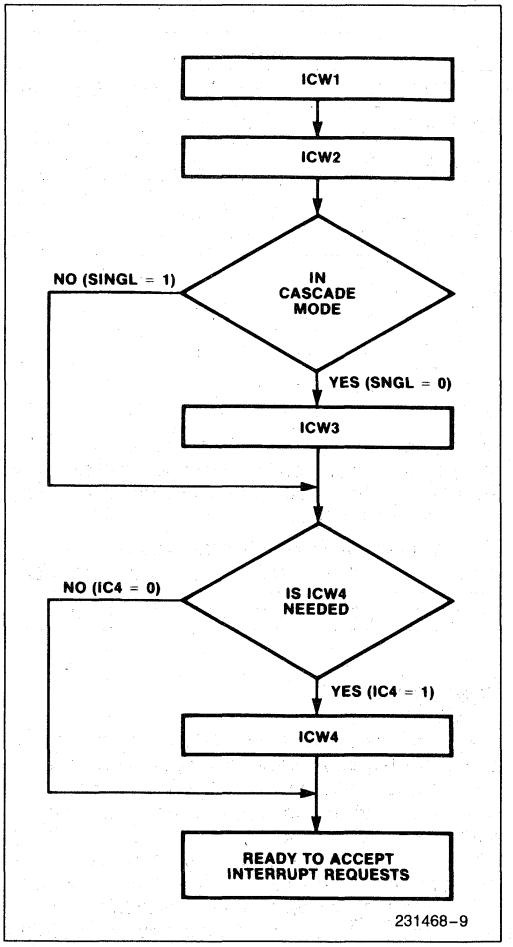


Figure 6. Initialization Sequence

Initialization Command Word 4 (ICW4)

SFNM: If SFNM = 1 the special fully nested mode is programmed.

BUF: If BUF = 1 the buffered mode is programmed. In buffered mode $\overline{SP}/\overline{EN}$ becomes an enable output and the master/slave determination is by M/S.

M/S: If buffered mode is selected: M/S = 1 means the 8259A is programmed to be a master, M/S = 0 means the 8259A is programmed to be a slave. If BUF = 0, M/S has no function.

AEOI: If AEOI = 1 the automatic end of interrupt mode is programmed.

μPM: Microprocessor mode: μPM = 0 sets the 8259A for MCS-80, 85 system operation, μPM = 1 sets the 8259A for 8086 system operation.

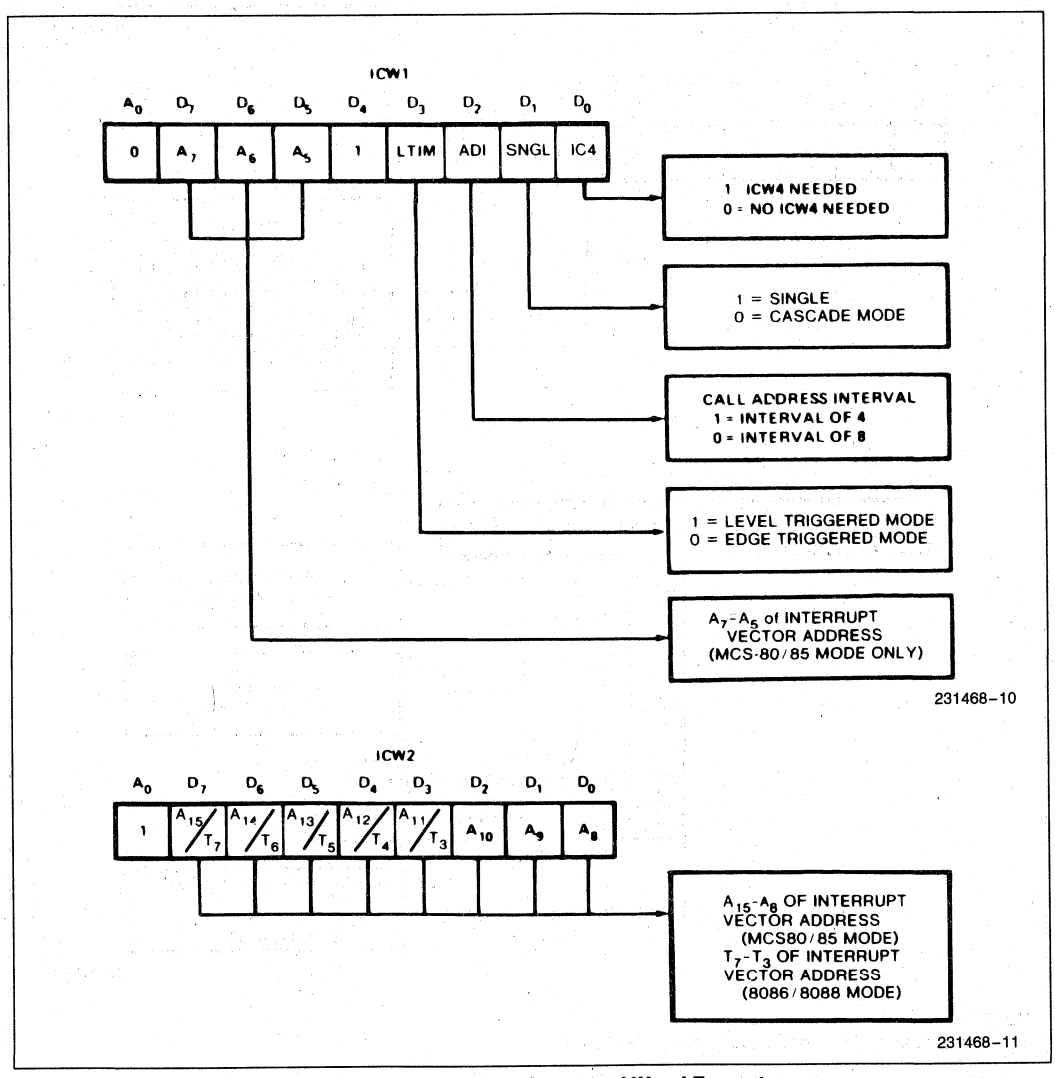
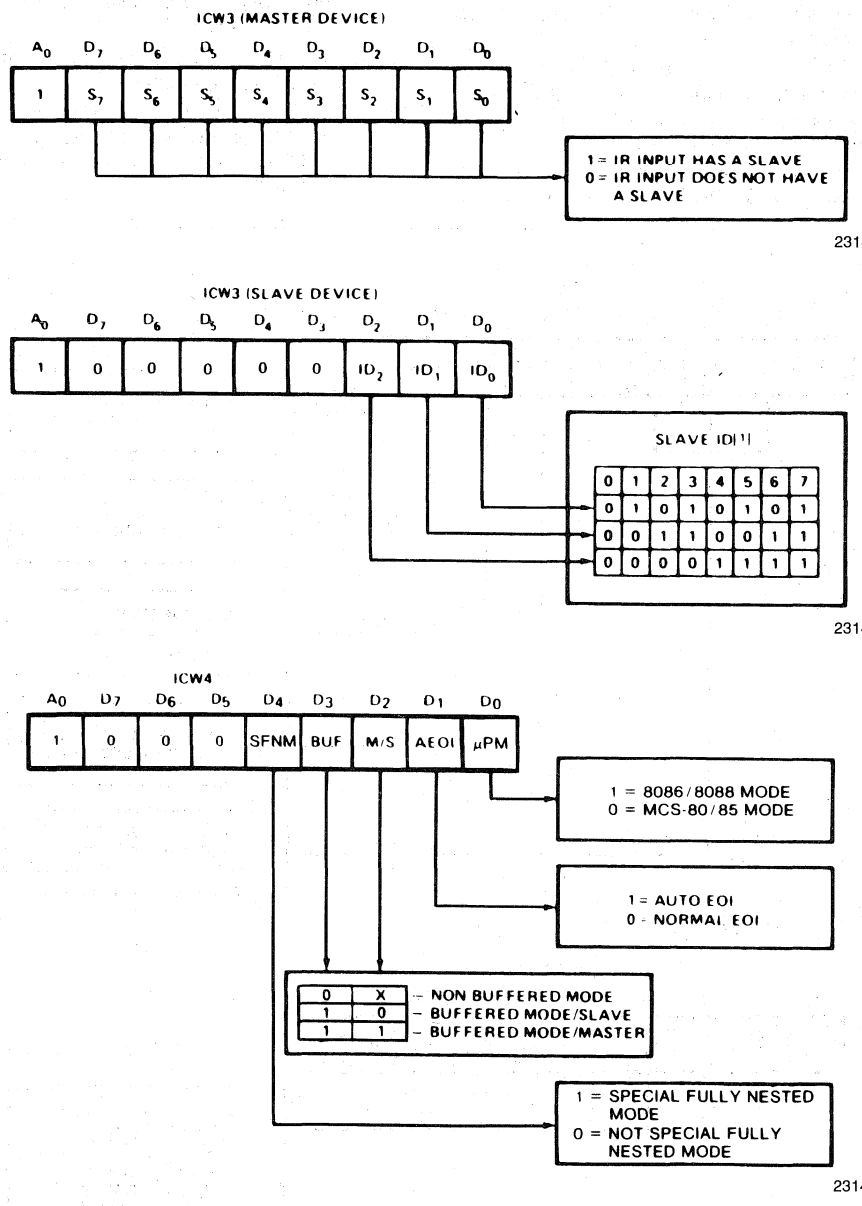


Figure 7. Initialization Command Word Format



NOTE:
Slave ID is equal to the corresponding master IR input.

Figure 7. Initialization Command Word Format (Continued)

OPERATION COMMAND WORDS (OCWS)

After the Initialization Command Words (ICWs) are programmed into the 8259A, the chip is ready to accept interrupt requests at its input lines. However, during the 8259A operation, a selection of algorithms can command the 8259A to operate in various modes through the Operation Command Words (OCWs).

Operation Control Words (OCWs)

OCW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0

OCW2

0	R	SL	EOI	0	0	L2	L1	L0

OCW3

0	0	ESMM	SMM	0	1	P	RR	RIS

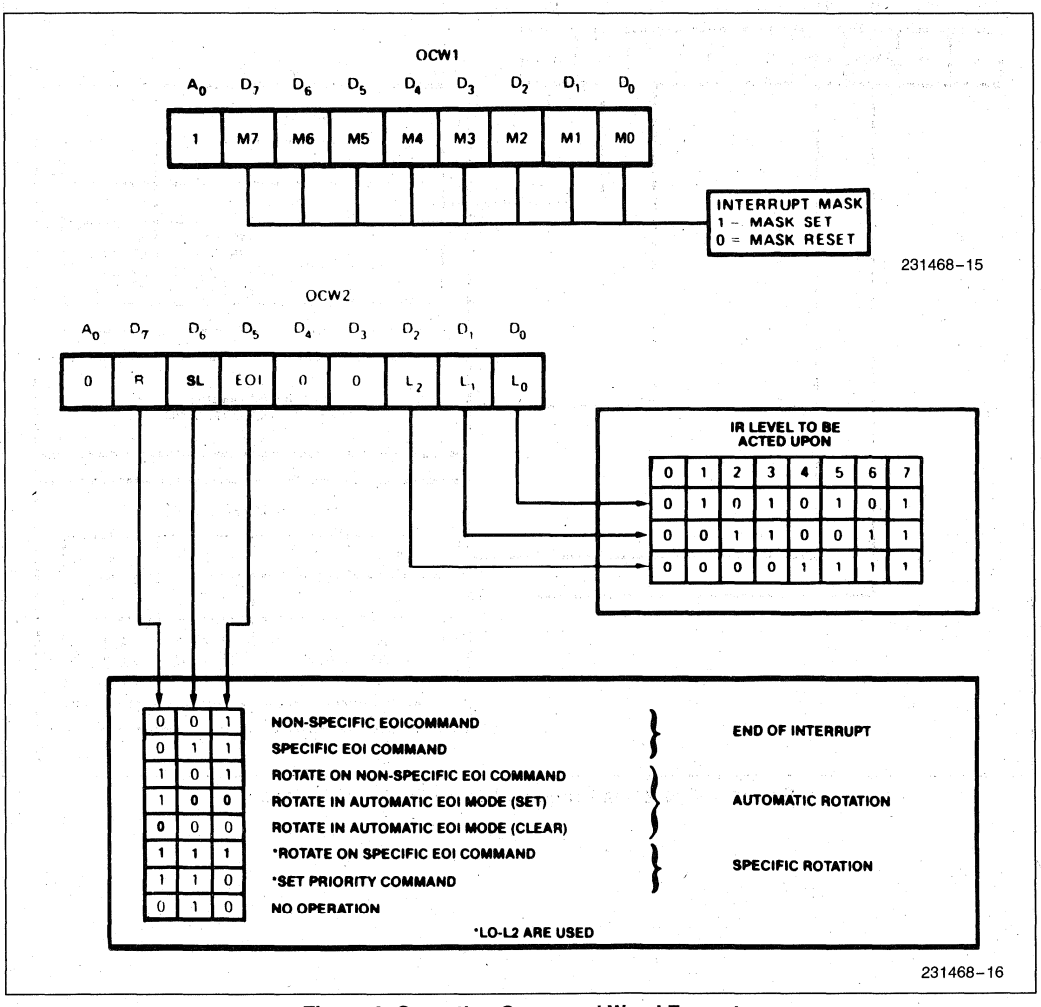


Figure 8. Operation Command Word Format

Operation Control Word 1 (OCW1)

OCW1 sets and clears the mask bits in the interrupt Mask Register (IMR). M_7–M_0 represent the eight mask bits. M = 1 indicates the channel is masked (inhibited), M = 0 indicates the channel is enabled.

Operation Control Word 2 (OCW2)

R, SL, EOI—These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

L_2, L_1, L_0—These bits determine the interrupt level acted upon when the SL bit is active.

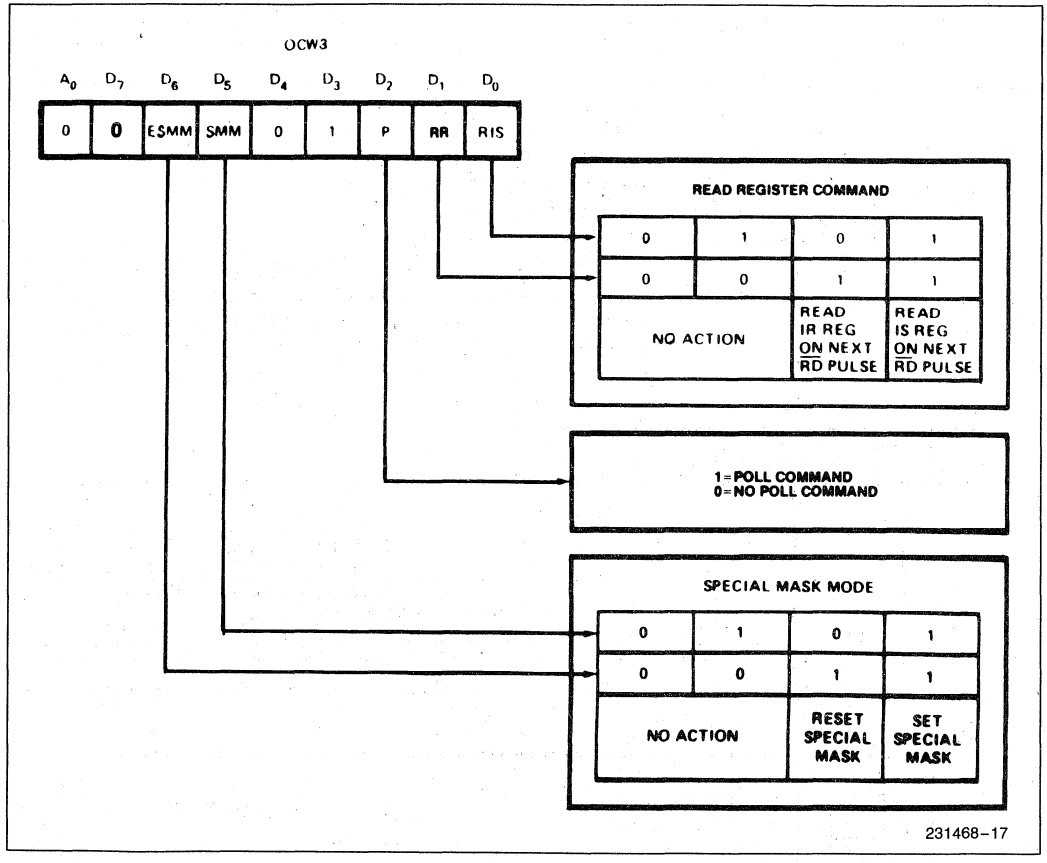


Figure 8. Operation Command Word Format (Continued)

Operation Control Word 3 (OCW3)

ESMM—Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0 the SMM bit becomes a "don't care".

SMM—Special Mask Mode. If ESMM = 1 and SMM = 1 the 8259A will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the 8259A will revert to normal mask mode. When ESMM = 0, SMM has no effect.

Fully Nested Mode

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority from 0 through 7 (0 highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (ISO-7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last INTA. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal Interupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IR0 has the highest prioirity and IR7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

End of Interrupt (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence $\overline{\text{INTA}}$ pulse (when AEOI bit in ICW1 is set) or by a command word that must be issued to the 8259A before returning from a service routine (EOI command). An EOI command must be issued twice if in the Cascade mode, once for the master and once for the corresponding slave.

There are two forms of EOI command: Specific and Non-Specific. When the 8259A is operated in modes which perserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued the 8259A will automatically reset the highest IS bit of those that are set, since in the fully nested mode the highest IS level was necessarily the last level acknowledged and serviced. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).

When a mode is used which may disturb the fully nested structure, the 8259A may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt must be issued which includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and L0–L2 is the binary level of the IS bit to be reset).

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the 8259A is in the Special Mask Mode.

Automatic End of Interrupt (AEOI) Mode

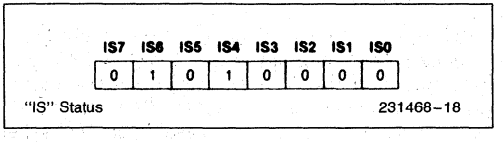
If AEOI = 1 in ICW4, then the 8259A will operate in AEOI mode continuously until reprogrammed by ICW4. in this mode the 8259A will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in MCS-80/85, second in 8086). Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single 8259A.

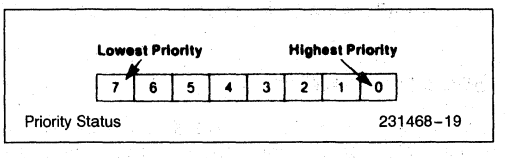
The AEOI mode can only be used in a master 8259A and not a slave. 8259As with a copyright date of 1985 or later will operate in the AEOI mode as a master or a slave.

Automatic Rotation (Equal Priority Devices)

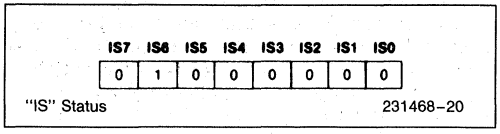
In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most *once*. For example, if the priority and "in service" status is:

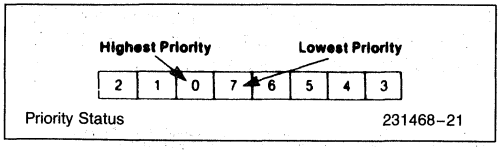
Before Rotate (IR4 the highest prioirity requiring service)





After Rotate (IR4 was serviced, all other priorities rotated correspondingly)





There are two ways to accomplish Automatic Rotation using OCW2, the Rotation on Non-Specific EOI Command (R = 1, SL = 0, EOI = 1) and the Rotate in Automatic EOI Mode which is set by (R = 1, SL = 0, EOI = 0) and cleared by (R = 0, SL = 0, EOI = 0).

Specific Rotation (Specific Priority)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

The Set Priority command is issued in OCW2 where: R = 1, SL = 1, L0–L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command by using the Rotate on Specific EOI command in OCW2 (R = 1, SL = 1, EOI = 1 and LO–L2 = IR level to receive bottom priority).

Interrupt Masks

Each Interrupt Request input can bem masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels operation.

Special Mask Mode

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the 8259A would have inhibited all lower priority requests with no easy way for the routine to enable them.

That is where the Special Mask Mode comes in. In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level *and enables* interrupts from *all other* levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectively enabled by loading the mask register.

The special Mask Mode is set by OWC3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

Poll Command

In Poll mode the INT output functions as it normally does. The microprocessor should ignore this output. This can be accomplished either by not connecting the INT output or by masking interrupts within the microprocessor, thereby disabling its interrupt input. Service to devices is achieved by software using a Poll command.

The Poll command is issued by setting P = '1" in OCW3. The 8259A treats the next $\overline{RD}$ pulse to the 8259A (i.e., $\overline{RD}$ = 0, $\overline{CS}$ = 0) as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from $\overline{WR}$ to $\overline{RD}$.

The word enabled onto the data bus during $\overline{RD}$ is:

D7	D6	D5	D4	D3	D2	D1	D0
I	—	—	—	—	W2	W1	W0

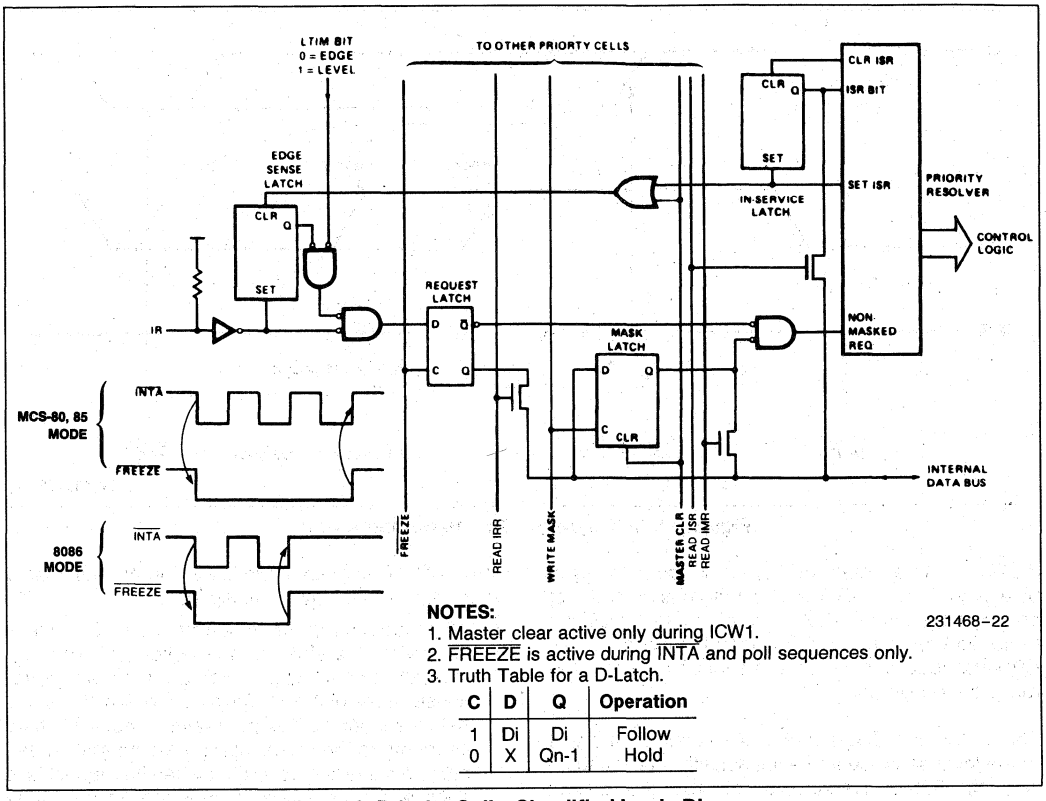
W0–W2: Binary code of the highest priority level requesting service.

I: Equal to "1" if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the $\overline{INTA}$ sequence is not needed (saves ROM space). Another application is to use the poll mode to expand the number of priority levels to more than 64.

Reading the 8259A Status

The input status of several internal registers can be read to update the user information on the system.



NOTES:
1. Master clear active only during ICW1.
2. $\overline{\text{FREEZE}}$ is active during $\overline{\text{INTA}}$ and poll sequences only.
3. Truth Table for a D-Latch.

231468-22

C	D	Q	Operation
1	Di	Di	Follow
0	X	Qn-1	Hold

Figure 9. Priority Cell—Simplified Logic Diagram

The following registers can be read via OCW3 (IRR and ISR or OCW1 [IMR]).

Interrupt Request Register (IRR): 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR.)

In-Service Register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt Command is issued.

Interrupt Mask Register: 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 0.)

The ISR can be read, when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 1).

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the 8259A "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.

After initialization the 8259A is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever $\overline{\text{RD}}$ is active and A0 = 1 (OCW1).

Polling overrides status read when P = 1, RR = 1 in OCW3.

Edge and Level Triggered Modes

This mode is programmed using bit 3 in ICW1.

If LTIM = '0', an interrupt request will be recognized by a low to high transition on an IR input. The IR input can remain high without generating another interrupt.

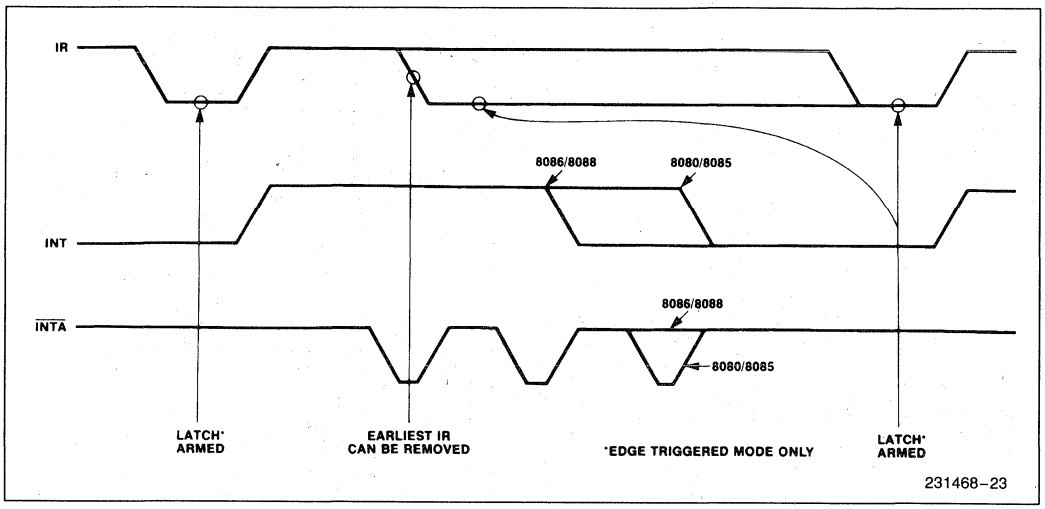


Figure 10. IR Triggering Timing Requirements

If LTIM = '1', an interrupt request will be recognized by a 'high' level on IR Input, and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued or the CPU interrupts is enabled to prevent a second interrupt from occurring.

The priority cell diagram shows a conceptual circuit of the level sensitive and edge sensitive input circuitry of the 8259A. Be sure to note that the request latch is a transparent D type latch.

In both the edge and level triggered modes the IR inputs must remain high until after the falling edge of the first $\overline{INTA}$. If the IR input goes low before this time a DEFAULT IR7 will occur when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IR inputs. To implement this feature the IR7 routine is used for "clean up" simply executing a return instruction, thus ignoring the interrupt. If IR7 is needed for other purposes a default IR7 can still be detected by reading the ISR. A normal IR7 interrupt will set the corresponding ISR bit, a default IR7 won't. If a default IR7 routine occurs during a normal IR7 routine, however, the ISR will remain set. In this case it is necessary to keep track of whether or not the IR7 routine was previously entered. If another IR7 occurs it is a default.

The Special Fully Nest Mode

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

a. When an interrupt request from a certain slave is in service this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IR's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

b. When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

Buffered Mode

When the 8259A is used in a large system where bus driving buffers are required on the data bus and the cascading mode is used, there exists the problem of enabling buffers.

The buffered mode will structure the 8259A to send an enable signal on $\overline{SP}/\overline{EN}$ to enable the buffers. In this mode, whenever the 8259A's data bus outputs are enabled, the $\overline{SP}/\overline{EN}$ output becomes active.

This modification forces the use of software programming to determine whether the 8259A is a master or a slave. Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW4 determines whether it is a master or a slave.

CASCADE MODE

The 8259A can be easily interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels.

The master controls the slaves through the 3 line cascade bus. The cascade bus acts like chip selects to the slaves during the $\overline{INTA}$ sequence.

In a cascade configuration, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will enable the corresponding slave to release the device routine address during bytes 2 and 3 of $\overline{INTA}$. (Byte 2 only for 8086/8088).

The cascade bus lines are normally low and will contain the slave address code from the trailing edge of the first $\overline{INTA}$ pulse to the trailing edge of the third pulse. Each 8259A in the system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: once for the master and once for the corresponding slave. An address decoder is required to activate the Chip Select (CS) input of each 8259A.

The cascade lines of the Master 8259A are activated only for slave inputs, non-slave inputs leave the cascade line inactive (low).

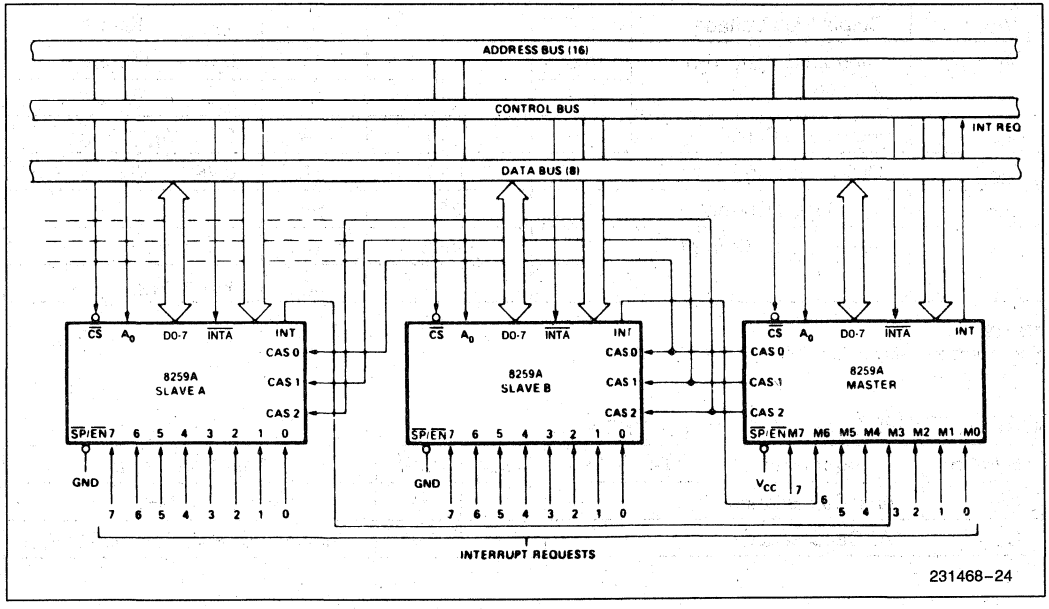


Figure 11. Cascading the 8259A

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . . 0°C to 70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on Any Pin
with Respect to Ground . . . . . −0.5V to +7V
Power Dissipation 1W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ± 10%

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0*	V_{CC} + 0.5V	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
$V_{OH(INT)}$	Interrupt Output High Voltage	3.5		V	I_{OH} = −100 μA
		2.4		V	I_{OH} = −400 μA
I_{LI}	Input Load Current	−10	+10	μA	0V ≤ V_{IN} ≤ V_{CC}
I_{LOL}	Output Leakage Current	−10	+10	μA	0.45V ≤ V_{OUT} ≤ V_{CC}
I_{CC}	V_{CC} Supply Current		85	mA	
I_{LIR}	IR Input Load Current		−300	μA	V_{IN} = 0
			10	μA	V_{IN} = V_{CC}

***NOTE:**
For Extended Temperature EXPRESS V_{IH} = 2.3V.

CAPACITANCE T_A = 25°C; V_{CC} = GND = 0V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	fc = 1 MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured Pins Returned to V_{SS}

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%

TIMING REQUIREMENTS

Symbol	Parameter	8259A		8259A-2		Units	Test Conditions
		Min	Max	Min	Max		
TAHRL	AO/$\overline{CS}$ Setup to $\overline{RD}$/$\overline{INTA}$ ↓	0		0		ns	
TRHAX	AO/$\overline{CS}$ Hold after $\overline{RD}$/$\overline{INTA}$ ↑	0		0		ns	
TRLRH	$\overline{RD}$ Pulse Width	235		160		ns	
TAHWL	AO/$\overline{CS}$ Setup to $\overline{WR}$ ↓	0		0		ns	
TWHAX	AO/$\overline{CS}$ Hold after $\overline{WR}$ ↑	0		0		ns	
TWLWH	$\overline{WR}$ Pulse Width	290		190		ns	
TDVWH	Data Setup to $\overline{WR}$ ↑	240		160		ns	
TWHDX	Data Hold after $\overline{WR}$ ↑	0		0		ns	
TJLJH	Interrupt Request Width (Low)	100		100		ns	See Note 1
TCVIAL	Cascade Setup to Second or Third $\overline{INTA}$ ↓ (Slave Only)	55		40		ns	
TRHRL	End of $\overline{RD}$ to Next $\overline{RD}$ End of $\overline{INTA}$ to Next $\overline{INTA}$ within an $\overline{INTA}$ Sequence Only	160		100		ns	
TWHWL	End of $\overline{WR}$ to Next $\overline{WR}$	190		100		ns	
*TCHCL	End of Command to Next Command (Not Same Command Type)	500		150		ns	
	End of $\overline{INTA}$ Sequence to Next $\overline{INTA}$ Sequence.	500		300			

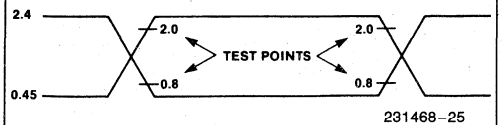
*Worst case timing for TCHCL in an actual microprocessor system is typically much greater than 500 ns (i.e. 8085A = 1.6 μs, 8085A-2 = 1 μs, 8086 = 1 μs, 8086-2 = 625 ns)

NOTE:
This is the low time required to clear the input latch in the edge triggered mode.

TIMING RESPONSES

Symbol	Parameter	8259A		8259A-2		Units	Test Conditions
		Min	Max	Min	Max		
TRLDV	Data Valid from $\overline{RD}$/$\overline{INTA}$ ↓		200		120	ns	C of Data Bus = 100 pF
TRHDZ	Data Float after $\overline{RD}$/$\overline{INTA}$ ↑	10	100	10	85	ns	C of Data Bus Max Test C = 100 pF Min Test C = 15 pF
TJHIH	Interrupt Output Delay		350		300	ns	
TIALCV	Cascade Valid from First $\overline{INTA}$ ↓ (Master Only)		565		360	ns	C_{INT} = 100 pF
TRLEL	Enable Active from $\overline{RD}$ ↓ or $\overline{INTA}$ ↓		125		100	ns	$C_{CASCADE}$ = 100 pF
TRHEH	Enable Inactive from $\overline{RD}$ ↑ or $\overline{INTA}$ ↑		150		150	ns	
TAHDV	Data Valid from Stable Address		200		200	ns	
TCVDV	Cascade Valid to Valid Data		300		200	ns	

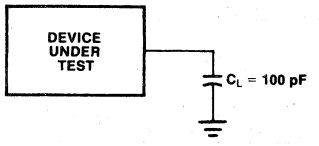
A.C. TESTING INPUT/OUTPUT WAVEFORM



231468-25

A.C. Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".

A.C. TESTING LOAD CIRCUIT

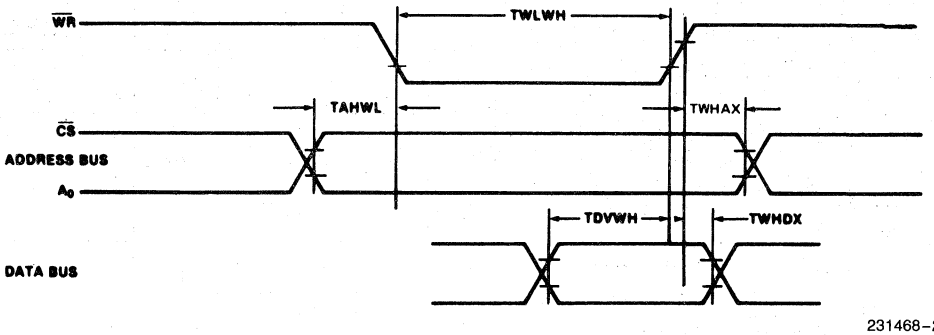


231468-26

C_L = 100 pF
C_L Includes Jig Capacitance

WAVEFORMS

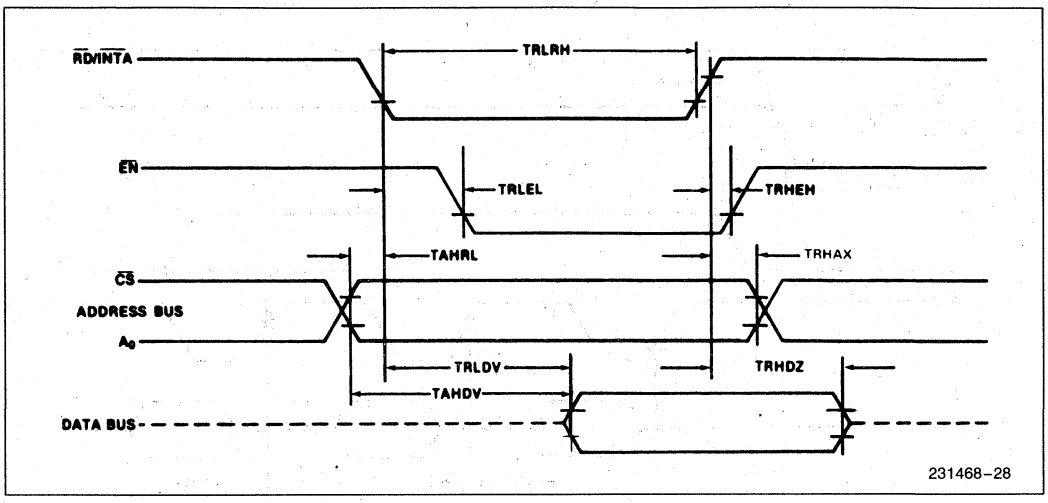
WRITE



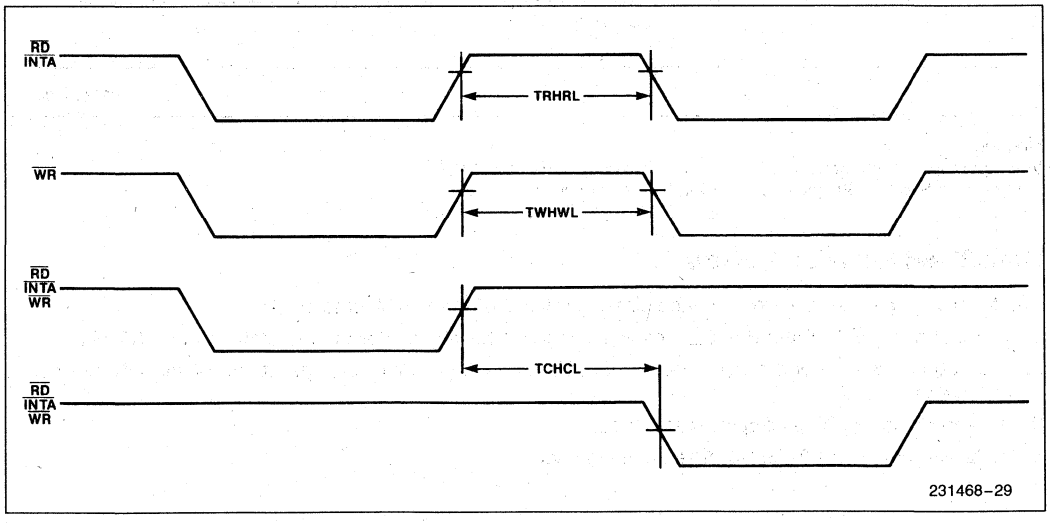
231468-27

WAVEFORMS (Continued)

READ/INTA

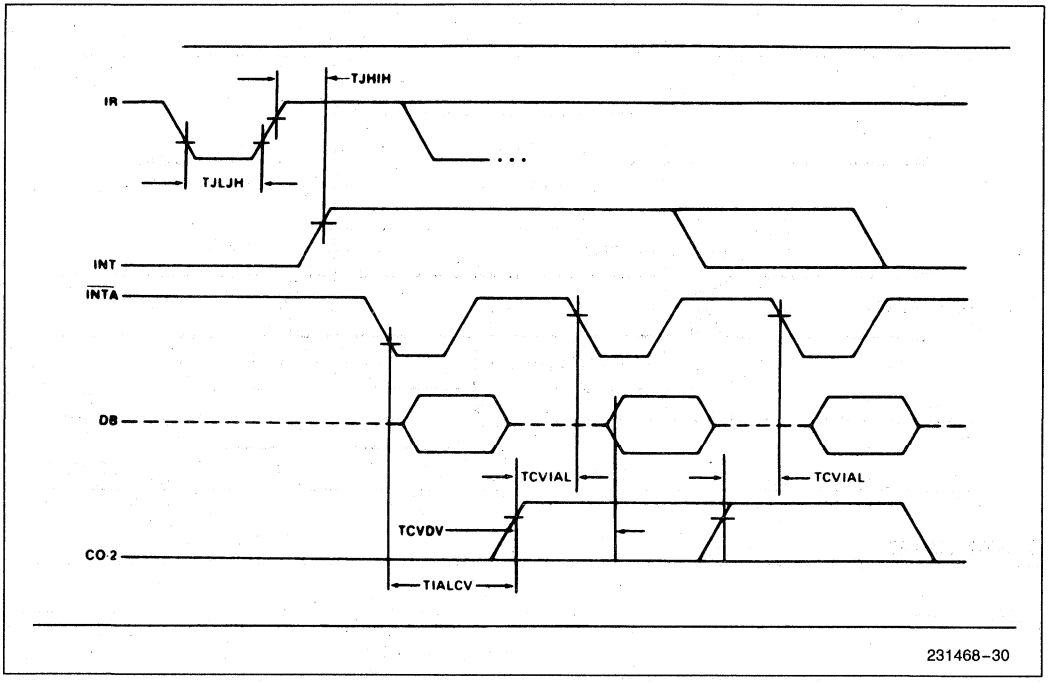


OTHER TIMING



WAVEFORMS (Continued)

$\overline{\text{INTA}}$ SEQUENCE



NOTES:
Interrupt output must remain HIGH at least until leading edge of first $\overline{\text{INTA}}$.
1. Cycle 1 in 8086, 8088 systems, the Data Bus is not active.

Data Sheet Revision Review

The following changes have been made since revision 2 of the 8259A data sheet.

1. The first paragraph of the Poll Command section was rewritten to clarify the status of the INT pin.
2. A paragraph was added to the Interrupt Sequence section to indicate the status of the INT pin during multiple interrupts.
3. A reference to PLCC packaging was added.
4. All references to the 8259A-8 have been deleted.

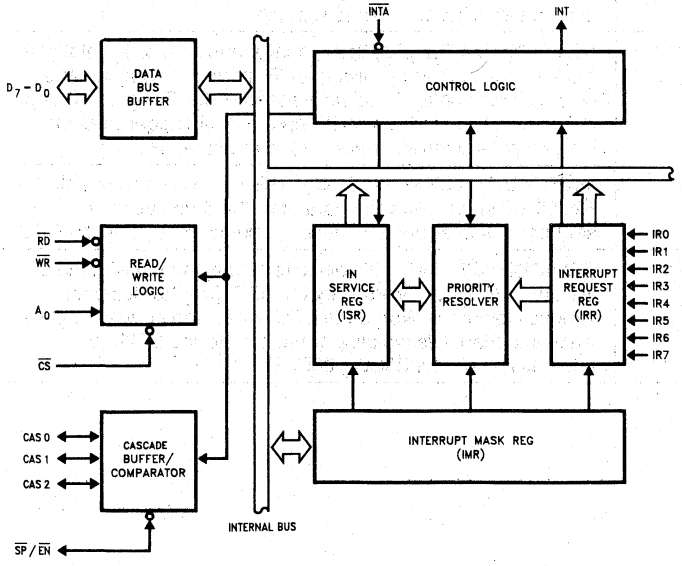
82C59A-2
CHMOS Programmable Interrupt Controller

- **Pin Compatible with NMOS 8259A-2**
- **Eight-Level Priority Controller**
- **Expandable to 64 levels**
- **Programmable Interrupt Modes**
- **Low Standby Power—10 μA**
- **Individual Request Mask Capability**
- **80C86/88 and 8080/85/86/88 Compatible**
- **Fully Static Design**
- **Single 5V Power Supply**
- **Available in 28-Pin Plastic DIP**
  (See Packaging Spec., Order #231369)

The Intel 82C59A-2 is a high performance CHMOS version of the NMOS 8259A-2 Priority Interrupt Controller. The 82C59A-2 is designed to relieve the system CPU from the task of polling in a multi-level priority interrupt system. The high speed and industry standard configuration of the 82C59A-2, make it compatible with microprocessors such as the 80C86/88, 8086/88 and 8080/85.

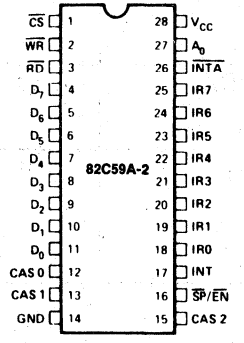
The 82C59A-2 can handle up to 8 vectored priority interrupts for the CPU and is cascadable to 64 without additional circuitry. It is designed to minimize the software and real time overhead in handling multi-level priority interrupts. Two modes of operation make the 82C59A-2 optimal for a variety of system requirements. Static CHMOS circuit design, requiring no clock input, insures low operating power. It is packaged in a 28-pin plastic DIP.

3



231201-1

Figure 1. Block Diagram



231201-2

Figure 2. Pin Configuration

October 1988
Order Number: 231201-004

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
V_{CC}	28	I	**SUPPLY:** +5V Supply.
GND	14	I	**GROUND.**
$\overline{CS}$	1	I	**CHIP SELECT:** A low on this pin enables $\overline{RD}$ and $\overline{WR}$ communication between the CPU and the 82C59A-2. INTA functions are independent of CS.
$\overline{WR}$	2	I	**WRITE:** A low on this pin when $\overline{CS}$ is low enables the 82C59A-2 to accept command words from the CPU.
$\overline{RD}$	3	I	**READ:** A low on this pin when $\overline{CS}$ is low enables the 82C59A-2 to release status onto the data bus for the CPU.
D_7–D_0	4–11	I/O	**BIDIRECTIONAL DATA BUS:** Control, status and interrupt-vector information is transferred via this bus.
CAS_0–CAS_2	12, 13, 15	I/O	**CASCADE LINES:** The CAS lines form a private 82C59A-2 bus to control a multiple 82C59A-2 structure. These pins are outputs for a master 82C59A-2 and inputs for a slave 82C59A-2.
$\overline{SP}/\overline{EN}$	16	I/O	**SLAVE PROGRAM/ENABLE BUFFER:** This is a dual function pin. When in the Buffered Mode it can be used as an output to control buffer transceivers (EN). When not in the buffered mode it is used as an input to designate a master (SP = 1) or slave (SP = 0).
INT	17	O	**INTERRUPT:** This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, thus it is connected to the CPU's interrupt pin.
IR_0–IR_7	18–25	I	**INTERRUPT REQUESTS:** Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode). Internal pull-up resistors are implemented on IR0–7.
$\overline{INTA}$	26	I	**INTERRUPT ACKNOWLEDGE:** This pin is used to enable 82C59A-2 interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
A_0	27	I	**AO ADDRESS LINE:** This pin acts in conjunction with the $\overline{CS}$, $\overline{WR}$, and $\overline{RD}$ pins. It is used by the 82C59A-2 to decipher various Command Words the CPU writes and status the CPU wishes to read. It is typically connected to the CPU A0 address line (A1 for 80C86, 80C88).

FUNCTIONAL DESCRIPTION

Interrupts in Microcomputer Systems

Microcomputer system design requires that I/O devices such as keyboards, displays, sensors and other components receive servicing in an efficient manner so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

The most common method of servicing such devices is the *Polled* approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuous polling cycle and that such a method would have a serious, detrimental effect on system throughput, thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desirable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, however, the processor would resume exactly where it left off.

This method is called *Interrupt*. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PIC, after issuing an Interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. This "pointer" is an address in a vectoring table and will often be referred to, in this document, as vectoring data.

The 82C59A-2

The 82C59A-2 is a device specifically designed for use in real time, interrupt driven microcomputer sys-

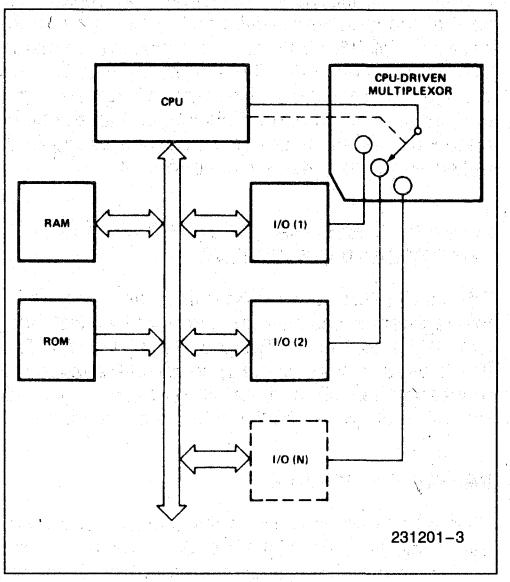


Figure 3a. Polled Method

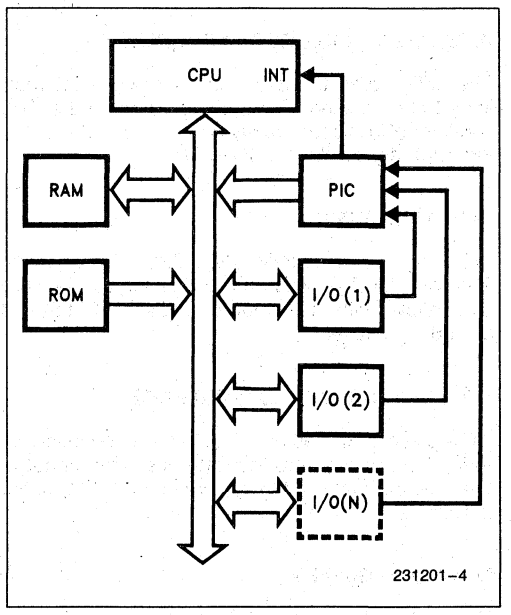


Figure 3b. Interrupt Method

tems. It manages eight levels or requests and has built-in features for expandability to other 82C59A-2's (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 82C59A-2 can be configured to match system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined as required, based on the total system environment.

INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service; and the ISR is used to store all the interrupt levels which are being serviced.

PRIORITY RESOLVER

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during $\overline{INTA}$ pulse.

INTERRUPT MASK REGISTER (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

INT (INTERRUPT)

This output goes directly to the CPU interrupt input. The V_{OH} level on this line is designed to be fully compatible with the 8080A, 8085A, 80C88 and 80C86 input levels.

$\overline{INTA}$ (INTERRUPT ACKNOWLEDGE)

$\overline{INTA}$ pulses will cause the 82C59A-2 to release vectoring information onto the data bus. The format of this data depends on the system mode (μPM) of the 82C59A-2.

DATA BUS BUFFER

This 3-state, bidirectional 8-bit buffer is used to interface the 82C59A-2 to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

READ/WRITE CONTROL LOGIC

The function of this block is to accept OUTput commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 82C59A-2 to be transferred onto the Data Bus.

$\overline{CS}$ (CHIP SELECT)

A LOW on this input enables the 82C59A-2. No reading or writing of the chip will occur unless the device is selected.

$\overline{WR}$ (WRITE)

A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 82C59A-2.

$\overline{RD}$ (READ)

A LOW on this input enables the 82C59A-2 to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.

A_0

This input signal is used in conjunction with $\overline{WR}$ and $\overline{RD}$ signals to write commands into the various command registers, as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

THE CASCADE BUFFER/COMPARATOR

This function block stores and compares the IDs of all 82C59A-2's used in the system. The associated three I/O pins (CAS0–2) are outputs when the 82C59A-2 is used as a master and are inputs when the 82C59A-2 is used as a slave. As a master, the 82C59A-2 sends the ID of the interrupting slave device onto the CAS0–2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive $\overline{INTA}$ pulses. (See section "Cascading the 82C59A-2".)

INTERRUPT SEQUENCE

The powerful features of the 82C59A-2 in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt depends on the type of CPU being used.

The events occur as follows in an MCS-80/85 system:

1. One or more of the INTERRUPT REQUEST Lines (IR7–0) are raised high, setting the corresponding IRR bit(s).

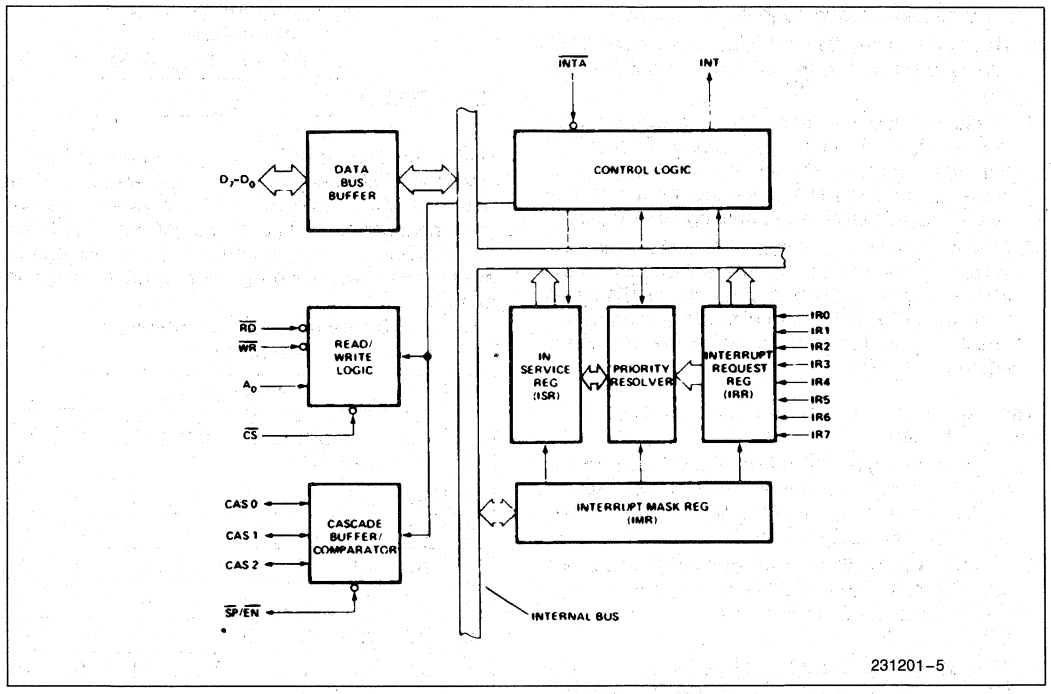


Figure 4. 82C59A-2 Block Diagram

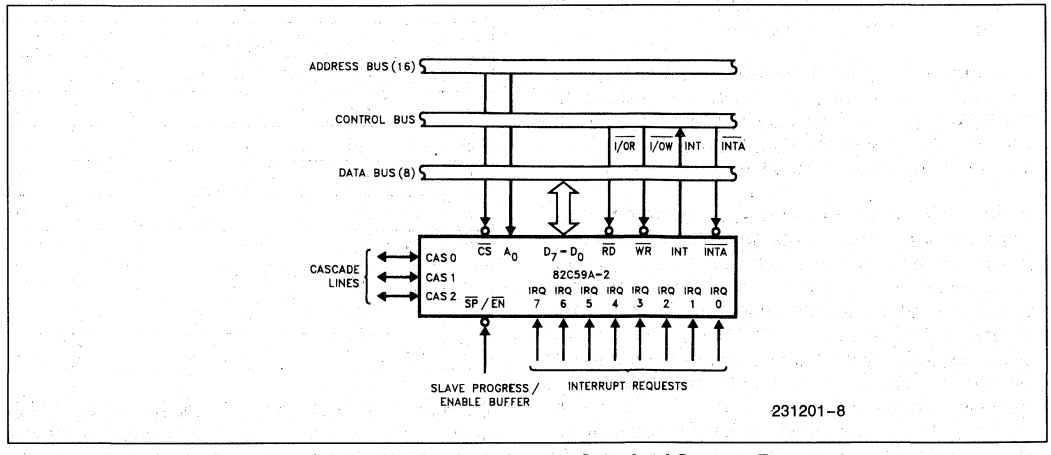


Figure 5. 82C59A-2 Interface to Standard System Bus

2. The 82C59A-2 evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an $\overline{INTA}$ pulse.
4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 82C59A-2 will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7–0 pins.
5. This CALL instruction will initiate two more $\overline{INTA}$ pulses to be sent to the 82C59A-2 from the CPU group.
6. These two $\overline{INTA}$ pulses allow the 82C59A-2 to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{INTA}$ pulse and the higher 8-bit address is released at the second $\overline{INTA}$ pulse.
7. This completes the 3-byte CALL instruction released by the 82C59A-2. In the AEOI mode the ISR bit is reset at the end of the third $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

The events occurring in an 80C86 system are the same until step 4.

4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 82C59A-2 does not drive the Data Bus during this cycle.
5. The 80C86 will initiate a second $\overline{INTA}$ pulse. During this pulse, the 82C59A-2 releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt is present at step 4 of either sequence (i.e., the request was too short in duration) the 82C59A-2 will issue an interrupt level 7. Both the vectoring bytes and the CAS lines will look like an interrupt level 7 was requested.

When the 82C59A-2 PIC receives an interrupt, INT becomes active and an interrupt acknowledge cycle is started. If a higher priority interrupt occurs between the two INTA pulses, the INT line goes inactive immediately after the second INTA pulse. After an unspecified amount of time the INT line is activated again to signify the higher priority interrupt waiting for service. This inactive time is not specified and can vary between parts. The designer should be aware of this consideration when designing a system which uses the 82C59A-2. It is recommended that proper asynchronous design techniques be followed.

INTERRUPT SEQUENCE OUTPUTS

MCS®-80, MCS-85

This sequence is timed by three $\overline{INTA}$ pulses. During the first $\overline{INTA}$ pulse the CALL opcode is enabled onto the data bus.

Content of First Interrupt Vector Byte

	D7	D6	D5	D4	D3	D2	D1	D0
CALL CODE	1	1	0	0	1	1	0	1

During the second $\overline{INTA}$ pulse the lower address of the appropriate service routine is enabled onto the data bus. When Interval = 4 bits A_5–A_7 are programmed, while A_0–A_4 are automatically inserted by the 82C59A-2. When Interval = 8 only A_6 and A_7 are programmed, while A_0–A_5 are automatically inserted.

Content of Second Interrupt Vector Byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

During the third INTA pulse the higher address of the appropriate service routine, which was programmed as byte 2 of the initialization sequence (A_8 – A_{15}), is enabled onto the bus.

Content of Third Interrupt Vector Byte

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

80C86, 80C88

80C86, 80C88 mode is similar to MCS-80 mode except that only two Interrupt Acknowledge cycles are issued by the processor and no CALL opcode is sent to the processor. The first interrupt acknowledge cycle is similar to that of MCS-80, 85 systems in that the 82C59A-2 uses it to internally freeze the state of the interrupts for priority resolution and as a master it issues the interrupt code on the cascade lines at the end of the INTA pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle in 80C86, 80C88 mode the master (or slave if so programmed) will send a byte of data to the processor with the acknowledged interrupt code composed as follows (note the state of the ADI mode control is ignored and A_5–A_{11} are unused in 80C86, 80C88 mode):

Content of Interrupt Vector Byte for 80C86, 80C88 System Mode

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

PROGRAMMING THE 82C59A-2

The 82C59A-2 accepts two types of command words generated by the CPU:

1. *Initialization Command Words (ICWs)*: Before normal operation can begin, each 82C59A-2 in the system must be brought to a starting point — by a sequence of 2 to 4 bytes timed by $\overline{WR}$ pulses.
2. *Operation Command Words (OCWs)*: These are the command words which command the 82C59A-2 to operate in various interrupt modes. These modes are:
   a. Fully nested mode
   b. Rotating priority mode
   c. Special mask mode
   d. Polled mode

The OCWs can be written into the 82C59A-2 anytime after initialization.

INITIALIZATION COMMAND WORDS (ICWS)

GENERAL

Whenever a command is issued with A0 = 0 and D4 = 1, this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the initialization sequence during which the following automatically occur.

a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IR) input must make a low-to-high transition to generate an interrupt.
b. The Interrupt Mask Register is cleared.
c. IR7 input is assigned priority 7.
d. The slave mode address is set to 7.
e. Special Mask Mode is cleared and Status Read is set to IRR.
f. If IC4 = 0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no Auto-EOI, MCS-80, 85 system).

***NOTE:**
Master/Slave in ICW4 is only used in the buffered mode.

INITIALIZATION COMMAND WORDS 1 AND 2 (ICW1, ICW2)

A_5–A_{15}: *Page starting address of service routines.* In an MCS 80/85 system, the 8 request levels will generate CALLs to 8 locations equally spaced in memory. These can be programmed to be spaced at intervals of 4 or 8 memory locations, thus the 8 routines will occupy a page of 32 or 64 bytes, respectively.

The address format is 2 bytes long (A_0–A_{15}). When the routine interval is 4, A_0–A_4 are automatically inserted by the 82C59A-2, while A_5–A_{15} are programmed externally. When the routine interval is 8, A_0–A_5 are automatically inserted by the 82C59A-2, while A_6–A_{15} are programmed externally.

The 8-byte interval will maintain compatibility with current software, while the 4-byte interval is best for a compact jump table.

In an 80C86, 80C88 system A_{15}–A_{11} are inserted in the five most significant bits of the vectoring

byte and the 82C59A-2 sets the three least significant bits according to the interrupt level. $A_{10}-A_5$ are ignored and ADI (Address Interval) has no effect:

LTIM: If LTIM = 1, then the 82C59A-2 will operate in the level interrupt mode. Edge detect logic on the interrupt inputs will be disabled.

ADI: CALL address interval. ADI = 1 then interval = 4; ADI = 0 then interval = 8.

SNGL: Single. Means that this is the only 82C59A-2 in the system. If SNGL = 1 no ICW3 will be issued.

IC4: If this bit is set — ICW4 has to be read. If ICW4 is not needed, set IC4 = 0.

INITIALIZATION COMMAND WORD 3 (ICW3)

This word is read only when there is more than one 82C59A-2 in the system and cascading is used, in which case SNGL = 0. It will load the 8-bit slave register. The functions of this register are:

a. In the master mode (either when SP = 1, or in buffered mode when M/S = 1 in ICW4) a "1" is set for each slave in the system. The master then will release byte 1 of the call sequence (for MCS-80/85 system) and will enable the corresponding slave to release bytes 2 and 3 (for 80C86, 80C88 only byte 2) through the cascade lines.

b. In the slave mode (either when $\overline{SP}$ = 0, or if BUF = 1 and M/S = 0 in ICW4) bits 2-0 identify the slave. The slave compares its cascade input with these bits and, if they are equal, bytes 2 and 3 of the call sequence (or just byte 2 for 80C86, 80C88 are released by it on the Data Bus.

INITIALIZATION COMMAND WORD 4 (ICW4)

SFNM: If SFNM = 1 the special fully nested mode is programmed.

BUF: If BUF = 1 the buffered mode is programmed. In buffered mode $\overline{SP}/\overline{EN}$ becomes an enable output and the master/slave determination is by M/S.

M/S: If buffered mode is selected: M/S = 1 means the 82C59A-2 is programmed to be a master, M/S = 0 means the 82C59A-2 is programmed to be a slave. If BUF = 0, M/S has no function.

AEOI: If AEOI = 1 the automatic end of interrupt mode is programmed.

μPM: Microprocessor mode: μPM = 0 sets the 82C59A-2 for MCS-80, 85 system operation, μPM = 1 sets the 82C59A-2 for 80C86 system operation.

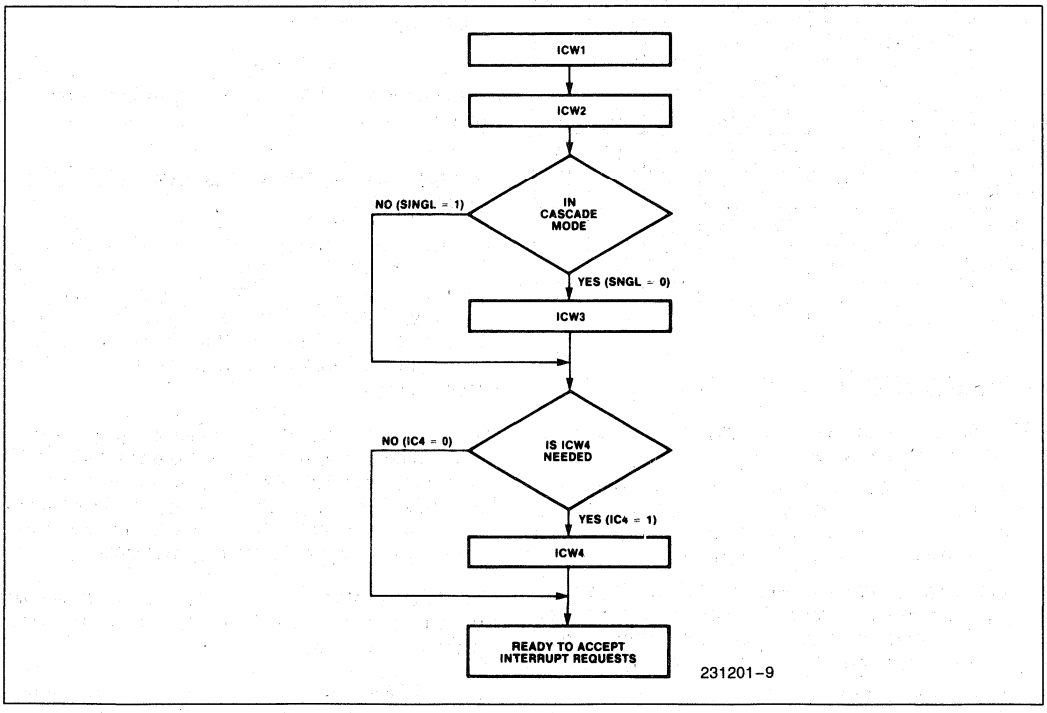
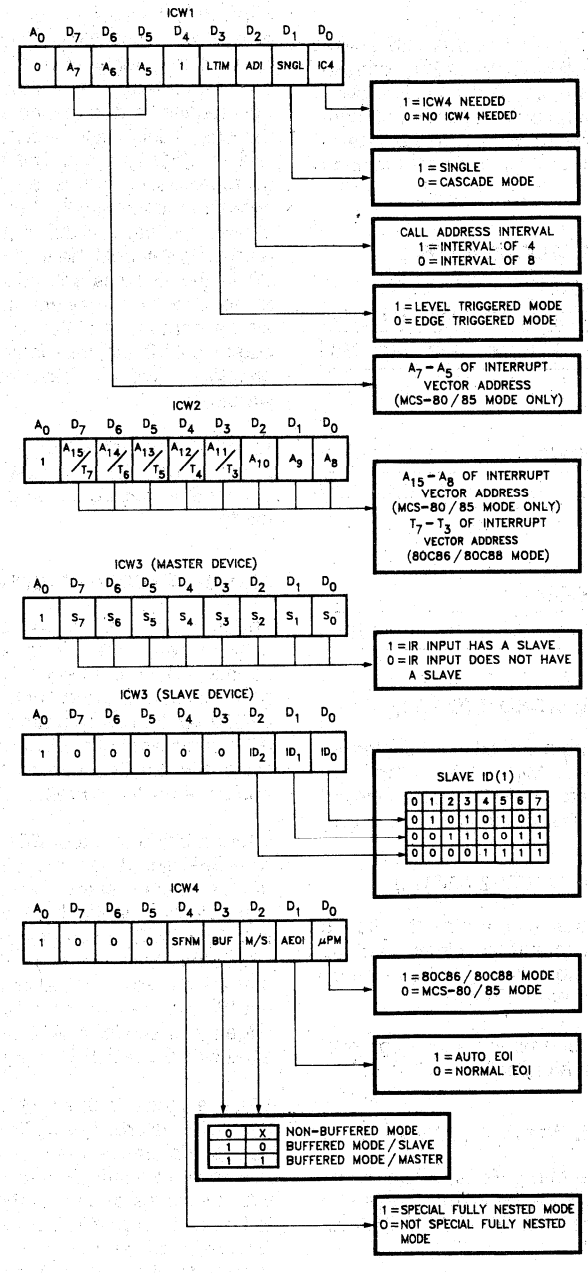


Figure 6. Initialization Sequence



NOTE:
Slave ID is equal to the corresponding master IR input.

Figure 7. Initialization Command Word Format

OPERATION COMMAND WORDS (OCWs)

After the initialization Command Words (ICWs) are programmed into the 82C59A-2, the chip is ready to accept interrupt requests at its input lines. However, during the 82C59A-2 operation, a selection of algorithms can command the 82C59A-2 to operate in various modes through the Operation Command Words (OCWs).

OPERATION CONTROL WORDS (OCWs)

OCW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0

OCW2

0	R	SL	EOI	0	0	L2	L1	L0

OCW3

0	0	ESMM	SMM	0	1	P	RR	RIS

OPERATION CONTROL WORD 1 (OCW1)

OCW1 sets and clears the mask bits in the interrupt Mask Register (IMR). M_7-M_0 represent the eight mask bits. M = 1 indicates the channel is masked (inhibited), M = 0 indicates the channel is enabled.

OPERATION CONTROL WORD 2 (OCW2)

R, SL, EOI — These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

L_2, L_1, L_0—These bits determine the interrupt level acted upon when the SL bit is active.

OPERATION CONTROL WORD 3 (OCW3)

ESMM — Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0 the SMM bit becomes a "don't care".

SMM — Special Mask Mode. If ESMM = 1 and SMM = 1 the 82C59A-2 will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the 82C59A-2 will revert to normal mask mode. When ESMM = 0, SMM has no effect.

FULLY NESTED MODE

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority form 0 through 7 (0 highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (ISO-7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic. End of Interrupt) bit is set, until the trailing edge of the last INTA. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

END OF INTERRUPT (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence $\overline{\text{INTA}}$ pulse (when AEOI bit in ICW4 is set) or by a command word that must be issued to the 82C59A-2 before returning from a service routine (EOI command). An EOI command must be issued twice if in the Cascade mode, once for the master and once for the corresponding slave.

There are two forms of EOI command: Specific and Non-Specific. When the 82C59A-2 is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued the 82C59A-2 will automatically reset the highest IS bit of those that are set, since in the fully nested mode the highest IS level was necessarily the last level acknowledged and serviced. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).

When a mode is used which may disturb the fully nested structure, the 82C59A-2 may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt must be issued which includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and L0-L2 is the binary level of the IS bit to be reset).

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the 82C59A-2 is in the Special Mask Mode.

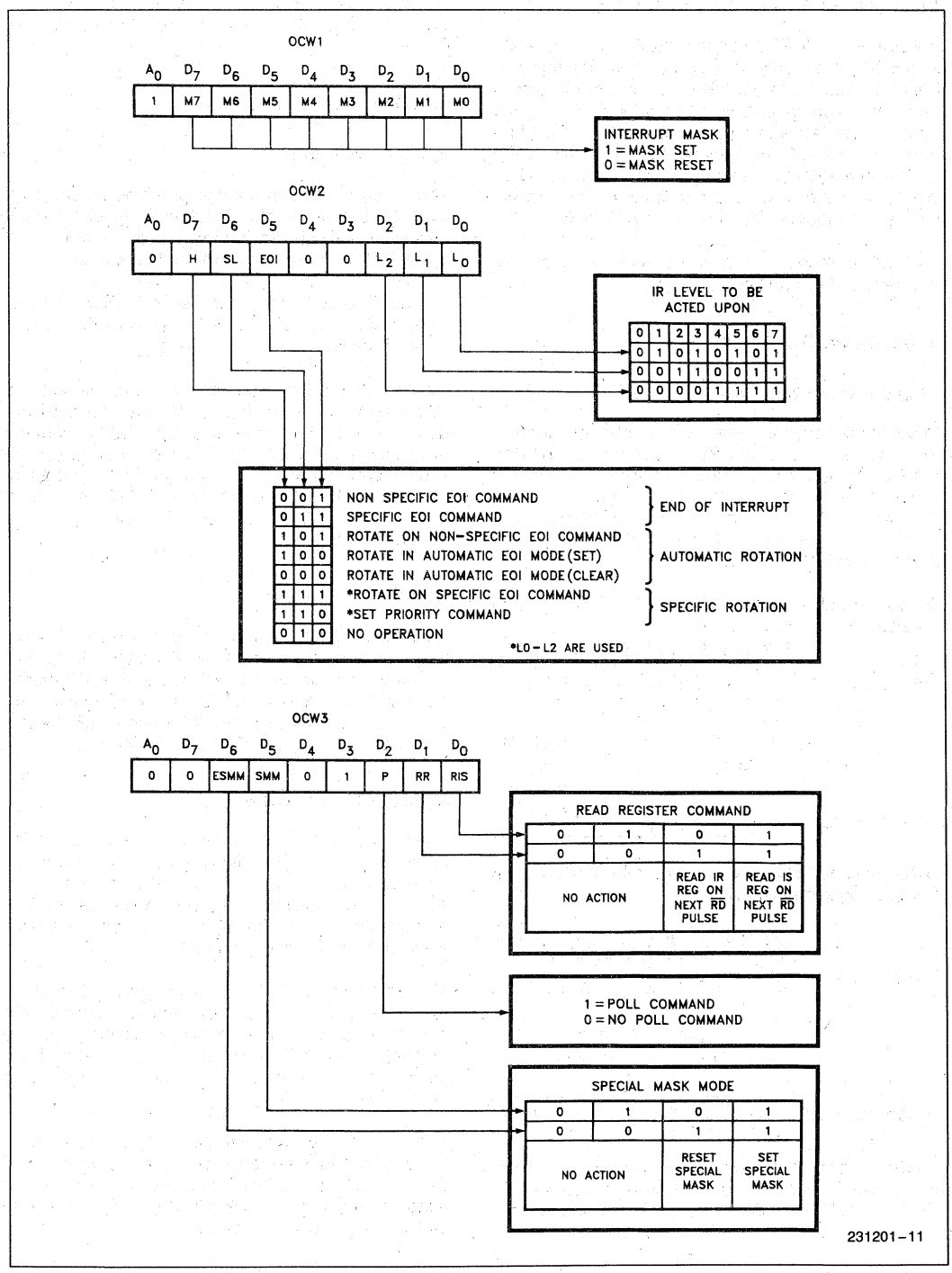


Figure 8. Operation Command Word Format

AUTOMATIC END OF INTERRUPT (AEOI) MODE

If AEOI = 1 in ICW4, then the 82C59A-2 will operate in AEOI mode continuously until reprogrammed by ICW4. In this mode the 82C59A-2 will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in MCS-80/85, second in 80C86/88). Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single 82C59A.

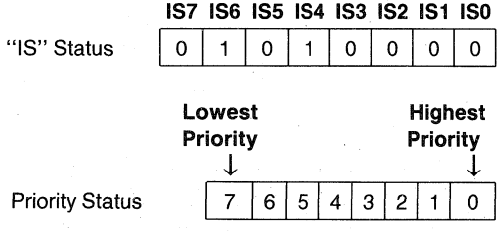
The AEOI mode can only be used in a master 82C59A and not a slave.

AUTOMATIC ROTATION

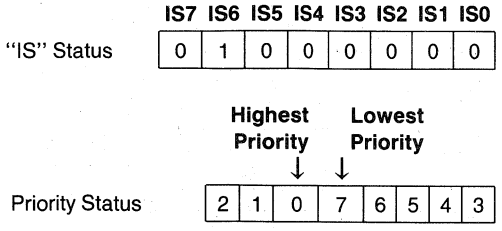
(Equal Priority Devices)

In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most *once*. For example, if the priority and "in service" status is:

Before Rotate (IR4 the highest priority requiring service)



After Rotate (IR4 was serviced, all other priorities rotated correspondingly)



There are two ways to accomplish Automatic Rotation using OCW2, the Rotation on Non-Specific EOI Command (R = 1, SL = 0, EOI = 1) and the Rotate in Automatic EOI Mode which is set by (R = 1, SL = 0, EOI = 0) and cleared by (R = 0, SL = 0, EOI = 0).

SPECIFIC ROTATION

(Specific Priority)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

The Set Priority command is issued in OCW2 where: R = 1, SL = 1; LO–L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command by using the Rotate on Specific EOI command in OCW2 (R = 1, SL = 1, EOI = 1 and LO–L2 = IR level to receive bottom priority).

INTERRUPT MASKS

Each Interrupt Request input can be masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels operation.

SPECIAL MASK MODE

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that if an interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the 82C59A-2 would have inhibited all lower priority requests with no easy way for the routine to enable them.

That is where the Special Mask Mode comes in. In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level *and enables* interrupts from *all other* levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectivity enabled by loading the mask register.

The special Mask Mode is set by OCW3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

POLL COMMAND

In Poll mode the INT output functions as it normally does. The microprocessor should ignore this output. This can be accomplished either by not connecting the INT output or by masking interrupts within the microprocessor, thereby disabling its interrupt input. Service to devices is achieved by software using a Poll command.

The Poll command is issued by setting P = "1" in OCW3. The 82C59A-2 treats the next $\overline{RD}$ pulse to the 82C59A-2 (i.e., $\overline{RD}$ = 0, $\overline{CS}$ = 0) as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from $\overline{WR}$ to $\overline{RD}$.

The word enabled onto the data bus during $\overline{RD}$ is:

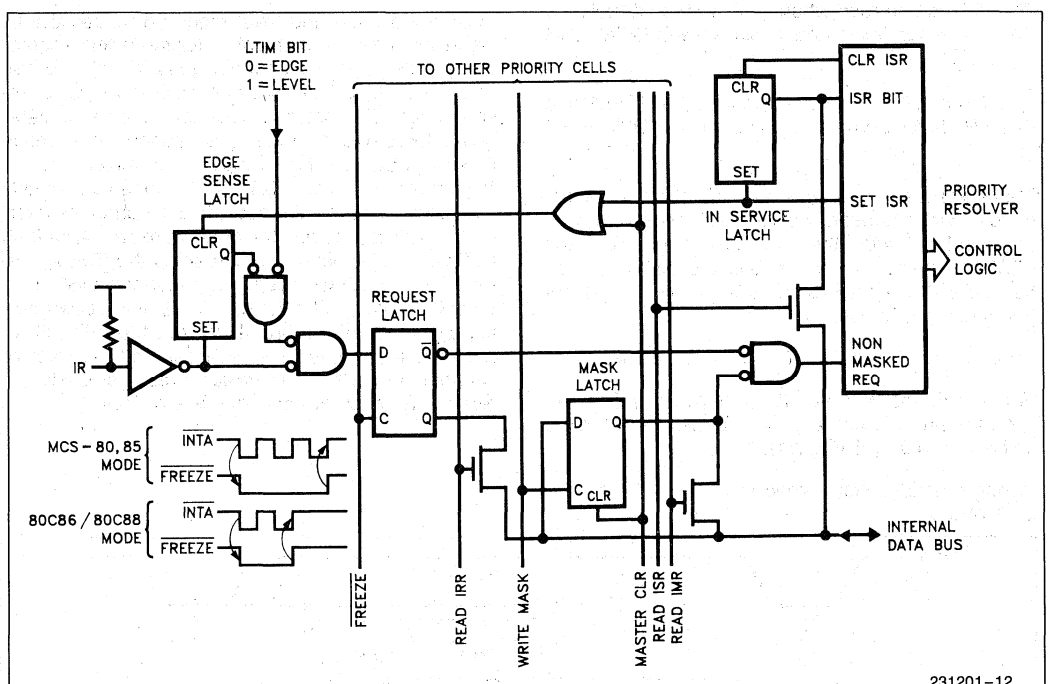
D7	D6	D5	D4	D3	D2	D1	D0
I	—	—	—	—	W2	W1	W0

WO–W2:

Binary code of the highest priority level requesting service.

I: Equal to a "1" if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the $\overline{INTA}$ sequence is not needed (saves ROM space). Another application is to use the poll mode to expand the number of priority levels to more than 64.



NOTES:
1. Master Clear active only during ICW1
2. Freeze/ is active during $\overline{INTA}$/and poll sequences only
3. Truth Table for D-Latch

C	D	Q	OPERATION
1	Di	Di	FOLLOW
0	X	Qn-1	HOLD

Figure 9. Priority Cell—Simplified Logic Diagram

READING THE 82C59A-2 STATUS

The input status of several internal registers can be read to update the user information on the system. The following registers can be read via OCW3 (IRR and ISR or OCW1 [IMR]).

Interrupt Request Register (IRR): 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR).

In-Service Register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt Command is issued.

Interrupt Mask Register: 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 0.)

The ISR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 1):

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the 82C59A-2 "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.

After initialization the 82C59A-2 is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever $\overline{RD}$ is active and AO = 1 (OCW1).

Polling overrides status read when P = 1, RR = 1 in OCW3.

EDGE AND LEVEL TRIGGERED MODES

This mode is programmed using bit 3 in ICW1.

If LTIM = '0', an interrupt request will be recognized by a low to high transition on an IR input. The IR input can remain high without generating another interrupt.

If LTIM = '1', an interrupt request will be recognized by a 'high' level on IR Input, and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued or the CPU interrupt is enabled to prevent a second interrupt from occurring.

The priority cell diagram shows a conceptual circuit of the level sensitive and edge sensitive input circuitry of the 82C59A-2. Be sure to note that the request latch is a transparent D type latch.

In both the edge and level triggered modes the IR inputs must remain high until after the falling edge of the first INTA. If the IR input goes low before this time a DEFAULT IR7 will occur when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IR inputs. To implement this feature the IR7 routine is used for "clean up" simply executing a return instruction, thus ignoring the interrupt. If IR7 is needed for other purposes a default IR7 can still be detected by reading the ISR. A normal IR7 interrupt will set the corresponding ISR bit, a default IR7 won't. If a default IR7 routine occurs during a normal IR7 routine, however, the ISR will remain set. In this case it is necessary to keep track of whether or not the IR7 routine was previously entered. If another IR7 occurs it is a default.

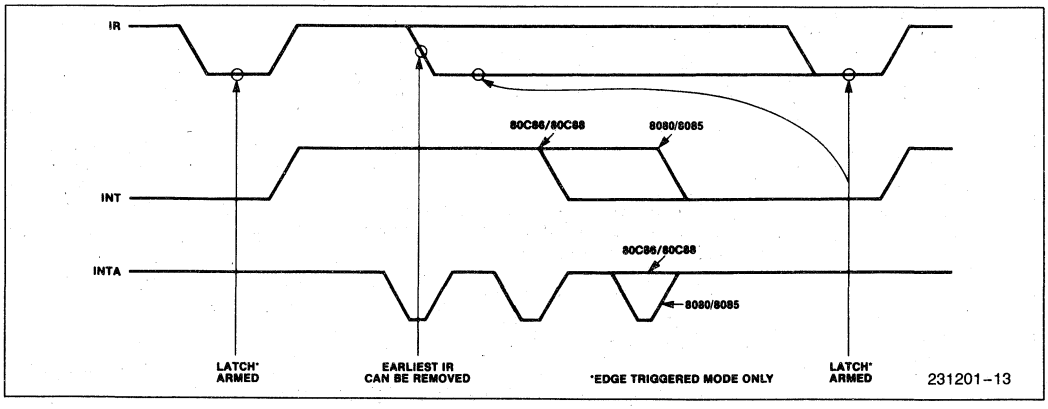


Figure 10. IR Triggering Timing Requirements

THE SPECIAL FULLY NESTED MODE

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

a. When an interrupt request from a certain slave is in service this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IR's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nestled mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

b. When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

BUFFERED MODE

When the 82C59A-2 is used in a large system where bus driving buffers are required on the data bus and the cascading mode is used, there exists the problem of enabling buffers.

The buffered mode will structure the 82C59A-2 to send an enable signal on $\overline{SP}/\overline{EN}$ to enable the buffers. In this mode, whenever the 82C59A-2's data bus outputs are enabled, the $\overline{SP}/\overline{EN}$ output becomes active.

This modification forces the use of software programming to determine whether the 82C59A-2 is a master or a slave. Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW3 determines whether it is a master or a slave.

CASCADE MODE

The 82C59A-2 can be easily interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels.

The master controls the slaves through the 3 line cascade bus. The cascade bus acts like chip selects to the slaves during the $\overline{INTA}$ sequence.

In a cascade configuration, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will enable the corresponding slave to release the device routine address during bytes 2 and 3 of INTA. (Byte 2 only for 80C86/80C88).

The cascade bus lines are normally low and will contain the slave address code from the trailing edge of the first INTA pulse to the trailing edge of the third pulse. Each 82C59A-2 in the system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: once for the master and once for the corresponding slave. An address decoder is required to activate the Chip Select (CS) input of each 82C59A-2.

The cascade lines of the Master 82C59A-2 are activated only for slave inputs, non slave inputs leave the cascade line inactive (low).

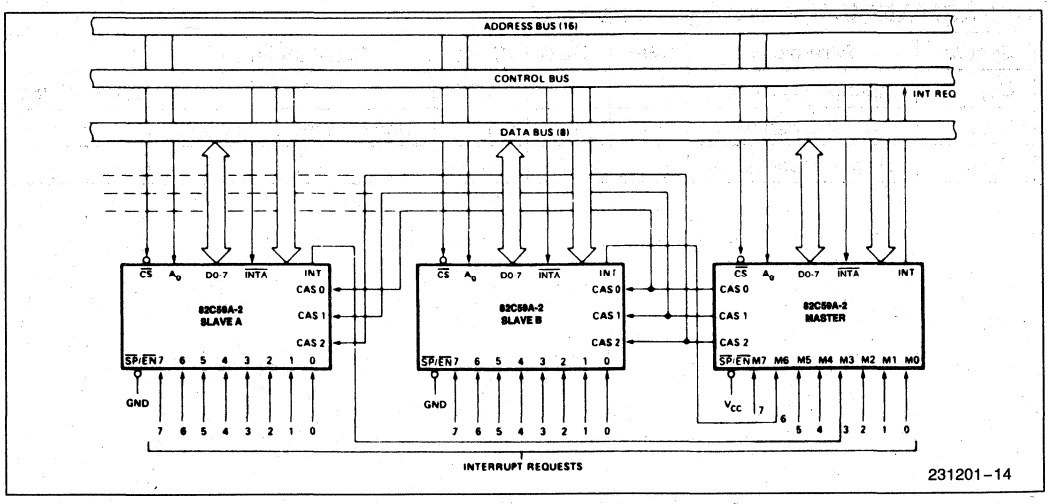


Figure 11. Cascading the 82C59A-2

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . . 0°C to 70°C
Storage Temperature . . . . . −65°C to + 150°C
Supply Voltage (w.r.t. ground) . . . . . −0.5 to 7.0V
Input Voltage (w.r.t. ground) . . . −0.5 to V_{CC} + 0.5V
Output Voltage (w.r.t. ground) . . −0.5 to V_{CC} + 0.5V
Power Dissipation . . . . . 0.9 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
I_{CCS}	Standby Supply Current		10	μA	$V_{IN} = V_{CC}$ or GND All IR = GND Outputs Unloaded V_{CC} = 5.5V
I_{CC}	Operating Supply Current		5	mA	(Note)
V_{IH}	Input High Voltage	2.2	V_{CC} + 0.5	V	
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.5 mA
V_{OH}	Output High Voltage	3.0 V_{CC} −0.4		V	I_{OH} = −2.5 mA I_{OH} = −100 μA
I_{LI}	Input Leakage Current		±1.0	μA	$0V \le V_{IN} \le V_{CC}$
I_{LO}	Output Leakage Current		±10	μA	$0V \le V_{OUT} \le V_{CC}$
I_{LIR}	IR Input Leakage Current		−300 +10	μA	V_{IN} = 0 $V_{IN} = V_{CC}$

NOTE:
Repeated data input with 80C86-2 timings.

CAPACITANCE T_A = 25°C; V_{CC} = GND = 0V

Symbol	Parameter	Min	Max	Units	Test Conditions	
C_{IN}	Input Capacitance		7	pF		fc = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins at GND	
C_{OUT}	Output Capacitance		15	pF		

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%

TIMING REQUIREMENTS

Symbol	Parameter	82C59A-2		Units	Test Conditions
		Min	Max		
TAHRL	A0/$\overline{CS}$ Setup to $\overline{RD}$/$\overline{INTA}$ ↓	10		ns	
TRHAX	A0/$\overline{CS}$ Hold after $\overline{RD}$/$\overline{INTA}$ ↑	5		ns	
TRLRH	$\overline{RD}$/$\overline{INTA}$ Pulse Width	160		ns	
TAHWL	A0/$\overline{CS}$ Setup to $\overline{WR}$ ↓	0		ns	
TWHAX	A0/$\overline{CS}$ Hold after $\overline{WR}$ ↑	0		ns	
TWLWH	$\overline{WR}$ Pulse Width	190		ns	
TDVWH	Data Setup to $\overline{WR}$ ↑	160		ns	
TWHDX	Data Hold after $\overline{WR}$ ↑	0		ns	
TJLJH	Interrupt Request Width (Low)	100		ns	(See Note)
TCVIAL	Cascade Setup to Second or Third $\overline{INTA}$ ↓ (Slave Only)	40		ns	
TRHRL	End of $\overline{RD}$ to next $\overline{RD}$ End of $\overline{INTA}$ to next $\overline{INTA}$ within an $\overline{INTA}$ sequence only	160		ns	
TWHWL	End of $\overline{WR}$ to next $\overline{WR}$	190		ns	
*TCHCL	End of Command to next Command (Not same command type) End of $\overline{INTA}$ sequence to next $\overline{INTA}$ sequence.	400		ns	

*Worst case timing for TCHCL in an actual microprocessor system is typically much greater than 400 ns (i.e. 8085A = 1.6 μs, 8085-A2 = 1 μs, 80C86 = 1 μs, 80C86-2 = 625 ns)

NOTE:
This is the low time required to clear the input latch in the edge triggered mode.

3

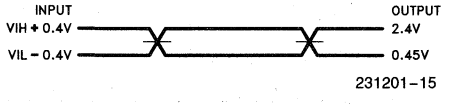
TIMING RESPONSES

Symbol	Parameter	8259A-2		Units	Test Conditions**
		Min	Max		
TRLDV	Data Valid from $\overline{RD}/\overline{INTA}$ ↓		120	ns	1
TRHDZ	Data Float after $\overline{RD}/\overline{INTA}$ ↑	10	85	ns	2
TJHIH	Interrupt Output Delay		300	ns	1
TIALCV	Cascade Valid from First $\overline{INTA}$ ↓ (Master Only)		360	ns	1
TRLEL	Enable Active from $\overline{RD}$ ↓ or $\overline{INTA}$ ↓		110	ns	1
TRHEH	Enable Inactive from $\overline{RD}$ ↑ or $\overline{INTA}$ ↑		150	ns	1
TAHDV	Data Valid from Stable Address		200	ns	1
TCVDV	Cascade Valid to Valid Data		200	ns	1

**Test Condition Definition Table

TEST CONDITION	V1	R1	R2	C1
1	1.7V	523Ω	OPEN	100 pf
2	4.5V	1.8 kΩ	1.8 kΩ	30 pf

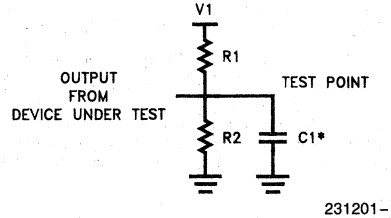
A.C. TESTING INPUT, OUTPUT WAVEFORM



231201-15

A.C. Testing: All input signals must switch between V_{IL} – 0.4V and V_{IH} + 0.4V. Input Rise and Fall Times must be ≤ 15 ns. All timing measurements are made at 2.4V and 0.45V.

A.C. TESTING LOAD CIRCUIT

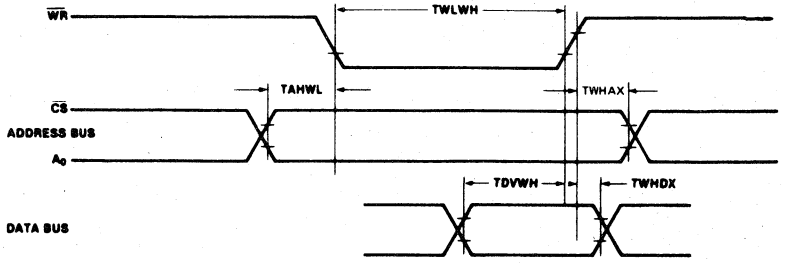


231201-16

*Includes Stray and Jig Capacitance

WAVEFORMS

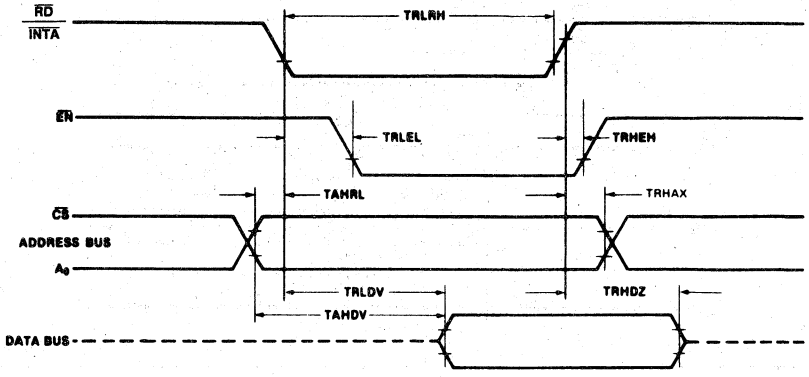
WRITE



231201-17

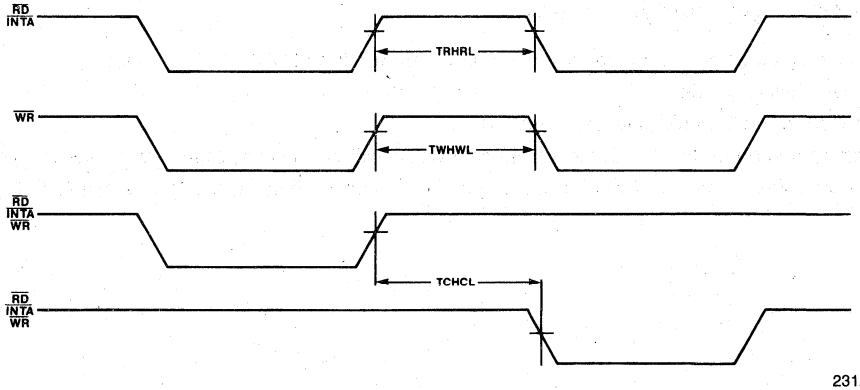
WAVEFORMS (Continued)

READ/INTA



231201-18

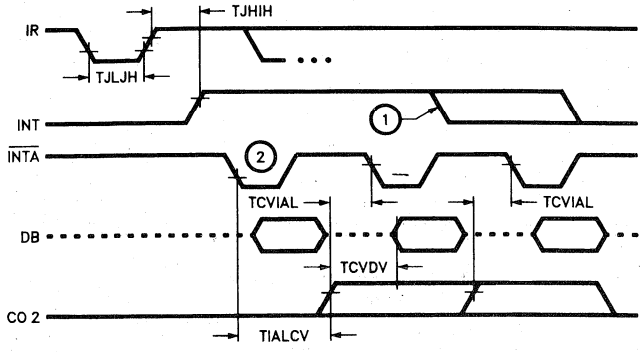
OTHER TIMING



231201-19

WAVEFORMS (Continued)

$\overline{\text{INTA}}$ SEQUENCE



231201-20

NOTES:
1. Interrupt output must remain HIGH at least until leading edge of first INTA.
2. Cycle 1 in 80C86 and 80C88 systems, the Data Bus is not active.

DATA SHEET REVISION REVIEW

The following changes have been made since revision 003 of the 82C59A-2 data sheet.

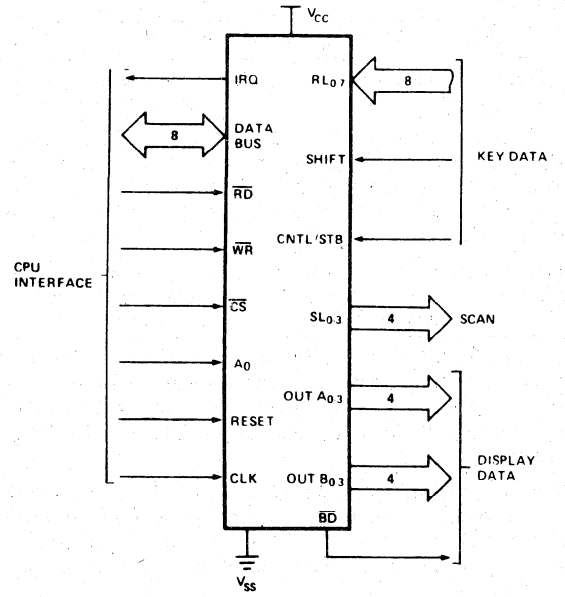
1. Preliminary was removed.
2. A reference to PLCC packaging was removed.
3. The first paragraph of the Poll Command section was rewritten to clarify the status of the INT pin.
4. A paragraph was added to the Interrupt Sequence section to indicate the status of the INT pin during multiple interrupts.

8279/8279-5
PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE

- **Simultaneous Keyboard Display Operations**
- **Scanned Keyboard Mode**
- **Scanned Sensor Mode**
- **Strobed Input Entry Mode**
- **8-Character Keyboard FIFO**
- **2-Key Lockout or N-Key Rollover with Contact Debounce**
- **Dual 8- or 16-Numerical Display**
- **Single 16-Character Display**
- **Right or Left Entry 16-Byte Display RAM**
- **Mode Programmable from CPU**
- **Programmable Scan Timing**
- **Interrupt Output on Key Entry**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**

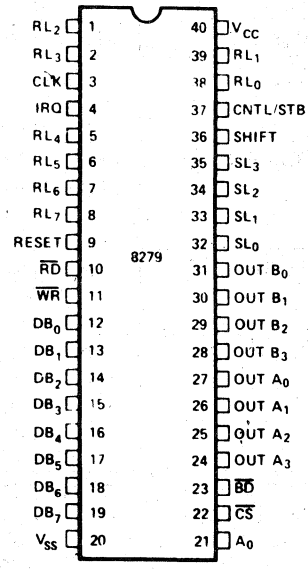
The Intel® 8279 is a general purpose programmable keyboard and display I/O interface device designed for use with Intel® microprocessors. The keyboard portion can provide a scanned interface to a 64-contact key matrix. The keyboard portion will also interface to an array of sensors or a strobed interface keyboard, such as the hall effect and ferrite variety. Key depressions can be 2-key lockout or N-key rollover. Keyboard entries are debounced and strobed in an 8-character FIFO. If more than 8 characters are entered, overrun status is set. Key entries set the interrupt output line to the CPU.

The display portion provides a scanned display interface for LED, incandescent, and other popular display technologies. Both numeric and alphanumeric segment displays may be used as well as simple indicators. The 8279 has 16x8 display RAM which can be organized into dual 16x4. The RAM can be loaded or interrogated by the CPU. Both right entry, calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.



290123-1

Figure 1. Logic Symbol



290123-2

Figure 2. Pin Configuration

September 1987
Order Number: 290123-002

HARDWARE DESCRIPTION

The 8279 is packaged in a 40 pin DIP. The following is a functional description of each pin.

Table 1. Pin Description

Symbol	Pin No.	Name and Function
DB_0–DB_7	19–12	**BI-DIRECTIONAL DATA BUS:** All data and commands between the CPU and the 8279 are transmitted on these lines.
CLK	3	**CLOCK:** Clock from system used to generate internal timing.
RESET	9	**RESET:** A high signal on this pin resets the 8279. After being reset the 8279 is placed in the following mode: 1) 16 8-bit character display—left entry. 2) Encoded scan keyboard—2 key lockout. Along with this the program clock prescaler is set to 31.
CS	22	**CHIP SELECT:** A low on this pin enables the interface functions to receive or transmit.
A_0	21	**BUFFER ADDRESS:** A high on this line indicates the signals in or out are interpreted as a command or status. A low indicates that they are data.
$\overline{RD}$, $\overline{WR}$	10–11	**INPUT/OUTPUT READ AND WRITE:** These signals enable the data buffers to either send data to the external bus or receive it from the external bus.
IRQ	4	**INTERRUPT REQUEST:** In a keyboard mode, the interrupt line is high when there is data in the FIFO/Sensor RAM. The interrupt line goes low with each FIFO/Sensor RAM read and returns high if there is still information in the RAM. In a sensor mode, the interrupt line goes high whenever a change in a sensor is detected.
V_{SS}, V_{CC}	20, 40	**GROUND AND POWER SUPPLY PINS.**
SL_0–SL_3	32–35	**SCAN LINES:** Scan lines which are used to scan the key switch or sensor matrix and the display digits. These lines can be either encoded (1 of 16) or decoded (1 of 4).
RL_0–RL_7	38, 39, 1, 2, 5–8	**RETURN LINE:** Return line inputs which are connected to the scan lines through the keys or sensor switches. They have active internal pullups to keep them high until a switch closure pulls one low. They also serve as an 8-bit input in the Strobed Input mode.
SHIFT	36	**SHIFT:** The shift input status is stored along with the key position on key closure in the Scanned Keyboard modes. It has an active internal pullup to keep it high until a switch closure pulls it low.
CNTL/STB	37	**CONTROL/STROBED INPUT MODE:** For keyboard modes this line is used as a control input and stored like status on a key closure. The line is also the strobe line that enters the data into the FIFO in the Strobed Input mode. (Rising Edge). It has an active internal pullup to keep it high until a switch closure pulls it low.
OUT A_0–OUT A_3 OUT B_0–OUT B_3	27-24 31–28	**OUTPUTS:** These two ports are the outputs for the 16 x 4 display refresh registers. The data from these outputs is synchronized to the scan lines (SL_0–SL_3) for multiplexed digit displays. The two 4 bit ports may be blanked independently. These two ports may also be considered as one 8-bit port.
$\overline{BD}$	23	**BLANK DISPLAY:** This output is used to blank the display during digit switching or by a display blanking command.

FUNCTIONAL DESCRIPTION

Since data input and display are an integral part of many microprocessor designs, the system designer needs an interface that can control these functions without placing a large load on the CPU. The 8279 provides this function for 8-bit microprocessors.

The 8279 has two sections: keyboard and display. The keyboard section can interface to regular typewriter style keyboards or random toggle or thumb switches. The display section drives alphanumeric displays or a bank of indicator lights. Thus the CPU is relieved from scanning the keyboard or refreshing the display.

The 8279 is designed to directly connect to the microprocessor bus. The CPU can program all operating modes for the 8279. These modes include:

Input Modes

- Scanned Keyboard—with encoded (8 x 8 key keyboard) or decoded (4 x 8 key keyboard) scan lines. A key depression generates a 6-bit encoding of key position. Position and shift and control status are stored in the FIFO. Keys are automatically debounced with 2-key lockout or N-key rollover.
- Scanned Sensor Matrix—with encoded (8 x 8 matrix switches) or decoded (4 x 8 matrix switches) scan lines. Key status (open or closed) stored in RAM addressable by CPU.
- Strobed Input—Data on return lines during control line strobe is transferred to FIFO.

Output Modes

- 8 or 16 character multiplexed displays that can be organized as dual 4-bit or single 8-bit (B_0 = D_0, A_3 = D_7).
- Right entry or left entry display formats.

Other features of the 8279 include:

- Mode programming from the CPU.
- Clock Prescaler
- Interrupt output to signal CPU when there is keyboard or sensor data available.
- An 8 byte FIFO to store keyboard information.
- 16 byte internal Display RAM for display refresh. This RAM can also be read by the CPU.

PRINCIPLES OF OPERATION

The following is a description of the major elements of the 8279 Programmable Keyboard/Display interface device. Refer to the block diagram in Figure 3.

I/O Control and Data Buffers

The I/O control section uses the $\overline{CS}$, A_0, $\overline{RD}$ and $\overline{WR}$ lines to control data flow to and from the various internal registers and buffers. All data flow to and from the 8279 is enabled by $\overline{CS}$. The character of the information, given or desired by the CPU, is identified by A_0. A logic one means the information is a command or status. A logic zero means the information is data. $\overline{RD}$ and $\overline{WR}$ determine the direction of data flow through the Data Buffers. The Data Buffers are bi-directional buffers that connect the internal bus to the external bus. When the chip is not selected ($\overline{CS}$ = 1), the devices are in a high impedance state. The drivers input during $\overline{WR} \bullet \overline{CS}$ and output during $\overline{RD} \bullet \overline{CS}$.

Control and Timing Registers and Timing Control

These registers store the keyboard and display modes and other operating conditions programmed by the CPU. The modes are programmed by presenting the proper command on the data lines with A_0 = 1 and then sending a $\overline{WR}$. The command is latched on the rising edge of $\overline{WR}$. The command is then decoded and the appropriate function is set. The timing control contains the basic timing counter chain. The first counter is a ÷ N prescaler that can be programmed to yield an internal frequency of 100 kHz which gives a 5.1 ms keyboard scan time and a 10.3 ms debounce time. The other counters divide down the basic internal frequency to provide the proper key scan, row scan, keyboard matrix scan, and display scan times.

Scan Counter

The scan counter has two modes. In the encoded mode, the counter provides a binary count that must be externally decoded to provide the scan lines for the keyboard and display. In the decoded mode, the scan counter decodes the least significant 2 bits and provides a decoded 1 of 4 scan. Note that when the keyboard is in decoded scan, so is the display. This means that only the first 4 characters in the Display RAM are displayed.

In the encoded mode, the scan lines are active high outputs. In the decoded mode, the scan lines are active low outputs.

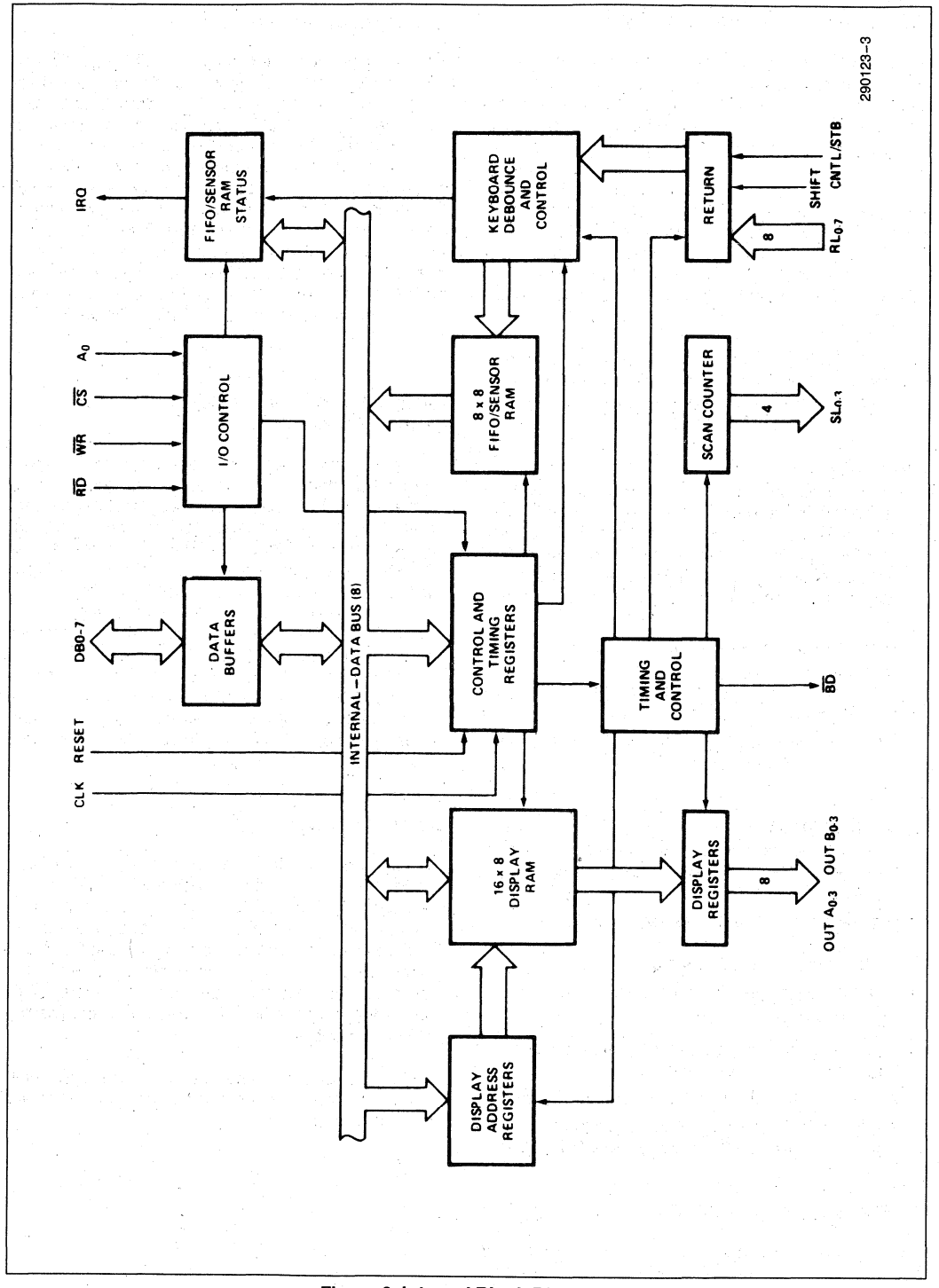


Figure 3. Internal Block Diagram

Return Buffers and Keyboard Debounce and Control

The 8 return lines are buffered and latched by the Return Buffers. In the keyboard mode, these lines are scanned, looking for key closures in that row. If the debounce circuit detects a closed switch, it waits about 10 ms to check if the switch remains closed. If it does, the address of the switch in the matrix plus the status of SHIFT and CONTROL are transferred to the FIFO. In the scanned Sensor Matrix modes, the contents of the return lines is directly transferred to the corresponding row of the Sensor RAM (FIFO) each key scan time. In Strobed Input mode, the contents of the return lines are transferred to the FIFO on the rising edge of the CNTL/STB line pulse.

FIFO/Sensor RAM and Status

This block is a dual function 8 x 8 RAM. In Keyboard or Strobed Input modes, it is a FIFO. Each new entry is written into successive RAM positions and each is then read in order of entry. FIFO status keeps track of the number of characters in the FIFO and whether it is full or empty. Too many reads or writes will be recognized as an error. The status can be read by an $\overline{RD}$ with $\overline{CS}$ low and A_0 high. The status logic also provides an IRQ signal when the FIFO is not empty. In Scanned Sensor Matrix mode, the memory is a Sensor RAM. Each row of the Sensor RAM is loaded with the status of the corresponding row of sensor in the sensor matrix. In this mode, IRQ is high if a change in a sensor is detected.

Display Address Registers and Display RAM

The Display Address Registers hold the address of the word currently being written or read by the CPU and the two 4-bit nibbles being displayed. The read/write addresses are programmed by CPU command. They also can be set to auto increment after each read or write. The Display RAM can be directly read by the CPU after the correct mode and address is set. The addresses for the A and B nibbles are automatically updated by the 8279 to match data entry by the CPU. The A and B nibbles can be entered independently or as one word, according to the mode that is set by the CPU. Data entry to the display can be set to either left or right entry. See Interface Considerations for details.

SOFTWARE OPERATION

8279 Commands

The following commands program the 8279 operating modes. The commands are sent on the Data Bus with $\overline{CS}$ low and A_0 high and are loaded to the 8279 on the rising edge of $\overline{WR}$.

Keyboard/Display Mode Set

	MSB							LSB
Code:	0	0	0	D	D	K	K	K

Where DD is the Display Mode and KKK is the Keyboard Mode.

DD

0	0	8 8-bit character display—Left entry
0	1	16 8-bit character display—Left entry*
1	0	8 8-bit character display—Right entry
1	1	16 8-bit character display—Right entry

For description of right and left entry, see Interface Considerations. Note that when decoded scan is set in keyboard mode, the display is reduced to 4 characters independent of display mode set.

KKK

0	0	0	Encoded Scan Keyboard—2 Key Lockout*
0	0	1	Decoded Scan Keyboard—2-Key Lockout
0	1	0	Encoded Scan Keyboard—N-Key Rollover
0	1	1	Decoded Scan Keyboard—N-Key Rollover
1	0	0	Encoded Scan Sensor Matrix
1	0	1	Decoded Scan Sensor Matrix
1	1	0	Strobed Input, Encoded Display Scan
1	1	1	Strobed Input, Decoded Display Scan

*Default after reset.

Program Clock

Code:	0	0	1	P	P	P	P	P

All timing and multiplexing signals for the 8279 are generated by an internal prescaler. This prescaler divides the external clock (pin 3) by a programmable integer. Bits PPPPP determine the value of this integer which ranges from 2 to 31. Choosing a divisor that yields 100 kHz will give the specified scan and

debounce times. For instance, if Pin 3 of the 8279 is being clocked by a 2 MHz signal, PPPPP should be set to 10100 to divide the clock by 20 to yield the proper 100 kHz operating frequency.

Read FIFO/Sensor RAM

Code:	0	1	0	AI	X	A	A	A

X = Don't Care

The CPU sets the 8279 for a read of the FIFO/Sensor RAM by first writing this command. In the Scan Keyboard Mode, the Auto-Increment flag (AI) and the RAM address bits (AAA) are irrelevant. The 8279 will automatically drive the data bus for each subsequent read (A_0 = 0) in the same sequence in which the data first entered the FIFO. All subsequent reads will be from the FIFO until another command is issued.

In the Sensor Matrix Mode, the RAM address bits AAA select one of the 8 rows of the Sensor RAM. If the AI flag is set (AI = 1), each successive read will be from the subsequent row of the sensor RAM.

Read Display RAM

Code:	0	1	1	AI	A	A	A	A

The CPU sets up the 8279 for a read of the Display RAM by first writing this command. The address bits AAAA select one of the 16 rows of the Display RAM. If the AI flag is set (A1 = 1), this row address will be incremented after each following read *or write* to the Display RAM. Since the same counter is used for both reading and writing, this command sets the next read or *write* address and the sense of the Auto-Increment mode for both operations.

Write Display RAM

Code:	1	0	0	AI	A	A	A	A

The CPU sets up the 8279 for a write to the Display RAM by first writing this command. After writing the command with A_0 = 1, all subsequent writes with A_0 = 0 will be to the Display RAM. The addressing and Auto-Increment functions are identical to those for the Read Display RAM. However, this command does not affect the source of subsequent Data Reads; the CPU will read from whichever RAM (Display of FIFO/Sensor) which was last specified. If, indeed, the Display RAM was last specified, the Write Display RAM will, nevertheless, change the next Read location.

Display Write Inhibit/Blanking

					A	B	A	B
Code:	1	0	1	X	IW	IW	BL	BL

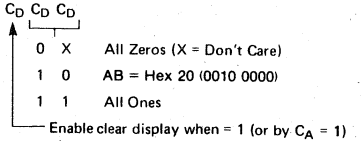
The IW Bits can be used to mask nibble A and nibble B in applications requiring separate 4-bit display ports. By setting the IW flag (IW = 1) for one of the ports, the port becomes marked so that entries to the Display RAM from the CPU do not affect that port. Thus, if each nibble is input to a BCD decoder, the CPU may write a digit to the Display RAM without affecting the other digit being displayed. It is important to note that bit B_0 corresponds to bit D_0 on the CPU bus, and that bit A_3 corresponds to bit D_7.

If the user wishes to blank the display, the BL flags are available for each nibble. The last Clear command issued determines the code to be used as a "blank." This code defaults to all zeros after a reset. Note that both BL flags must be set to blank a display formatted with a single 8-bit port.

Clear

Code:	1	1	0	C_D	C_D	C_D	C_F	C_A

The C_D bits are available in this command to clear all rows of the Display RAM to a selectable blanking code as follows:



290123-13

During the time the Display RAM is being cleared (~ 160 μs), it may not be written to. The most significant bit of the FIFO status word is set during this time. When the Display RAM becomes available again, it automatically resets.

If the C_F bit is asserted (C_F = 1), the FIFO status is cleared and the interrupt output line is reset. Also, the Sensor RAM pointer is set to row 0.

C_A, the Clear All bit, has the combined effect of C_D and C_F; it uses the C_D clearing code on the Display RAM and also clears FIFO status. Furthermore, it resynchronizes the internal timing chain.

End Interrupt/Error Mode Set

Code: | 1 | 1 | 1 | E | X | X | X | X | X = Don't care

For the sensor matrix modes this command lowers the IRQ line and enables further writing into RAM. (The IRQ line would have been raised upon the detection of a change in a sensor value. This would have also inhibited further writing into the RAM until reset).

For the N-key rollover mode—if the E bit is programmed to "1" the chip will operate in the special Error mode. (For further details, see Interface Considerations Section.)

Status Word

The status word contains the FIFO status, error, and display unavailable signals. This word is read by the CPU when A_0 is high and $\overline{CS}$ and $\overline{RD}$ are low. See Interface Considerations for more detail on status word.

Data Read

Data is read when A_0, $\overline{CS}$ and $\overline{RD}$ are all low. The source of the data is specified by the Read FIFO or Read Display commands. The trailing edge of $\overline{RD}$ will cause the address of the RAM being read to be incremented if the Auto-Increment flag is set. FIFO reads always increment (if no error occurs) independent of AI.

Data Write

Data that is written with A_0, $\overline{CS}$ and $\overline{WR}$ low is always written to the Display RAM. The address is specified by the latest Read Display or Write Display command. Auto-Incrementing on the rising edge of $\overline{WR}$ occurs if AI is set by the latest display command.

INTERFACE CONSIDERATIONS

Scanned Keyboard Mode, 2-Key Lockout

There are three possible combinations of conditions that can occur during debounce scanning. When a key is depressed, the debounce logic is set. Other depressed keys are looked for during the next two scans. If none are encountered, it is a single key depression and the key position is entered into the FIFO along with the status of CNTL and SHIFT lines. If the FIFO was empty, IRQ will be set to signal the CPU that there is an entry in the FIFO. If the FIFO was full, the key will not be entered and the error flag will be set. If another closed switch is encountered, no entry to the FIFO can occur. If all other keys are released before ths one, then it will be entered to the FIFO. If this key is released before any other, it will be entirely ignored. A key is entered to the FIFO only once per depression, no matter how many keys were pressed along with it or in what order they were released. If two keys are depressed within the debounce cycle, it is a simultaneous depression. Neither key will be recognized until one key remains depressed alone. The last key will be treated as a single key depression.

Scanned Keyboard Mode, N-Key Rollover

With N-key Rollover each key depression is treated independently from all others. When a key is depressed, the debounce circuit waits 2 keyboard scans and then checks to see if the key is still down. If it is , the key is entered into the FIFO. Any number of keys can be depressed and another can be recognized and entered into the FIFO. If a simultaneous depression occurs, the keys are recognized and entered according to the order the keyboard scan found them.

Scanned Keyboard—Special Error Modes

For N-key rollover mode the user can program a special error mode. This is done by the "End Interrupt/Error Mode Set" command. The debounce cycle and key-validity check are as in normal N-key mode. If during a *single debounce cycle*, two keys are found depressed, this is considered a simultaneous multiple depression, and sets an error flag. This flag will prevent any further writing into the FIFO and will set interrupt (if not yet set). The error flag could be read in this mode by reading the FIFO STATUS word. (See "FIFO STATUS" for further details.) The error flag is reset by sending the normal CLEAR command with CF = 1.

Sensor Matrix Mode

In Sensor Matrix mode, the debounce logic is inhibited. The status of the sensor switch is inputted directly to the Sensor RAM. In this way the Sensor RAM keeps an image of the state of the switches in the sensor matrix. Although debouncing is not provided, this mode has the advantage that the CPU knows how long the sensor was closed and when it

was released. A keyboard mode can only indicate a validated closure. To make the software easier, the designer should functionally group the sensors by row since this is the format in which the CPU will read them.

The IRQ line goes high if any sensor value change is detected at the end of a sensor matrix scan. The IRQ line is cleared by the first data read operation if the Auto-Increment flag is set to zero, or by the End Interrupt command if the Auto-Increment flag is set to one.

NOTE:
Multiple changes in the matrix Addressed by (SL_0–$_3$ = 0) may cause multiple interrupts. (SL_0 = 0 in the Decoded Mode.) Reset may cause the 8279 to see multiple changes.

Data Format

In the Scanned Keyboard mode, the character entered into the FIFO corresponds to the position of the switch in the keyboard plus the status of the CNTL and SHIFT lines (non-inverted). CNTL is the MSB of the character and SHIFT is the next most significant bit. The next three bits are from the scan counter and indicate the row the key was found in. The last three bits are from the column counter and indicate to which return line the key was connected.

MSB							LSB
CNTL	SHIFT	SCAN			RETURN		

SCANNED KEYBOARD DATA FORMAT

In Sensor Matrix mode, the data on the return lines is entered directly in the row of the Sensor RAM that corresponds to the row in the matrix being scanned. Therefore, each switch position maps directly to a Sensor RAM position. The SHIFT and CNTL inputs are ignored in this mode. Note that switches are not necessarily the only thing that can be connected to the return lines in this mode. Any logic that can be triggered by the scan lines can enter data to the return line inputs. Eight multiplexed input ports could be tied to the return lines and scanned by the 8279.

MSB							LSB
RL_7	RL_6	RL_5	RL_4	RL_3	RL_2	RL_1	RL_0

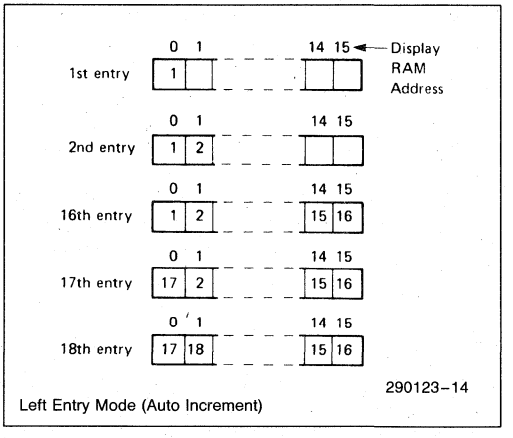
In Strobed Input mode, the data is also entered to the FIFO from the return lines. The data is entered by the rising edge of a CNTL/STB line pulse. Data can come from another encoded keyboard or simple switch matrix. The return lines can also be used as a general purpose strobed input.

MSB							LSB
RL_7	RL_6	RL_5	RL_4	RL_3	RL_2	RL_1	RL_0

Display

Left Entry

Left Entry mode is the simplest display format in that each display position directly corresponds to a byte (or nibble) in the Display RAM. Address 0 in the RAM is the left-most display character and address 15 (or address 7 in 8 character display) is the right most display character. Entering characters from position zero causes the display to fill from the left. The 17th (9th) character is entered back in the left most position and filling again proceeds from there.

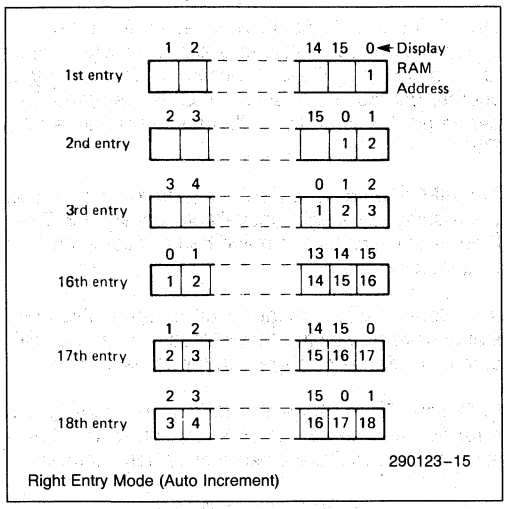


290123-14

Left Entry Mode (Auto Increment)

Right Entry

Right entry is the method used by most electronic calculators. The first entry is placed in the right most display character. The next entry is also placed in the right most character after the display is shifted left one character. The left most character is shifted off the end and is lost.

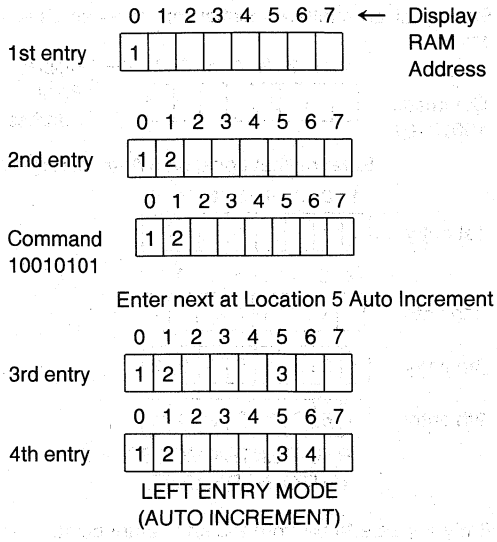


Right Entry Mode (Auto Increment)

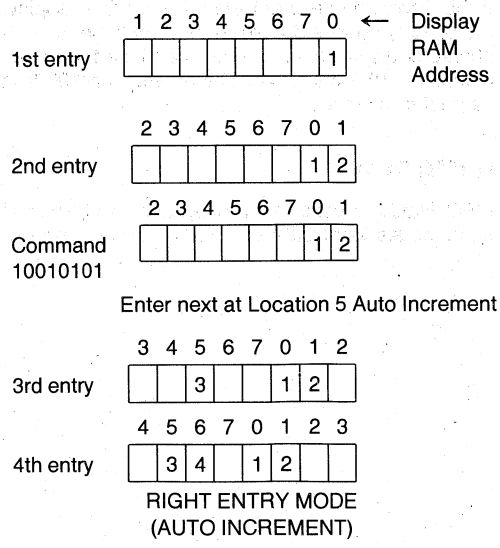
Note that now the display position and register address do not correspond. Consequently, entering a character to an arbitrary position in the Auto Increment mode may have unexpected results. Entry starting at Display RAM address 0 with sequential entry is recommended.

Auto Increment

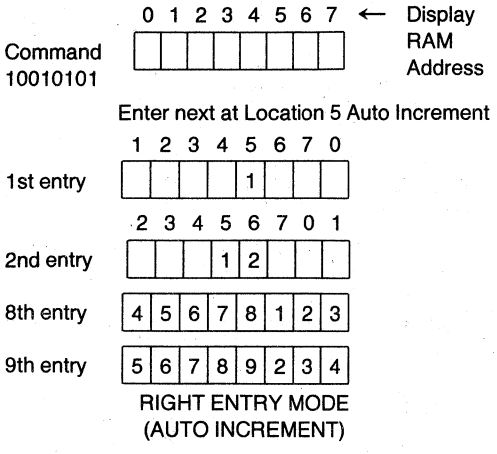
In the Left Entry mode, Auto Incrementing causes the address where the CPU will next write to be incremented by one and the character appears in the next location. With non-Auto Incrementing the entry is both to the same RAM address and display position. Entry to an arbitrary address in the Auto Increment mode has no undesirable side effects and the result is predictable:



In the Right Entry mode, Auto Incrementing and non Incrementing have the same effect as in the Left Entry except if the address sequence is interrupted.



Starting at an arbitrary location operates as shown below:



Entry appears to be from the intial entry point.

8/16 Character Display Formats

If the display mode is set to an 8 character display, the on duty-cycle is double what it would be for a 16 character display (e.g., 5.1 ms scan time for 8 characters vs. 10.3 ms for 16 characters with 100 kHz internal frequency).

G. FIFO Status

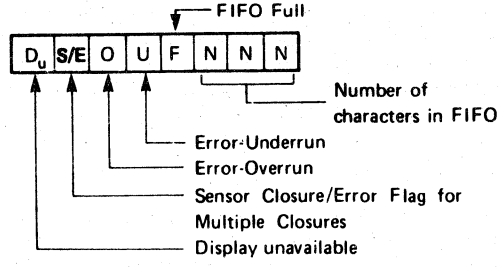
FIFO status is used in the Keyboard and Strobed Input modes to indicate the number of characters in the FIFO and to indicate whether an error has occurred. There are two types of errors possible: overrun and underrun. Overrun occurs when the entry of another character into a full FIFO is attempted. Underrun occurs when the CPU tries to read an empty FIFO.

The FIFO status word also has a bit to indicate that the Display RAM was unavailable because a Clear Display or Clear All command had not completed its clearing operation.

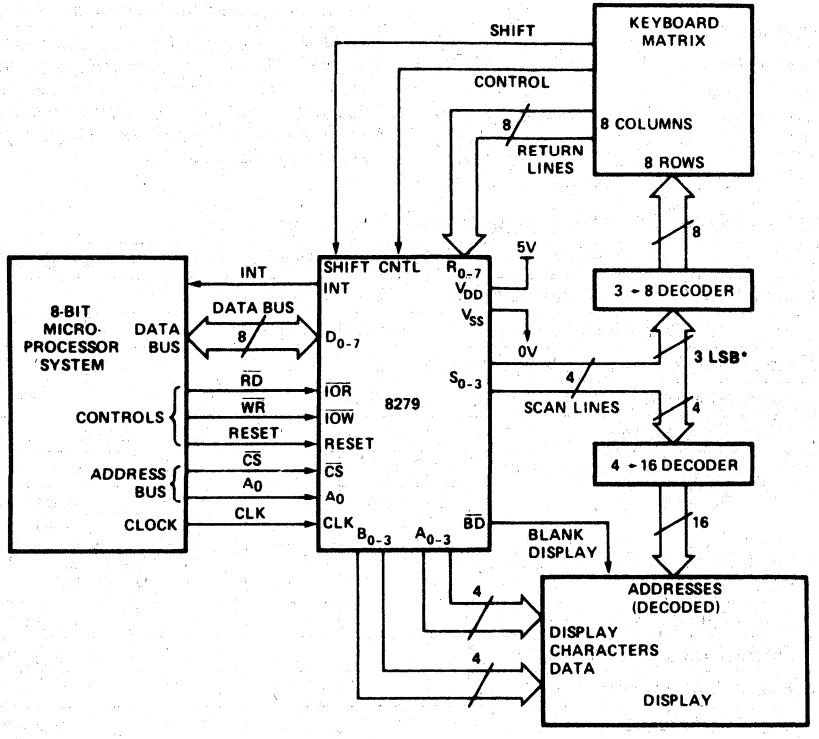
In a Sensor Matrix mode, a bit is set in the FIFO status word to indicate that at least one sensor closure indication is contained in the Sensor RAM.

In Special Error Mode the S/E bit is showing the error flag and serves as an indication to whether a simultaneous multiple closure error has occurred.

FIFO STATUS WORD



290123-4



*Do not drive the keyboard decoder with the MSB of the scan lines.

Figure 4. System Block Diagram

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature .....0°C to 70°C
Storage Temperature.....−65°C to 125°C
Voltage on any Pin with Respect to Ground.....−0.5V to +7V
Power Dissipation .....1 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C V_{SS} = 0V (Note 3)*

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL1}	Input Low Voltage for Return Lines	−0.5	1.4	V	
V_{IL2}	Input Low Voltage for All Others	−0.5	0.8	V	
V_{IH1}	Input High Voltage for Return Lines	2.2		V	
V_{IH2}	Input High Voltage for All Others	2.0		V	
V_{OL}	Output Low Voltage		0.45	V	(Note 1)
V_{OH1}	Output High Voltage on Interrupt Line	3.5		V	(Note 2)
V_{OH2}	Other Outputs	2.4			I_{OH} = −400 μA 8279-5 −100 μA 8279
I_{IL1}	Input Current on Shift, Control and Return Lines		+10 −100	μA μA	$V_{IN} = V_{CC}$ V_{IN} = 0V
I_{IL2}	Input Leakage Current on All Others		±10	μA	$V_{IN} = V_{CC}$ to 0V
I_{OFL}	Output Float Leakage		±10	μA	$V_{OUT} = V_{CC}$ to 0.45V
I_{CC}	Power Supply Current		120	mA	
C_{IN}	Input Capacitance		10	pF	f_C = 1 MHz Unmeasured Pins Returned to V_{SS}(6)
C_{OUT}	Output Capacitance		20	pF	

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{SS} = 0V (Note 3)*

Bus Parameters

READ CYCLE

Symbol	Parameter	8279		8279-5		Unit
		Min	Max	Min	Max	
t_{AR}	Address Stable Before $\overline{READ}$	50		0		ns
t_{RA}	Address Hold Time for $\overline{READ}$	5		0		ns
t_{RR}	$\overline{READ}$ Pulse Width	420		250		ns
t_{RD}(4)	Data Delay from $\overline{READ}$		300		150	ns
t_{AD}(4)	Address to Data Valid		450		250	ns
t_{DF}	$\overline{READ}$ to Data Floating	10	100	10	100	ns
t_{RCY}	Read Cycle Time	1		1		μs
t_{AW}	Address Stable Before $\overline{WRITE}$	50		0		ns
t_{WA}	Address Hold Time for $\overline{WRITE}$	20		0		ns

A.C. CHARACTERISTICS (Continued)

WRITE CYCLE

Symbol	Parameter	8279		8279-5		Unit
		Min	Max	Min	Max	
t_{WW}	$\overline{WRITE}$ Pulse Width	400		250		ns
t_{DW}	Data Set Up Time for $\overline{WRITE}$	300		150		ns
t_{WD}	Data Hold Time for $\overline{WRITE}$	40		0		ns
t_{WCY}	Write Cycle Time	1		1		μs

OTHER TIMINGS

Symbol	Parameter	8279		8279-5		Unit
		Min	Max	Min	Max	
$t_{\phi W}$	Clock Pulse Width	230		120		ns
t_{CY}	Clock Period	500		320		ns

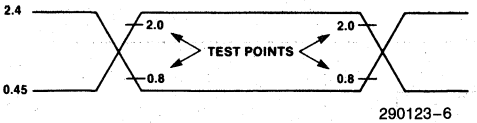
Keyboard Scan Time ..... 5.1 ms
Keyboard Debounce Time ..... 10.3 ms
Key Scan Time ..... 80 μs
Display Scan Time ..... 10.3 ms

Digit-on Time ..... 480 μs
Blanking Time ..... 160 μs
Internal Clock Cycle(5) ..... 10 μs

NOTES:
1. 8279, I_{OL} = 1.6 mA; 8279-5, I_{OL} = 2.2 mA.
2. I_{OH} = −100 μA
3. 8279, V_{CC} = +5V ±5%; 8279-5, V_{CC} = +5V ±10%
4. 8279, C_L = 100 pF; 8279-5, C_L = 150 pF.
5. The Prescaler should be programmed to provide a 10 μs internal clock cycle.
6. Sampled not 100% tested. T_A = 25°C.

* For Extended Temperature EXPRESS, use M8279A electrical parameters.

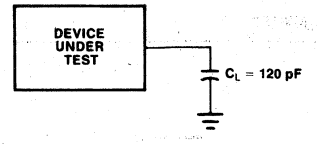
A.C. TESTING INPUT, OUTPUT WAVEFORM



290123-6

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT

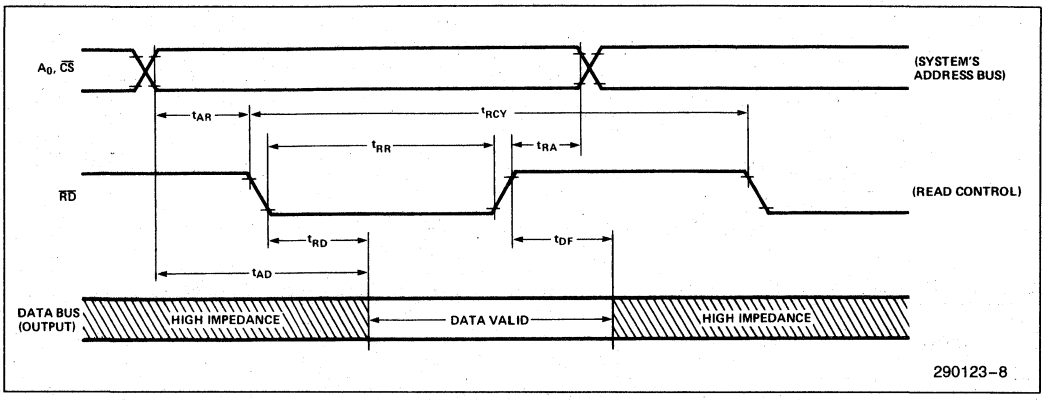


290123-7

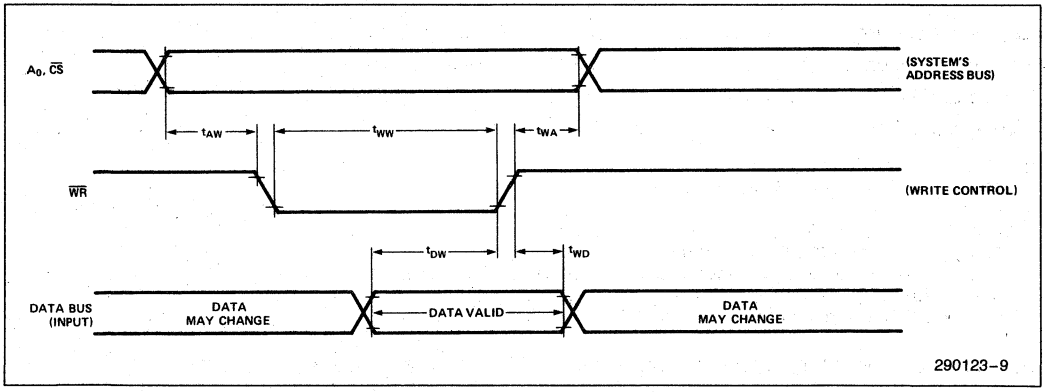
C_L = 120 pF
C_L Includes Jig Capacitance

WAVEFORMS

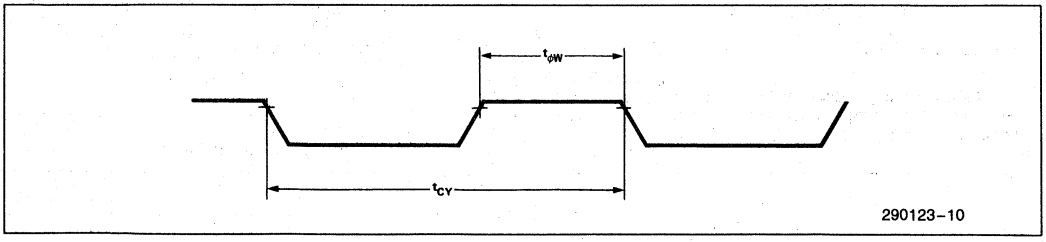
READ OPERATION



WRITE OPERATION

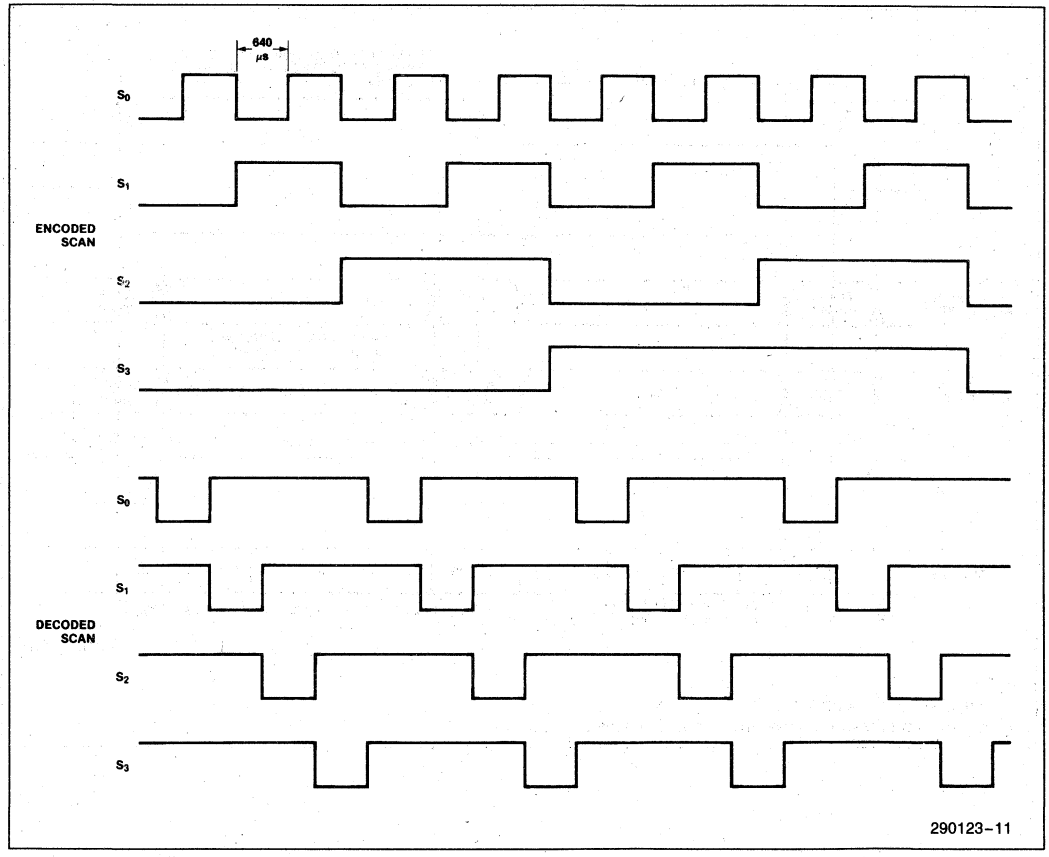


CLOCK INPUT



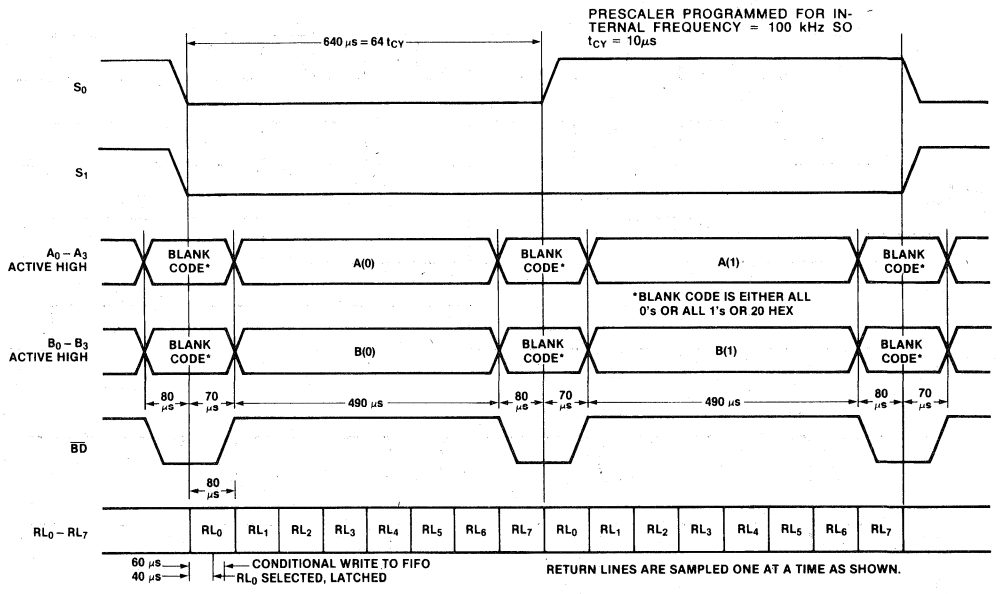
WAVEFORMS (Continued)

SCAN



WAVEFORMS (Continued)

DISPLAY



290123-12

NOTE:
Shown is encoded scan left entry
S_2–S_3 are not shown but they are simply S_1 divided by 2 and 4.

82389
MESSAGE PASSING COPROCESSOR
A MULTIBUS® II BUS INTERFACE CONTROLLER

- **Highly Integrated VLSI Device**
  - **— Single-Chip Interface for the Parallel System Bus (IEEE 1296)**
  - **— Interrupt Handling/Bus Arbitration Functions**
  - **— Dual-Buffer Input and Output DMA Capabilities**
  - **— Nine 32-Byte High Speed FIFOs**
- **Multiple Interface Support**
  - **— Complete Protocol Support of the PSB Bus (Message Passing)**
  - **— Processor Independent Interface (8-, 16-, or 32-Bit CPU)**
  - **— Low-Cost 8-Bit Microcontroller Interface**
  - **— Dual-Port Memory Interface**
- **High Performance Coprocessing Functions**
  - **— Offloads CPU for Communication and Bus Interfacing**
  - **— 40 Megabytes/sec Burst Transfer Speed**
  - **— Optimized for Real-Time Response (Max. 900 ns for 32-Byte Interrupt Packet)**
- **CMOS Technology**
- **149-Pin PGA Package (15 x 15 Grid)**

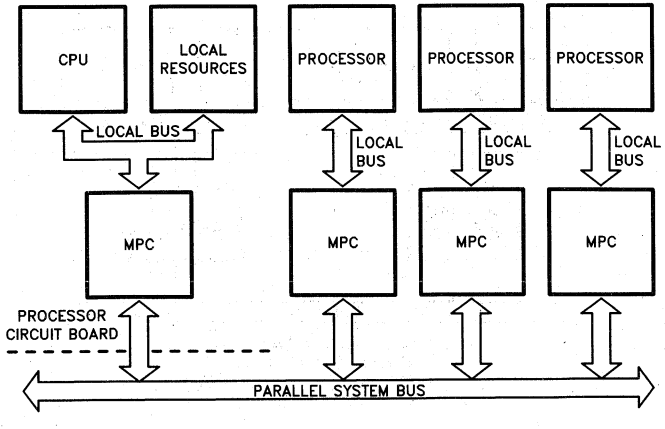
3

The MPC 82389 is a highly integrated VLSI device that maximizes the performance of a Multibus® II based multiprocessor system. It integrates the functions of bus arbitration, data transmit packetizing, error handling and interrupt control. Because of these integrated functions, the host CPU can be offloaded to utilize the maximum bus performance and subsequently increase the system throughput. The MPC 82389 also supports geographic addressing by providing access to the local interconnect registers for reference and control.

The MPC 82389 is designed to interface with an 8-, 16-, or 32-bit processor. The Parallel System Bus (PSB) performance is not affected by the CPU buswidth or bandwidth. The data on the PSB is burst transferred at the maximum bus speed of 40 Megabytes/second regardless of CPU bus performance. Such performance is possible due to decoupling of the CPU from the PSB.

This data sheet is supplemented by a *MPC User's Manual*, Intel literature number 176526-002. The *MPC User's Manual* provides detailed information regarding hardware and software board design information. In addition, the IEEE 1296 specification can provide more information regarding the MULTIBUS II bus architecture.

MULTIPROCESSOR ARCHITECTURE



290145-1

April 1990
Order Number: 290145-004

1.0 MPC 82389 INTRODUCTION

The 82389 Message Passing Coprocessor (MPC) is a highly integrated CMOS VLSI device which interfaces any microprocessor to the MULTIBUS II Parallel System Bus (PSB). The PSB is defined for easy access and sharing of resources in a processing environment which allows the existence of both intelligent and non-intelligent add-in boards. The MPC complements the MULTIBUS II environment by providing an optimized interface for the PSB at its maximum bandwidth. The MPC also offloads the host CPU, thus increasing system throughput, by providing the necessary bus arbitration, message passing protocol, error handling and interrupt control for a MULTIBUS II system. Figure 1-1 shows an example of the MPC's message passing performance.

1.1 Functional Overview

The MPC 82389 is a bus interface controller which offloads the host CPU for interprocessor communication on the PSB. The MPC 82389 features four interfaces which support a variety of data transfer operations.

1.1.1 MPC 82389 INTERFACES

The three primary interfaces to the MPC (PSB Interface, Host CPU Interface and Interconnect Interface) all function asynchronously to one another. This is accomplished through the use of internal latches and FIFOs that allow references to occur simultaneously on all interfaces. In addition to the three primary interfaces, the MPC contains a Dual-Port Interface which provides compatibility with past system implementations and software.

—PSB Interface
The PSB Interface is the synchronized, shared data pathway in the MULTIBUS II system.

—Host CPU Interface
The Host CPU Interface is a set of addressable registers and ports that is the private pathway for the local microprocessor on the MULTIBUS II board.

—Interconnect Interface
The Interconnect Interface provides a path for added board functionality that is independent from the host CPU.

—Dual-Port Interface
The Dual-Port Interface supports shared memory references.

1.1.2 MAJOR OPERATIONS

—Unsolicited and Solicited Message Passing
The unsolicited and solicited message passing protocol is an interprocessor communication protocol which allows an intelligent agent* on the PSB to communicate with another agent without any CPU intervention at full PSB speed.

—PSB Memory and I/O Single Cycle Access
The MPC performs single cycle read/write transfers from the host to memory and I/O locations across the PSB. The MPC handles bus arbitration, parity generation and error detection without CPU intervention.

—Local Interconnect Access
The host CPU and other agents on the PSB can access local interconnect space via the MPC.

*An agent is any device with an interface to the PSB.

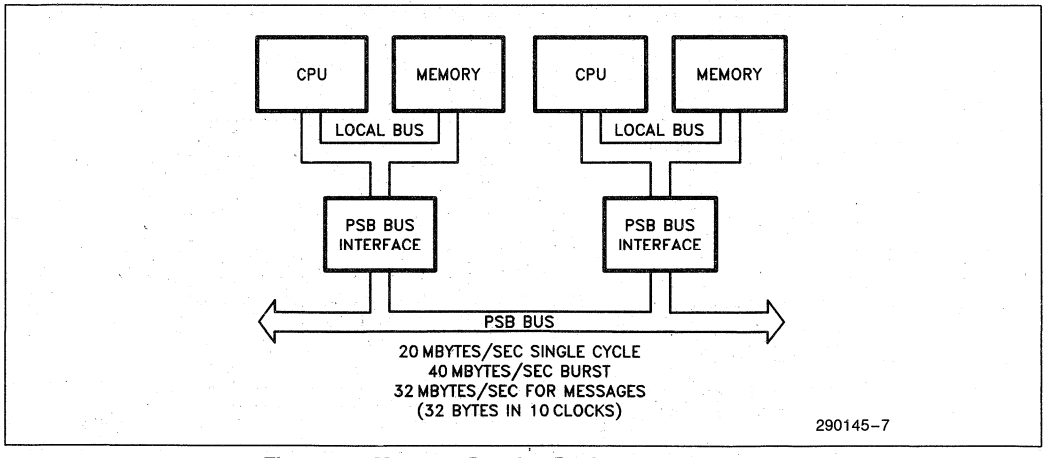


Figure 1-1. Message Passing Performance Example

—Remote Interconnect Access
The MPC enables the host CPU to access remote interconnect locations assigned to other PSB agents.

—Dual-Port Memory Access Support
Other PSB agents can access dual-port memory via the MPC.

—Central Services Module (CSM) support
The MPC has a minimal set of built-in CSM support features which allow the CSM to be incorporated into any MULTIBUS II board design.

2.0 MESSAGE PASSING PROTOCOL

The MULTIBUS II architecture designates the data transfer protocol between agents on the PSB as message passing. Message passing allows agents to transfer variable amounts of data at maximum PSB speed. The MPC fully supports the PSB's standardized message passing protocol. The entire handshaking procedure between agents on the PSB is handled by the MPC without CPU intervention.

There are two types of messages that can be transmitted from one agent to another: Unsolicited Messages and Solicited Messages.

2.1 Unsolicited Messages

Unsolicited messages are short, fixed-length messages that can arrive unexpectedly. Unsolicited messages can be transmitted without explicit buffer allocation and without synchronization between sending and receiving agents on the PSB. Unsolicited messages are often referred to as intelligent or virtual interrupts, since they can be used as a signaling mechanism between boards, replacing traditional system interrupts and freeing the CPU from having to poll for information. In addition, unsolicited messages allow for up to 28 bytes of user data.

2.2 Solicited Messages

Solicited messages are used to transfer large amounts of data. Up to 16 Mbytes (less 1 byte) of data can be transferred in a single solicited message transmission sequence. Solicited message transfers require the receiving agent to explicitly allocate a buffer. Buffer negotiation between sending and receiving agents is handled using unsolicited messages as follows:

- A buffer request message initiates a solicited message transfer. It requests the receiving agent to allocate a buffer large enough to hold the solicited data.
- A buffer grant message must be returned by the receiving agent before the solicited data can be transferred. The buffer grant informs the sending agent's MPC that a buffer has been allocated and indicates that the receiving agent's MPC is ready to begin the data transfer.
- A buffer reject message is returned by the receiving agent if a buffer for the solicited data cannot be provided. In this case, the rejection is final, and no further action is required.

If a DMA controller handles the solicited message transfer, DMA controller setup is also needed. Typically, the sending agent programs its DMA controller immediately before sending a buffer request, and the receiving agent programs its DMA controller immediately before sending a buffer grant.

Once solicited buffer negotiation is complete (the sending agent's MPC has received a buffer grant), the agents transfer the data without further intervention. The data is sent as a series of solicited packets on the sending agent's local bus. The MPCs perform transfer and routing across the PSB automatically. At the end of the solicited data transfer, both the sending and receiving agents get a completion indication from their local MPC.

3.0 MPC 82389 INTERFACES

The MPC 82389 features a total of 4 interfaces. The three primary interfaces are the Host CPU Interface, PSB Interface and the Interconnect Interface. The MPC also has a Dual-Port Memory Interface which provides compatibility with past system implementations and software. Figure 3-1 shows the four MPC bus interfaces.

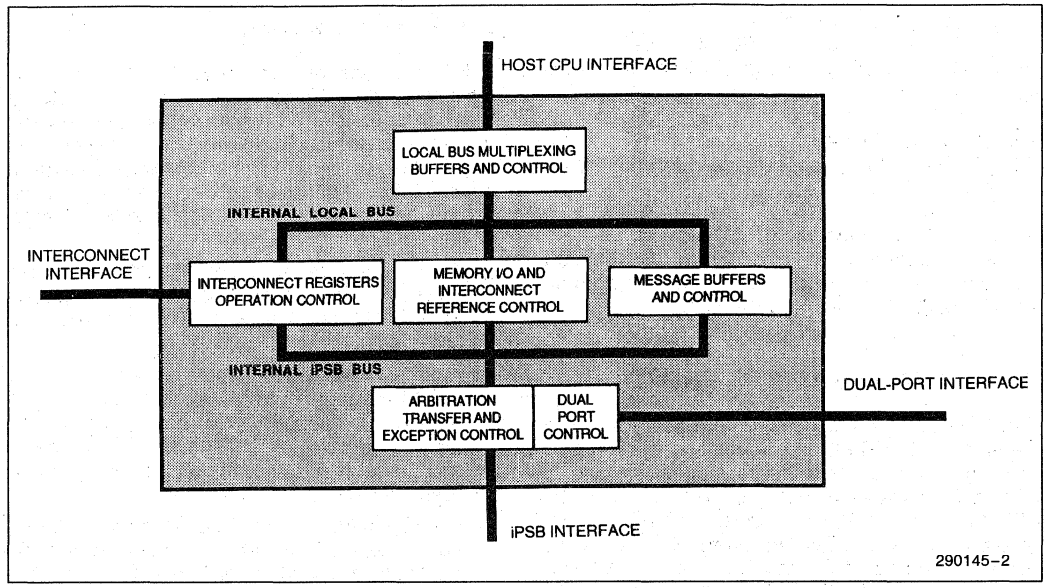


Figure 3-1. MPC Bus Interfaces

3.1 Host CPU Interface

The Host CPU Interface connects an 8-, 16-, or 32-bit processor to the MPC. The Host CPU Interface supports direct references to memory, I/O, and interconnect address space on the PSB. The entire Host CPU Interface is composed of three sub-interfaces: Register Sub-Interface, Reference Sub-Interface and DMA Sub-Interface.

—Register Sub-Interface
The Register Sub-Interface is composed of a bank of 8-bit registers on the Host CPU Interface. These registers provide the configuration, status and command interface for the host CPU. A host register operation is independent from operations which may be in progress at the MPC's other interfaces. However, some host register operations are dependent on the internal state of the MPC. In host register operations, the maximum duration is decided by the strobe width. Thus, the number of wait states required at the local interface is under the control of the host CPU.

—Reference Sub-Interface
The Reference Sub-Interface supports direct references to memory, I/O, and interconnect address space on the PSB. Memory and I/O references are initiated by the CPU to the MPC. The MPC responds to a memory or I/O reference by putting the CPU on hold while arbitrating for the PSB. The CPU is held in wait states until the reference is complete or until a bus exception condition occurs on the PSB. The Reference Sub-Interface supports both read and write operations to the registers. The local interconnect address space is differentiated from the interconnect address on the PSB by the bit pattern stored in the MPC's slot address register.

—DMA Sub-Interface
The DMA Sub-Interface supports data transfers between the local memory and the MPC during solicited message operations. The DMA Interface is designed to support either two-cycle or fly-by (single-cycle) read/write transfers. For two-cycle operations, the DMA controller performs one cycle into memory and another cycle to the MPC; a read command is used to get data from the MPC and a write command is used to put data into the MPC. Fly-by operations allow data to be transferred during a single bus cycle; a fly-by transfer will use a write command to get data from the MPC (corresponding to a memory write) and a read command to put data into the MPC (corresponding to a memory read). The higher performance possible with fly-by transfers mandates the alignment of data on 4-byte boundaries.

3.2 Parallel System Bus Interface

The Parallel System Bus (PSB) Interface is a full 32-bit interface to other boards in the MULTIBUS II chassis. The PSB Interface supports PSB arbitration, data transfer and error handling.

—Parallel System Bus Arbitration
The MPC begins PSB access arbitration upon a request which is generated inside the MPC. This request could be the result of a synchronized PSB memory, I/O or interconnect reference request or a message packet transmit request from the CPU.

—Data Transfer
The PSB Interface contains all the address/data lines and necessary control signals for data transfer. These control signals provide the control mechanism between agents during transfer operations.

—Error Handling
The MPC monitors errors generated during data transfer operations. The MPC recognizes data integrity problems on the PSB and bus timeout conditions.

3.3 Interconnect Interface

The Interconnect Interface is an independent 8-bit communication interface which allows the MPC to be connected to a microcontroller. (It is highly recommended that an 8051 or similar microcontroller be used on the Interconnect Interface.) This microcontroller will perform tasks such as board configuration at startup and local diagnostics.

The interconnect space of an agent is the only required bus space by the IEEE 1296 specification and has a 512-byte register range. Within this space the microcontroller can store the local operating and configuration parameters associated with the agent. For example, local diagnostics can be executed out of the microcontroller and the results posted in the interconnect space.

Local resources on an agent gain access to interconnect space through the MPC's interconnect bus. A microcontroller connects to the interconnect bus for intelligent handling of interconnect operations. All interconnect bus signals are asynchronous to the bus clock and to the local bus signals.

3.4 Basic Implementation with the MPC 82389

Figure 3-2 shows a basic implementation of the MPC 82389. Included in this implementation is the Interconnect Interface, the Host CPU Interface and the PSB Interface.

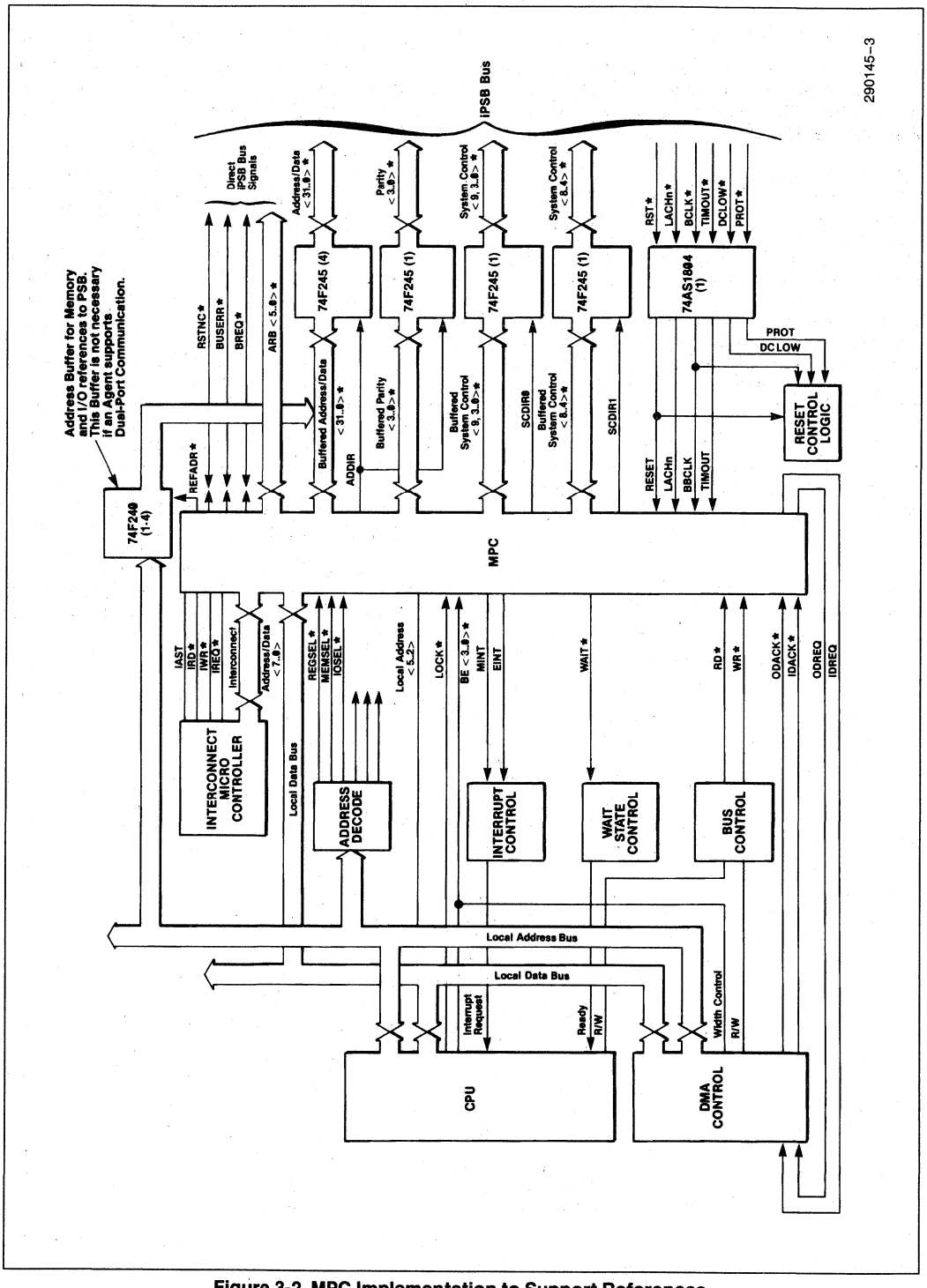


Figure 3-2. MPC Implementation to Support References

3.5 Dual-Port Interface

The Dual-Port Interface supports shared memory accesses between agents on the PSB. In order to fully implement dual-port memory, some additional dual-port memory controller logic is required. Figure 3-3 shows an example of the MPC implemented with dual-port memory.

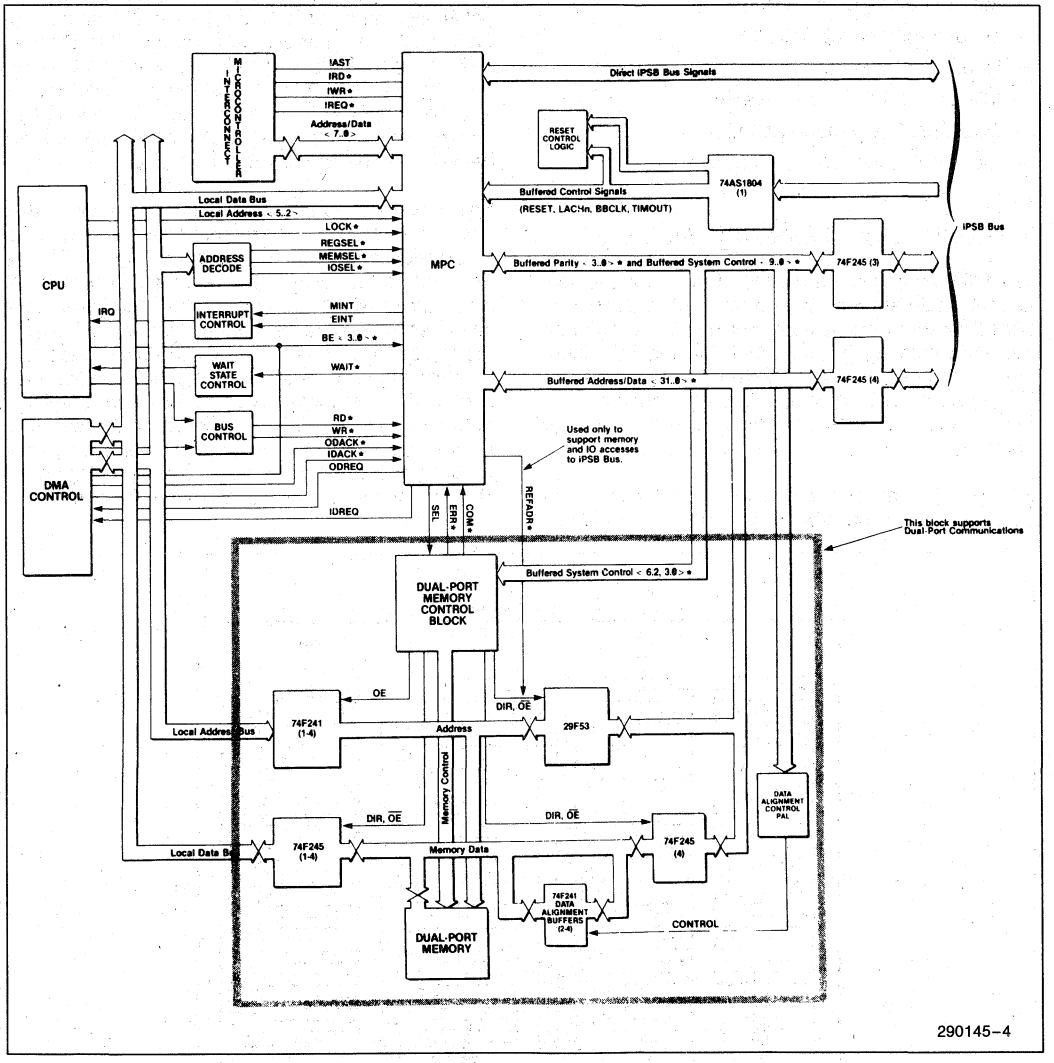


Figure 3-3. The MPC Implemented with Dual-Port Memory

4.0 MPC 82389 OPERATIONS

The primary function of the MPC 82389 is MULTIBUS II message passing. In addition to message passing, the MPC performs the following functions:

— Memory and I/O Reference
— Local Interconnect Reference
— Remote Interconnect Reference
— Interconnect Replier Operations
— Dual-Port Replier Operations
— Central Services Module Support

4.1 MULTIBUS II Message Passing

The MPC manages the routing of message packets as they flow between the interfaces of each MULTIBUS II agent in the system. For message traffic on the PSB, message decode logic on the PSB input bus determines message routing through the MPC. For the Host CPU Interface and Interconnect Interface, the MPC defines a signal protocol for message passing.

MULTIBUS II messages, both unsolicited and solicited, are transferred through nine dedicated internal FIFO buffers between the Host CPU Interface and PSB Interface. Unsolicited messages are intelligent (also called virtual) interrupts which notify the receiving agent to prepare for the receipt of solicited messages. Unsolicited messages use the Transmit/Error FIFO and the Receive FIFO. The Transmit FIFO holds a 32-byte packet for transmittal across the PSB. If there is an error in transmission, the Transmit FIFO becomes the Error FIFO, where the errant message can be read back along with error status. The Receive FIFO is a circular queue of four 32-byte buffers from which unsolicited messages are received from the PSB by the host CPU.

Solicited messages consist of information data packets which are transmitted between agents. Solicited messages use the Solicited Input FIFO and Solicited Output FIFO. These FIFOs are dual 32-byte buffers which are used for the temporary storage of solicited data packets as they travel between the Host CPU Interface and the PSB Interface. The solicited output header logic attaches header information to the solicited data packet before sending it onto the PSB. All FIFOs are able to operate independently and concurrently, thus creating a true multitasking message passing environment. Figure 4-1 shows the nine dedicated internal FIFO buffers.

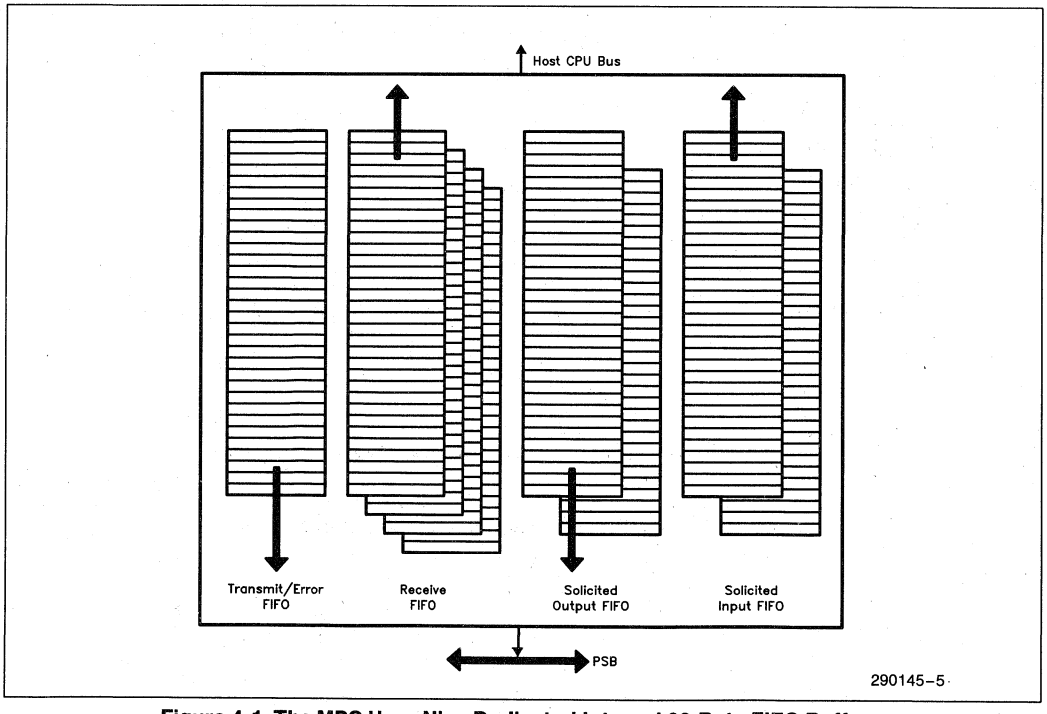


Figure 4-1. The MPC Uses Nine Dedicated Internal 32-Byte FIFO Buffers

4.1.1 UNSOLICITED TRANSMIT/RECEIVE

Unsolicited message passing sequences occur between the Host CPU Interface and the PSB Interface using FIFOs internal to the MPC. FIFO status is available on the Host CPU Interface and in state machines internal to the MPC. On the Host CPU Interface, host register operations write bytes to the Transmit FIFO and read bytes from the Receive FIFO. On the PSB, the MPC manages the emptying and filling of the Transmit and Receive FIFOs using MULTIBUS II message passing protocol and the Transmit and Receive FIFOs on another agent's MPC. For detailed information about message passing protocol across the PSB, see the *IEEE 1296 High Performance Synchronous 32-bit Bus Standard.*

4.1.2 SOLICITED INPUT/OUTPUT

Solicited transfers are pre-negotiated using unsolicited message sequences. Dedicated FIFOs (Solicited Input FIFO and Solicited Output FIFO) are then used for the transfer of solicited data packets. This allows large amounts of data to be moved between agents independently of unsolicited messages. In most cases, the solicited transfer occurs under DMA control, freeing the host CPU to handle other activities. The DMA controller uses the input channel DMA request/acknowledge and output channel DMA request/acknowledge signals along with the read/write signal to stream the data from/to the solicited FIFOs. On the PSB, the data is transferred in bursts using MULTIBUS II message passing protocol and similar solicited FIFOs on another agent's MPC. The MPCs add header information to the packets on the PSB, indicating source, destination and length. Data transfers through the solicited FIFOs can be set up for 8, 16 or 32 bits of data width on the Host CPU Interface, but occur at full 32-bit width on the PSB.

4.2 Memory and I/O References

Remote memory or I/O reference operations are Host CPU Interface operations that involve an access through the MPC to a resource across the PSB. This resource can be a dumb memory or I/O board. The remote reference can only be done through the MPC as a single cycle operation (no block transfers) to the remote resource and can involve an unknown number of wait states. Many MULTIBUS II CPU boards use an alternate path (such as the iLBX™ bus found on Intel iSBC® boards) that is an independent extension of the local bus for full-speed and block transfer operations.

The host CPU initiates a memory or I/O reference by activating memory select ($\overline{\text{MEMSEL}}$†) or I/O select ($\overline{\text{IOSEL}}$), A<5–2>, $\overline{\text{BE}}$<3–0>, with a $\overline{\text{RD}}$ or $\overline{\text{WR}}$ strobe. If necessary, $\overline{\text{LOCK}}$ is activated to allow back-to-back accesses across the PSB, holding all other agents off the memory or I/O resource. The MPC activates its $\overline{\text{WAIT}}$ output to indicate that the operation is in progress.

The data for reference operation proceeds through the MPC and PSB to a memory or I/O address on another agent. A data path from D<31–0> through the buffered address/data bus ($\overline{\text{BAD}}$<31–0>) is used for the data transfer. Data is latched internally in a reference data latch. Parity is generated to the PSB on BPAR <3–0> for the data on each write operation and checked on data read. Completion of the operation is indicated when the MPC deactivates the $\overline{\text{WAIT}}$ output.

The memory or I/O address for the reference operation is routed around the MPC through an external reference address latch. This latch is controlled by the REFADR signal from the MPC.

4.3 Local Interconnect Reference

A local interconnect reference operation is an access by the host CPU to the interconnect records maintained by the local interconnect microcontroller. The geographic interconnect address is preloaded into a pair of registers internal to the MPC. The upper 5 bits of the interconnect address determine whether the operation is local or remote. A data path from D<7–0> to the interconnect address/data bus (IAD<7–0>) is used. The microcontroller uses the interconnect request ($\overline{\text{IREQ}}$) output to sense the request. The request is serviced by the interconnect microcontroller through a sequence of accesses to registers within the MPC using the interconnect address strobe (IAST), interconnect read ($\overline{\text{IRD}}$), and interconnect write ($\overline{\text{IWR}}$) strobes, and the IAD multiplexed bus. The $\overline{\text{WAIT}}$ signal is used as for memory and I/O references to indicate completion of the local interconnect reference operation.

4.4 Remote Interconnect Reference

A remote interconnect reference is an access by the host CPU to interconnect space on another agent. The host CPU requests a remote interconnect reference by writing the interconnect address to the same register used in the local interconnect request, except that the upper 5 bits of the interconnect address indicate the slot address of another agent on the PSB. The data flows through the MPC as in a remote memory or I/O reference, except that the data transfer occurs only on D<7–0>. The remote microcontroller services the request through an interconnect replier operation.

†* indicates that the signal is active low.

4.5 Interconnect Replier Operations

When another agent performs a remote interconnect reference request, it gains access to local interconnect space through the MPC. The MPC decodes an interconnect request on the PSB for a slot ID match and signals the interconnect microcontroller independently of the local bus interface. The microcontroller then handles the request in the same way as a local interconnect request.

4.6 Dual-Port Replier Operations

Other agents can access dual-port memory via the MPC. A memory access request on the PSB is decoded by the MPC for an address range match and serviced by the dual-port controller (external circuitry must be provided). The MPC provides only the handshaking path. Data transfer occurs directly on the $\overline{BAD}$ bus. If a bus exception occurs while a dual-port memory reference is in progress, the MPC will signal the dual-port controller to terminate the operation.

4.7 Central Services Modular Support

The IEEE 1296 specification defines the Central Services Module (CSM) that resides in Slot 0 of a MULTIBUS II system. The CSM is responsible for these functions:

— reset sequencing (generates reset signal on the PSB)

— assignment of card slot and arbitration IDs during reset initialization

— generation of system wide clocks for all agents (bus clocks and time of day)

— generation of bus timeout

— battery back-up of system constants (host ID, time of day, etc.)

The MPC has a minimal set of built-in CSM support features that allow the incorporation of CSM into any MULTIBUS II board design. The MPC, interconnect microcontroller, and a small amount of external circuitry can fully implement the CSM automatically when the board is inserted into Card Slot 0.

4.7.1 ADDITIONAL CSM REQUIREMENTS

In addition to the interconnect microcontroller and the MPC, the following functions must be provided through external logic:

— clock generation

— PSB reset generation

— cold/warm start detection

— PSB timeout generation

The clock generator provides the bus clock ($\overline{BCLK}$) and central clock ($\overline{CCLK}$) signals to the PSB. The reset generator provides the hardware reset line ($\overline{RESET}$) to all agents on the PSB. Cold/warm start detection circuitry distinguishes between a power-up reset and a warm-start reset; on power-up the CSM assigns arbitration and slot IDs. The PSB timeout function determines when the PSB is hung.

See the *MPC User's Manual* (Intel literature number 176526-002) and the *CSM\002 Hardware Reference Manual* (Intel literature number 459706-001) for more information about the CSM.

5.0 MPC 82389 PIN DESCRIPTIONS

This section describes each signal pin (or group of pins) on the MPC. Emphasis is placed on giving as much information as possible to ease the task of designing hardware associated with the MPC signal pins. The pins are described in terms of these functional groups:

- PSB interface
- local bus (host CPU) interface
- dual-port memory control
- interconnect bus interface

5.1 PSB Signals

The PSB signals provide the interface to other boards in the MULTIBUS II chassis. Very little support circuitry is required for this part of the board. Only high-current drivers and reset control logic is needed. Some MPC signal pins have built-in open collector high-current drivers that allow connection directly to the PSB. For complete information on the PSB, see the *IEEE 1296 High Performance Synchronous 32-bit Bus Standard* document.

PSB signals fall into five groups, depending on function:

- arbitration operation signal group
- address/data bus signal group
- system control signal group
- central control signal group
- exception operation signal group

Unless otherwise stated, all PSB signals are synchronous to the bus clock.

NOTE:
High current drivers used to drive the buffered address/data ($\overline{BAD}$) bus should be controlled with minimal logic. This is to limit propagation delays and avoid possible bus contention problems. Ensure that the placement of these drivers and the MPC is done as close to the PSB (the P1 connector on a MULTIBUS II board) as possible to minimize signal stub lengths and capacitive loading.

5.1.1 ARBITRATION OPERATION SIGNAL GROUP

These MPC pins are used by an agent to obtain exclusive access to the PSB. They are all high-current drive, open-collector signals. Below is a description of each signal.

$\overline{BREQ}$ (Bus Request). $\overline{BREQ}$ is a bidirectional open-collector signal that connects directly to the PSB. As an input to the MPC, it indicates that agents are awaiting access to the bus. As an output, the MPC asserts $\overline{BREQ}$ to request PSB access.

$\overline{ARB}$<5-0> (Arbitration). $\overline{ARB}$<5-0> are bidirectional, open-collector signals that connect directly to the PSB. $\overline{ARB}$<5-0> are used (during normal operation) to identify the mode and arbitration priority of an agent during an arbitration cycle. During system initialization (while reset is active), the central services module (CSM) drives these signals to initialize slot and arbitration IDs.

5.1.2 ADDRESS/DATA BUS SIGNAL GROUP

This signal group includes a 32-bit multiplexed address/data path ($\overline{BAD}$<31-0>) and the byte parity signals ($\overline{BPAR}$<3-0>). These signals require buffering through bus transceivers before connection to the PSB. This signal group also includes the bus transceiver control signals (ADDIR and $\overline{REFADR}$).

$\overline{BAD}$<31-0> (Buffered Address/Data). $\overline{BAD}$<31-0> are the 32 bidirectional, multiplexed address/data signals that provide the interface to the PSB address/data bus ($\overline{AD}$) when buffered through 74F245 or equivalent bus transceivers.

NOTE:
Do not use pull-up resistors to drive the BAD bus high. If pull-up resistors are present, the MPC cannot guarantee valid logic states with proper timing.

$\overline{BPAR}$<3-0> (Buffered Parity). $\overline{BPAR}$ are four signals that provide parity for the 32-bit $\overline{BAD}$ bus. These bidirectional lines connect to the PSB $\overline{PAR}$<3-0> signals through a 74F245 or equivalent transceiver. These signals are used to receive byte parity for incoming data and to drive byte parity for outgoing data.

ADDIR (Address/Data Direction). ADDIR is an output that provides direction control over the bus transceivers buffering the $\overline{BAD}$<31-0> and $\overline{BPAR}$<3-0> signals. In the high state, this signal causes the transceivers to drive address/data information along with parity onto the PSB. In the low state, this signal causes address/data information and parity to be received from the PSB.

$\overline{REFADR}$ (Reference Address Enable). $\overline{REFADR}$ is an output used to enable external reference address buffers during reference operations. Asserting this signal places the reference address onto the $\overline{BAD}$ bus. The address path enabled by this signal is only used for memory and I/O reference operations to the PSB. It is not used during message passing or for PSB references to interconnect space.

5.1.3 SYSTEM CONTROL SIGNAL GROUP

The system control signal group on the PSB provides a control mechanism between agents during transfer operations.

$\overline{BSC}$<9-0> (Buffered System Control). $\overline{BSC}$<9-0> is a group of ten bidirectional signals that connect to the PSB through 74F245 or equivalent transceivers. Agents on the PSB use these signals for commands or status, depending on the phase of the operation. The function of each of these lines during request and reply phases of transfer operations is summarized in Table 5-1.

Table 5-1. Summary of $\overline{BSC}$ Signal Functions

Signal	Request Phase	Reply Phase
$\overline{BSC0}$	Bus Owner in Request Phase	Bus Owner in Reply Phase
$\overline{BSC1}$	LOCK	LOCK
$\overline{BSC2}$	Data Width	End-of-Transfer
$\overline{BSC3}$	Data Width	Bus Owner Ready
$\overline{BSC4}$	Address Space	Replying Agent Ready
$\overline{BSC5}$	Address Space	Agent Status
$\overline{BSC6}$	Read/Write Data Transfer	Agent Status
$\overline{BSC7}$	Reserved	Agent Status
$\overline{BSC8}$	Even Parity on BSC<7-4>	Even Parity on BSC<7-4>
$\overline{BSC9}$	Even Parity on BSC<3-0>	Even Parity on BSC<3-0>

NOTE:
The end-of-transfer (EOT) handshake in single-cycle operations is indicated by BSC<4,3,2> as follows: the requesting MPC drives $\overline{BSC}$<3,2> and waits for the replier to drive $\overline{BSC4}$; when the replier responds, the EOT handshake is complete.

SCDIR<1,0> (System Control Direction). SCDIR<1,0> are output signals that provide direction control of the 74F245 transceivers driving and receiving $\overline{BSC}$<9-0>. SCDIR0 provides control for $\overline{BSC}$<9,3-0>, while SCDIR 1 provides control for $\overline{BSC}$<8-4>. When either signal is high, the bus transceiver drives $\overline{BSC}$ signals onto the PSB. When either signal is low, signals on the PSB are driven onto the $\overline{BSC}$ lines.

5.1.4 CENTRAL CONTROL SIGNAL GROUP

The central control signal group provides bus status and control information for devices operating on the PSB. The CSM, residing in slot 0 of the MULTIBUS II backplane, generates $\overline{BCLK}$, LACHn, and RESET.

BBCLK (Buffered Bus Clock). BBCLK is received by the MPC to synchronize all operations on the PSB. This input should be connected to $\overline{BCLK}$ (on the PSB) using a 74AS1804 or equivalent inverting buffer. The falling edge of $\overline{BCLK}$ provides all system timing references. BBCLK normally has a fixed operating frequency of 10 MHz.

NOTE:
$\overline{BCLK}$ can be varied from DC to 10 MHz. You may use this feature for single-stepping on the PSB during debugging.

LACHn (ID Latch). LACHn is an input signal used during initialization of slot and arbitration IDs (where "n" is the slot number). When the RESET signal is active, LACHn indicates when a slot or arbitration ID is available and should be latched. LACHn is an active high input and should be connected to the $\overline{LACHn}$ signal on the PSB with a 74AS1804 or equivalent inverting buffer.

RESET. Reset is an input that places the MPC in a known state. Only the parts of the MPC involved with initialization of slot and arbitration IDs remain unaffected. RESET is an active high input and should be connected to the $\overline{RST}$ signal on the PSB with a 74AS1804 or equivalent inverting buffer.

If the MPC is used in a CSM implementation, the interconnect microcontroller and some external logic controls RESET. On power up, the CSM generates the RESET signal to the backplane. Within a few clock cycles, receiving MPCs complete their internal reset. Table 5-2 summarizes the states of MPC signal outputs while the RESET signal is active.

Table 5-2. Signal States During Reset

Signal	Reset State	Signal	Reset State
$\overline{BREQ}$	Z(H)	$\overline{ARB}$<5–0>	Z(H)
$\overline{BAD}$<31–0>	Z	D<31–0>	Z
ADDIR	L	$\overline{SEL}$	H
$\overline{REFADR}$	H	$\overline{WAIT}$	H
$\overline{BSC}$<9–0>	Z	ODREQ, IDREQ	L
SCDIR<1,0>	L	MINT, EINT	L
$\overline{BUSERR}$	Z(H)	$\overline{RSTNC}$	L

NOTE:
H = Electrical high state.
L = Electrical low state.
Z = High impedance (tri-state).

$\overline{RSTNC}$ (Reset Not Complete). Agents assert $\overline{RSTNC}$ during reset to extend the initialization time period beyond the time that RESET allows. $\overline{RSTNC}$ is a bidirectional OR-tied signal on the PSB that is low when one or more agents have not completed their reset requirements. Agents cannot perform bus operations while $\overline{RSTNC}$ is asserted. However, agents may access local interconnect space if your firmware implementation allows such access. $\overline{RSTNC}$ is an open-collector signal with high-current drive that connects directly to the PSB.

5.1.5 Exception Operation Signal Group

The exception operation signal group indicates exception errors on the PSB.

$\overline{BUSERR}$ (Bus Error). The MPC asserts $\overline{BUSERR}$ when a data integrity problem on the PSB is detected during a transfer operation. Possible problems are: detection of a parity error on the $\overline{BAD}$ bus or $\overline{BSC}$ lines, or a protocol error associated with the $\overline{BSC}$ lines. $\overline{BUSERR}$ is a bidirectional, open-collector signal with high current drive that connects directly to the PSB.

TIMOUT (Timeout). TIMOUT, as an input from the PSB, is used to detect a bus timeout condition. The CSM activates this signal when it determines that an agent is taking too much time asserting a handshake signal, or if a bus owner has maintained bus ownership for an excessive length of time. The exact amount of time is a fixed value relative to BBCLK that is approximately 10,000 clock cycles (1 ms @ 10 MHz). TIMOUT is an active high input to the MPC and must be connected to the $\overline{TIMOUT}$ signal of the PSB through a 74AS1804 or equivalent inverting buffer.

When the MPC is configured for CSM operation, TIMOUT becomes an output, generating the timeout condition to all agents on the PSB. In this case, the TIMOUT pin should be connected to the PSB by a 74F242 driver or equivalent.

5.2 Dual-Port Memory Control Signals

The MPC provides these signals ($\overline{SEL}$, $\overline{COM}$, $\overline{ERR}$) to support dual-port memory. In order to fully implement dual-port memory, some additional dual-port memory controller logic is required.

$\overline{SEL}$ (Select). The $\overline{SEL}$ output indicates that a dual-port memory access is in progress. $\overline{SEL}$ initiates dual-port operations and may be used to enable the dual-port data buffers onto the $\overline{BAD}$ bus. When the MPC receives the EOT handshake, or if the MPC detects an exception, it deactivates $\overline{SEL}$.

$\overline{COM}$ (Complete). $\overline{COM}$ is an input to the MPC. The dual-port memory controller asserts $\overline{COM}$ to indicate completion of a dual-port access. $\overline{COM}$ is assumed to be synchronous to the bus clock. After the memory controller has asserted $\overline{COM}$, the MPC asserts the replier ready ($\overline{BSC4}$) signal on the next bus clock. The memory controller cannot deassert $\overline{COM}$ until the EOT handshake is complete on the PSB. This requires that the memory controller monitor the PSB for the EOT handshake.

$\overline{ERR}$ (Error). $\overline{ERR}$, an input to the MPC, is asserted by the dual-port memory controller to signal a memory data parity error. $\overline{ERR}$ must be stable (high or low) whenever $\overline{COM}$ is asserted. The MPC responds to this signal by completing the replier handshake on the PSB using a *data error* agent error code. This signal may be asynchronous to the bus clock since it is qualified by the $\overline{COM}$ signal.

5.3 Local Bus Signals

The MPC local bus allows many types of microprocessors, perhaps with differing data widths, byte alignment, and bit ordering, to connect to the MULTIBUS II PSB. This microprocessor is often referred to as the *host CPU* on the MULTIBUS II processor board. The MPC has five signal groups on the local bus:

- data bus
- address/status signals
- transfer control
- interrupt signals
- DMA control lines

5.3.1 DATA BUS

The local data bus is the signal path for data transfers between the host CPU and the MPC.

D<31–0>. D<31–0> is the 32-bit local data bus. Although this is a 32-bit interface, the MPC allows operation with processors using 8-, 16-, or 32-bit data busses.

NOTE:
Intel CPU architecture defines bit 0 and byte 0 as least significant. When connecting non-Intel processors to the MPC local data bus, it is important that this bit and byte ordering be maintained across the PSB. This allows agents of differing CPU types to work together in a single chassis. If byte-swapping is needed, see the discussion of the *byte enable* (BE<3–0>) signal pins.

5.3.2 ADDRESS/STATUS SIGNALS

The address/status signals select or identify all MPC operations over the local bus.

A<5–2> (Address). The address inputs select MPC registers for message and interconnect space operations. A1 and A0 are omitted to provide a consistent register address for all data bus width options. A <5–2> are qualified by $\overline{RD}$ or $\overline{WR}$ and therefore must be stable within the specified set-up and hold window.

$\overline{MEMSEL}$ (Memory Select). This MPC input signal tells the MPC that the current operation is a memory reference across the PSB. $\overline{MEMSEL}$ is qualified by $\overline{RD}$ or $\overline{WR}$ and therefore must be stable within the specified set-up and hold window.

NOTE:
$\overline{MEMSEL}$, $\overline{IOSEL}$, $\overline{REGSEL}$ $\overline{IDACK}$, and $\overline{ODACK}$ are mutually exclusive. In order to be valid, no more than one should be active during the same set-up and hold window.

$\overline{IOSEL}$ (I/O Select). This input signal tells the MPC that the current operation is an I/O reference to the PSB. $\overline{IOSEL}$ is qualified by $\overline{RD}$ or $\overline{WR}$ and therefore must be stable within the specified set-up and hold window.

$\overline{REGSEL}$ (Register Select). This input signal is used to identify MPC register operations. $\overline{REGSEL}$ is qualified by $\overline{RD}$ or $\overline{WR}$ and therefore must be stable within the specified set-up and hold window.

$\overline{LOCK}$. This input signal allows back-to-back operations to be performed on the PSB or local interconnect space. When the bus owner activates $\overline{LOCK}$, all other agents are held off the PSB or local resource until $\overline{LOCK}$ is deactivated.

$\overline{BE}$<3–0> (Byte Enable). These input signals, generated by the host CPU or DMA controller, validate bytes on the data bus. $\overline{BE}$<3–0> are qualified by $\overline{RD}$ or $\overline{WR}$ and therefore must be stable within the specified set-up and hold window. $\overline{BE}$<3–0> correspond to data bytes 3 through 0 on the data bus (where byte 3 is D<31–24>). For remote reference operations, only combinations supported by the IEEE 1296 specification are valid.

A 32-bit local bus requires that all byte enable and data signals are used. For 16-bit local buses, $\overline{BE1}$ and $\overline{BE2}$ are used to indicate which of the two bytes will contain valid data, and only D<15–0> are used. For 8-bit local bus operations, $\overline{BE1}$ and $\overline{BE0}$ are used to select which byte of the PSB will carry the valid data byte. This mode uses ony D<7–0> (on the local bus). Note that during all read operations, the MPC drives all data lines (D<31–0>). Consecutive accesses to message FIFOs must be in ascending byte sequence 0, 1, 2, 3 in any non-overlapping combination.

Table 5-3 shows the valid byte enable combinations for both the local data bus (D<31–0>) and the PSB ($\overline{AD}$<31–0>):

Table 5-3. Valid Byte Enable Combinations

$\overline{BE3}$	$\overline{BE2}$	$\overline{BE1}$	$\overline{BE0}$	D31–24	D23–16	D15–8	D7–0	AD31–24	AD23–16	AD15–8	AD7–0
L	L	L	L	V3	V2	V1	V0	V3	V2	V1	V0
L	L	L	H	V3	V2	V1	X	V3	V2	V1	X
H	L	L	L	X	V2	V1	V0	X	V2	V1	V0
L	L	H	H	V3	V1	X	X	X	X	V3	V2
H	L	L	H	X	V2	V1	X	X	V2	V1	X
H	H	L	L	X	X	V1	V0	X	X	V1	V0
L	H	H	H	V3	X	X	X	X	X	V3	X
H	L	H	H	X	V2	X	X	X	X	X	V2
H	H	L	H	X	X	V1	X	X	X	V1	X
H	H	H	L	X	X	X	V0	X	X	X	V0
L	H	L	H	X	X	X	V0	X	X	V0	X
L	H	H	L	X	X	X	V0	X	X	X	V0

NOTES:
L = Electrical low state (active)
H = Electrical high state (inactive)
Vn = Valid data bytes
X = Active bytes with undefined data

3

For the 32-bit host interface, legal combinations of byte enables form *byte lanes:* the paths where valid data bytes are present during a single transfer on the local data bus (as well as in the MULTIBUS II environment). Non-Intel Microprocessors can use byte lanes to perform byte-swapping or other data manipulations in hardware. The figure below illustrates the legal byte lanes as they relate to byte enable combinations:

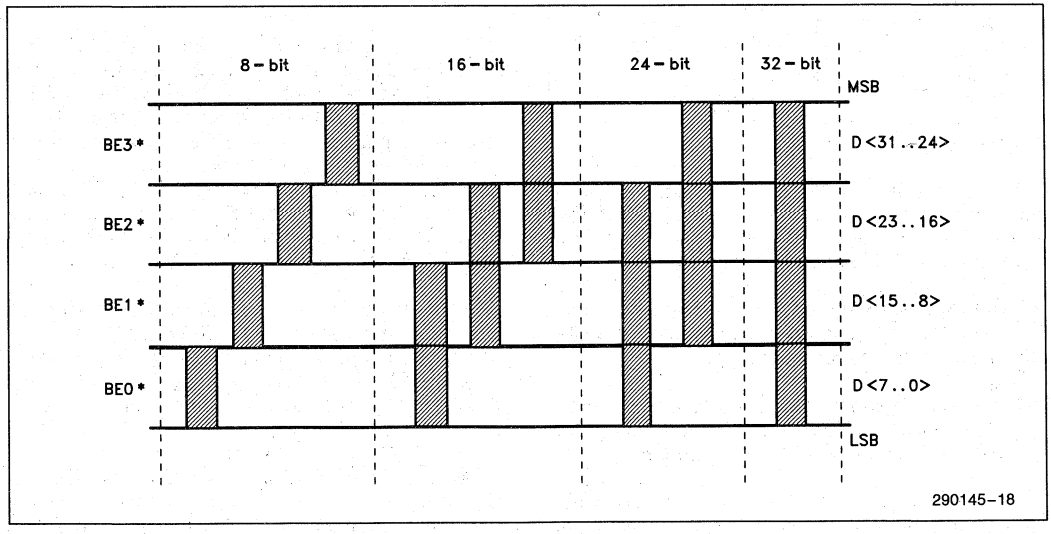


Figure 5-4. Byte Lanes

Each shaded box in Figure 5-4 represents a valid byte lane for a given combination of $\overline{BE}$ during a single read or write operation. There are four types of byte lanes: 8-bit, 16-bit, 24-bit and 32-bit. Bit and byte ordering follow the Intel standard of bit or byte 0 as least significant. Assume that invalid byte lanes contain any value of data (i.e. non-constant). Take precautions (masking in software, etc.) to ensure that invalid data does not cause problems.

When using a DMA controller to handle solicited data transfers to/from local memory, misalignment of data in memory and resulting partial packets are handled using the $\overline{BE}$ lines. The DMA interface of the MPC provides support by only incrementing internal pointers (or detecting completion) when the proper byte-enable signal is active. Table 5-4 shows which $\overline{BE}$ line the MPC recognizes for partial packets:

Table 5-4. Byte Enable Usage for DMA Control

DMA Width	Bytes Remaining	Byte Enable Recognized
32-bit	>3	$\overline{BE3}$
32-bit	3	$\overline{BE2}$
32-bit	2	$\overline{BE1}$
32-bit	1	$\overline{BE0}$
16-bit	>1	$\overline{BE1}$
16-bit	1	$\overline{BE0}$
8-bit	>0	$\overline{BE0}$

5.3.3 TRANSFER CONTROL SIGNALS

Transfer operation control to the MPC over the local bus is provided by two command signals and a wait signal. This handshake provides fully interlocked (two-sided handshake) operation.

$\overline{RD}$ (Read). This input signal starts a read operation. $\overline{RD}$ must transition cleanly, since it is used to qualify other signals in the read operation.

$\overline{WR}$ (Write). This input signal starts a write operation. $\overline{WR}$ must transition cleanly, since it is used to qualify other signals in the write operation.

$\overline{WAIT}$. $\overline{WAIT}$ is an MPC output signal used to extend a transfer operation. The signal will be used by the MPC for all accesses that require synchronization to another resource. It is activated when a command goes active and deactivated when the operation is completed.

5.3.4 INTERRUPT SIGNALS

Interrupt signals are used to inform the host CPU that the MPC requires service. The MPC generates two signals: one for message operations and one for reference errors.

MINT (Message Interrupt). The MINT output signal is used for all message-related signaling to the host CPU. This includes the arrival of an unsolicited message, the availability of the transmit FIFO, the completion of a solicited transfer, and an error-on message transfer.

EINT (Error Interrupt). The EINT output signal is used to signal all errors related to memory, I/O, or interconnect space operations. Internal registers in the MPC provide exact details of the error via interconnect space.

5.3.5 DMA CONTROL SIGNALS

The MPC provides four DMA control signals that connect with an external DMA controller.

ODREQ (Output Channel DMA Request). ODREQ is an output signal that enables DMA transfers to the MPC (i.e., output to the PSB). This signal behaves as a normal DMA request line during solicited message output operations. ODREQ is activated during the transfer phase of a solicited message operation when the solicited output FIFO is empty. The DMA controller responds to ODREQ by moving data from local memory to the FIFO for transfer to a receiving agent on the PSB.

IDREQ (Input Channel DMA Request). IDREQ is an output signal that enables DMA transfers from the MPC (i.e. input from the PSB). This signal behaves as a normal DMA request line during solicited message input operations. IDREQ is activated during the transfer phase of a solicited message operation when the solicited input FIFO is full. The DMA controller responds to ODREQ by moving data from the FIFO to local memory. When the FIFO is emptied, IDREQ is deactivated.

$\overline{ODACK}$ (Output Channel DMA Acknowledge). $\overline{ODACK}$ is generated by the DMA controller in response to an output channel DMA request. $\overline{ODACK}$ is qualified by $\overline{RD}$ or $\overline{WR}$ and therefore must be stable within the specified set-up and hold window.

NOTE:
$\overline{MEMSEL}$, $\overline{IOSEL}$, $\overline{REGSEL}$ $\overline{IDACK}$, and $\overline{ODACK}$ are mutually exclusive. In order to be valid, no more than on should be active during the same set-up and hold window.

$\overline{IDACK}$ (Input Channel DMA Acknowledge). $\overline{IDACK}$ is generated by the DMA controller in response to an input channel DMA request. $\overline{IDACK}$ is qualified by $\overline{RD}$ or $\overline{WR}$ and therefore must be stable within the specified set-up and hold window.

5.4 Interconnect Bus Signals

Brief descriptions of the interconnect bus signal pins are given here. For more information on using the interconnect microcontroller, see the *MPC User's Manual*, Chapter 5, "Interconnect Programming" (Order number 176526-002).

IAD<7-0> (Interconnect Address/Data). IAD<7-0> is an 8-bit, bidirectional, multiplexed address and data bus intended to interface directly to a microcontroller. In addition to the MPC, other interconnect accessible local resources can be connected to this bus.

$\overline{\text{IREQ}}$ (Interconnect Request). The MPC asserts this output signal when an interconnect operation has been requested from either the local bus or the PSB. The MPC asserts $\overline{\text{IREQ}}$ to the interconnect microcontroller at different times for read and write operations. For a read operation, $\overline{\text{IREQ}}$ is asserted immediately after detecting an address match between the requested address and an internal register. For a write operation, $\overline{\text{IREQ}}$ is delayed until valid data is available (i.e., $\overline{\text{BSC3}}$ is asserted). In either case, if the local bus interface has locked the local interconnect space, $\overline{\text{IREQ}}$ is inhibited.

IAST (Interconnect Address Strobe). IAST is a signal from the microcontroller that tells the MPC that a valid address is on the interconnect bus. IAST may be directly connected to the ALE (Address Latch Enable or equivalent) output of most microcontrollers. IAST must provide clean transitions.

$\overline{\text{IRD}}$ (Interconnect Bus Read). The microcontroller asserts $\overline{\text{IRD}}$ to perform a read operation to one of the MPC interconnect interface registers. $\overline{\text{IRD}}$ must provide clean transitions.

NOTE:
When $\overline{\text{IRD}}$ and $\overline{\text{IWR}}$ are activated at the same time, *all* MPC outputs are disabled. Use this feature to disable the MPC in board test applications.

$\overline{\text{IWR}}$ (Interconnect Write). The microcontroller asserts $\overline{\text{IWR}}$ to perform a write operation to one of the MPC interconnect interface registers. $\overline{\text{IWR}}$ must provide clean transitions.

6.2 Package Dimensions

The MPC 82389 is packaged in a 149-pin Ceramic Pin Grid Array (PGA). The pins are arranged 0.100 inch (2.54 mm) center-to-center, in a 15 x 15 matrix. Please refer to Figure 6-3 for case outlines.

A wide variety of sockets are available including the zero-insertion force socket for prototyping.

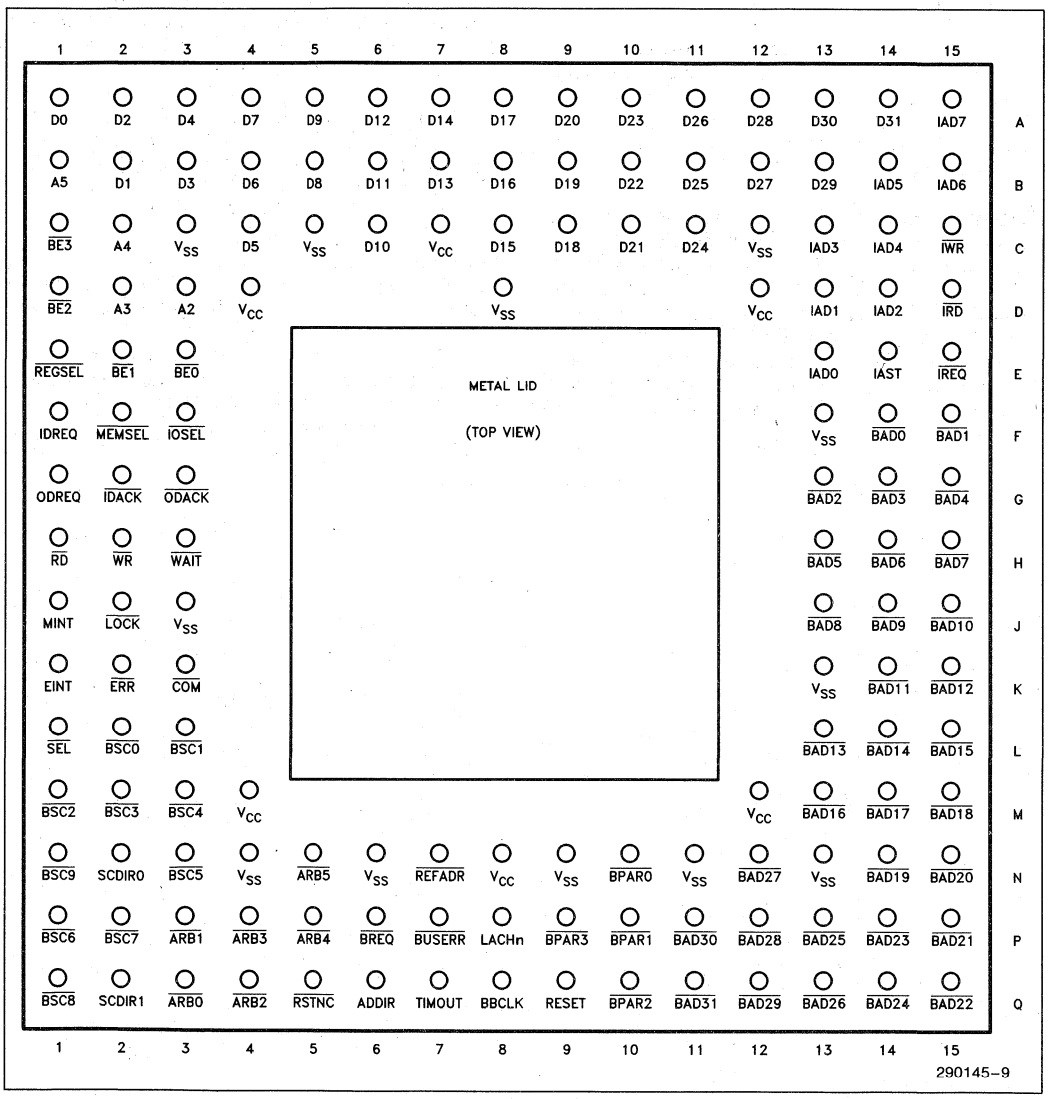


Figure 6-1. MPC 82389 Pinout—View from Top Side

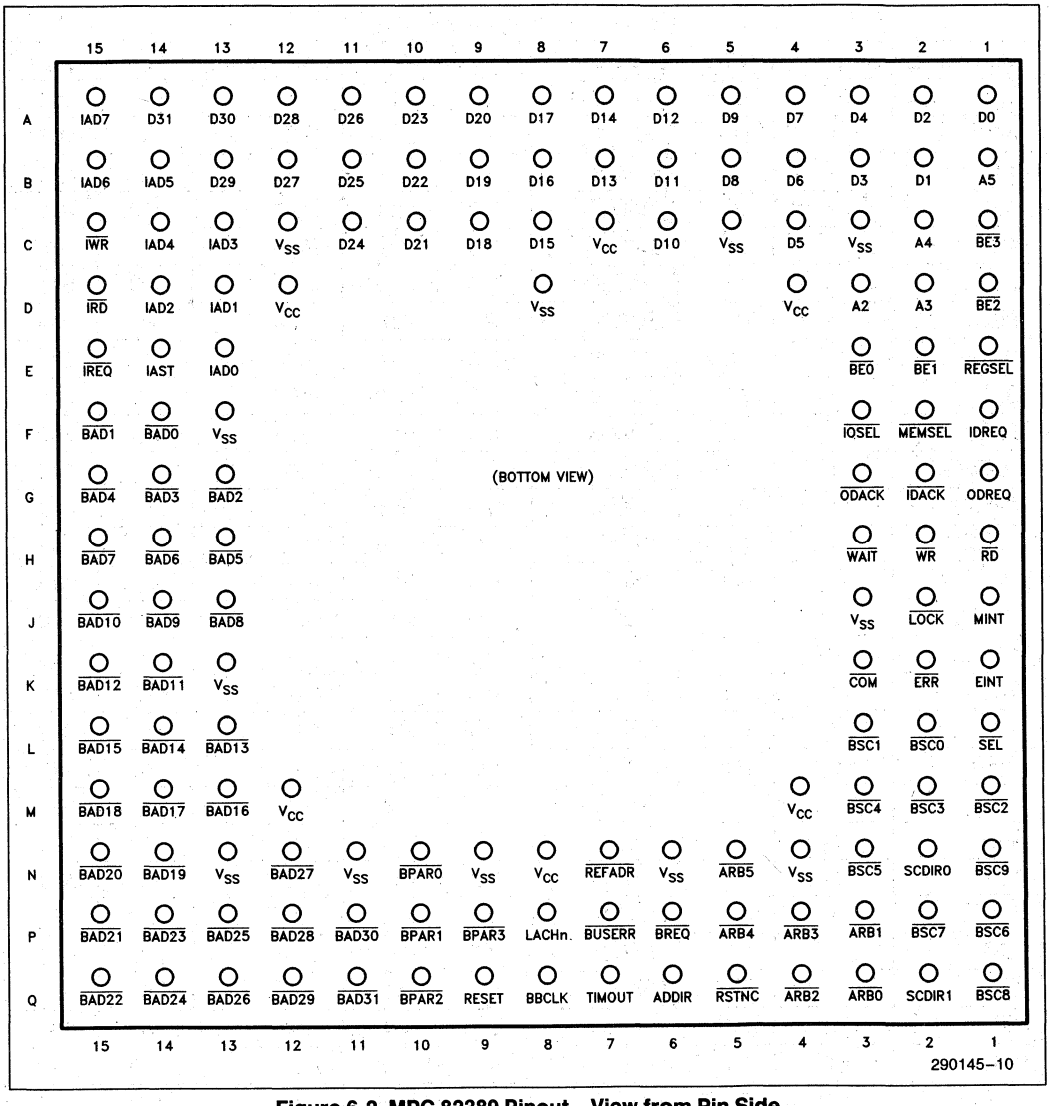


Figure 6-2. MPC 82389 Pinout—View from Pin Side

Table 6-1. MPC Signal Summary

Mnemonic	Type	Pin #	Mnemonic	Type	Pin #	Mnemonic	Type	Pin #
V_{CC}		D4	REFADR	O	N7	IAST	I	E14
A5	I	B1	ADDIR	O	Q6	$\overline{IRD}$	I	D15
A4	I	C2	$\overline{BPAR3}$	I/O	P9	$\overline{IWR}$	I	C15
A3	I	D2	$\overline{BAD31}$	I/O	Q11	IAD7	I/O	A15
A2	I	D3	$\overline{BAD30}$	I/O	P11	IAD6	I/O	B15
$\overline{BE3}$	I	C1	$\overline{BAD29}$	I/O	Q12	IAD5	I/O	B14
$\overline{BE2}$	I	D1	$\overline{BAD28}$	I/O	P12	IAD4	I/O	C14
$\overline{BE1}$	I	E2	$\overline{BAD27}$	I/O	N12	IAD3	I/O	C13
$\overline{BE0}$	I	E3	$\overline{BAD26}$	I/O	Q13	IAD2	I/O	D14
$\overline{IOSEL}$	I	F3	$\overline{BAD25}$	I/O	P13	IAD1	I/O	D13
$\overline{MEMSEL}$	I	F2	$\overline{BAD24}$	I/O	Q14	IAD0	I/O	E13
$\overline{REGSEL}$	I	E1	$\overline{BAD23}$	I/O	P14	V_{CC}		D12
$\overline{IDACK}$	I	G2	$\overline{BAD22}$	I/O	Q15	V_{SS}		C12
$\overline{ODACK}$	I	G3	$\overline{BAD21}$	I/O	P15	D31	I/O	A14
IDREQ	O	F1	$\overline{BAD20}$	I/O	N15	D30	I/O	A13
ODREQ	O	G1	$\overline{BAD19}$	I/O	N14	D29	I/O	B13
$\overline{WR}$	I	H2	$\overline{BAD18}$	I/O	M15	D28	I/O	A12
$\overline{RD}$	I	H1	$\overline{BAD17}$	I/O	M14	D27	I/O	B12
$\overline{WAIT}$	O	H3	$\overline{BAD16}$	I/O	M13	D26	I/O	A11
V_{SS}		J3	$\overline{BAD15}$	I/O	L15	D25	I/O	B11
MINT	O	J1	$\overline{BAD14}$	I/O	L14	D24	I/O	C11
EINT	O	K1	$\overline{BAD13}$	I/O	L13	D23	I/O	A10
$\overline{LOCK}$	I	J2	$\overline{BAD12}$	I/O	K15	D22	I/O	B10
$\overline{ERR}$	I	K2	$\overline{BAD11}$	I/O	K14	D21	I/O	C10
$\overline{SEL}$	O	L1	$\overline{BAD10}$	I/O	J15	D20	I/O	A9
$\overline{COM}$	I	K3	$\overline{BAD9}$	I/O	J14	D19	I/O	B9
$\overline{BSC9}$	I/O	N1	$\overline{BAD8}$	I/O	J13	D18	I/O	C9
$\overline{BSC8}$	I/O	Q1	$\overline{BAD7}$	I/O	H15	D17	I/O	A8
$\overline{BSC7}$	I/O	P2	$\overline{BAD6}$	I/O	H14	D16	I/O	B8
$\overline{BSC6}$	I/O	P1	$\overline{BAD5}$	I/O	H13	D15	I/O	C8
$\overline{BSC5}$	I/O	N3	$\overline{BAD4}$	I/O	G15	D14	I/O	A7
$\overline{BSC4}$	I/O	M3	$\overline{BAD3}$	I/O	G14	D13	I/O	B7
$\overline{ARB3}$	I/O, OC	P4	V_{SS}		N13	D2	I/O	A2

Table 6-1. MPC Signal Summary (Continued)

Mnemonic	Type	Pin #	Mnemonic	Type	Pin #	Mnemonic	Type	Pin #
$\overline{BSC3}$	I/O	M2	$\overline{BAD2}$	I/O	G13	D12	I/O	A6
$\overline{BSC2}$	I/O	M1	$\overline{BAD1}$	I/O	F15	D11	I/O	B6
$\overline{BSC1}$	I/O	L3	$\overline{BAD0}$	I/O	F14	D10	I/O	C6
$\overline{BSC0}$	I/O	L2	$\overline{BPAR2}$	I/O	Q10	D9	I/O	A5
SCDIR1	O	Q2	$\overline{BPAR1}$	I/O	P10	D8	I/O	B5
SCDIR0	O	N2	$\overline{BPAR0}$	I/O	N10	D7	I/O	A4
V_{CC}		M4	V_{CC}		N8	D6	I/O	B4
V_{SS}		N4	V_{SS}		N9	D5	I/O	C4
$\overline{ARB5}$	I/O, OC	N5	V_{SS}		N11	D4	I/O	A3
$\overline{ARB4}$	I/O, OC	P5	V_{CC}		M12	D3	I/O	B3
$\overline{ARB2}$	I/O, OC	Q4	V_{SS}		F13	D1	I/O	B2
$\overline{ARB1}$	I/O, OC	P3	V_{SS}		K13	D0	I/O	A1
$\overline{ARB0}$	I/O, OC	Q3	BBCLK	I	Q8	V_{CC}		C7
V_{SS}		N6	LACHn	I	P8	V_{SS}		D8
$\overline{BREQ}$	I/O, OC	P6	RESET	I	Q9	V_{SS}		C5
TIMOUT	I/O	Q7	$\overline{RSTNC}$	I/O, OC	Q5	$\overline{BUSERR}$	I/O, OC	P7
$\overline{IREQ}$	O	E15	V_{SS}		C3			

NOTES:
I = input
O = output
I/O = input/output
OC = open-collector
* = active-low

3

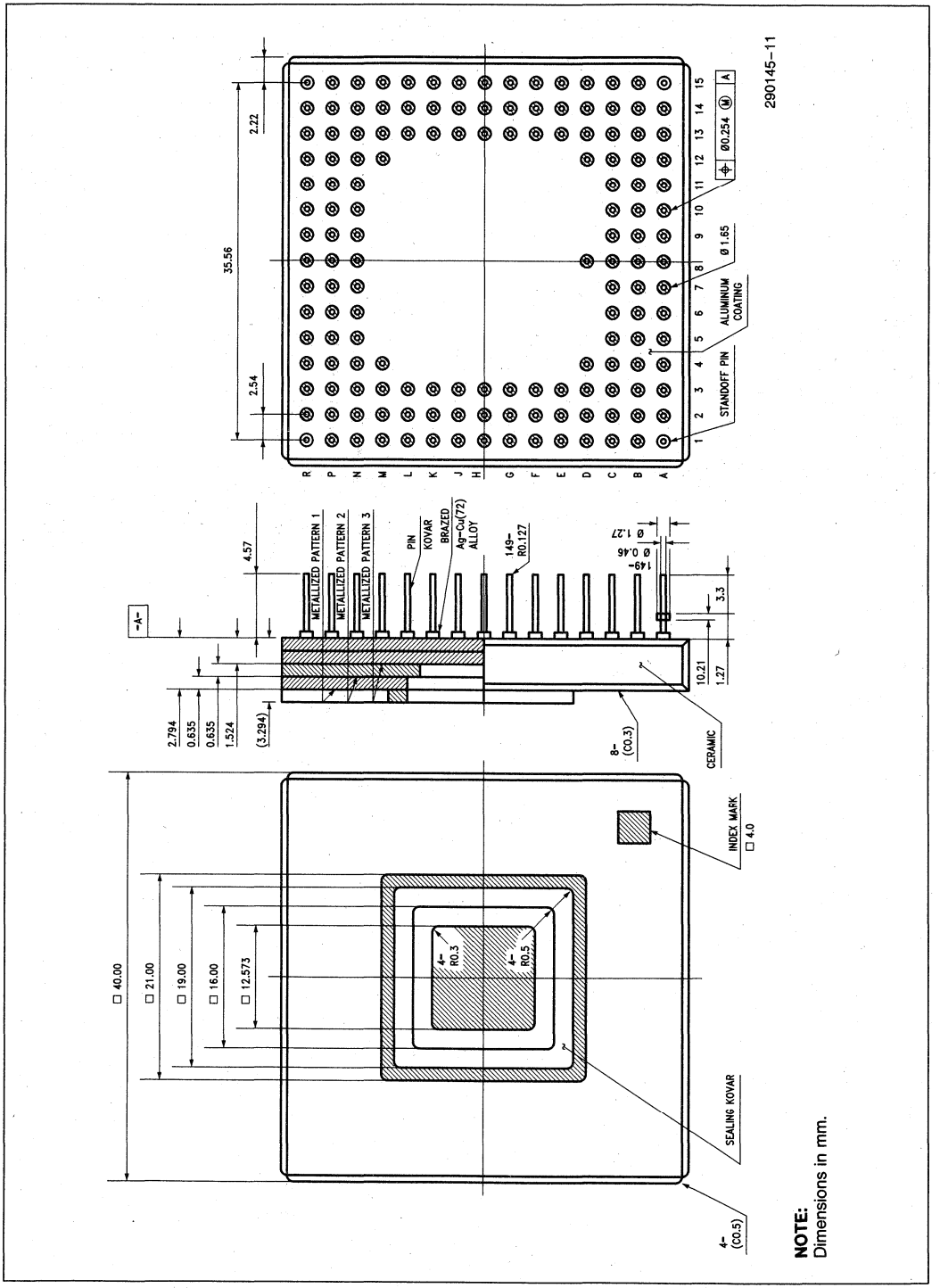


Figure 6-3. 149-Pin PGA Package Dimensions

7.0 MPC 82389 ELECTRICAL DATA

This section provides detailed A.C. and D.C. specifications for the MPC 82389.

7.1 Maximum Ratings

Operating Temperature (Under Bias) ..... −10°C to +85°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin ..... −0.5V to V_{CC} + 0.5V
Power Dissipation 2.5W

NOTE:
Stresses above those listed may cause permanent damage to the device. This is a stress rating only and functional operation at these or any other conditions above those listed in the operational sections of this specification is not implied.

Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Although the MPC 82389 contains protective circuitry to resist damage from static electrical discharges, always take precautions against high static voltages or electric fields.

7.2 D.C. Specifications V_{CC} = 5.0V ±5%, T_A = 0°C to +70°C

Table 7-1. D.C. Specifications

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{OL1}	Output Low Voltage		0.45	V	I_{OL} Max
V_{OL2}	Output Low Voltage Open Collector		0.55	V	I_{OL} Max
V_{OH}	Output High Voltage	2.4		V	I_{OH} Max
I_{CC}	Power Supply Current		400	mA	
I_L	Input Leakage Current		±10	µA	$0V \le V_{IN} \le V_{CC}$
I_{L1}	Open Collector		±100	µA	$0.4V \le V_{IN} \le 2.4V$
	Leakage Current		±400	µA	$0V \le V_{IN} \le V_{CC}$
I_{L2}	BBCLK Input Leakage Current		±100	µA	$0V \le V_{IN} \le V_{CC}$
I_{OL}	Output Low Current	4.0		mA	V_{OL} = 0.45V
I_{OL1}	Open Collector Output Low Current	60.0		mA	V_{OL} = 0.55V
I_{OL2}	ADDIR and $\overline{\text{REFADR}}$ Output Low Current	8.0		mA	V_{OL} = 0.45V
I_{OH}	Output High Current	−1.0		mA	V_{OH} = 2.4V
C_I	Input Capacitance		10	pF	f_C = 1 MHz, 25°C (Note 1)
C_{IO}	I/O Capacitance		20	pF	f_C = 1 MHz, 25°C (Note 1)
C_{CLK}	Clock Input Capacitance		15	pF	f_C = 1 MHz, 25°C (Note 1)
C_{OC}	Open Collector Capacitance		20	pF	f_C = 1 MHz, 25°C (Note 1)

NOTE:
1. Sampled only, not 100% tested.

7.3 A.C. Specifications

The A.C. specifications for the MPC 82389 are specified in Tables 7-2, 7-3 and 7-4 and Figures 7-2, 7-3, 7-4 and 7-5. Figure 7-1 specifies the test points for measuring the A.C. parameters. Table 7-2 and Figures 7-2 and 7-3 specify the A.C. parameters for the host CPU bus. Table 7-3 and Figure 7-4 specify the A.C. parameters for the interconnect bus. Table 7-4 and Figure 7-5 specify the A.C. parameters for the PSB. Figure 7-6 defines the test load for the A.C. specifications.

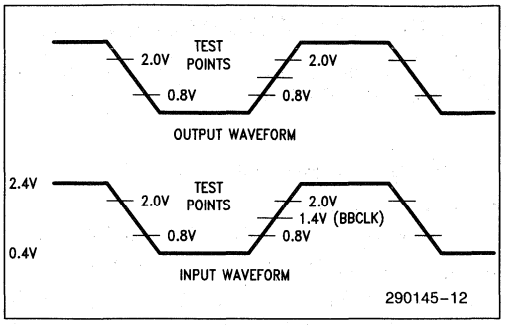


Figure 7-1. A.C. Test Waveforms

Table 7-2. Host CPU Bus A.C. Specifications (V_{CC} = 5V ±5%, T_A = 0°C to +70°C)

Symbol	Parameter	Min	Max	Units	Test Conditions
t_1	Address and $\overline{BE}$ Setup to Command Active	20		ns	
	Select and DACK Setup to Command Active	18		ns	
t_2	Address, $\overline{BE}$, Select and DACK Hold from Command Active	5		ns	
t_3	Time between Commands	24		ns	
t_4	Command Inactive to Read Data Disable (Note 5)		15	ns	
t_5	Read Data Hold from Command Inactive	3		ns	
t_6	Read Data Enable from Command Active	0		ns	
t_7	$\overline{WAIT}$ Active from Command Active		20	ns	C_L = 50 pF
t_8	Command Inactive from $\overline{WAIT}$ Inactive	0		ns	
t_9	$\overline{WAIT}$ Inactive to Read Data Valid		25	ns	C_L = 90 pF
t_{10}	Command Active to Write Data Valid		200	ns	
t_{11}	Write Data Hold from $\overline{WAIT}$ Inactive	0		ns	
t_{12}	Command Active to $\overline{LOCK}$ Active (Note 1)		100	ns	
t_{13}	$\overline{LOCK}$ Hold from $\overline{WAIT}$ Inactive (Note 2)	0		ns	
t_{14}	Command Active Time	42		ns	
t_{15}	Read Data Valid from Command Active		42	ns	C_L = 90 pF
t_{16}	Write Data Setup to Command Inactive —Registers —DMA	 20 20		 ns ns	
t_{17}	Write Data Hold from Command Inactive	3		ns	
t_{18}	Command Active to MINT or DREQ Inactive (Notes 3, 4)		42	ns	C_L = 50 pF
t_{19}	Command Active to DREQ Inactive (Note 4)		25	ns	C_L = 50 pF

NOTES:
1. Required to guarantee locking of resource.
2. Required to guarantee resource remains locked.
3. MINT deassertion only if no other sources are pending.
4. For DREQ inactive timing, t_{19} applies to a normal last transfer deassert condition and t_{18} to an error deassert condition.
5. Disable condition occurs when the output current becomes less than the input leakage specification.

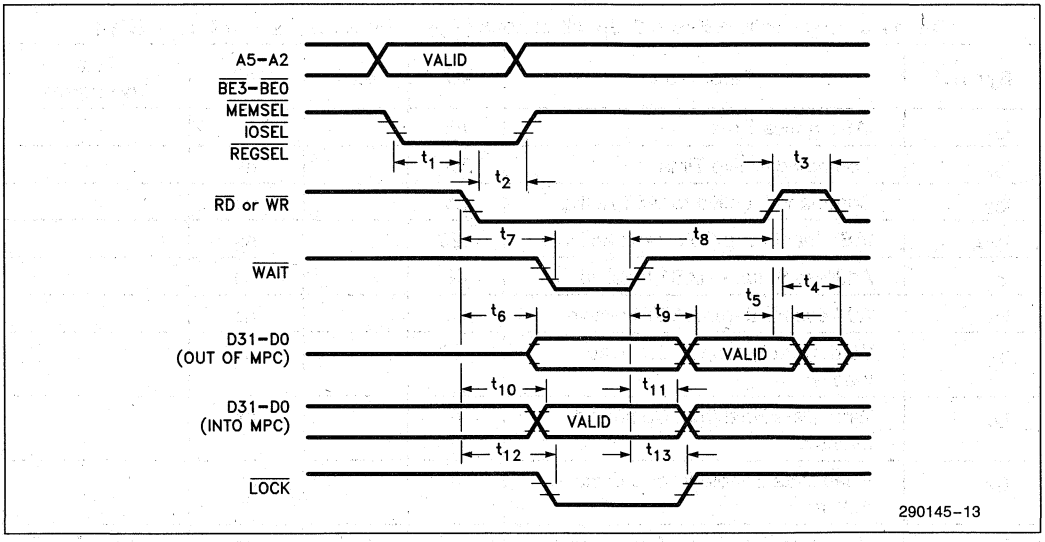


Figure 7-2. Host CPU Interface Reference Operation Timing

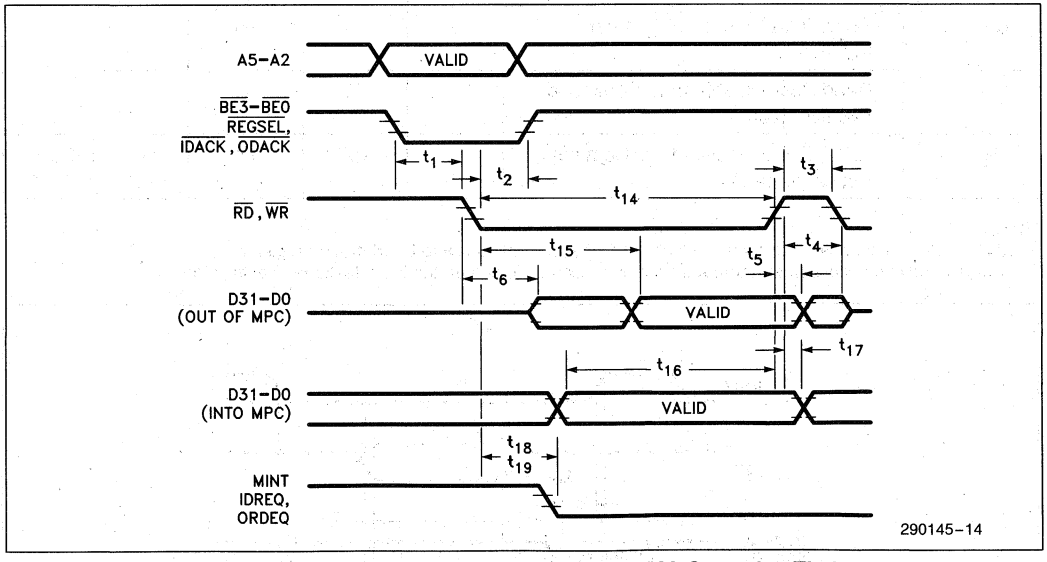
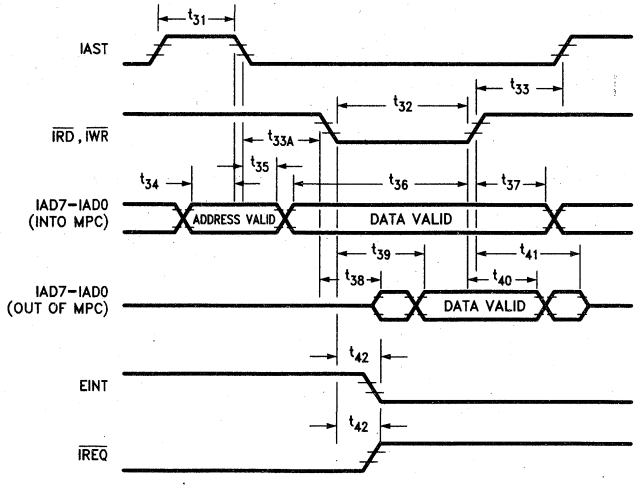


Figure 7-3. Host CPU Interface Register and DMA Operation Timing

Table 7-3. Interconnect Bus A.C. Specifications (V_{CC} = 5V ±5%, T_A = 0°C to +70°C)

Symbol	Parameter	Min	Max	Units	Test Conditions
t_{31}	IAST Active Time	85		ns	
t_{32}	Command Active Time	250		ns	
t_{33}	Command Inactive to IAST Active	25		ns	
t_{33A}	IAST Inactive to Command Active	120		ns	
t_{34}	Address Setup to IAST Inactive	40		ns	
t_{35}	Address Hold from IAST Inactive	20		ns	
t_{36}	Write Data Setup to Command Inactive	120		ns	
t_{37}	Write Data Hold from Command Inactive	5		ns	
t_{38}	Read Data Enable from Command Active	0		ns	
t_{39}	Read Data Valid from Command Active		120	ns	C_L = 150 pF
t_{40}	Read Data Hold from Command Inactive	0		ns	
t_{41}	Read Data Disable from Command Inactive (Note 2)		30	ns	
t_{42}	EINT, $\overline{IREQ}$ Inactive from Command Active (Note 1)		100	ns	C_L = 150 pF

NOTES:
1. EINT inactive only on write to error register. $\overline{IREQ}$ inactive only on write to arbitration register.
2. Disable condition occurs when the output current becomes less than the input leakage specification.



290145-15

Figure 7-4. Interconnect Bus Timing

Table 7-4. PSB Interface A.C. Specifications (V_{CC} = 5V ±5%, T_A = 0°C to +70°C)

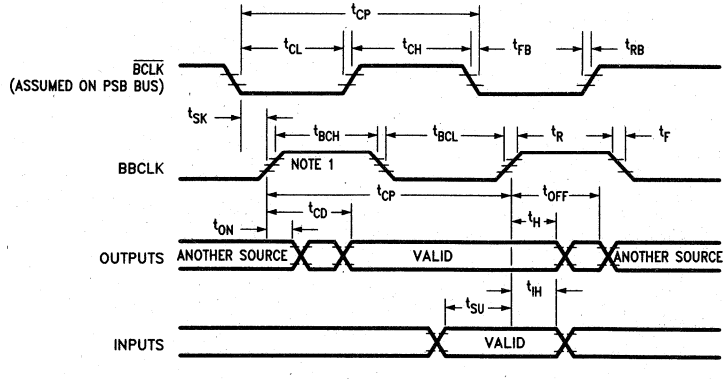
Symbol	Parameter	Min	Max	Units	Test Conditions
t_{CP}	Clock Period	99.9		ns	
*t_{CL}	$\overline{BCLK}$ Low Time	40		ns	
*t_{CH}	$\overline{BCLK}$ High Time	40		ns	
t_{BCL}	BBCLK Low Time	38		ns	
t_{BCH}	BBCLK High Time	38		ns	
t_{RB}	$\overline{BCLK}$ Rise Time	1	5	ns	
t_{FB}	$\overline{BCLK}$ Fall Time	1	2	ns	
t_R	BBCLK Rise Time	0.5	1	ns	
t_F	BBCLK Fall Time	0.5	1	ns	
t_{SK}	$\overline{BCLK}$ to BBCLK Skew (Note 1)	−0.5	4.0	ns	
t_{CD}	Clock to Output Delay				
	$\overline{BREQ}$, $\overline{BUSERR}$, $\overline{RSTNC}$ (Note 2)		36	ns	C_L = 500 pF
	$\overline{ARB5}$–$\overline{ARB0}$ (Notes 2, 3)		36	ns	C_L = 500 pF
	$\overline{BAD31}$–$\overline{BAD0}$, $\overline{BSC7}$–$\overline{BSC0}$		29	ns	C_L = 75 pF
	$\overline{BPAR3}$–$\overline{BPAR0}$, $\overline{BSC9}$, $\overline{BSC8}$		29	ns	C_L = 50 pF
	SCDIR0, SCDIR1 (H to L)		19	ns	C_L = 25 pF
	(L to H)		21	ns	C_L = 25 pF
	ADDIR (L to H)		21	ns	C_L = 50 pF
	(H to L)		27	ns	C_L = 50 pF
	$\overline{REFADR}$		29	ns	C_L = 75 pF
	$\overline{SEL}$		29	ns	C_L = 50 pF
t_H	Hold Time from Clock				
	$\overline{BREQ}$, $\overline{BUSERR}$, $\overline{RSTNC}$	6.5		ns	C_L = 25 pF
	$\overline{ARB5}$–$\overline{ARB0}$ (Note 3)	6.5		ns	C_L = 25 pF
	$\overline{BAD31}$–$\overline{BAD0}$, $\overline{BPAR3}$–$\overline{BPAR0}$	5.0		ns	C_L = 15 pF
	$\overline{BSC9}$–$\overline{BSC0}$	4.0		ns	C_L = 15 pF
	SCDIR0, SCDIR1	4.0		ns	C_L = 15 pF
	ADDIR	5.0		ns	C_L = 25 pF
	$\overline{REFADR}$	4.0		ns	C_L = 25 pF
	$\overline{SEL}$	4.0		ns	C_L = 15 pF
t_{ON}	Turn On Delay from Clock (Note 4)				
	$\overline{BREQ}$, $\overline{BUSERR}$, $\overline{RSTNC}$	6.5		ns	
	$\overline{ARB5}$–$\overline{ARB0}$ (Note 1)	6.5		ns	
	$\overline{BAD31}$–$\overline{BAD0}$, $\overline{BPAR3}$–$\overline{BPAR0}$	5.0		ns	
	$\overline{BSC9}$–$\overline{BSC0}$	4.0		ns	
t_{OFF}	Turn Off Delay from Clock (Note 5)				
	$\overline{BREQ}$, $\overline{BUSERR}$, $\overline{RSTNC}$		36	ns	
	$\overline{ARB5}$–$\overline{ARB0}$ (Note 3)		36	ns	
	$\overline{BAD31}$–$\overline{BAD0}$, $\overline{BPAR3}$–$\overline{BPAR0}$		29	ns	
	$\overline{BSC9}$–$\overline{BSC0}$		29	ns	

*t_{CL} and t_{CH} are MULTIBUS II specifications.

Table 7-4. PSB Interface A.C. Specifications (V_{CC} = 5V ±5%, T_A = 0°C to +70°C) (Continued)

Symbol	Parameter	Min	Max	Units	Test Conditions
t_{SU}	Input Setup Time to Clock				
	$\overline{BREQ}$, $\overline{BUSERR}$, $\overline{RSTNC}$	22		ns	
	$\overline{ARB5}$–$\overline{ARB0}$ (Note 3)	40		ns	
	$\overline{BAD31}$–$\overline{BAD0}$, $\overline{BPAR3}$–$\overline{BPAR0}$	24		ns	
	$\overline{BSC9}$–$\overline{BSC0}$	24		ns	
	TIMEOUT, LACHn, RESET	24		ns	
	$\overline{COM}$, $\overline{ERR}$	40		ns	
t_{IH}	Input Hold Time from Clock				
	$\overline{BREQ}$, $\overline{BUSERR}$, $\overline{RSTNC}$	0		ns	
	$\overline{ARB5}$–$\overline{ARB0}$ (Note 3)	0		ns	
	$\overline{BAD31}$–$\overline{BAD0}$, $\overline{BPAR3}$–$\overline{BPAR0}$	3		ns	
	$\overline{BSC9}$–$\overline{BSC0}$	2		ns	
	TIMEOUT, LACHn, RESET	2		ns	
	$\overline{COM}$, $\overline{ERR}$	3		ns	

NOTES:
1. The clock timings are provided to reference the MPC specification to the PSB specifications. These specifications assume a 74AS1804 or equivalent buffer.
2. The 500 pF load is a distributed load as defined in the PSB specification. The open drain signals are designed such that the output delay and bus loss meets the PSB specification requirement.
3. The $\overline{ARB5}$–$\overline{ARB0}$ signal timings are with respect to the first and last clock of the arbitration period. Details can be found in the PSB specification. Also, the arbitration logic has been designed to meet the loop delay specification accounting for the full path of input to output plus bus loss.
4. Minimum turn on times are measured the same way as hold times. Specifically, the logic level driven by another device on the previous clock cycle must not be disturbed.
5. Maximum turn off times are measured to the condition where the output leakage current becomes less than the input leakage specification.
6. All stated capacitances are based on design requirements. Production test limitations may require some parameters to be tested under a different condition.



NOTE:
1. SAMPOINT point for BBCLK is 1.4V.

Figure 7-5. PSB Interface Timing

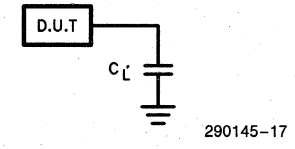


Figure 7-6. A.C. Test Load

8.0 REFERENCE DOCUMENTS

Part Number	Title Description
176526-002	MPC User's Manual
146077	MULTIBUS® II Architecture Specifications
149299	Interconnect Interface Specifications
149300	MULTIBUS® II MPC External Product Specifications
149247	MULTIBUS® II Transport Protocol Specifications
459706-001	CSM/002 Hardware Reference Manual

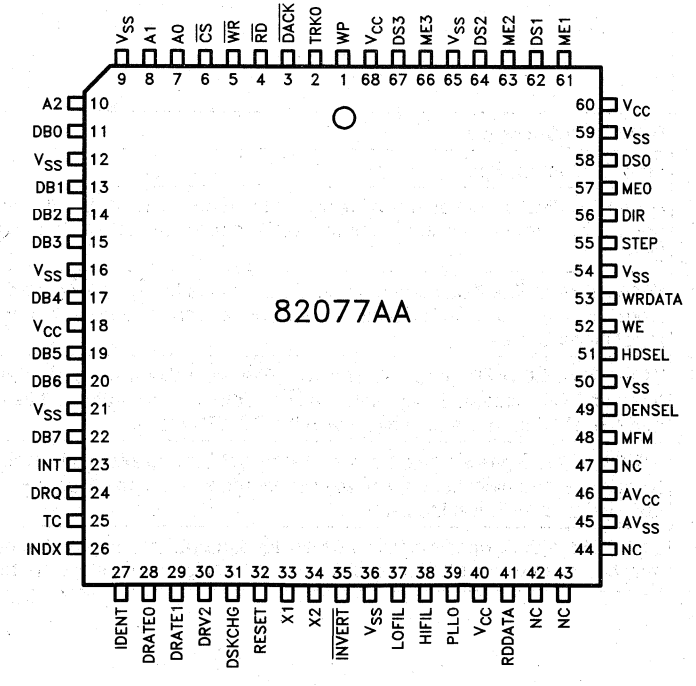
Floppy Disk Controller 4

82077AA
CHMOS SINGLE-CHIP FLOPPY DISK CONTROLLER

- **Single-Chip Floppy Disk Solution**
  - **— 100% PC AT* Compatible**
  - **— 100% PS/2* Compatible**
  - **— 100% PS/2 Model 30 Compatible**
  - **— Integrated Drive and Data Bus Buffers**
- **Integrated Analog Data Separator**
  - **— 250 Kbits/sec**
  - **— 300 Kbits/sec**
  - **— 500 Kbits/sec**
  - **— 1 Mbits/sec**
- **High Speed Processor Interface**
- **Perpendicular Recording Support**
- **Integrated Tape Drive Support**
- **12 mA Host Interface Drivers, 40 mA Disk Drivers**
- **Four Fully Decoded Drive Select and Motor Signals**
- **Programmable Write Precompensation Delays**
- **Addresses 256 Tracks Directly, Supports Unlimited Tracks**
- **16 Byte FIFO**
- **68-Pin PLCC**

The 82077AA floppy disk controller has completely integrated all of the logic required for floppy disk control. The 82077AA, a 24 MHz crystal, a resistor package and a device chip select implements a PC AT or PS/2 solution. All programmable options default to compatible values. The dual PLL data separator has better performance than most board level/discrete PLL implementations. The FIFO allows better system performance in multi-master systems (e.g. PS/2, EISA).

The 82077AA is fabricated with Intel's CHMOS III technology and is available in a 68-lead PLCC (plastic) package.



290166-1

Figure 1. 82077AA Pinout

*PS/2 and PC AT are trademarks of IBM.

October 1990
Order Number: 290166-003

Table 1. 82077AA Pin Description

Symbol	Pin#	I/O	Description
HOST INTERFACE			
RESET	32	I	**RESET:** A high level places the 82077AA in a known idle state. All registers are cleared except those set by the Specify command.
$\overline{CS}$	6	I	**CHIP SELECT:** Decodes base address range and qualifies $\overline{RD}$ and $\overline{WR}$ inputs.
A0 A1 A2	7 8 10	I	**ADDRESS:** Selects one of the host interface registers:

A2	A1	A0		Register
0	0	0	R	Status Register A
0	0	1	R	Status Register B
0	1	0	R/W	Digital Output Register
0	1	1	R/W	Tape Drive Register
1	0	0	R	Main Status Register
1	0	0	W	Data Rate Select Register
1	0	1	R/W	Data (FIFO)
1	1	0		Reserved
1	1	1	R	Digital Input Register
1	1	1	W	Configuration Control Register

DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7	11 13 14 15 17 19 20 22	I/O	**DATA BUS:** Data bus with 12 mA drive
$\overline{RD}$	4	I	**READ:** Control signal
$\overline{WR}$	5	I	**WRITE:** Control signal
DRQ	24	O	**DMA REQUEST:** Requests service from a DMA controller. Normally active high, but goes to high impedance in AT and Model 30 modes when the appropriate bit is set in the DOR.
$\overline{DACK}$	3	I	**DMA ACKNOWLEDGE:** Control input that qualifies the $\overline{RD}$, $\overline{WR}$ inputs in DMA cycles. Normally active low, but is disabled in AT and Model 30 modes when the appropriate bit is set in the DOR.
TC	25	I	**TERMINAL COUNT:** Control line from a DMA controller that terminates the current disk transfer. TC is accepted only while $\overline{DACK}$ is active. This input is active high in the AT, and Model 30 modes and active low in the PS/2™ mode.
INT	23	O	**INTERRUPT:** Signals a data transfer in non-DMA mode and when status is valid. Normally active high, but goes to high impedance in AT, and Model 30 modes when the appropriate bit is set in the DOR.
X1 X2	33 34		**CRYSTAL 1,2:** Connection for a 24 MHz fundamental mode parallel resonant crystal. X1 may be driven with a MOS level clock and X2 would be left unconnected.

Table 1. 82077AA Pin Description (Continued)

Symbol	Pin #	I/O	Description
HOST INTERFACE (Continued)			
IDENT	27	I	**IDENTITY:** Upon Hardware RESET, this input (along with MFM pin) selects between the three interface modes. After RESET, this input selects the type of drive being accessed and alters the level on DENSEL. The MFM pin is also sampled at Hardware RESET, and then becomes an output again. Internal pull-ups on MFM permit a no connect.

IDENT	MFM	INTERFACE
1	1 or NC	AT Mode
1	0	ILLEGAL
0	1 or NC	PS/2 Mode
0	0	Model 30 Mode

AT MODE: Major options are: enables DMA Gate logic, TC is active high, Status Registers A & B not available.
PS/2 MODE: Major options are: No DMA Gate logic, TC is active low, Status Registers A & B are available.
MODEL 30 MODE: Major options are: enable DMA Gate logic, TC is active high, Status Registers A & B available.
After Hardware reset this pin determines the polarity of the DENSEL pin. IDENT at a logic level of "1", DENSEL will be active high for high (500 Kbps/1 Mbps) data rates (typically used for 5.25" drives). IDENT at a logic level of "0", DENSEL will be active low for high data rates (typically used for 3.5" drives).

DISK CONTROL (All outputs have 40 mA drive capability)			
$\overline{\text{INVERT}}$	35	I	**INVERT:** Strapping option. Determines the polartity of **all** signals in this section. Should be strapped to ground when using the internal buffers and these signals become active LOW. When strapped to VCC, these signals become active high and external inverting drivers and receivers are required.
ME0 ME1 ME2 ME3	57 61 63 66	O	**ME0–3:** Decoded Motor enables for drives 0–3. The motor enable pins are directly controlled via the Digital Output Register.
DS0 DS1 DS2 DS3	58 62 64 67	O	**DRIVE SELECT 0–3:** Decoded drive selects for drives 0–3. These outputs are decoded from the select bits in the Digital Output Register and gated by ME0–3.
HDSEL	51	O	**HEAD SELECT:** Selects which side of a disk is to be used. An active level selects side 1.
STEP	55	O	**STEP:** Supplies step pulses to the drive.
DIR	56	O	**DIRECTION:** Controls the direction the head moves when a step signal is present. The head moves toward the center if active.
WRDATA	53	O	**WRITE DATA:** FM or MFM serial data to the drive. Precompensation value is selectable through software.
WE	52	O	**WRITE ENABLE:** Drive control signal that enables the head to write onto the disk.

Table 1. 82077AA Pin Description (Continued)

Symbol	Pin #	I/O	Description
DISK CONTROL (All outputs have 40 mA drive capability) (Continued)			
DENSEL	49	O	**DENSITY SELECT:** Indicates whether a low (250/300 Kbps) or high (500 Kbps/1 Mbps) data rate has been selected.
DSKCHG	31	I	**DISK CHANGE:** This input is reflected in the Digital Input Register.
DRV2	30	I	**DRIVE2:** This indicates whether a second drive is installed and is reflected in Status Register A.
TRK0	2	I	**TRACK0:** Control line that indicates that the head is on track 0.
WP	1	I	**WRITE PROTECT:** Indicates whether the disk drive is write protected.
INDX	26	I	**INDEX:** Indicates the beginning of the track.
PLL SECTION			
RDDATA	41	I	**READ DATA:** Serial data from the disk. INVERT also affects the polarity of this signal.
HIFIL	38	I/O	**HIGH FILTER:** Analog reference signal for internal data separator compensation. This should be filtered by an external capacitor to LOFIL.
LOFIL	37	I/O	**LOW FILTER:** Low noise ground return for the reference filter capacitor.
MFM	48	I/O	**MFM:** At Hardware RESET, aids in configuring the 82077AA. Internal pull-up allows a no connect if a "1" is required. After reset this pin becomes an output and indicates the current data encoding/decoding mode (Note: If the pin is held at logic level "0" during hardware RESET it must be pulled to "1" after reset to enable the output. The pin can be released on the falling edge of hardware RESET to enable the output). MFM is active high (MFM). MFM may be left tied low after hardware reset, in this case the MFM function will be disabled.
DRATE0 DRATE1	28 29	O	**DATARATE0–1:** Reflects the contents of bits 0,1 of the Data Rate Register. (Drive capability of +6.0 mA @ 0.4V and −4.0 mA @ 2.4V)
PLL0	39	I	**PLL0:** This input optimizes the data separator, for either floppy disks or tape drives. A "1" (or V_{CC}) selects the floppy mode, a "0" (or GND) selects tape mode.
MISCELLANEOUS			
VCC	18 40 60 68		**Voltage:** +5V
GND	9 12 16 21 36 50 54 59 65		**Ground**
AVCC	46		**Analog Supply**
AVSS	45		**Analog Ground**
NC	42 43 44 47		**No Connection:** These pins **MUST** be left unconnected.

1.0 INTRODUCTION

The 82077AA is a true single-chip floppy disk, and tape drive controller for the PC-AT and PS/2. The 82077AA, a 24 MHz crystal, a resistor package and a chip select implement a complete design. All drive control signals are fully decoded and have 40 mA drive buffers with selectable polarity. Signals returned from the drive are sent through on-chip input buffers with hysteresis for noise immunity. The integrated analog data separator needs no external compensation yet allows for a wide motor speed variation with exceptionally low soft error rates. The microprocessor interface has a 12 mA drive buffer on the data bus plus 100% hardware register compatibility for PC-AT's and PS/2's. The 16-byte FIFO with programmable thresholds is extremely useful in multi-master systems (PS/2, EISA) or systems with a large amount of bus latency.

Upon reset, (Pin 32) the 82077AA defaults to 8272A functionality. New features are either selected via hardware straps or new commands. Figure 1-1 is a block diagram of the 82077AA.

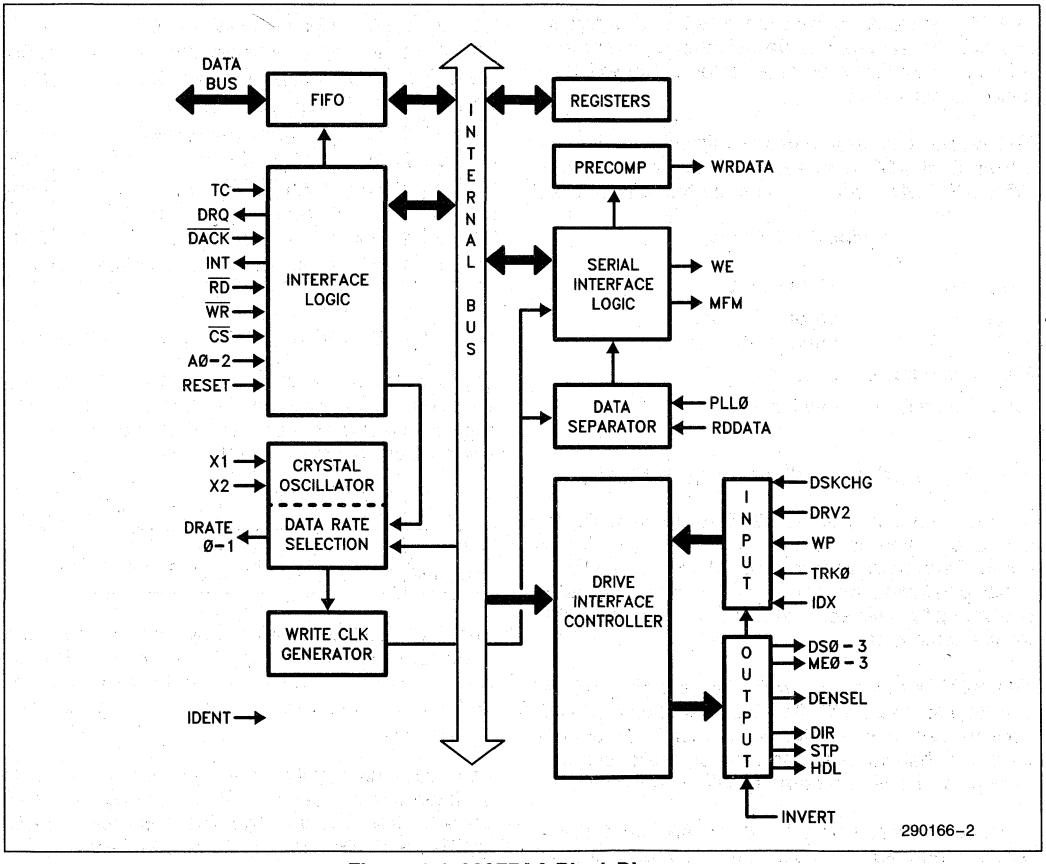


Figure 1-1. 82077AA Block Diagram

1.1 Oscillator

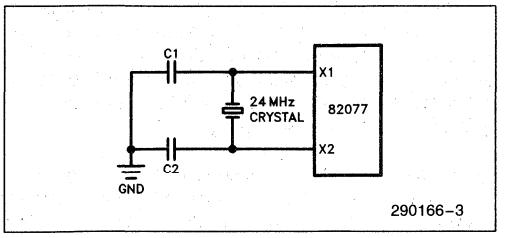


Figure 1-2. Crystal Oscillator Circuit

The 24 MHz clock can be supplied either by a crystal or a MOS level square wave. All internal timings are referenced to this clock or a scaled count which is data rate dependent.

The crystal oscillator must be allowed to run for 10 ms after VCC has reached 4.5V or exiting the POWERDOWN mode to guarantee that it is stable.

Crystal Specifications

Frequency:	24 MHz ±0.1%
Mode:	Parallel Resonant Fundamental Mode
Series Resistance:	Less than 40Ω
Shunt Capacitance:	Less than 5 pF

1.2 Perpendicular Recording Mode

An added capability of the 82077AA is the ability to interface directly to perpendicular recording floppy drives. Perpendicular recording differs from the traditional longitudinal method by orienting the magnetic bits vertically. This scheme packs in more data bits for the same area.

The 82077AA with perpendicular recording drives can read standard 3.5" floppies as well as read and write perpendicular media. Some manufacturers offer drives that can read and write standard and perpendicular media in a perpendicular media drive.

A single command puts the 82077AA into perpendicular mode. All other commands operate as they normally do. The perpendicular mode requires the 1 Mbps data rate of the 82077AA. At this data rate, the FIFO eases the host interface bottleneck due to the speed of data transfer to or from the disk.

2.0 MICROPROCESSOR INTERFACE

The interface consists of the standard asynchronous signals: $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, A0–A3, INT, DMA control and a data bus. The address lines select between configuration registers, the FIFO and control/status registers. This interface can be switched between PC AT, Model 30, or PS/2™ normal modes. The PS/2™ register sets are a superset of the registers found in a PC-AT.

2.1 Status, Data and Control Registers

The base address range is supplied via the $\overline{CS}$ pin. For PC-AT or PS/2 designs this would be 3F0 Hex to 3F7 Hex.

A2	A1	A0		Register	
0	0	0	R	Status Register A	SRA
0	0	1	R	Status Register B	SRB
0	1	0	R/W	Digital Output Register	DOR
0	1	1	R/W	Tape Drive Register	TSR
1	0	0	R	Main Status Register	MSR
1	0	0	W	Data Rate Select Register	DSR
1	0	1	R/W	Data (FIFO)	FIFO
1	1	0		Reserved	
1	1	1	R	Digital Input Register	DIR
1	1	1	W	Configuration Control Register	CCR

2.1.1a STATUS REGISTER A (SRA, PS/2 MODE)

This register is read-only and monitors the state of the interrupt pin and several disk interface pins. This register is part of the register set, and is not accessible in PC-AT mode.

7	6*	5	4*	3	2*	1*	0
INT PENDING	$\overline{DRV2}$	STEP	$\overline{TRK0}$	HDSEL	$\overline{INDX}$	$\overline{WP}$	DIR

The INT PENDING bit is used by software to monitor the state of the 82077AA INTERRUPT pin. The bits marked with a "*" reflect the state of drive signals on the cable and are independent of the state of the $\overline{INVERT}$ pin.

As a read-only register, there is no default value associated with a reset other than some drive bits will change with a reset. The INT PENDING, STEP, HDSEL, and DIR bits will be low after reset.

2.1.1b STATUS REGISTER A (SRA, MODEL 30 MODE)

7	6	5	4	3	2	1	0
INT PENDING	DRQ	STEP F/F	TRKO	$\overline{HDSEL}$	INDEX	WP	$\overline{DIR}$

This register has the following changes in PS/2 Model 30 Mode. Disk interface pins (Bits 0, 1, 2, 3, & 4) are inverted from PS/2 Mode. The DRQ bit

monitors the status of the DMA Request pin. The STEP bit is latched with the Step output going active and is cleared with a read to the DIR register, Hardware or Software RESET.

2.1.2a STATUS REGISTER B (SRB, PS/2 MODE)

This register is read-only and monitors the state of several disk interface pins. This register is part of the PS/2 register set, and is not accessible in PC-AT mode.

7	6	5	4	3*	2	1	0
1	1	DRIVE SEL 0	WRDATA TOGGLE	RDDATA TOGGLE	WE	MOT EN1	MOT EN0

As the only drive input, RDDATA TOGGLE's activity is independent of the INVERT pin level and reflects the level as seen on the cable.

The two TOGGLE bits do not read back the state of their respective pins directly. Instead, the pins drive a Flip/Flop which produces a wider and more reliably read pulse. Bits 6 and 7 are undefined and always return a 1.

After any reset, the activity on the TOGGLE pins are cleared. Drive select and Motor bits cleared by the RESET pin and not software resets.

2.1.2b STATUS REGISTER B (SRB, MODEL 30 MODE)

7	6	5	4	3	2	1	0
$\overline{DRV2}$	$\overline{DS1}$	$\overline{DS0}$	WRDATA F/F	RDDATA F/F	WE F/F	$\overline{DS3}$	$\overline{DS2}$

This register has the following changes in Model 30 Mode. Bits 0, 1, 5, and 6 return the decoded value of the Drive Select bits in the DOR register. Bits 2, 3, and 4 are set by their respective active going edges and are cleared by reading the DIR register. The WRDATA bit is triggered by raw WRDATA signals and is not gated by WE. Bits 2, 3, and 4 are cleared to a low level by either Hardware or Software RESET.

2.1.3 DIGITAL OUTPUT REGISTER (DOR)

The Digital Output Register contains the drive select and motor enable bits, a reset bit and a $\overline{DMA\ GATE}$ bit.

7	6	5	4	3	2	1	0
MOT EN3	MOT EN2	MOT EN1	MOT EN0	$\overline{DMA\ GATE}$	$\overline{RESET}$	DRIVE SEL 1	DRIVE SEL 0

The MOT ENx bits directly control their respective motor enable pins (ME0–3). A one means the pin is active, the $\overline{INVERT}$ pin determines the active level. The DRIVE SELx bits are decoded to provide four drive select lines and only one may be active at a time. A one is active and the $\overline{INVERT}$ pin determines the level on the cable. Standard programming practice is to set both MOT ENx and DRIVE SELx bits at the same time.

Table 2-1 lists a set of DOR values to activate the drive select and motor enable for each drive.

Table 2-1. Drive Activation Values

Drive	DOR Value
0	1CH
1	2DH
2	4EH
3	8FH

The $\overline{DMAGATE}$ bit is enabled only in PC-AT and Model 30 Modes. If $\overline{DMAGATE}$ is set low, the INT and DRQ outputs are tristated and the $\overline{DACK}$ and TC inputs are disabled. $\overline{DMAGATE}$ set high will enable INT, DRQ, TC, and $\overline{DACK}$ to the system. In PS/2 Mode $\overline{DMAGATE}$ has no effect upon INT, DRQ, TC or $\overline{DACK}$ pins and they are always active.

This $\overline{RESET}$ bit clears the basic core of the 82077AA and the FIFO circuits when the LOCK bit is set to "0" (see section 5.3.2 for LOCK bit definition). Once set, it remains set until the user clears this bit. This bit is set by a chip reset and the 82077AA is held in a reset state until the user clears this bit. The $\overline{RESET}$ bit has no effect upon this register.

2.1.4 TAPE DRIVE REGISTER (TDR)

This register allows the user to assign tape support to a particular drive during initialization. Any future references to that drive number automatically invokes tape support. This register is cleared by Hardware reset, Software resets have no effect.

7	6	5	4	3	2	1	0
—	—	—	—	—	—	Tape SEL1	Tape SEL0

Bits 2 thru 7 are not writable and remain tristated if read. The Tape Select bits are Hardware RESET to 0's, making Drive 0 not available for tape support. Drive 0 is "reserved" for the floppy boot drive.

Tape SEL1	Tape SEL0	DRIVE SELECTED
0	0	None
0	1	1
1	0	2
1	1	3

The tuning of the PLL for tape characteristics can also be done in hardware. If a 0 (GND) is applied to pin 39 (PLL0) the PLL is optimized for tape drives, a 1 (V_{CC}) optimizes the PLL for floppies. This hardware selection mechanism overrides the software selection scheme. A typical hardware application would route the Drive Select pin used for tape drive support to pin 39 (PLL0).

2.1.5 DATARATE SELECT REGISTER (DSR)

This register is included for compatibility with the 82072 floppy controller and is write-only. Changing the data rate changes the timings of the drive control signals. To ensure that drive timings are not violated when changing data rates, choose a drive timing such that the fastest data rate will not violate the timing.

7	6	5	4	3	2	1	0
S/W RESET	POWER DOWN	0	PRE-COMP 2	PRE-COMP 1	PRE-COMP 0	DRATE SEL 1	DRATE SEL 0

This register is the same as used in the 82072 except that the internal/external PLL select bit is removed. It is recommended that bit 5 be written with a 0 for compatibility.

S/W RESET behaves the same as DOR RESET except that this reset is self clearing.

POWER DOWN deactivates the internal clocks and shuts off the oscillator. Disk control pins are put in an inactive state. All input signals must be held in a valid state (D.C. level 1 or 0). POWER DOWN is exited by activating one of the reset functions.

PRECOMP 0–2 adjusts the WRDATA output to the disk to compensate for magnetic media phenomena known as bit shifting. The data patterns that are susceptible to bit shifting are well understood and the 82077AA compensates the data pattern as it is written to the disk. The amount of precompensation is dependent upon the drive and media but in most cases the default value is acceptable.

The 82077AA starts precompensating the data pattern starting on Track 0. The CONFIGURE command can change the track that precompensating starts on. Table 2-2 lists the precompensation values that can be selected and Table 2-3 lists the default precompensation values. The default value is selected if the three bits are zeros.

DRATE 0–1 select one of the four data rates as listed in Table 2-4. The default value is 250 Kbps upon a chip ("Hardware") reset. Other ("Software") Resets do not affect the DRATE or PRECOMP bits.

Table 2-2. Precompensation Delays

PRECOMP 432	Precompensation Delay
111	0.00 ns—DISABLED
001	41.67 ns
010	83.34 ns
011	125.00 ns
100	166.67 ns
101	208.33 ns
110	250.00 ns
000	DEFAULT

Table 2-3. Default Precompensation Delays

Data Rate	Precompensation Delays
1 Mbps	41.67 ns
500 Kbps	125 ns
300 Kbps	125 ns
200 Kbps	125 ns

Table 2-4. Data Rates

DRATESEL 1	DRATESEL 0	DATA RATE MFM	DATA RATE FM
1	1	1 Mbps	Illegal
0	0	500 Kbps	250 Kbps
0	1	300 Kbps	150 Kbps
1	0	250 Kbps	125 Kbps

2.1.6 MAIN STATUS REGISTER (MSR)

The Main Status Register is a read-only register and is used for controlling command input and result output for all commands.

7	6	5	4	3	2	1	0
RQM	DIO	NON DMA	CMD BSY	DRV 3 BUSY	DRV 2 BUSY	DRV 1 BUSY	DRV 0 BUSY

RQM—Indicates that the host can transfer data if set to a 1. No access is permitted if set to a 0.

DIO—Indicates the direction of a data transfer once RQM is set. A 1 indicates a read and a 0 indicates a write is required.

NON-DMA—This mode is selected in the SPECIFY command and will be set to a 1 during the execution phase of a command. This is for polled data transfers and helps differentiate between the data transfer phase and the reading of result bytes.

COMMAND BUSY—This bit is set to a one when a command is in progress. This bit will go active after the command byte has been accepted and goes inactive at the end of the results phase. If there is no result phase (SEEK, RECALIBRATE commands), this bit is returned to a 0 after the last command byte.

DRV x BUSY—These bits are set to ones when a drive is in the seek portion of a command, including seeks, and recalibrates.

2.1.7 FIFO (DATA)

All command parameter information and disk data transfers go through the FIFO. The FIFO is 16 bytes in size and has programmable threshold values. Data transfers are governed by the RQM and DIO bits in the Main Status Register.

The FIFO defaults to an 8272A compatible mode after a "Hardware" reset (Reset via pin 32). "Software" Resets (Reset via DOR or DSR register) can also place the 82077AA into 8272A compatible mode if the LOCK bit is set to "0" (See section 5.3.2 for the definition of the LOCK bit). This maintains PC-AT hardware compatibility. The default values can be changed through the CONFIGURE command (enable full FIFO operation with threshold control). The advantage of the FIFO is that it allows the system a larger DMA latency without causing a disk error. Table 2.5 gives several examples of the delays with a FIFO. The data is based upon the following formula:

$$\text{Threshold\#} \times \left| \frac{1}{\text{DATA RATE}} \times 8 \right| - 1.5\ \mu s = \text{DELAY}$$

Table 2-5. FIFO Service Delay

FIFO Threshold Examples	Maximum Delay to Servicing at 1 Mbps Data Rate
1 byte	1 × 8 μs − 1.5 μs = 6.5 μs
2 bytes	2 × 8 μs − 1.5 μs = 14.5 μs
8 bytes	8 × 8 μs − 1.5 μs = 62.5 μs
15 bytes	15 × 8 μs − 1.5 μs = 118.5 μs

FIFO Threshold Examples	Maximum Delay to Servicing at 500 Kbps Data Rate
1 byte	1 × 16 μs − 1.5 μs = 14.5 μs
2 bytes	2 × 16 μs − 1.5 μs = 30.5 μs
8 bytes	8 × 16 μs − 1.5 μs = 126.5 μs
15 bytes	15 × 16 μs − 1.5 μs = 238.5 μs

At the start of a command, the FIFO action is always disabled and command parameters must be sent based upon the RQM and DIO bit settings. As the 82077AA enters the command execution phase, it clears the FIFO of any data to ensure that invalid data is not transferred.

An overrun or underrun will terminate the current command and the transfer of data. Disk writes will complete the current sector by generating a 00 pattern and valid CRC.

2.1.8a DIGITAL INPUT REGISTER (DIR, PC-AT MODE)

This register is read only in all modes. In PC-AT mode only bit 7 is driven, all other bits remain tristated.

7	6	5	4	3	2	1	0
DSK CHG	—	—	—	—	—	—	—

DSKCHG monitors the pin of the same name and reflects the opposite value seen on the disk cable, regardless of the value of $\overline{\text{INVERT}}$.

2.1.8b DIGITAL INPUT REGISTER (DIR, PS/2 MODE)

7	6	5	4	3	2	1	0
DSK CHG	1	1	1	1	DRATE SEL1	DRATE SEL0	$\overline{\text{HIGH}}$ $\overline{\text{DENS}}$

The following is changed in PS/2 Mode: Bits 6, 5, 4, and 3 return a value of "1", and the DRATE SEL1-0 return the value of the current data rate selected (see Table 2-4 for values).

$\overline{\text{HIGH DENS}}$ is low whenever the 500 Kbps or 1 Mbps data rates are selected. This bit is independent of the effects of the IDENT and $\overline{\text{INVERT}}$ pins.

Table 2-6 shows the state of the DENSEL pin when $\overline{\text{INVERT}}$ is low.

Table 2-6. DENSEL Encoding

Data Rate	IDENT*	DENSEL
1 Mbps	0	0
	1	1
500 Kbps	0	0
	1	1
300 Kbps	0	1
	1	0
250 Kbps	0	1
	1	0

*After ("Hardware") Chip Reset

This pin is set high after a pin RESET and is unaffected by DOR and DSR resets.

2.1.8c DIGITAL INPUT REGISTER (DIR, MODEL 30 MODE)

7	6	5	4	3	2	1	0
$\overline{DSK}$ $\overline{CHG}$	0	0	0	$\overline{DMA}$ $\overline{GATE}$	NOPREC	DRATE SEL1	DRATE SEL0

The following is changed in Model 30 Mode: Bits 6, 5, and 4 return a value of "0", and Bit 7 ($\overline{DSKCHG}$) is inverted in Model 30 Mode.

Bit 3 reflects the value of $\overline{DMAGATE}$ bit set in the DOR register.

Bit 2 reflects the value of NOPREC bit set in the CCR register.

2.1.9a CONFIGURATION CONTROL REGISTER (CCR, PC AT and PS/2 MODES)

This register sets the datarate and is write only. In the PC-AT it is named the DSR.

7	6	5	4	3	2	1	0
—	—	—	—	—	—	DRATE SEL1	DRATE SEL0

Refer to the table in the Data Rate Select Register for values. Unused bits should be set to 0.

2.1.9b CONFIGURATION CONTROL REGISTER (CCR, MODEL 30 MODE)

7	6	5	4	3	2	1	0
—	—	—	—	—	NOPREC	DRATE SEL1	DRATE SEL0

NOPREC has no function, and is reset to "0" with a Hardware RESET only.

2.2 RESET

There are three sources of reset on the 82077AA; the RESET pin, a reset generated via a bit in the DOR and a reset generated via a bit in the DSR. All resets take the 82077AA out of the power down state.

On entering the reset state, all operations are terminated and the 82077AA enters an idle state. Activating reset while a disk write activity is in progress will corrupt the data and CRC.

On exiting the reset state, various internal registers are cleared, and the 82077AA waits for a new command. Drive polling will start unless disabled by a new CONFIGURE command.

2.2.1 RESET PIN ("HARDWARE") RESET

The RESET pin is a global reset and clears all registers except those programmed by the SPECIFY command. The DOR Reset bit is enabled and must be cleared by the host to exit the reset state.

2.2.2 DOR RESET vs DSR RESET ("SOFTWARE" RESET)

These two resets are functionally the same. The DSR Reset is included to maintain 82072 compatibility. Both will reset the 8272 core which affects drive status information. The FIFO circuits will also be reset if the LOCK bit is a "0" (See section 5.3.2 for the definition of the LOCK bit). The DSR Reset clears itself automatically while the DOR Reset requires the host to manually clear it. DOR Reset has precedence over the DSR Reset. The DOR Reset is set automatically upon a pin RESET. The user must manually clear this reset bit in the DOR to exit the reset state.

2.3 DMA Transfers

DMA transfers are enabled with the SPECIFY command and are initiated by the 82077AA by activating the DRQ pin during a data transfer command. The FIFO is enabled directly by asserting $\overline{DACK}$ and addresses need not be valid.

3.0 DRIVE INTERFACE

The 82077AA has integrated all of the logic needed to interface to a floppy disk or tape drives which use floppy interface. All drive outputs have 40 mA drive capability and all inputs use a receive buffer with hysteresis. The internal analog data separator requires no external components, yet allows for an extremely wide capture range with high levels of read-data jitter, and ISV. The designer needs only to run the 82077AA disk drive signals to the disk or tape drive connector.

3.1 Cable Interface

The $\overline{\text{INVERT}}$ pin selects between using the internal buffers on the 82077AA or user supplied inverting buffers. $\overline{\text{INVERT}}$ pulled to V_{CC} disables the internal buffers; pulled to ground will enable them. There is no need to use external buffers with the 82077AA in typical PC applications.

The polarity of the DENSEL pin is controlled through the IDENT pin, after hardware reset. For 5.25″ drives a high on DENSEL tells the drive that either the 500 Kbps or 1 Mbps data rate is selected. For some 3.5″ drives the polarity of DENSEL changes to a low for high data rates. See **Table 2-6 DENSEL Encoding** for $\overline{\text{IDENT}}$ pin settings.

Additionally, the two types of drives have different electrical interfaces. Generally, the 5.25″ drive uses open collector drivers and the 3.5″ drives (as used on PS/2) use totem-pole drivers. The output buffers on the 82077AA do not change between open collector or totem-pole, they are always totem-pole. For design information on interfacing 5.25″ and 3.5″ drives to a single 82077AA, refer to Section 9.

3.2 Data Separator

The function of the data separator is to lock onto the incoming serial read data. When lock is achieved the serial front end logic of the chip is provided with a clock which is synchronized to the read data. The synchronized clock, called Data Window, is used to internally sample the serial data. One state of Data Window is used to sample the data portion of the bit cell, and the alternate state samples the clock portion. Serial to parallel conversion logic separates the read data into clock and data bytes.

To support reliable disk/tape reads the data separator must track fluctuations in the read data frequency. Frequency errors primarily arise from two sources: motor rotation speed variation and instantaneous speed variation (ISV). A second condition, and one that opposes the ability to track frequency shifts is the response to bit jitter.

The internal data separator consists of two analog phase lock loops (PLLs) as shown in Figure 3-1. The two PLLs are referred to as the reference PLL and the data PLL. The reference PLL (the master PLL) is used to bias the data PLL (the slave PLL). The reference PLL adjusts the data PLL's operating point as a function of process, junction temperature and supply voltage. Using this architecture it was possible to eliminate the need for external trim components.

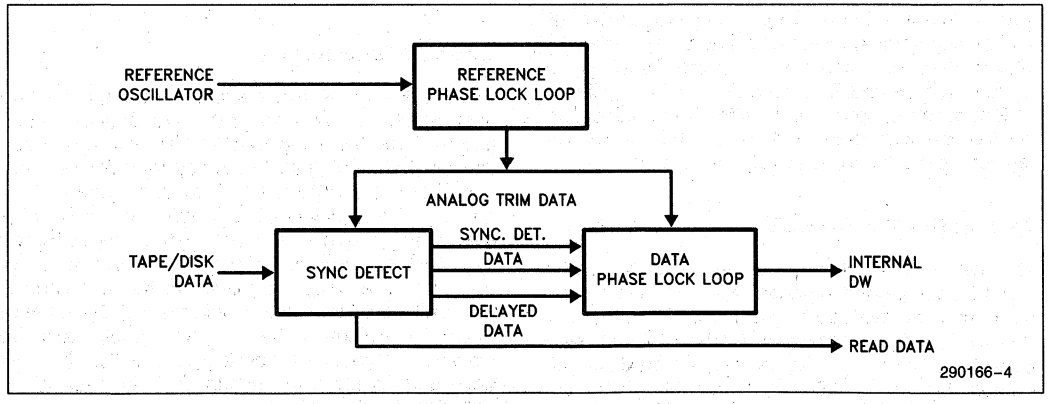


Figure 3-1. Data Separator Block Diagram

3.2.1 PHASE LOCK LOOP OVERVIEW

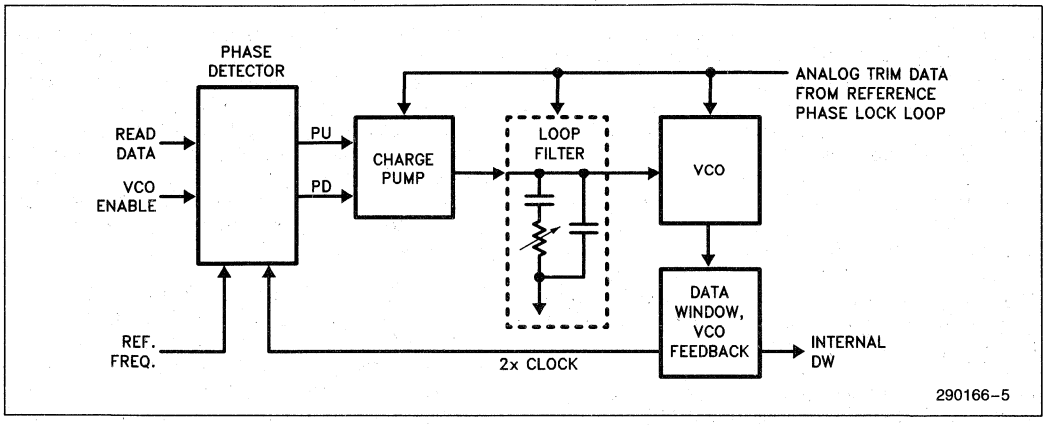


Figure 3-2. Data PLL

Figure 3-2 shows the data PLL. The reference PLL has control over the loop gain by its influence on the charge pump and the VCO. In addition the reference PLL controls the loop filter time constant. As a result the closed loop transfer function of the data PLL is controlled, and immune to the first order, to environmental factors and process variation.

Systems with analog PLLs are often very sensitive to noise. In the design of this data separator many steps were taken to avoid noise sensitivity problems. The analog section of the chip has a separate VSS pin (AVSS) which should be connected externally to a noise free ground. This provides a clean basis for VSS referenced signals. In addition many analog circuit features were employed to make the overall system as insensitive to noise as possible.

3.2.1 JITTER TOLERANCE

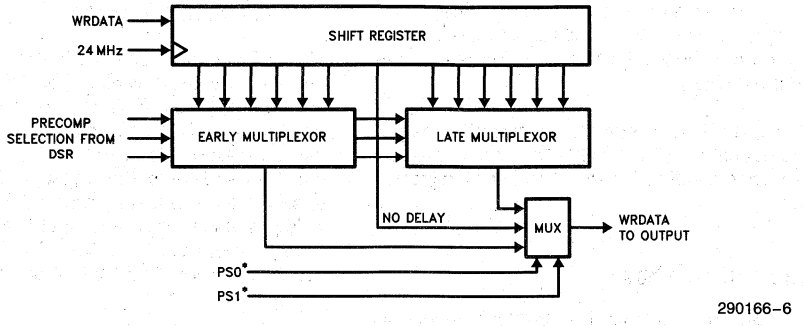
The jitter immunity of the system is dominated by the data PLL's response to phase impulses. This is measured as a percentage of the theoretical data window by dividing the maximum readable bit shift by a ¼ bitcell distance. For instance, if the maximum allowable bit shift is 300 ns for a 500 Kbps data stream, the jitter tolerance is 60%. The graph in Figures 12-1 thru 12-4 and 13-1 thru 13-4 of the Data Separator Characteristics sections illustrate the jitter tolerance of the 82077AA across each frequency range.

3.2.2 LOCKTIME (tLOCK)

The lock, or settling time of the data PLL is designed to be 64 bit times. This corresponds to 4 sync bytes in the FM mode and 8 sync bytes in the MFM mode. This value assumes that the sync field jitter is 5% the bit cell or less. This level of jitter should be easily achieved for a constant bit pattern, since intersymbol interference should be equal, thus nearly eliminating random bit shifting.

3.2.3 CAPTURE RANGE

Capture Range is the maximum frequency range over which the data separator will acquire phase lock with the incoming RDDATA signal. In a floppy disk environment, this frequency variation is composed of two components: drive motor speed error and ISV. Frequency is a factor which may determine the maximum level of the ISV (Instantaneous Speed Variation) component. In general, as frequency increases the allowed magnitude of the ISV component will decrease. When determining the capture range requirements, the designer should take the maximum amount of frequency error for the disk drive and double it to account for media switching between drives.



NOTE:
PS0,1 are 8272A control signals but are not available as outputs on the 82077AA.

Figure 3-3. Precompensation Block Diagram

3.2.4 REFERENCE FILTER

To provide a clean bias voltage for the internal data separator, two pins have been provided to filter this signal. It is recommended to place a 0.0047 uF capacitor between HIFIL and LOFIL to filter the reference signal. A smaller capacitance will reduce the effectiveness of the filter and could result in a lower jitter tolerance. Conversely, a larger capacitance has the potential to further improve jitter tolerance, but will result in an increased settling time after a change in data rate. For instance, a filter capacitor of 0.005 uF will yield a settling time of approximately 500 microseconds. Since HIFIL generates a relatively low current signal (approximately 10 uA), care also needs to be taken to avoid external leakage on this pin. The quality of the capacitor, solder flux, grease, and dirt can all impact the amount of leakage on the board.

3.3 Write Precompensation

The write precompensation logic is used to minimize bit shifts in the RDDATA stream from the disk drive. The shifting of bits is a known phenomena of magnetic media and is dependent upon the disk media AND the floppy drive.

The 82077AA monitors the bit stream that is being sent to the drive. The data patterns that require precompensation are well known. Depending upon the pattern, the bit is shifted either early or late (or not at all) relative to the surrounding bits. Figure 3-3 is a block diagram of the internal circuit.

The top block is a 13-bit shift register with the no delay tap being in the center. This allows 6 levels of early and late shifting with respect to nominal. The shift register is clocked at the main clock rate (24 MHz). The output is fed into 2 multiplexors—one for early and one for late. A final stage of multiplexors combines the early, late and normal data stream back into one which is the WRDATA output.

4.0 CONTROLLER PHASES

For simplicity, command handling in the 82077AA can be divided into three phases: Command, Execution and Result. Each phase is described in the following secions.

4.1 Command Phase

After a reset, the 82077AA enters the command phase and is ready to accept a command from the host. For each of the commands, a defined set of command code bytes and parameter bytes has to be written to the 82077AA before the command phase is complete (Please refer to Section 5.0 for the command descriptions). These bytes of data must be transferred in the order prescribed.

Before writing to the 82077AA, the host must examine the RQM and DIO bits of the Main Status Register. RQM, DIO must be equal to "1" and "0" respectively before command bytes may be written. RQM is set false by the 82077AA after each write cycle until the received byte is processed. The 82077AA asserts RQM again to request each parameter byte of

the command, unless an illegal command condition is detected. After the last parameter byte is received, RQM remains "0", and the 82077AA automatically enters the next phase as defined by the command definition.

The FIFO is disabled during the command phase to retain compatibility with the 8272A, and to provide for the proper handling of the "Invalid Command" condition.

4.2 Execution Phase

All data transfers to or from the 82077AA occur during the execution phase, which can proceed in DMA or non-DMA mode as indicated in the SPECIFY command.

Each data byte is transferred by an INT or DRQ depending on the DMA mode. The CONFIGURE command can enable the FIFO and set the FIFO threshold value.

The following paragraphs detail the operation of the FIFO flow control. In these descriptions, <threshold> is defined as the number of bytes available to the 82077AA when service is requested from the host, and ranges from 1 to 16. The parameter FIFOTHR which the user programs is one less, and ranges from 0 to 15.

A low threshold value (i.e. 2) results in longer periods of time between service requests, but requires faster servicing of the request, for both read and write cases. The host reads (writes) from (to) the FIFO until empty (full), then the transfer request goes inactive. The host must be very responsive to the service request. This is the desired case for use with a "fast" system.

A high value of threshold (i.e. 12) is used with a "sluggish" system by affording a long latency period after a service request, but results in more frequent service requests.

4.2.1 NON-DMA MODE, TRANSFERS FROM THE FIFO TO THE HOST

The INT pin and RQM bits in the Main Status Register are activated when the FIFO contains (16–<threshold>) bytes, or the last bytes of a full sector transfer have been placed in the FIFO. The INT pin can be used for interrupt driven systems and RQM can be used for polled sytems. The host must respond to the request by reading data from the FIFO. This process is repeated until the last byte is transferred out of the FIFO. The 82077AA will deactivate the INT pin and RQM bit when the FIFO becomes empty.

4.2.2 NON-DMA MODE, TRANSFERS FROM THE HOST TO THE FIFO

The INT pin and RQM bit in the Main Status Register are activated upon entering the execution phase of data transfer commands. The host must respond to the request by writing data into the FIFO. The INT pin and RQM bit remain true until the FIFO becomes full. They are set true again when the FIFO has <threshold> bytes remaining in the FIFO. The INT pin will also be deactivated if TC and DACK# both go inactive. The 82077AA enters the result phase after the last byte is taken by the 82077AA from the FIFO (i.e. FIFO empty condition).

4.2.3 DMA MODE, TRANSFERS FROM THE FIFO TO THE HOST

The 82077AA activates the DRQ pin when the FIFO contains (16–<threshold>) bytes, or the last byte of a full sector transfer has been placed in the FIFO. The DMA controller must respond to the request by reading data from the FIFO. The 82077AA will deactivate the DRQ pin when the FIFO becomes empty. DRQ goes inactive after DACK# goes active for the last byte of a data transfer (or on the active edge of RD#, on the last byte, if no edge is present on DACK#). A data underrun may occur if DRQ is not removed in time to prevent an unwanted cycle.

4.2.4 DMA MODE, TRANSFERS FROM THE HOST TO THE FIFO

The 82077AA activates the DRQ pin when entering the execution phase of the data transfer commands. The DMA controller must respond by activating the DACK# and WR# pins and placing data in the FIFO. DRQ remains active until the FIFO becomes full. DRQ is again set true when the FIFO has <threshold> bytes remaining in the FIFO. The 82077AA will also deactivate the DRQ pin when TC becomes true (qualified by DACK#), indicating that no more data is required. DRQ goes inactive after DACK# goes active for the last byte of a data transfer (or on the active edge of WR# of the last byte, if no edge is present on DACK#). A data overrun may occur if DRQ is not removed in time to prevent an unwanted cycle.

4.2.5 DATA TRANSFER TERMINATION

The 82077AA supports terminal count explicitly through the TC pin and implicitly through the underrun/overrun and end-of-track (EOT) functions. For full sector transfers, the EOT parameter can define the last sector to be transferred in a single or multi-sector transfer. If the last sector to be transferred is a partial sector, the host can stop transferring the data in mid-sector, and the 82077AA will continue to complete the sector as if a hardware TC was received. The only difference between these implicit functions and TC is that they return "abnormal termination" result status. Such status indications can be ignored if they were expected.

Note that when the host is sending data to the FIFO of the 82077AA, the internal sector count will be complete when 82077AA reads the last byte from its side of the FIFO. There may be a delay in the removal of the transfer request signal of up to the time taken for the 82077AA to read the last 16 bytes from the FIFO. The host must tolerate this delay.

4.3 Result Phase

The generation of INT determines the beginning of the result phase. For each of the commands, a defined set of result bytes has to be read from the 82077AA before the result phase is complete. (Refer to Section 5.0 on command descriptions.) These bytes of data must be read out for another command to start.

RQM and DIO must both equal "1" before the result bytes may be read from the FIFO. After all the result bytes have been read, the RQM and DIO bits switch to "1" and "0" respectively, and the CB bit is cleared. This indicates that the 82077AA is ready to accept the next command.

5.0 COMMAND SET/DESCRIPTIONS

Commands can be written whenever the 82077AA is in the command phase. Each command has a unique set of needed parameters and status results. The 82077AA checks to see that the first byte is a valid command and, if valid, proceeds with the command. If it was invalid, the next time the RQM bit in the MSR register is a "1" the DIO and CB bits will also be "1", indicating the FIFO must be read. A result byte of 80H will be read out of the FIFO, indicating an invalid command was issued. After reading the result byte from the FIFO the 82077AA will return to the command phase. Table 5-1 is a summary of the Command set.

Table 5-1. 82077AA Command Set

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
READ DATA										
Command	W	MT	MFM	SK	0	0	1	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior to Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data transfer between the FDD and system
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID information after Command execution
	R					R				
	R					N				
READ DELETED DATA										
Command	W	MT	MFM	SK	0	1	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior to Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data transfer between the FDD and system
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID information after Command execution
	R					R				
	R					N				
WRITE DATA										
Command	W	MT	MFM	0	0	0	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior to Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data transfer between the system and FDD
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID information after Command execution
	R					R				
	R					N				

Table 5-1. 82077AA Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
WRITE DELETED DATA										
Command	W	MT	MFM	0	0	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior to Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data transfer between the FDD and system
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID information after Command execution
	R					R				
	R					N				
READ TRACK										
Command	W	0	MFM	0	0	0	0	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior to Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data transfer between the FDD and system. FDC reads all of cylinders contents from index hole to EOT
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID information after Command execution
	R					R				
	R					N				
VERIFY										
Command	W	MT	MFM	SK	1	0	1	1	0	Command Codes
	W	EC	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior to Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL/SC				
Execution										No data transfer takes place
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID information after Command execution
	R					R				
	R					N				
VERSION										
Command	W	0	0	0	1	0	0	0	0	Command Code
Result	R	1	0	0	1	0	0	0	0	Enhanced Controller

4

Table 5-1. 82077AA Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
FORMAT TRACK										
Command	W	0	MFM	0	0	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				N					Bytes/Sector
	W				SC					Sectors/Cylinder
	W				GPL					Gap 3
	W				D					Filler Byte
Execution For Each Sector Repeat:	W				C					Input Sector Parameters
	W				H					
	W				R					
	W				N					
										82077AA formats an entire cylinder
Result	R				ST 0					Status information after Command execution
	R				ST 1					
	R				ST 2					
	R				Undefined					
	R				Undefined					
	R				Undefined					
	R				Undefined					
RECALIBRATE										
Command	W	0	0	0	0	0	1	1	1	Command Codes
	W	0	0	0	0	0	0	DS1	DS0	
Execution										Head retracted to Track 0 Interrupt
SENSE INTERRUPT STATUS										
Command	W	0	0	0	0	1	0	0	0	Command Codes
Result	R				ST 0					Status information at the end of each seek operation
	R				PCN					
SPECIFY										
Command	W	0	0	0	0	0	0	1	1	Command Codes
	W	SRT					HUT			
	W	HLT							ND	
SENSE DRIVE STATUS										
Command	W	0	0	0	0	0	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Result	R				ST 3					Status information about FDD
SEEK										
Command	W	0	0	0	0	1	1	1	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				NCN					
Execution										Head is positioned over proper Cylinder on Diskette
CONFIGURE										
Command	W	0	0	0	1	0	0	1	1	Configure Information
	W	0	0	0	0	0	0	0	0	
	W	0	EIS	EFIFO	POLL			FIFOTHR		
	W				PRETRK					
RELATIVE SEEK										
Command	W	1	DIR	0	0	1	1	1	1	
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				RCN					

Table 5-1. 82077AA Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
DUMPREG										
Command	W	0	0	0	0	1	1	1	0	*Note
Execution										Registers placed in FIFO
Result	R				PCN-Drive 0					
	R				PCN-Drive 1					
	R				PCN-Drive 2					
	R				PCN-Drive 3					
	R		SRT				HUT			
	R				HLT				ND	
	R				SC/EOT					
	R	LOCK	0	D_3	D_2	D_1	D_0	GAP	WGATE	
	R	0	EIS	EFIFO	POLL		FIFOTHR			
	R				PRETRK					
READ ID										
Command	W	0	MFM	0	0	1	0	1	0	Commands
	W	0	0	0	0	0	HDS	DS1	DS0	
Execution										The first correct ID information on the Cylinder is stored in Data Register
Result	R				ST 0					Status information after Command execution
	R				ST 1					
	R				ST 2					
	R				C					
	R				H					Disk status after the Command has completed.
	R				R					
	R				N					
PERPENDICULAR MODE										
Command	W	0	0	0	1	0	0	1	0	Command Codes
	W	OW	0	D_3	D_2	D_1	D0	GAP	WGATE	
LOCK										
Command	W	LOCK	0	0	1	0	1	0	0	Command Code
Result	R	0	0	0	LOCK	0	0	0	0	
INVALID										
Command	W				Invalid Codes					Invalid Command Codes (NoOp — 82077AA goes into Standby State)
Result	R				ST 0					ST 0 = 80H

SC is returned if the last command that was issued was the FORMAT command. EOT is returned if the last command was a READ or WRITE.

NOTE:
These bits are used internally only. They are not reflected in the Drive Select pins. It is the users responsibility to maintain correspondence between these bits and the Drive Select pins (DOR).

PARAMETER ABBREVIATIONS

Symbol	Description
C	Cylinder address. The currently selected cylinder address, 0 to 255.
D_0, D_1 D_2, D_3	Drive Select 0-3. Designates which drives are Perpendicular drives, a "1" indicating Perpendicular drive.
D	Data pattern. The pattern to be written in each sector data field during formatting.
DIR	Direction control. If this bit is 0, then the head will step out from the spindle during a relative seek. If set to a 1, the head will step in toward the spindle.
DS0, DS1	Disk Drive Select.

DS1	DS0	
0	0	drive 0
0	1	drive 1
1	0	drive 2
1	1	drive 3

Symbol	Description
DTL	Special sector size. By setting N to zero (00), DTL may be used to control the number of bytes transferred in disk read/write commands. The sector size (N = 0) is set to 128. If the actual sector (on the diskette) is larger than DTL, the remainder of the actual sector is read but is not passed to the host during read commands; during write commands, the remainder of the actual sector is written with all zero bytes. The CRC check code is calculated with the actual sector. When N is not zero, DTL has no meaning and should be set to FF HEX.
EC	Enable Count. When this bit is "1" the "DTL" parameter of the Verify Command becomes SC (Number of sectors per track).
EFIFO	Enable FIFO. When this bit is 0, the FIFO is enabled. A "1" puts the 82077AA in the 8272A compatible mode where the FIFO is disabled.
EIS	Enable implied seek. When set, a seek operation will be performed before executing any read or write command that requires the C parameter in the command phase. A "0" disables the implied seek.
EOT	End of track. The final sector number of the current track.
GAP	Alters Gap 2 length when using Perpendicular Mode.
GPL	Gap length. The gap 3 size. (Gap 3 is the space between sectors excluding the VCO synchronization field).
H/HDS	Head address. Selected head: 0 or 1 (disk side 0 or 1) as encoded in the sector ID field.
HLT	Head load time. The time interval that 82077AA waits after loading the head and before initiating a read or write operation. Refer to the SPECIFY command for actual delays.
HUT	Head unload time. The time interval from the end of the execution phase (of a read or write command) until the head is unloaded. Refer to the SPECIFY command for actual delays.
Lock	Lock defines whether EFIFO, FIFOTHR, and PRETRK parameters of the CONFIGURE command can be reset to their default values by a "Software Reset" (Reset made by setting the proper bit in the DSR or DOR registers).
MFM	MFM/FM mode selector. A one selects the double density (MFM) mode. A zero selects single density (FM) mode.
MT	Multi-track selector. When set, this flag selects the multi-track operating mode. In this mode, the 82077AA treats a complete cylinder, under head 0 and 1, as a single track. The 82077AA operates as if this expanded track started at the first sector under head 0 and ended at the last sector under head 1. With this flag set, a multitrack read or write operation will automatically continue to the first sector under head 1 when the 82077AA finishes operating on the last sector under head 0.
N	Sector size code. This specifies the number of bytes in a sector. If this parameter is "00", then the sector size is 128 bytes. The number of bytes transferred is determined by the DTL parameter. Otherwise the sector size is (2 raised to the "N'th" power) times 128. All values up to "07" hex are allowable. "07"h would equal a sector size of 16k. It is the users responsibility to not select combinations that are not possible with the drive.

N	Sector Size
00	128 bytes
01	256 bytes
02	512 bytes
03	1024 bytes
..	...
07	16 Kbytes

NCN	New cylinder number. The desired cylinder number.
ND	Non-DMA mode flag. When set to 1, indicates that the 82077AA is to operate in the non-DMA mode. In this mode, the host is interrupted for each data transfer. When set to 0, the 82077AA operates in DMA mode, interfacing to a DMA controller by means of the DRQ and DACK# signals.
OW	The bits denoted D_0, D_1, D_2, and D_3 of the **PERPENDICULAR MODE** command can only be overwritten when the OW bit is set to "1".
PCN	Present cylinder number. The current position of the head at the completion of SENSE INTERRUPT STATUS command.
POLL	Polling disable. When set, the internal polling routine is disabled. When clear, polling is enabled.
PRETRK	Precompensation start track number. Programmable from track 00 to FFH.

Symbol	Description
R	Sector address. The sector number to be read or written. In multi-sector transfers, this parameter specifies the sector number of the first sector to be read or written.
RCN	Relative cylinder number. Relative cylinder offset from present cylinder as used by the RELATIVE SEEK command.
SC	Number of sectors. The number of sectors to be initialized by the FORMAT command. The number of sectors to be verified during a Verify Command, when EC is set.
SK	Skip flag. When set to 1, sectors containing a deleted data address mark will automatically be skipped during the execution of READ DATA. If READ DELETED is executed, only sectors with a deleted address mark will be accessed. When set to "0", the sector is read or written the same as the read and write commands.
SRT	Step rate interval. The time interval between step pulses issued by the 82077AA. Programmable from 0.5 to 8 milliseconds, in increments of 0.5 ms at the 1 Mbit data rate. Refer to the SPECIFY command for actual delays.
ST0 ST1 ST2 ST3	Status register 0–3. Registers within the 82077AA that store status information after a command has been executed. This status information is available to the host during the result phase after command execution.
WGATE	Write gate alters timing of WE, to allow for pre-erase loads in perpendicular drives.

5.1 Data Transfer Commands

All of the READ DATA, WRITE DATA and VERIFY type commands use the same parameter bytes and return the same results information. The only difference being the coding of bits 0–4 in the first byte.

An implied seek will be executed if the feature was enabled by the CONFIGURE command. This seek is completely transparent to the user. The Drive Busy bit for the drive will go active in the Main Status Register during the seek portion of the command. If the seek portion fails, it will be reflected in the results status normally returned for a READ/WRITE DATA command. Status Register 0 (ST0) would contain the error code and C would contain the cylinder on which the seek failed.

5.1.1 READ DATA

A set of nine (9) bytes is required to place the 82077AA into the Read Data Mode. After the READ DATA command has been issued, the 82077AA loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the SPECIFY command), and begins reading ID Address Marks and ID fields. When the sector address read off the diskette matches with the sector address specified in the command, the 82077AA reads the sector's data field and transfers the data to the FIFO.

After completion of the read operation from the current sector, the sector address is incremented by one, and the data from the next logical sector is read and output via the FIFO. This continuous read function is called "Multi-Sector Read Operation". Upon receipt of TC, or an implied TC (FIFO overrun/underrun), the 82077AA stops sending data, but will continue to read data from the current sector, check the CRC bytes, and at the end of the sector terminate the READ DATA Command.

N determines the number of bytes per sector (see Table 5-2 below). If N is set to zero, the sector size is set to 128. The DTL value determines the number of bytes to be transferred. If DTL is less than 128, the 82077AA transfers the specified number of bytes to the host. For reads, it continues to read the entire 128 byte sector and checks for CRC errors. For writes it completes the 128 byte sector by filling in zeroes. If N is not set to 00 Hex, DTL should be set to FF Hex, and has no impact on the number of bytes transferred.

Table 5-2. Sector Sizes

N	Sector Size
00	128 bytes
01	256 bytes
02	512 bytes
03	1024 bytes
..	...
07	16 Kbytes

The amount of data which can be handled with a single command to the 82077AA depends upon MT (multi-track) and N (Number of bytes/sector).

Table 5-3. Effects of MT and N Bits

MT	N	Max. Transfer Capacity	Final Sector Read from Disk
0	1	256 × 26 = 6,656	26 at side 0 or 1
1	1	256 × 52 = 13,312	26 at side 1
0	2	512 × 15 = 7,680	15 at side 0 or 1
1	2	512 × 30 = 15,360	15 at side 1
0	3	1024 × 8 = 8,192	8 at side 0 or 1
1	3	1024 × 16 = 16,384	16 at side 1

The Multi-Track function (MT) allows the 82077AA to read data from both sides of the diskette. For a particular cylinder, data will be transferred starting at Sector 1, Side 0 and completing at the last sector of the same track at Side 1.

If the host terminates a read or write operation in the 82077AA, then the ID information in the result phase is dependent upon the state of the MT bit and EOT byte. Refer to Table 5-6.

At the completion of the READ DATA Command, the head is not unloaded until after the Head Unload Time Interval (specified in the SPECIFY command) has elapsed. If the host issues another command before the head unloads then the head settling time may be saved between subsequent reads.

If the 82077AA detects a pulse on the IDX pin twice without finding the specified sector (meaning that the diskette's index hole passes through index detect logic in the drive twice), the 82077AA sets the IC code in Status Register 0 to "01" (Abnormal termination), and sets the ND bit in Status Register 1 to "1" indicating a sector not found, and terminates the READ DATA Command.

After reading the ID and Data Fields in each sector, the 82077AA checks the CRC bytes. If a CRC error occurs in the ID or data field, the 82077AA sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the DE bit flag in Status Register 1 to "1", sets the DD bit in Status Register 2 to "1" if CRC is incorrect in the ID field, and terminates the READ DATA Command.

Table 5-4 below describes the affect of the SK bit on the READ DATA command execution and results.

Table 5-4. Skip Bit vs READ DATA Command

SK Bit Value	Data Address Mark Type Encountered	Results		
		Sector Read?	CM Bit of ST2 Set?	Description of Results
0	Normal Data	Yes	No	Normal Termination.
0	Deleted Data	Yes	Yes	Address Not Incremented. Next Sector Not Searched For.
1	Normal Data	Yes	No	Normal Termination.
1	Deleted Data	No	Yes	Normal Termination Sector Not Read ("Skipped").

Except where noted in Table 5-4, the C or R value of the sector address is automatically incremented (see Table 5-6).

5.1.2 READ DELETED DATA

This command is the same as the READ DATA command, only it operates on sectors that contain a Deleted Data Address Mark at the beginning of a Data Field.

Table 5-5 describes the affect of the SK bit on the READ DELETED DATA command execution and results.

Table 5-5. Skip Bit vs READ DELETED DATA Command

SK Bit Value	Data Address Mark Type Encountered	Results		
		Sector Read?	CM Bit of ST2 Set?	Description of Results
0	Normal Data	Yes	Yes	Address Not Incremented. Next Sector Not Searched For.
0	Deleted Data	Yes	No	Normal Termination.
1	Normal Data	No	Yes	Normal Termination Sector Not Read ("Skipped").
1	Deleted Data	Yes	No	Normal Termination.

Except where noted in Table 5-5 above, the C or R value of the sector address is automatically incremented (See Table 5-6).

5.1.3 READ TRACK

This command is similar to the READ DATA command except that the entire data field is read continuously from each of the sectors of a track. Immediately after encountering a pulse on the IDX pin, the 82077AA starts to read all data fields on the track as continuous blocks of data without regard to logical sector numbers. If the 82077AA finds an error in the ID or DATA CRC check bytes, it continues to read data from the track and sets the appropriate error bits at the end of the command. The 82077AA compares the ID information read from each sector with the specified value in the command, and sets the ND flag of Status Register 1 to a "1" if there is no comparison. Multi-track or skip operations are not allowed with this command. The MT and SK bits (Bits D7 and D5 of the first command byte respectively) should always be set to "0".

This command terminates when the EOT specified number of sectors have been read. If the 82077AA

Table 5-6. Result Phase Table

MT	Head	Final Sector Transferred to Host	ID Information at Result Phase			
			C	H	R	N
0	0	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	NC	01	NC
	1	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	NC	01	NC
1	0	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	NC	LSB	01	NC
	1	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	LSB	01	NC

NC: no change, the same value as the one at the beginning of command execution.
LSB: least significant bit, the LSB of H is complemented.

does not find an ID Address Mark on the diskette after the second occurrence of a pulse on the IDX pin, then it sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the MA bit in Status Register 1 to "1", and terminates the command.

5.1.4 WRITE DATA

After the WRITE DATA command has been issued, the 82077AA loads the head (if it is in the unloaded state), waits the specified head load time if unloaded (defined in the SPECIFY command), and begins reading ID Fields. When the sector address read from the diskette matches the sector address specified in the command, the 82077AA reads the data from the host via the FIFO, and writes it to the sector's data field.

After writing data into the current sector, the 82077AA computes the CRC value and writes it into the CRC field at the end of the sector transfer. The Sector Number stored in "R" is incremented by one, and the 82077AA continues writing to the next data field. The 82077AA continues this "Multi-Sector Write Operation". Upon receipt of a terminal count signal or if a FIFO over/under run occurs while a data field is being written, then the remainder of the data field is filled with zeros.

The 82077AA reads the ID field of each sector and checks the CRC bytes. If it detects a CRC error in one of the ID Fields, it sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the DE bit of Status Register 1 to "1", and terminates the WRITE DATA command.

The WRITE DATA command operates in much the same manner as the READ DATA command. The following items are the same. Please refer to the READ DATA Command for details:

- Transfer Capacity
- EN (End of Cylinder) bit
- ND (No Data) bit
- Head Load, Unload Time Interval
- ID information when the host terminates the command.
- Definition of DTL when N = 0 and when N does not = 0.

5.1.5 WRITE DELETED DATA

This command is almost the same as the WRITE DATA command except that a Deleted Data Address Mark is written at the beginning of the Data Field instead of the normal Data Address Mark. This command is typically used to mark a bad sector containing an error on the floppy disk.

5.1.6 VERIFY

The VERIFY command is used to verify the data stored on a disk. This command acts exactly like a READ DATA command except that no data is transferred to the host. Data is read from the disk, CRC computed and checked against the previously stored value.

Because no data is transferred to the host, TC (pin 25) cannot be used to terminate this command. By setting the EC bit to "1" an implicit TC will be issued to the 82077AA. This implicit TC will occur when the SC value has decrement to 0 (an SC value of 0 will verify 256 sectors). This command can also be terminated by setting the EC bit to "0" and the EOT value equal to the final sector to be checked. If EC is set to "0" DTL/SC should be programmed to 0FFH. Refer to Table 5-6 and Table 5-7 for information concerning the values of MT and EC versus SC and EOT value.

Definitions:

Sectors Per Side = Number of formatted sectors per each side of the disk.

Sectors Remaining = Number of formatted sectors left which can be read, including side 1 of the disk if MT is set to "1".

Table 5-7. Verify Command Result Phase Table

MT	EC	SC/EOT Value	Termination Result
0	0	SC = DTL EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
0	0	SC = DTL EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
0	1	SC ≤ # Sectors Remaining AND EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
0	1	SC > # Sectors Remaining OR EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
1	0	SC = DTL EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
1	0	SC = DTL EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
1	1	SC ≤ # Sectors Remaining AND EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
1	1	SC > # Sectors Remaining OR EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid

NOTE:
If MT is set to "1" and the SC value is greater than the number of remaining formatted sectors on Side 0, verifying will continue on Side 1 of the disk.

5.1.7 FORMAT TRACK

The FORMAT command allows an entire track to be formatted. After a pulse from the IDX pin is detected, the 82077AA starts writing data on the disk including Gaps, Address Marks, ID Fields and Data Fields, per the IBM System 34 or 3740 format (MFM or FM respectively). The particular values that will be written to the gap and data field are controlled by the values programmed into N, SC, GPL, and D which are specified by the host during the command phase. The data field of the sector is filled with the data byte specified by D. The ID Field for each sector is supplied by the host; that is, four data bytes per sector are needed by the 82077AA for C, H, R, and N (cylinder, head, sector number and sector size respectively).

After formatting each sector, the host must send new values for C, H, R and N to the 82077AA for the next sector on the track. The R value (sector number) is the only value that must be changed by the host after each sector is formatted. This allows the disk to be formatted with nonsequential sector addresses (interleaving). This incrementing and formatting continues for the whole track until the 82077AA encounters a pulse on the IDX pin again and it terminates the command.

Table 5-8 contains typical values for gap fields which are dependent upon the size of the sector and the number of sectors on each track. Actual values can vary due to drive electronics.

Table 5-8. Typical Values for Formatting

		Sector Size	N	SC	GPL1	GPL2
5.25″ Drives	FM	128	00	12	07	09
		128	00	10	10	19
		512	02	08	18	30
		1024	03	04	46	87
		2048	04	02	C8	FF
		4096	05	01	C8	FF
		...	...			
	MFM	256	01	12	0A	0C
		256	01	10	20	32
		512*	02	09	2A	50
		1024	03	04	80	F0
		2048	04	02	C8	FF
		4096	05	01	C8	FF
		...	...			
3.5″ Drives	FM	128	0	0F	07	1B
		256	1	09	0F	2A
		512	2	05	1B	3A
	MFM	256	1	0F	0E	36
		512**	2	09	1B	54
		1024	3	05	35	74

GPL1 = suggested GPL values in read and write commands to avoid splice point between data field and ID field of contiguous sections.
GPL2 = suggested GPL value in FORMAT TRACK command.
*PC-AT values (typical)
**PS/2 values (typical). Applies with 1.0 MB and 2.0 MB drives.

NOTE:
All values except Sector Size are in Hex.

4

5.1.7.1 Format Fields

GAP 4a	SYNC	IAM		GAP 1	SYNC	IDAM		C	H	S	N	C	GAP 2	SYNC	DATA AM		DATA	C	GAP 3	GAP 4b
80x 4E	12x 00	3x C2	FC	50x 4E	12x 00	3x A1	FE	Y L	D	E C	O	R C	22x 4E	12x 00	3x A1	FB F8		R C		

Figure 5-1. System 34 Format Double Density

GAP 4a	SYNC	IAM	GAP 1	SYNC	IDAM	C	H	S	N	C	GAP 2	SYNC	DATA AM	DATA	C	GAP 3	GAP 4b
40x FF	6x 00	FC	26x FF	6x 00	FE	Y L	D	E C	O	R C	11x FF	6x 00	FB or F8		R C		

Figure 5-2. System 3740 Format Single Density

GAP 4a	SYNC	IAM		GAP 1	SYNC	IDAM		C	H	S	N	C	GAP 2	SYNC	DATA AM		DATA	C	GAP 3	GAP 4b
80x 4E	12x 00	3x C2	FC	50x 4E	12x 00	3x A1	FE	Y L	D	E C	O	R C	41x 4E	12x 00	3x A1	FB F8		R C		

Figure 5-3. Perpendicular Format

5.2 Control Commands

Control commands differ from the other commands in that no data transfer takes place. Three commands generate an interrupt when complete; READ ID, RECALIBRATE and SEEK. The other control commands do not generate an interrupt.

5.2.1 READ ID

The READ ID command is used to find the present position of the recording heads. The 82077AA stores the values from the first ID Field it is able to read into its registers. If the 82077AA does not find an ID Address Mark on the diskette after the second occurrence of a pulse on the IDX pin, it then sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the MA bit in Status Register 1 to "1", and terminates the command.

The following commands will generate an interrupt upon completion. They do not return any result bytes. It is highly recommended that control commands be followed by the SENSE INTERRUPT STATUS command. Otherwise, valuable interrupt status information will be lost.

5.2.2 RECALIBRATE

This command causes the read/write head within the 82077AA to retract to the track 0 position. The 82077AA clears the contents of the PCN counter, and checks the status of the TRK0 pin from the FDD. As long as the TRK0 pin is low, the DIR pin remains 0 and step pulses are issued. When the TRK0 pin goes high, the SE bit in Status Register 0 is set to "1", and the command is terminated. If the TRK0 pin is still low after 79 step pulses have been issued, the 82077AA sets the SE and the EC bits of Status Register 0 to "1", and terminates the command. Disks capable of handling more than 80 tracks per side may require more than one RECALIBRATE command to return the head back to physical Track 0.

The RECALIBRATE command does not have a result phase. SENSE INTERRUPT STATUS command must be issued after the RECALIBRATE command to effectively terminate it and to provide verification of the head position (PCN). During the command phase of the recalibrate operation, the 82077AA is in the BUSY state, but during the execution phase it is in a NON BUSY state. At this time another RECALIBRATE command may be issued, and in this manner, parallel RECALIBRATE operations may be done on up to 4 drives at once.

Upon power up, the software must issue a RECALIBRATE command to properly initialize all drives and the controller.

5.2.3 SEEK

The read/write head within the drive is moved from track to track under the control of the SEEK Command. The 82077AA compares the PCN which is the current head position with the NCN and performs the following operation if there is a difference:

—PCN < NCN: Direction signal to drive set to "1" (step in), and issues step pulses.

—PCN > NCN: Direction signal to drive set to "0" (step out), and issues step pulses.

The rate at which step pulses are issued is controlled by SRT (Stepping Rate Time) in the SPECIFY command. After each step pulse is issued, NCN is compared against PCN, and when NCN = PCN, then the SE bit in Status Register 0 is set to "1", and the command is terminated.

During the command phase of the seek or recalibrate operation, the 82077AA is in the BUSY state, but during the execution phase it is in the NON BUSY state.

Note that if implied seek is not enabled, the read and write commands should be preceded by:

1) SEEK command; Step to the proper track
2) SENSE INTERRUPT STATUS command; Terminate the Seek command
3) READ ID. Verify head is on proper track
4) Issue READ/WRITE command.

The SEEK command does not have a result phase. Therefore, it is highly recommended that the SENSE INTERRUPT STATUS Command be issued after the SEEK command to terminate it and to provide verification of the head position (PCN). The H bit (Head Address) in ST0 will always return a "0". When exiting POWERDOWN mode, the 82077AA clears the PCN value and the status information to zero. Prior to issuing the POWERDOWN command, it is highly recommended that the user service all pending interrupts through the SENSE INTERRUPT STATUS command.

5.2.4 SENSE INTERRUPT STATUS

An interrupt signal on INT pin is generated by the 82077AA for one of the following reasons:

1. Upon entering the Result Phase of:
   a. READ DATA Command
   b. READ TRACK Command
   c. READ ID Command
   d. READ DELETED DATA Command
   e. WRITE DATA Command
   f. FORMAT TRACK Command
   g. WRITE DELETED DATA Command
   h. VERIFY Command
2. End of SEEK, RELATIVE SEEK or RECALIBRATE Command
3. 82077AA requires a data transfer during the execution phase in the non-DMA Mode

The SENSE INTERRUPT STATUS command resets the interrupt signal and via the IC code and SE bit of Status Register 0, identifies the cause of the interrupt. If a SENSE INTERRUPT STATUS command is issued when no active interrupt condition is present, the status register ST0 will return a value of 80H (invalid command).

Table 5-9. Interrupt Identification

SE	IC	Interrupt Due To
0	11	Polling
1	00	Normal Termination of SEEK or RECALIBRATE command
1	01	Abnormal Termination of SEEK or RECALIBRATE command

The SEEK, RELATIVE SEEK and the RECALIBRATE commands have no result phase. SENSE INTERRUPT STATUS command must be issued immediately after these commands to terminate them and to provide verification of the head position (PCN). The H (Head Address) bit in ST0 will always return a "0". If a SENSE INTERRUPT STATUS is not issued, the drive, will continue to be BUSY and may effect the operation of the next command.

5.2.5 SENSE DRIVE STATUS

SENSE DRIVE STATUS obtains drive status information. It has no execution phase and goes directly to the result phase from the command phase. STATUS REGISTER 3 contains the drive status information.

5.2.6 SPECIFY

The SPECIFY command sets the initial values for each of the three internal timers. The HUT (Head Unload Time) defines the time from the end of the execution phase of one of the read/write commands to the head unload state. The SRT (Step Rate Time) defines the time interval between adjacent step pulses. Note that the spacing between the first and second step pulses may be shorter than the remaining step pulses. The HLT (Head Load Time) defines the time between the Head Load signal goes high and the read, write operation starts. The values change with the data rate speed selection and are documented in Table 5-10. The values are the same for MFM and FM.

Table 5-10. Drive Control Delays (ms)

	HUT				SRT			
	1M	500K	300K	250K	1M	500K	300K	250K
0	128	256	426	512	8.0	16	26.7	32
1	8	16	26.7	32	7.5	15	25	30
..	..	..	..	..	..	..	..	..
E	112	224	373	448	1.0	2	3.33	4
F	120	240	400	480	0.5	1	1.67	2

	HLT			
	1M	500K	300K	250K
00	128	256	426	512
01	1	2	3.3	4
02	2	4	6.7	8
..	..	..	..	.
7F	126	252	420	504
7F	127	254	423	508

The choice of DMA or NON-DMA operations is made by the ND bit. When this bit is "1", the NON-DMA mode is selected, and when ND is "0", the DMA mode is selected. In DMA mode, data transfers are signalled by the DRQ pin. Non-DMA mode uses the RQM bit and the INT pin to signal data transfers.

5.2.7 CONFIGURE

Issued to select the special features of the 82077AA. A CONFIGURE command need not be issued if the default values of the 82077AA meet the system requirements.

CONFIGURE DEFAULT VALUES:

EIS —No Implied Seeks

EFIFO —FIFO Disabled

POLL —Polling Enabled

FIFOTHR —FIFO Threshold Set to 1 Byte

PRETRK —Pre-Compensation Set to Track 0

EIS—Enable implied seek. When set to "1", the 82077AA will perform a SEEK operation before executing a read or write command. Defaults to no implied seek.

EFIFO—A "1" puts the FIFO into the 8272A compatible mode where the FIFO is disabled. This means data transfers are asked for on a byte by byte basis. Defaults to "1", FIFO disabled. The threshold defaults to one.

POLL—Disable polling of the drives. Defaults to "0", polling enabled. When enabled, a single interrupt is generated after a RESET. No polling is performed while the drive head is loaded and the head unload delay has not expired.

FIFOTHR—The FIFO threshold in the execution phase of read or write commands. This is programmable from 1 to 16 bytes. Defaults to one byte. A "00" selects one byte "0F" selects 16 bytes.

PRETRK—Pre-compensation start track number. Programmable from track 0 to 255. Defaults to track 0. A "00" selects track 0, "FF" selects 255.

5.2.8 VERSION

The VERSION command checks to see if the controller is an enhanced type or the older type (8272A/765A). A value of 90 H is returned as the result byte, defining an enhanced FDD controller is in use. No interrupts are generated.

5.2.9 RELATIVE SEEK

The command is coded the same as for SEEK, except for the MSB of the first byte and the DIR bit.

DIR Head Step Direction Control.

DIR	Action
0	Step Head Out
1	Step Head In

RCN Relative Cylinder Number that determines how many tracks to step the head in or out from the current track number.

The RELATIVE SEEK command differs from the SEEK command in that it steps the head the absolute number of tracks specified in the command instead of making a comparison against an internal register. The SEEK command is good for drives that support a maximum of 256 tracks. RELATIVE SEEKs cannot be overlapped with other RELATIVE SEEKs. Only one RELATIVE SEEK can be active at a time. Bit 4 of Status Register 0 (EC) will be set if RELATIVE SEEK attempts to step outward beyond Track 0.

As an example, assume that a floppy drive has 300 useable tracks and that the host needs to read track 300 and the head is on any track (0–255). If a SEEK command was issued, the head would stop at track 255. If a RELATIVE SEEK command was issued, the 82077AA would move the head the specified number of tracks, regardless of the internal cylinder position register (but would increment the register). If the head had been on track 40 (D), the maximum track that the 82077AA could position the head on using RELATIVE SEEK, would be 296 (D), the initial track, +256 (D). The maximum count that the head can be moved with a single RELATIVE SEEK command is 256 (D).

The internal register, PCN, would overflow as the cylinder number crossed track 255 and would contain 40 (D). The resulting PCN value is thus (NCN + PCN) mod 256. Functionally, the 82077AA starts counting from 0 again as the track number goes above 255(D). It is the users responsibility to compensate 82077AA functions (precompensation track number) when accessing tracks greater than 255. The 82077AA does not keep track that it is working in an "extended track area" (greater than 255). Any command issued would use the current PCN value except for the RECALIBRATE command which only looks for the TRACK0 signal. RECALIBRATE would return an error if the head was farther than 79 due to its limitation of issuing a maximum 80 step pulses. The user simply needs to issue a second RECALIBRATE command. The SEEK command and implied seeks will function correctly within the 44 (D) track (299–255) area of the "extended track area". It is the users responsibility not to issue a new track position that would exceed the maximum track that is present in the extended area.

To return to the standard floppy range (0–255) of tracks, a RELATIVE SEEK would be issued to cross the track 255 boundary.

A RELATIVE SEEK can be used instead of the normal SEEK but the host is required to calculate the difference between the current head location and the new (target) head location. This may require the host to issue a READ ID command to ensure that the head is physically on the track that software assumes it to be. Different 82077AA commands will return different cylinder results which may be difficult to keep track of with software without the READ ID command.

5.2.10 DUMPREG

The DUMPREG command is designed to support system run-time diagnostics and application software development and debug.

5.2.11 PERPENDICULAR MODE COMMAND

The PERPENDICULAR MODE command should be issued prior to executing READ/WRITE/FORMAT commands that access a disk drive with perpendicular recording capability. With this command, the length of the Gap2 field and VCO enable timing can be altered to accommodate the unique requirements of these drives. Table 5-11 describes the effects of the WGATE and GAP bits for the PERPENDICULAR MODE command. Upon a reset, the 82077AA will default to the conventional mode (WGATE = 0, GAP = 0).

Table 5-11 Effects of WGATE and GAP Bits

GAP	WGATE	MODE	VCO Low Time after Index Pulse	Length of Gap2 Format Field	Portion of Gap2 Written by Write Data Operation	Gap2 VCO Low Time for Read Operations
0	0	Conventional Mode	33 Bytes	22 Bytes	0 Bytes	24 Bytes
0	1	Perpendicular Mode (500 Kbps Data Rate)	33 Bytes	22 Bytes	19 Bytes	24 Bytes
1	0	Reserved (Conventional)	33 Bytes	22 Bytes	0 Bytes	24 Bytes
1	1	Perpendicular Mode (1 Mbps Data Rate)	18 Bytes	41 Bytes	38 Bytes	43 Bytes

Selection of the 500 Kbps and 1 Mbps perpendicular modes is independent of the actual data rate selected in the Data rate Select Register. The user must ensure that the two data rates remain consistent.

The Gap2 and VCO timing requirements for perpendicular recording type drives are dictated by the design of the read/write head. In the design of this head, a pre-erase head precedes the normal read/write head by a distance of 200 micrometers. This works out to about 38 bytes at a 1 Mbps recording density. Whenever the write head is enabled by the Write Gate signal the pre-erase head is also activated at the same time. Thus, when the write head is initially turned on, flux transitions recorded on the media for the first 38 bytes will not be preconditioned with the pre-erase head since it has not yet been activated. To accommodate this head activation and deactivation time, the Gap2 field is expanded to a length of 41 bytes. The format field shown in Figure 5-3 illustrates the change in the Gap2 field size for the perpendicular format.

On the read back by the 82077AA, the controller must begin synchronization at the beginning of the Sync field. For the conventional mode, the internal PLL VCO is enabled (VCOEN) approximately 24 bytes from the start of the Gap2 field. But when the controller operates in the 1 Mbps perpendicular mode (WGATE = 1, GAP = 1), VCOEN goes active after 43 bytes to accommodate the increased Gap2 field size. For both cases, an approximate 2 byte cushion is maintained from the beginning of the sync field for the purposes of avoiding write splices in the presence of motor speed variation.

For the WRITE DATA case, the 82077AA activates Write Gate at the beginning of the sync field under the conventional mode. The controller then writes a new sync field, data address mark, data field, and CRC as shown in Figure 5-1. With the pre-erase head of the perpendicular drive, the write head must be activated in the Gap2 field to insure a proper write of the new sync field. For the 1 Mbps perpendicular mode (WGATE = 1, GAP = 1), 38 bytes will be written in the Gap2 space. Since the bit density is proportional to the data rate, 19 bytes will be written in the Gap2 field for the 500 Kbps perpendicular mode (WGATE = 1, GAP = 0).

It should be noted that none of the alterations in Gap2 size, VCO timing, or Write Gate timing affect normal program flow. The information provided here is just for background purposes and is not needed for normal operation. Once the PERPENDICULAR MODE command is invoked, 82077AA software behavior from the user standpoint is unchanged.

5.3 Command Set Enhancements

The PERPENDICULAR MODE and DUMPREG commands were enhanced along with the addition of a new LOCK command. These enhancements are explained in this section of the data sheet. The commands were enhanced/added in order to provide protection against older software application package which could inadvertently cause system compatibility problems. The modifications/addition are fully backward compatible with the older 82077AAs which do not support the enhancements. All 82077AAs will support these enhancements as of Q1/1991. For more information regarding which 82077AA do or do not support the enhancements please contact your local Intel Sales office.

5.3.1 PERPENDICULAR Mode

The PERPENDICULAR MODE Command is enhanced to allow the system designers to designate specific drives as Perpendicular recording drives. This enhancement is made so that the system designer does not have to worry about older application software packages which bypass their system's FDC (Floppy Disk Controller) routines. The enhancement will also allow data transfers between Conventional and Perpendicular drives without having to issue PERPENDICULAR MODE commands between the accesses of the two different drives, nor having to change write pre-compensation values. The following is an explanation of how this enhancement is implemented:

Old PERPENDICULAR MODE command:

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
PERPENDICULAR MODE										
Command	W	0	0	0	1	0	0	1	0	Command
	W	0	0	0	0	0	0	GAP	WGATE	Code

NOTE:
For the definition of GAP and WGATE bits see Table 5-11 and Section 5.2.11 of the data sheet.
For the Enhanced PERPENDICULAR MODE command definition see Table 5-1.

With the old implementation, the user must properly program both the PERPENDICULAR MODE command and write pre-compensation value before accessing either a Conventional or Perpendicular drive. These programmed values apply to all drives (D0–D3) which the 82077AA may access. It should also be noted that any form of RESET "Hardware" or "Software" will configure the PERPENDICULAR MODE command for Conventional mode (GAP and WGATE = "0").

With the enhanced implementation, both the GAP and WGATE bits have the same affects as the old implementation except for when they are both programmed for value of "0" (Conventional mode). For the case when both GAP and WGATE equal "0" the PERPENDICULAR MODE command will have the following effect on the 82077AA: 1) If any of the new bits D0, D1, D2, and D3 are programmed to "1" the corresponding drive will automatically be programmed for Perpendicular mode (ie: GAP2 being written during a write operation, the programmed Data Rate will determine the length of GAP2.), and data will be written with 0 ns write pre-compensation. 2) any of the new bits (D0–D3) that are programmed for "0" the designated drive will be programmed for Conventional Mode and data will be written with the currently programmed write pre-compensation value. 3) Bits D0, D1, D2, and D3 can only be over written when the OW bit is written as a "1". The status of these bits can be determined by interpreting the eight result byte of the enhanced DUMPREG Command (See Section 5.3.3). (Note: if either the GAP or WGATE bit is a "1", then bits D0–D3 are ignored.)

"Software" and "Hardware" RESET will have the following effects on the enhanced PERPENDICULAR MODE command:

1) "Software" RESETs (Reset via DOR or DSR registers) will only clear GAP and WGATE bits to "0", D3, D2, D1, and D0 will retain their previously programmed values.
2) "Hardware" RESETs (Reset via pin 32) will clear all bits (GAP, Wgate, D0, D1, D2, and D3) to "0" (All Drives Conventional Mode).

5.3.2 LOCK

In order to protect a system with long DMA latencies against older application software packages that can disable the 82077AA's FIFO the following LOCK Command has been added to the 82077AA's command set: [Note: This command should only be used by the system's FDC routines, and ISVs (Independent Software Vendors) should refrain from using it. If an ISV's application calls for having the 82077AA FIFO disabled a CONFIGURE Command should be used to toggle the EFIFO (Enable FIFO) bit. ISV can determine the value of the LOCK bit by interpreting the eighth result byte of an DUMPREG Command (See Section 5.3.3).]

The LOCK command defines whether EFIFO, FIFOTHR, and PRETRK parameters of the CONFIGURE command can be RESET by the DOR and DSR registers. When the LOCK bit is set to a "1" all subsequent "software" RESETs by the DOR and DSR registers will not change the previously set parameter values in the CONFIGURE command. When the LOCK bit is set to a "0" "software" RESETs by the DOR or DSR registers will return these parameters to their default values (See Section 5.2.7). All "hardware" Resets by pin 32 will set the LOCK bit to a "0" value, and will return EFIFO, FIFOTHR, and PRETRK to their default values. A Status byte is returned immediately after issuing the command byte. This Status byte reflects the value of the Lock bit set by the command byte. (Note: No interrupts are generated at the end of this command.)

5.3.3 Enhanced DUMPREG Command

To accommodate the new LOCK command and enhanced PERPENDICULAR MODE command the eighth result byte of DUMPREG command has been modified in the following manner:

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
		DUMPREG								
Result	R	Eighth Result Byte — Undefined —								Old Enhanced
	R	LOCK	0	D3	D2	D1	D0	GAP	WGATE	

NOTES:
1. Data bit 7 reflects the status of the new LOCK bit set by the LOCK Command.
2. Data Bits D0–D5 reflect the status for bits D3, D2, D1, D0, GAP and WGATE set by the PERPENDICULAR MODE Command.

6.0 STATUS REGISTER ENCODING

The contents of these registers are available only through a command sequence.

6.1 Status Register 0

Bit No.	Symbol	Name	Description
7, 6	IC	Interrupt Code	00-Normal termination of command. The specified command was properly executed and completed without error. 01-Abnormal termination of command. Command execution was started, but was not successfully completed. 10-Invalid command. The requested command could not be executed. 11-Abnormal termination caused by Polling.
5	SE	Seek End	The 82077AA completed a SEEK or RECALIBRATE command, or a READ or WRITE with implied seek command.
4	EC	Equipment Check	The TRK0 pin failed to become a "1" after: 1. 80 step pulses in the RECALIBRATE command. 2. The RELATIVE SEEK command causes the 82077AA to step outward beyond Track 0.
3	—	—	Unused. This bit is always "0".
2	H	Head Address	The current head address.
1, 0	DS1, 0	Drive Select	The current selected drive.

6.2 Status Register 1

Bit No.	Symbol	Name	Description
7	EN	End of Cylinder	The 82077AA tried to access a sector beyond the final sector of the track (255D). Will be set if TC is not issued after Read or Write Data Command.
6	—	—	Unused. This bit is always "0".
5	DE	Data Error	The 82077AA detected a CRC error in either the ID field or the data field of a sector.
4	OR	Overrun/ Underrun	Becomes set if the 82077AA does not receive CPU or DMA service within the required time interval, resulting in data overrun or underrun.
3	—	—	Unused. This bit is always "0".
2	ND	No Data	Any one of the following: 1. READ DATA, READ DELETED DATA command, the 82077AA did not find the specified sector. 2. READ ID command, the 82077AA cannot read the ID field without an error. 3. READ TRACK command, the 82077AA cannot find the proper sector sequence.
1	NW	Not Writable	WP pin became a "1" while the 82077AA is executing a WRITE DATA, WRITE DELETED DATA, or FORMAT TRACK command.
0	MA	Missing Address Mark	Any one of the following: 1. The 82077AA did not detect an ID address mark at the specified track after encountering the index pulse from the IDX pin twice. 2. The 82077AA cannot detect a data address mark or a deleted data address mark on the specified track.

6.3 Status Register 2

Bit No.	Symbol	Name	Description
7	—	—	Unused. This bit is always "0".
6	CM	Control Mark	Any one of the following: 1. READ DATA command, the 82077AA encounters a deleted data address mark. 2. READ DELETED DATA command, the 82077AA encounters a data address mark.
5	DD	Data Error in Data Field.	The 82077AA detected a CRC error in the data field.
4	WC	Wrong Cylinder	The track address from the sector ID field is different from the track address maintained inside the 82077AA.
3	—	—	Unused. This bit is always "0".
2	—	—	Unused. This bit is always "0".
1	BC	Bad Cylinder	The track address from the sector ID field is different from the track address maintained inside the 82077AA and is equal to FF hex which indicates a bad track with a hard error according to the IBM soft-sectored format.
0	MD	Missing Data Address Mark	The 82077AA cannot detect a data address mark or a deleted data address mark.

6.4 Status Register 3

Bit No.	Symbol	Name	Description
7	—	—	Unused. This bit is always "0".
6	WP	Write Protected	Indicates the status of the WP pin.
5	—	—	Unused. This bit is always "1".
4	T0	TRACK 0	Indicates the status of the TRK0 pin.
3	—	—	Unused. This bit is always "1".
2	HD	Head Address	Indicates the status of the HDSEL pin.
1, 0	DS1, 0	Drive Select	Indicates the status of the DS1, DS0 pins.

7.0 COMPATIBILITY

The 82077AA was designed with software compatibility in mind. It is a fully backwards compatible solution with the older generation 8272A and NEC765A/B disk controllers. The 82077AA also implements on-board registers for compatibility with the Personal System/2s as well as PC/AT and PC/XT floppy disk controller subsystems. Upon a hardware reset of the 82077AA, all registers, functions and enhancements default to a PS/2, PC/AT, or PS/2 Model 30 compatible operating mode depending on how the IDENT and MFM pins are sampled during Hardware Reset.

7.1 Register Set Compatibility

The register set contained within the 82077AA is a culmination of hardware registers based on the architectural growth of the IBM personal computer line. Table 7-1 indicates the registers required for compatibility based on the type of computer.

Table 7-1. 82077AA Register Support

82077AA Register	8272A	82072	PC/XT	PC/AT	PS/2	Mod 30
SRA					X	X
SRB					X	X
DOR			X	X	X	X
MSR	X	X	X	X	X	X
DSR		X				
Data (FIFO)	X	X	X	X	X	X
DIR				X	X	X
CCR		X*		X	X	X

*CCR is emulated by DSR in an 82072 PC/AT design.

7.2 PS/2 vs. AT vs. Model 30 Mode

To maintain compatibility between PS/2, PC/AT, and Model 30 environments the IDENT and MFM pins are provided. The 82077AA is placed into the proper mode of operations upon Hardware RESET with the appropriate settings of the IDENT and MFM pins. The proper settings of the IDENT and MFM pins are described in IDENT's pin description. Differences between the three modes are described in the following sections.

7.2.1 PS/2 MODE

IDENT strapped low causes the polarity of DENSEL to be active low for high (500 Kbps/1 Mbps) data rates (typically used for 3.5" drives). This polalrity of DENSEL assumes INVERT# to be low. A comprehensive description of DENSEL behavior is given in Table 2-6.

The $\overline{\text{DMAGATE}}$ bit in the Digital Output Register (DOR) will not cause the DRQ or INT output signals to tristate. This maintains consistency with the operation of the floppy disk controller subsystem in the PS/2 architecture.

TC is an active low input signal that is internally qualified by $\overline{\text{DACK}}$ being active low.

7.2.2 PC/AT MODE

IDENT strapped high causes the polarity of DENSEL to be active high for high (500 Kbps/1 Mbps) data rates (typically used for 5.25" drives). This polarity of DENSEL assumes INVERT# to be low. A comprehensive description of DENSEL behavior is given in Table 2-6.

If the $\overline{\text{DMAGATE}}$ bit is written to a "0" in the Digital Output Register (DOR), DRQ and INT will tristate. If $\overline{\text{DMAGATE}}$ is written to a "1", then DRQ and INT will be driven appropriately by the 82077AA.

TC is an active high input signal that is internally qualified by DACK# being active low.

7.2.3 MODEL 30 MODE

IDENT strapped low causes the polarity of DENSEL to be active low for high (500 Kbps/1 Mbps) data rates (typically used for 3.5" drives). This polarity of DENSEL assumes INVERT# to be low. A comprehensive description of DENSEL behavior is given in Table 2-6.

$\overline{\text{DMAGATE}}$ and TC function the same as in PC/AT Mode.

7.3 Compatibility with the FIFO

The FIFO of the 82077AA is designed to be transparent to non-FIFO disk controller software developed on the older generation 8272A standard. Operation of the 82077AA FIFO can be broken down into two tiers of compatibility. For first tier compatibility, the FIFO is left in the default disabled condition upon a "Hardware" reset (via pin 32). In this mode the FIFO operates in a byte mode and provides complete compability with non-FIFO based software. For second tier compatibility, the FIFO is enabled via the CONFIGURE command. When the FIFO is enabled, it will temporarily enter a byte mode during the command and result phase of disk controller operation. This allows for compatible operation when interrogating the Main Status Register (MSR) for the purpose of transferring a byte at a time to or from the disk controller. For normal disk controller applications, the system designer can still take advantage of the FIFO for time critical data transfers during the execution phase and not create any conflicts with non-FIFO software during the command or result phase.

In some instances, use of the FIFO in any form has conflicted with certain specialized software. An example of a compatibility conflict using the FIFO is with software that monitors the progress of a data transfer during the execution phase. If the software assumed the disk controller was operating in a single byte mode and counted the number of bytes transferred to or from the disk controller to trigger some time dependent event on the disk media (i.e. head position over a specific data field), the same software will not have an identical time relationship if the FIFO is enabled. This is because the FIFO allows data to be queued up, and then burst trans-

ferred across the host bus. To accommodate software of this type, it is recommended that the FIFO be disabled.

7.4 Drive Polling

The 82077AA supports the polling mode of the older generation 8272A. This mode is enabled upon a reset and can be disabled via the CONFIGURE command. This mode is supported for the sole purpose of providing backwards compatibility with software that expects it's presence.

The intended purpose of drive polling dates back to 8″ drives as a means to monitor any change in status for each disk drive present in the system. Each of the drives is selected for a period of time and its READY signal sampled. After a delay, the next drive is selected. Since the 82077AA does not support READY in this capacity (internally tied true), the polling sequence is only simulated and does not affect the drive select lines (DS0–DS3) when it is active. If enabled, it occurs whenever the 82077AA is waiting for a command or during SEEKs and RECALIBRATEs (but not IMPLIED SEEKs). Each drive is assumed to be not ready after a reset and a "ready" value for each drive is saved in an internal register as the simulated drive is polled. An interrupt will be generated on the first polling loop because of the initial "not ready" status. This interrupt must be followed with a SENSE INTERRUPT STATUS command from the host to clear the interrupt condition for each of the four logical drives.

8.0 PROGRAMMING GUIDELINES

Programming the 82077AA is identical to any other 8272A compatible disk controller with the exception of some additional commands. For the new designer it is useful to provide some guidelines on how to program the 82077AA. A typical disk operation involves more than issuing a command and waiting for the results. The control of the floppy disk drive is a low level operation that requires software intervention at different stages. New commands and features have been added to the 82077AA to reduce the complexity of this software interface.

8.1 Command and Result Phase Handshaking

Before a command or parameter byte can be issued to the 82077AA, the Main Status Register (MSR) must be interrogated for a ready status and proper FIFO direction. A typical floppy controller device driver should contain a subroutine for sending command or parameter bytes. For this discussion, the routine will be called "Send_byte" with the flowchart shown in Figure 8-1.

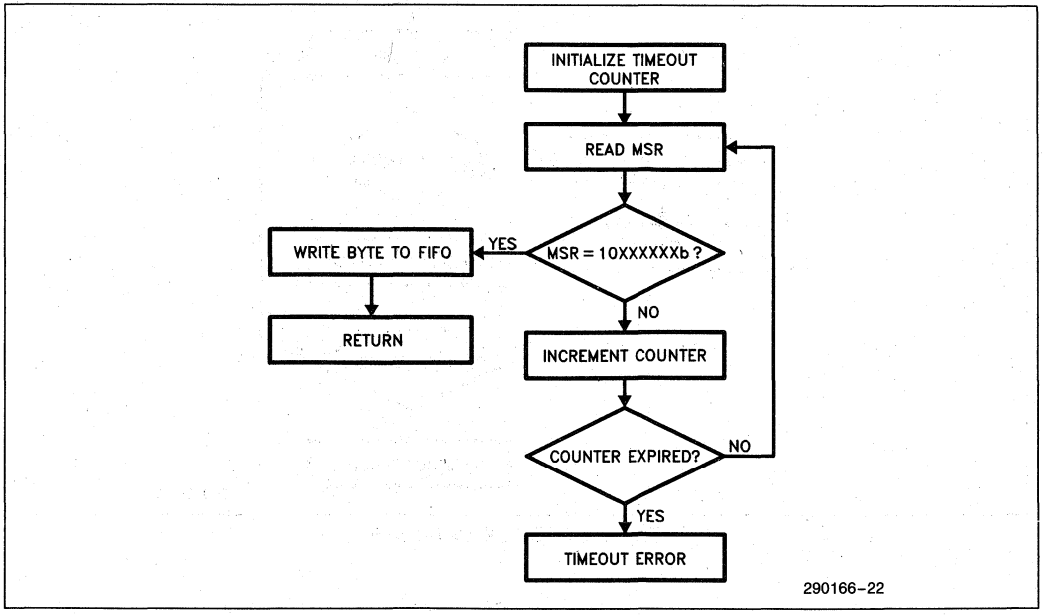


Figure 8-1. Send_Byte Routine

The routine loops until RQM is 1 and DIO is 0 indicating a ready status and FIFO direction is inward. If this condition is true, the 82077AA is ready to accept a command or parameter byte. A timeout counter is used to insure software response within a reasonable amount of time in case of no response by the 82077AA. As a note, the programmer must be careful how the maximum delay is chosen to avoid unnecessary timeouts. For example, if a new command is issued when the 82077AA is in the middle of a polling routine, the MSR will not indicate a ready status for the next parameter byte until the polling sequence completes the loop. This could cause a delay between the first and second bytes of up to 250 μs (@ 250 Kbps). If polling is disabled, this maximum delay is 175 μs. There should also be enough timeout margin to accommodate a shift of the software to a higher speed system. A timeout value that results in satisfactory operation on a 16 MHz CPU might fail when the software is moved to a system with a 25 MHz CPU. A recommended solution is to derive the timeout counter from a system hardware counter that is fixed in frequency from CPU clock to CPU clock.

For reading result bytes from the 82077AA, a similar routine is used. Figure 8-2 illustrates the flowchart for the routine "Get_byte". The MSR is polled until RQM is 1 and DIO is 1, which indicates a ready status and outward FIFO direction. At this point, the host can read a byte from the FIFO. As in the Send_byte routine, a timout counter should be incorporated in case of a disk controller lock-up condition. For example, if a disk was not inserted into the disk drive at the time of a read operation, the controller would fail to receive the index pulse and lock-up since the index pulses are required for termination of the execution phase.

8.2 Initialization

Initializing the 82077AA involves setting up the appropriate configuration after a reset. Parameters set by the SPECIFY command are undefined after a system reset and will need to be reinitialized. CONFIGURE command parameters default to a known state after a system reset but will need to be reinitialized if the system requirements are different from the default settings. The flowchart for the recommended initialization sequence of the 82077AA is shown in Figure 8-3.

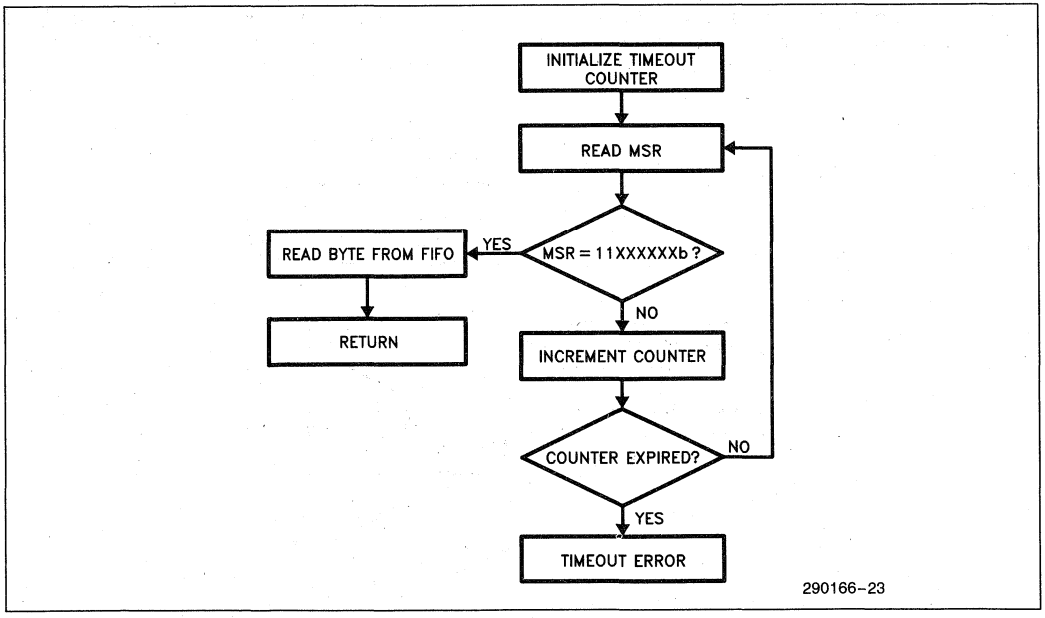


Figure 8-2. Get_Byte Routine

Following a reset of the 82077AA, the Configuration Control Register (CCR) should be reinitialized for the appropriate data rate. An external reset via the RESET pin will cause the data rate and write precompensation values to default to 250 Kbps (10b) and 125 ns (000b) respectively. Since the 125 ns write precompensation value is optimal for the 5¼" and 3½" disk drive environment, most applications will not require the value to be changed in the initialization sequence. As a note, a software reset issued via the DOR or DSR will not affect the data rate or write precompensation values. But it is recommended as a safe programming practice to always program the data rate after a reset, regardless of the type.

Since polling is enabled after a reset of the 82077AA, four SENSE INTERRUPT STATUS commands need to be issued afterwards to clear the status flags for each drive. The flowchart in Figure 8-3 illustrates how the software clears each of the four interrupt status flags internally queued by the 82077AA. It should be noted that although four SENSE INTERRUPT STATUS commands are issued, the INT pin is only active until the first SENSE INTERRUPT STATUS command is executed.

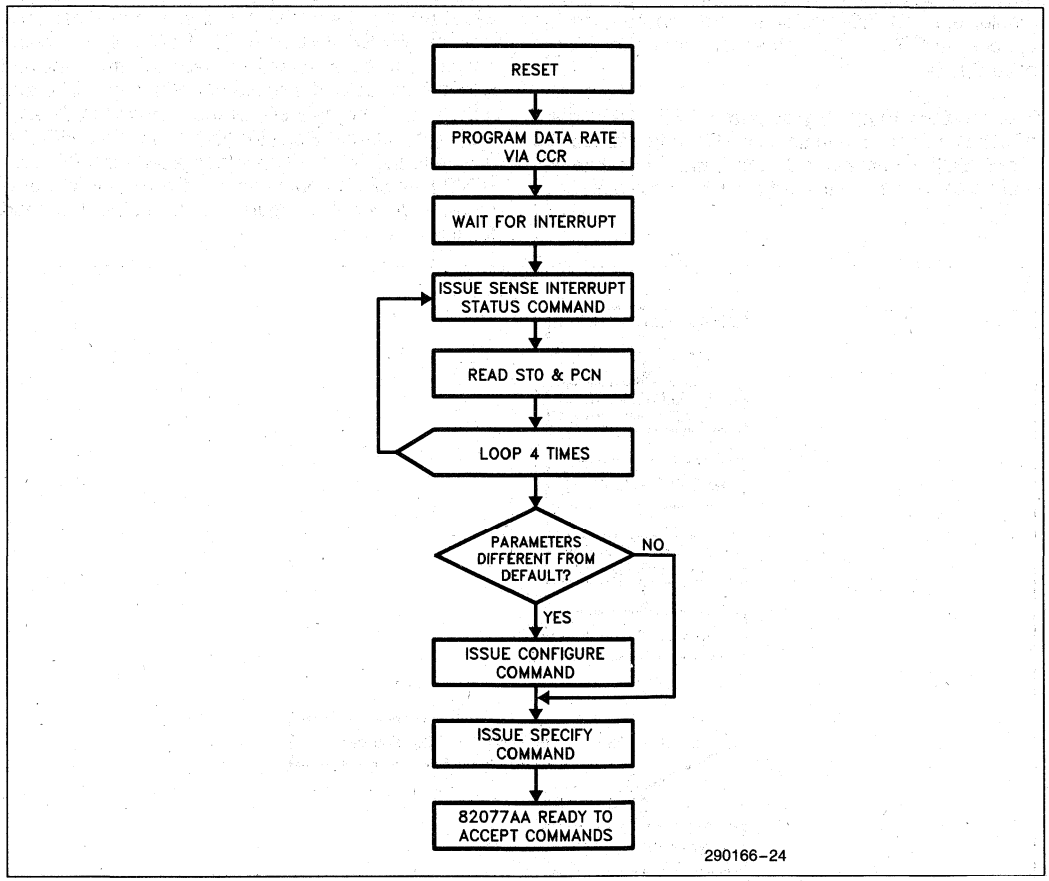


Figure 8-3. Initialization Flowchart

As a note, if the CONFIGURE command is issued within 250 μs of the trailing edge of reset (@ 1 Mbps), the polling mode of the 82077AA can be disabled before the polling initiated interrupt occurs. Since polling stops when the 82077AA enters the command phase, it is only time critical up to the first byte of the CONFIGURE command. If disabled in time, the system software no longer needs to issue the four SENSE INTERRUPT STATUS commands to clear the internal interrupt flags normally caused by polling.

The CONFIGURE command should also be issued if the system requirements are different from the default settings (as described in Section 5.2.7). For example, the CONFIGURE command can be used to enable the FIFO, set the threshold, and enable Implied Seeks.

The non-DMA mode flag, step rate (SRT), head load (HLT), and head unload times (HUT) programmed by the SPECIFY command do not default to a known state after a reset. This behavior is consistent with the 8272A and has been preserved here for compatibility. Thus, it is necessary to always issue a SPECIFY command in the initialization routine.

8.3 Recalibrates and Seeks

Commands that position the disk head are different from the typical READ/WRITE/FORMAT command in the sense that there is no result phase. Once a RECALIBRATE, SEEK, or RELATIVE SEEK command has been issued, the 82077AA will return a ready status in the Main Status Register (MSR) and perform the head positioning operation as a background task. When the seek is complete, the 82077AA will assert the INT signal to request service. A SENSE INTERRUPT STATUS command should then be asserted to clear the interrupt and read the status of the operation. Since the drive and motor enable signals are directly controlled through the Digital Output Register (DOR) on the 82077AA, a write to the DOR will need to precede the RECALIBRATE or SEEK command if the drive and motor is not already enabled. Figure 8-4 shows the flow chart for this operation.

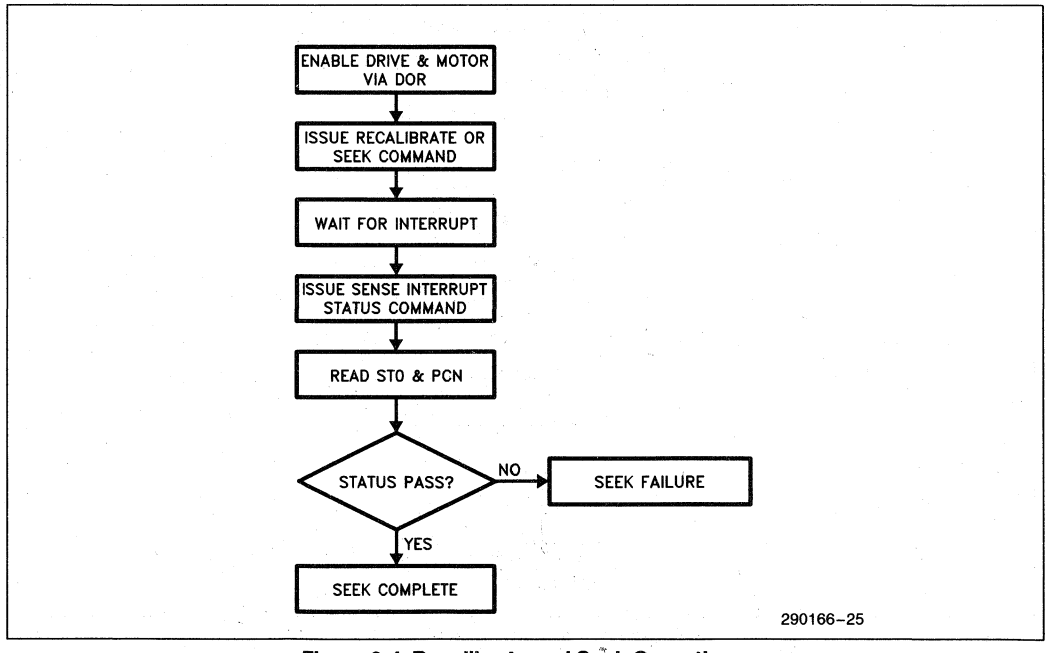


Figure 8-4. Recalibrate and Seek Operations

8.4 Read/Write Data Operations

A read or write data operation requires several steps to complete successfully. The motor needs to be turned on, the head positioned to the correct cylinder, the DMA controller initialized, the read or write command initiated, and an error recovery scheme implemented. The flowchart in Figure 8-5 highlights a recommended algorithm for performing a read or write data operation.

Before data can be transferred to or from the diskette, the disk drive motor must be brought up to speed. For most 3½" disk drives, the spin-up time is 300 ms, while the 5¼" drive usually requires about 500 ms due to the increased moment of inertia associated with the larger diameter diskette.

One technique for minimizing the motor spin-up delay in the read data case is to begin the read operation immediately after the motor is turned on. When the motor is not initially up to speed, the internal data separator will fail to lock onto the incoming data stream and report a failure in the status registers. The read operation is then repeated until successful status is obtained. There is no risk of a data integrity problem since the data field is CRC validated. But, it is not recommended to use this technique for the write data operation even though it requires successful reading of the ID field before the write takes place. The data separator performance of the 82077AA is such that locking to the data stream could take place while the motor speed variation is still significant. This could result in errors when an attempt is made to read the disk media by other disk controllers that have a narrower incoming data stream frequency bandwidth.

After the motor has been turned on, the matching data rate for the media inserted into the disk drive should then be programmed to the 82077AA via the Configuration Control Register (CCR). The 82077AA is designed to allow a different data rate to be programmed arbitrarily without disrupting the integrity of the device. In some applications, it is required to automatically determine the recorded data rate of the inserted media. One technique for doing this is to perform a READ ID operation at each available data rate until a successful status is returned in the result phase.

If implied seeks are not enabled, the disk drive head must be positioned over the correct cylinder by executing a SEEK command. After the seek is complete, a head settling time needs to be asserted before the read or write operation begins. For most drives, this delay should be a minimum of 15 ms. When using implied seeks, the minimum head settling time can be enforced by the head load time (HLT) parameter designated in the SPECIFY command. For example, a HLT value of 8 will yield an effective head settling time of 16 ms for a programmed data rate of 500 Kbps. Of course if the head is already positioned over the correct cylinder, the head settling time does not need to be enforced.

The DMA controller is then initialized for the data transfer and the read or write command is executed. Typically the DMA controller will assert Terminal Count (TC) when the data transfer is complete. The 82077AA will then complete the current data transfer and assert the INT signal signifying it has entered the result phase. The result phase can also be entered by the 82077AA if an error is encountered or the last sector number equals the End of Track (EOT) parameter.

Based on the algorithm in Figure 8-5, if an error is encountered after reading the result bytes, two more retries are performed by reinitializing the DMA controller and re-issuing the read or write data command. A persisting failure could indicate the seek operation did not achieve proper alignment between the head and the track. The disk head should then be recalibrated and the seek repeated for a maximum of two more tries. Unsuccessful operation after this point should be reported as a disk failure to the operating system.

8.5 Formatting

The disk formatting procedure involves positioning the head on each track and creating a fixed format field used for organizing the data fields. The flowchart in Figure 8-6 highlights the typical format procedure.

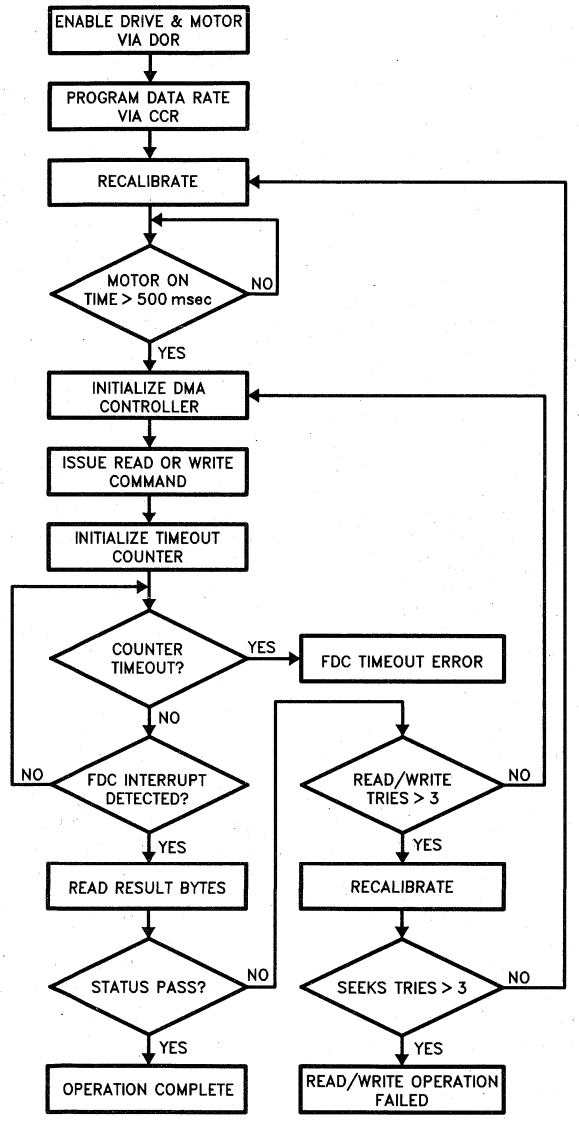


Figure 8-5. Read/Write Operation

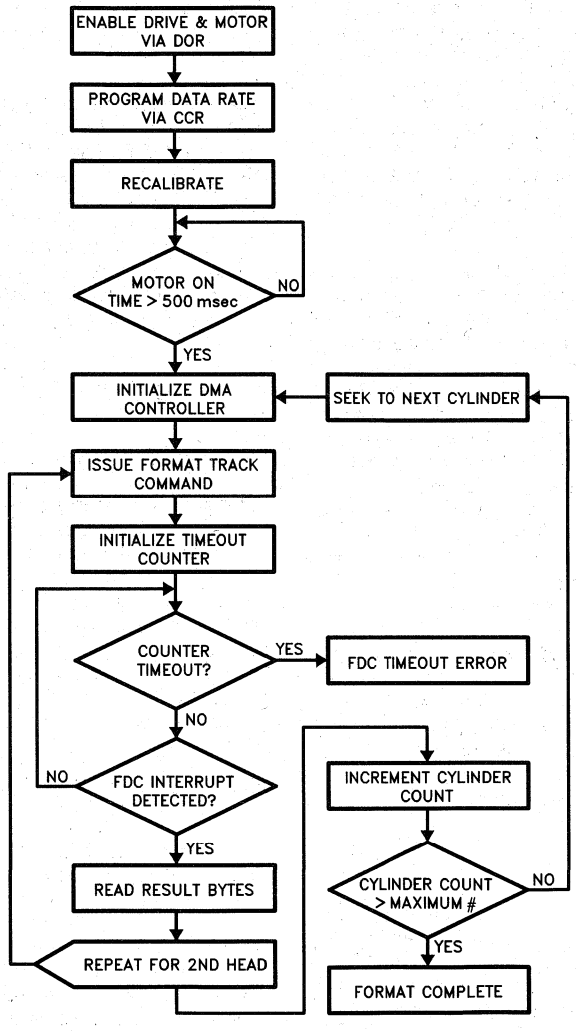


Figure 8-6 Formatting

After the motor has been turned on and the correct data rate programmed, the disk head is recalibrated to track 0. The disk is then allowed to come up to speed via a 500 ms delay. It is important the disk speed has stabilized before the actual formatting to avoid any data rate frequency variations. Since the format fields contain critical information used by the data separator of the disk controller for synchronization purposes, frequency stability of the data stream is imperative for media interchangeability among different systems.

The ID field data created on the disk during the format process is provided by the DMA controller during the execution phase. The DMA controller is initialized to send the C, H, R and N values for each sector ID field. For example, to format cylinder 7, on head 1, with 9 sectors, and a sector size of 2 (512 bytes), the DMA controller should be programmed to transfer 36 bytes (9 sectors x 4 bytes per sector) with the following data field: 7,1,1,2, 7,1,2,2, 7,1,3,2, . . . 7,1,9,2. Since the values provided to the 82077AA during the execution phase of the format command are directly recorded as the ID fields on the disk, the data contents can be arbitrary. Some forms of copy protection have been implemented by taking advantage of this capability.

After each head for a cylinder has been formatted, a seek operation to the next cylinder is performed and the format process is repeated. Since the FORMAT TRACK command does not have implied seek capability, the SEEK command must be used. Also, as discussed in Section 8-2, the head settling time needs to be adhered to after each seek operation.

8.6 Verifies

In some applications, the sector data needs to be verified immediately after each write operation. The verify technique historically used with the 8272A or 82072 disk controller involved reinitializing the DMA controller to perform a read transfer or verify transfer (DACK# is asserted but not RD#) immediately after each write operation. A read command is then to be issued to the disk controller and the resulting status indicates if the CRC validated the previously written data. This technique has the drawback of requiring additional software intervention by having to reprogram the DMA controller between each sector write operation. The 82077AA supports this older verify technique but also provides a new VERIFY command that does not require the use of the DMA controller.

To verify a write data transfer or format track operation using the VERIFY command, the software simply issues the command with the same format as a READ DATA command but without the support of the DMA controller. The 82077AA will then perform a disk read operation without a host data transfer. The CRC will be calculated for each sector read and compared against the value stored on the disk. When the VERIFY command is complete, the status register will report any detected CRC errors.

9.0 DESIGN APPLICATIONS

9.1 PC/AT Floppy Disk Controller

This section presents a design application of a PC/AT compatible floppy disk controller. With an 82077AA, a 24 MHz crystal, a resistor package, and a device chip select, a complete floppy disk controller can be built. The 82077AA integrates all the necessary building blocks for a reliable and low cost solution. But before we discuss the design application using the 82077AA, it is helpful to describe the architecture of the original IBM PC/AT floppy disk controller design that uses the 8272A.

9.1.1 PC/AT FLOPPY DISK CONTROLLER ARCHITECTURE

The standard IBM PC/AT floppy disk controller using the 8272A requires 34 devices for a complete solution. The block diagram in Figure 9-1 illustrates the complexity of the disk controller. A major portion of this logic involves the design of the data separator. The reliability of the disk controller is primarily dictated by the performance and stability of the data separator. Discrete board level analog phase lock loops generally offer good bit jitter margins but suffer from instability and tuning problems in the manufacturing stage if not carefully designed. While digital data separator designs offer stability and generally a lower chip count, they suffer from poor performance in the recovery of data.

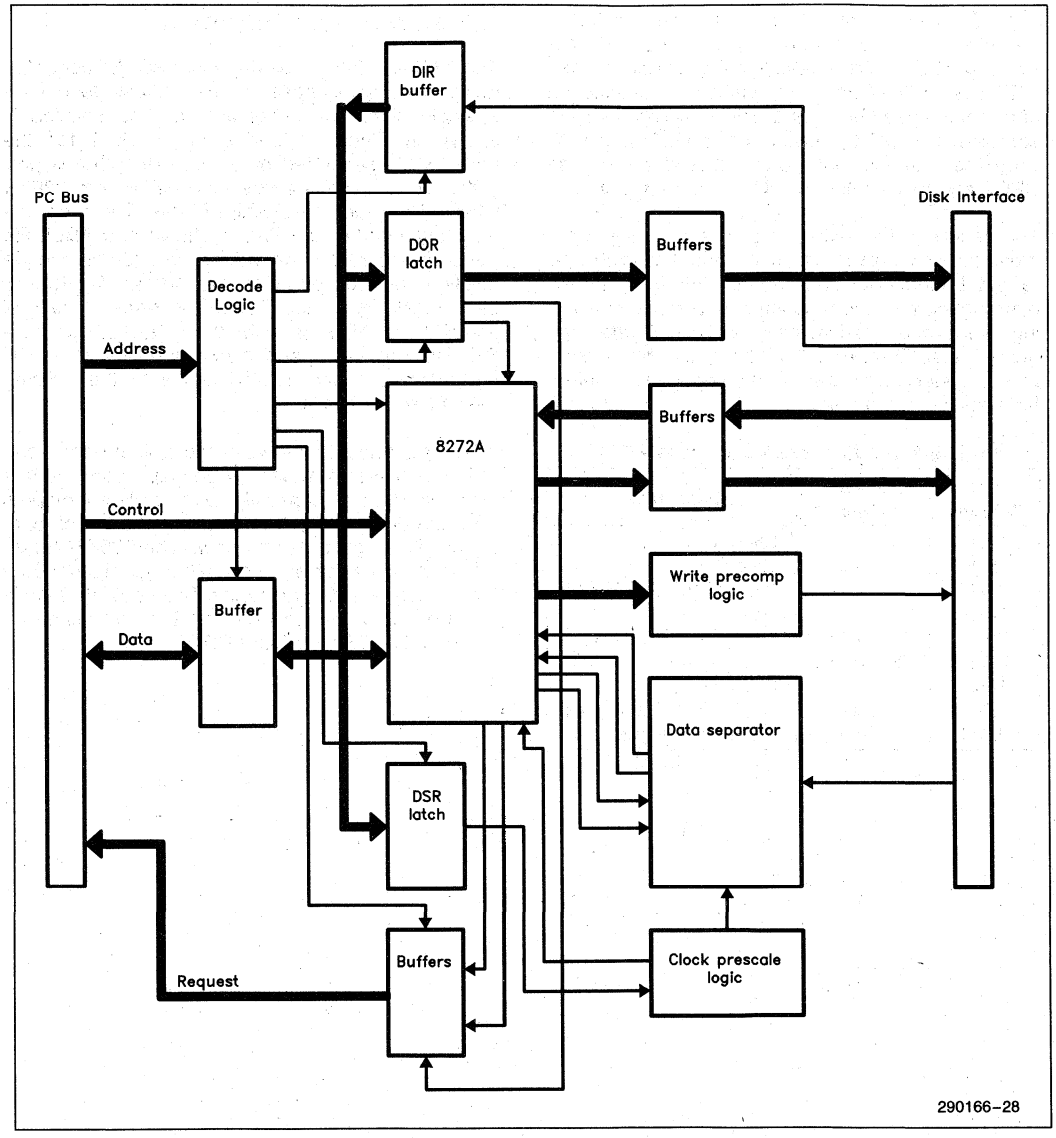


Figure 9-1. Standard IBM PC/AT Floppy Disk Controller

Table 9-1 indicates the drive and media types the IBM PC/AT disk controller can support. This requires the data separator to operate at three different data rates: 250 Kbps, 300 Kbps and 500 Kbps. Clocks to the data separator and disk controller need to be prescaled correspondingly to accommodate each of these data rates. The clock prescaling is controlled by the Data rate Select Register (DSR). Supporting all three data rates can compromise the performance of the phase lock loop (PLL) if steps are not taken in the design to adjust the performance parameters of the PLL with the data rate.

Table 9-1. Standard PC/AT Drives and Media Formats

Capacity	Drive Speed	Data Rate	Sectors	Cylinders
360 Kbyte	300 RPM	250 Kbps	9	40
*360 Kbyte	360 RPM	300 Kbps	9	40
1.2 Mbyte	360 RPM	500 Kbps	15	80

*360 Kbyte diskette in a 1.2 Mbyte drive.

The PC/AT disk controller provides direct control of the drive selects and motors via the Digital Output Register (DOR). As a result, drive selects on the 8272A are not utilized. This places drive selection and motor speed-up control responsibility with the software. The DOR is also used to perform a software reset of the disk controller and tristate the DRQ2 and IRQ6 output signals on the PC bus.

The design of the disk controller also requires address decode logic for the disk controller and register set, buffering for both the disk interface and PC bus, support for write precompensation and monitoring of the disk change signal via a separate read only register (DIR). An I/O address map of the complete register set for the PC/AT floppy disk controller is shown in Table 9-2.

Table 9-2. I/O Address Map for the PC/AT

I/O Address	Access Type	Description
3F0H	—	Unused
3F1H	—	Unused
3F2H	Write	Digital Output Register
3F3H	—	Unused
3F4H	Read	Main Status Register
3F5H	Read/Write	Data Register
3F6H	—	Unused
3F7H	Write	Data Rate Select Register
3F7H	Read	Digital Input Register

9.1.2 82077AA PC/AT SOLUTION

The 82077AA integrates the entire PC/AT controller design with the exception of the address decode on a single chip. The schematic for this solution is shown in Figure 9-2. The chip select for the 82077AA is generated by a 16L8 PAL that is programmed to decode addresses 03F0H thru 03F7H when AEN (Address Enable) is low. The programming equation for the PAL is shown in a ABEL file format in Figure 9-3. An alternative address decode solution could be provided by using a 74LS133 13 input NAND gate and 74LS04 inverter to decode A3–A14 and AEN. Although the PC/AT allows for a 64K I/O address space, decoding down to a 32K I/O address space is sufficient with the existing base of add-in cards.

A direct connection between the disk interface and the 82077AA is provided by on-chip output buffers with a 40 mA sink capability. Open collector outputs from the disk drive are terminated at the disk controller with a 150Ω resistor pack. The 82077AA disk interface inputs contain a schmitt trigger input structure for higher noise immunity. The host interface is a similar direct connection with 12 mA sink capabilities on DB0–DB7, INT and DRQ.

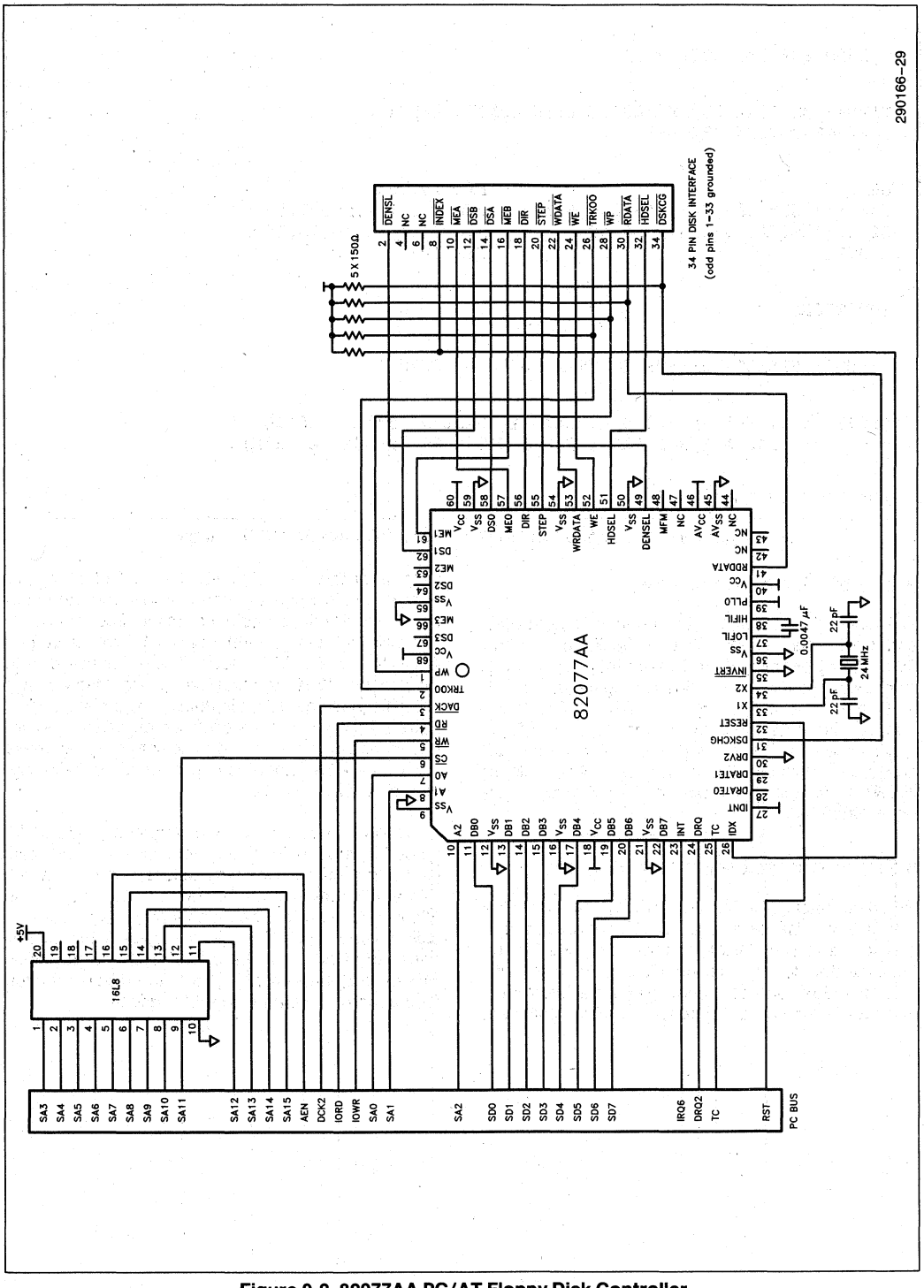


Figure 9-2. 82077AA PC/AT Floppy Disk Controller

```
MODULE PCAT077_LOGIC;

TITLE "82077AA PC/AT FLOPPY DISK CONTROLLER';
PCAT077 DEVICE "P16L8';

GND,VCC                              PIN 10,20;
SA3,SA4,SA5,SA6,SA7,SA8,SA9,SA10     PIN 1,2,3,4,5,6,7,8;
SA11,SA12,SA13,SA14,SA15,AEN         PIN 9,11,13,14,15,16;
CS077_                               PIN 12;

EQUATIONS

"" CHIP SELECT FOR THE 82077AA (3F0H -- 3F7H)

CS077_ = !(!SA15 & !SA14 & !SA13 & !SA12 & !SA11 & !SA10 &
          SA9 & SA8 & SA7 & SA6 & SA5 & SA4 & !SA3 & !AEN);

END PCAT077_LOGIC
```

Figure 9-3. PAL Equation File for a PC/AT Compatible FDC Board

9.2 3.5″ Drive Interfacing

The 82077AA is designed to interface to both 3.5″ and 5.25″ disk drives. This is facilitated by the 82077AA by orienting IDENT to get the proper polarity of DENSEL for the disk drive being used. Typically DENSEL is active high for high (500 Kbps/ 1 Mbps) data rates on 5.25″ drives. And DENSEL is typically active low for high data rates on 3.5″ drives. A complete description of how to orient IDENT to get the proper polarity for DENSEL is given in Table 2-6.

9.2.1 3.5″ DRIVES UNDER THE AT MODE

When interfacing the 82077AA floppy disk controller with a 3.5″ disk drive in a PC/AT application, it is possible that two design changes will need to be implemented for the design discussed in Section 9.1. Most 3.5″ disk drives incorporate a totem pole interface structure as opposed to open collector. Outputs of the disk drive will drive both high or low voltage levels when the drive is selected, and float only when the drive has been deselected. These totem pole outputs generally can only sink or source 4 mA of current. As a result, it is recommended to replace the 150Ω termination resistor pack with a 4.7 KΩ package to pull floating signals inactive. Some other 3.5″ drives do have an open collector interface, but have limited sink capability. In these cases, the drive manufacturer manuals usually suggest a 1 KΩ termination.

A second possible change required under "AT mode" operation involves high capacity 3.5″ disk drives that utilize a density select signal to switch between media recorded at a 250 Kbps and 500 Kbps data rate. The polarity of this signal is typically inverted for 3.5″ drives versus 5.25″ drives. Thus, an inverter can be added between the DENSEL output of the 82077AA and the disk drive interface connector when using 3.5″ drives.

But drives that do not support both data rates or drives with an automatic density detection feature via an optical sensor do not require the use of the DENSEL signal.

Another method is to change the polarity of IDENT with a drive select signal. ORing RESET with the drive select signal (DS0–3) used for the 3.5″ disk drive will produce the proper polarity for DENSEL (assuming INVERT# is low).

9.2.2 3.5″ DRIVES UNDER THE PS/2 MODES

If IDENT is strapped to ground, the DENSEL output signal polarity will reflect a typical 3.5″ drive mode of operation. That is, DENSEL will be high for 250 Kbps or 300 Kbps and low for 500 Kbps or 1 Mbps (assuming INVERT# is low). Thus the only change from the disk interface shown in Figure 9-2 is to replace the 150Ω termination resistor pack with a value of about 10 KΩ. This will prevent excessive current consumption on the CMOS inputs of the 82077AA by pulling them inactive when the drive(s) are deselected.

9.2.3 COMBINING 5.25″ AND 3.5″ DRIVES

If 5.25″ and 3.5″ drives are to be combined in a design, then steps need to be taken to avoid contention problems on the disk interface. Since 3.5″ drives do not have a large sink capability, the 150Ω termination resistor pack required by 5.25″ drives cannot be used with the 3.5″ drive. To accommodate both drives with the same disk controller, the outputs of the 3.5″ drive should be buffered before connecting to the 82077AA disk interface inputs. The 82077AA inputs are then connected to the necessary resistive termination load for the 5.25″ interface.

The block diagram in Figure 9-4 highlights how a combined interface could be designed. In this example, the 5.25″ drive is connected to drive select 0 (DS0) and the 3.5″ drive is connected to drive select 1 (DS1). DS1 is also used to enable a 74LS244 buffer on the output signals of the 3.5″ drive. The drive select logic of the 82077AA is mutually exclusive and prevents the activation of the buffer and 5.25″ drive at the same time. Since the 74LS244 has an I_{OL} of 24 mA, the termination resistor should be increased to 220Ω. This could impact the reliability of the 5.25″ drive interface if the cable lengths are greater than 5 feet.

To accommodate the polarity reversal of the DENSEL signal for 3.5″ drives, it is routed through an inverter for the 3.5″ drive interface. A 1 KΩ pull-up should be placed on the output of the inverter to satisfy the I_{OH} requirements for the 3.5″ drive when using a 74LS04.

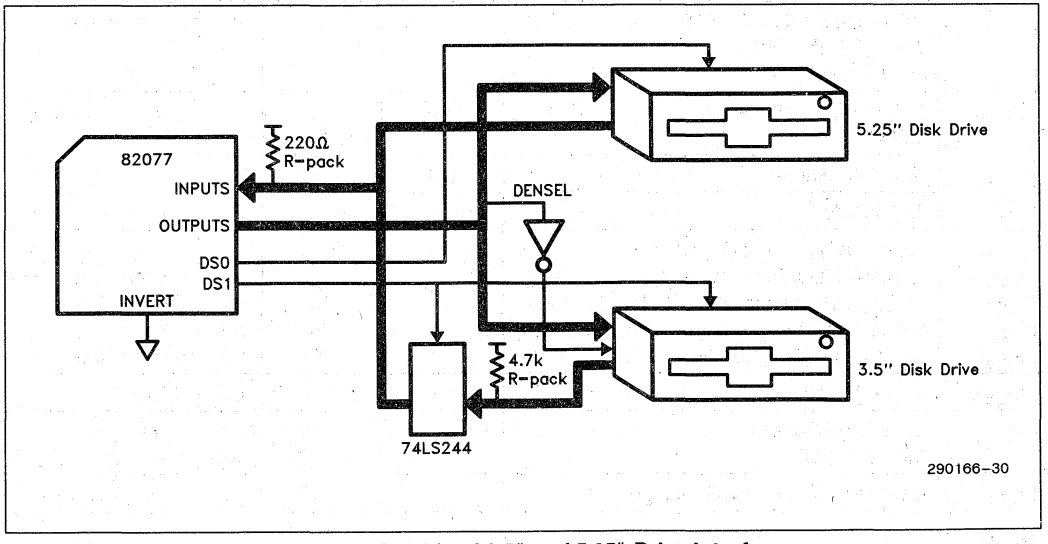


Figure 9-4. Combined 3.5″ and 5.25″ Drive Interface

10.0 D.C. SPECIFICATIONS

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

10.1 Absolute Maximum Ratings

Storage Temperature ..... −65°C to +150°C
Supply Voltage ..... −0.5 to +8.0V
Voltage on Any Input ..... GND − 2V to 6.5V
Voltage on Any Output.. GND − 0.5V to VCC+0.5V
Power Dissipation ..... 1 Watt

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

10.2 D.C. Characteristics

T_A = 0°C to = 70°C, V_{CC} = +5V ±10%, V_{SS} = AV_{SS} = 0V

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{ILC}	Input Low Voltage, X1	−0.5	0.8	V	
V_{IHC}	Input High Voltage, X1	3.9	V_{CC} +0.5	V	
V_{IL}	Input Low Voltage (all pins except X1)	−0.5	0.8	V	
V_{IH}	Input High Voltage (all pins except X1)	2.0	V_{CC} +0.5	V	
V_{OL}	Output Low Voltage MFM		0.4	V	I_{OL} = 2.5 mA
	DRATE0−1		0.4	V	I_{OL} = 6.0 mA
	DB0−7, INT and DRQ		0.4	V	I_{OL} = 12 mA
	ME0−3, DS0−3, DIR, STP WRDATA, WE, HDSEL and DENSEL		0.4	V	I_{OL} = 40 mA
V_{OH}	Output High Voltage MFM	3.0		V	I_{OH} = −2.5 mA
	All Other Outputs	3.0		V	I_{OH} = −4.0 mA
	All Outputs	V_{CC} − 0.4		V	I_{OH} = −100 µA
	V_{CC} Supply Current (Total)				
I_{CC1}	1 Mbps Data Rate, V_{IL} = V_{SS}, V_{IH} = V_{CC}		45	mA	(Notes 1, 2)
I_{CC2}	1 Mbps Data Rate, V_{IL} = 0.45, V_{IH} = 2.4		50	mA	(Notes 1, 2)
I_{CC3}	500 Kbps Data Rate, V_{IL} = V_{SS}, V_{IH} = V_{CC}		35	mA	(Notes 1, 2)
I_{CC4}	500 Kbps Data Rate, V_{IL} = 0.45, V_{IH} = 2.4		40	mA	(Notes 1, 2)
I_{CCSB}	I_{CC} in Powerdown		1.5	mA	(Note 3)
I_{IL}	Input Load Current (all input pins)		10	µA	V_{IN} = V_{CC}
			−10	µA	V_{IN} = 0V
I_{OFL}	Data Bus Output Float Leakage		±10	µA	0.45 < V_{OUT} < V_{CC}

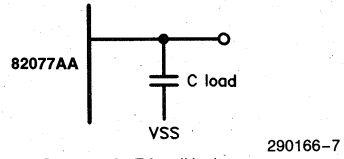
NOTES:
1. The data bus are the only inputs that may be floated.
2. Tested while reading a sync field of "00".
3. V_{IL} = V_{SS}, V_{IH} = V_{CC}; Outputs not connected to D.C. loads.

Capacitance

C_{IN}	Input Capacitance	10	pF	F = 1 MHz, T_A = 25°C
C_{IN1}	Clock Input Capacitance	20	pF	Sampled, not 100% Tested
$C_{I/O}$	Input/Output Capacitance	20	pF	

NOTE:
All pins except pins under test are tied to AC ground.

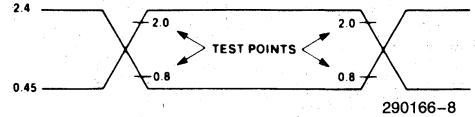
LOAD CIRCUIT



290166-7

C_{load} = 50 pF for all logic outputs,
100 pF for the data bus.

A. C. TESTING INPUT, OUTPUT WAVEFORM



290166-8

11.0 A.C. SPECIFICATIONS

T_A = 0°C to 70°C, V_{CC} = +5V ±10%, V_{SS} = AV_{SS} = 0V

Symbol	Parameter	Min	Max	Unit
CLOCK TIMINGS				
t1	Clock Rise Time		10	ns
	Clock Fall Time		10	ns
t2	Clock High Time(7)	16	26	ns
t3	Clock Low Time(7)	16	26	ns
t4	Clock Period	41.66	41.66	ns
t5	Internal Clock Period(3)			
HOST READ CYCLES				
t7	Address Setup to $\overline{RD}$	5		ns
t8	$\overline{RD}$ Pulse Width	90		ns
t9	Address Hold from RD	0		ns
t10	Data Valid from $\overline{RD}$(12)		80	ns
t11	Command Inactive	60		ns
t12	Output Float Delay		35	ns
t13	INT Delay from RD		t5 + 125	ns
t14	Data Hold from $\overline{RD}$	5		ns

4

A.C. SPECIFICATIONS (Continued)

T_A = 0°C to 70°C, V_{CC} = +5V ±10%, V_{SS} = AV_{SS} = 0V

Symbol	Parameter	Min	Max	Unit
HOST WRITE CYCLES				
t15	Address Setup to $\overline{WR}$	5		ns
t16	$\overline{WR}$ Pulse Width	90		ns
t17	Address Hold from WR	0		ns
t18	Command Inactive	60		ns
t19	Data Setup to WR	70		ns
t20	Data Hold from WR	0		ns
t21	INT Delay from WR		t5 + 125	ns
DMA CYCLES				
t22	DRQ Cycle Period(1)	6.5		µs
t23	DACK to DRQ Inactive		75	ns
t24	RD to DRQ Inactive(4)		100	ns
t25	DACK Setup to $\overline{RD}$, $\overline{WR}$	5		ns
t26	DACK Hold from RD, WR	0		ns
t27	DRQ to $\overline{RD}$, $\overline{WR}$ Active(1)	0	6	µs
t28	Terminal Count Width(10)	50		ns
t29	TC to DRQ Inactive		150	ns
RESET				
t30	"Hardware" Reset Width(5)	170		t4
t30a	"Software" Reset Width(5)	(Note 11)		µs
t31	Reset to Control Inactive		2	µs
WRITE DATA TIMING				
t32	Write Data Width(6)			ns
DRIVE CONTROL				
t35	DIR Setup to STEP(14)	1.0		µs
t36	DIR Hold from STEP	10		µs
t37	STEP Active Time (High)	2.5		µs
t38	STEP Cycle Time(2)			µs
t39	INDEX Pulse Width	5		t5
t41	WE to HDSEL Change	(Note 13)		ms

A.C. SPECIFICATIONS (Continued)

T_A = 0°C to 70°C, V_{CC} = +5V ±10%, V_{SS} = AV_{SS} = 0V

Symbol	Parameter	Min	Max	Unit
READ DATA TIMING				
t40	Read Data Pulse Width	50		ns
f44	PLL Data Rate 82077AA-1, 82077AA(15)		1M	bit/s
	82077AA-5(15)		500K	bit/s
t44	Data Rate Period 1/f44			
tLOCK	Lockup Time		64	t44

NOTES:

1. This timing is for FIFO threshold = 1. When FIFO threshold is N bytes, the value should be multiplied by N and subtract 1.5 µs. The value shown is for 1 Mbps, scales linearly with data rate.
2. This value can range from 0.5 ms to 8.0 ms and is dependent upon data rate and the Specify command value.
3. Many timings are a function of the selected data rate. The nominal values for the internal clock period (t5) for the various data rates are:

1 Mbps	3 x oscillator period = 125 ns
500 Kbps	6 x oscillator period = 250 ns
300 Kbps	10 x oscillator period = 420 ns
250 Kbps	12 x oscillator period = 500 ns

4. If $\overline{DACK}$ transitions before $\overline{RD}$, then this specification is ignored. If there is no transition on $\overline{DACK}$, then this becomes the DRQ inactive delay.
5. Reset requires a stable oscillator to meet the minimum active period.
6. Based on the internal clock period (t_5). For various data rates, the Write Data Width minimum values are:

1 Mbps	5 x oscillator period −50 ns = 150 ns
500 Kbps	10 x oscillator period −50 ns = 360 ns
300 Kbps	16 x oscillator period −50 ns = 615 ns
250 Kbps	19 x oscillator period −50 ns = 740 ns

7. Test points for clock high time are 3.5V. Due to transitional times, clock high time max and clock low time max cannot be met simultaneously. Clock high time min and clock low time max cannot be met simultaneously.
8. Based on internal clock period (t_5).
9. Jitter tolerance is defined as: $\frac{\text{Maximum bit shift from nominal position}}{\text{1/4 period of nominal data rate}} \times 100\%$

It is a measure of the allowable bit jitter that may be present and still be correctly detected. The data separator jitter tolerance is measured under dynamic conditions that jitters the bit stream according to a reverse precompensation algorithm.

10. TC width is defined as the time that both TC and DACK are active.
11. The minimum Reset active period for a Software Reset is dependent on the Data Rate, after the 82077AA has been properly reset using the t30 spec. The minimum Software Reset period then becomes:

1 Mbps	21 x t4 = 0.875 µs
500 Kbps	42 x t4 = 1.75 µs
300 Kbps	70 x t4 = 2.9 µs
250 Kbps	84 x t4 = 3.5 µs

12. Status Register's status bits which are not latched may be updated during a Host read operation.
13. The minimum MFM values for WE to HDSEL change (t41) for the various data rates are:

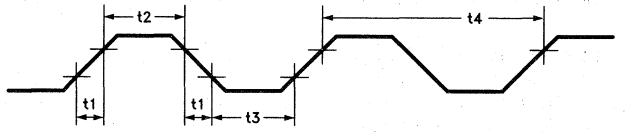
1 Mbps	0.5 ms + [8 x GPL]
500 Kbps	1.0 ms + [16 x GPL]
300 Kbps	1.6 ms + [26.66 x GPL]
250 Kbps	2.0 ms + [32 x GPL]

GPL is the size of gap 3 defined in the sixth byte of a Write Command.

14. This timing is a function of the selected data rate as follows:
    1 Mbps 1.0 μs Min
    500 Kbps 2.0 μs Min
    300 Kbps 3.3 μs Min
    250 Kbps 4.0 μs Min
15.

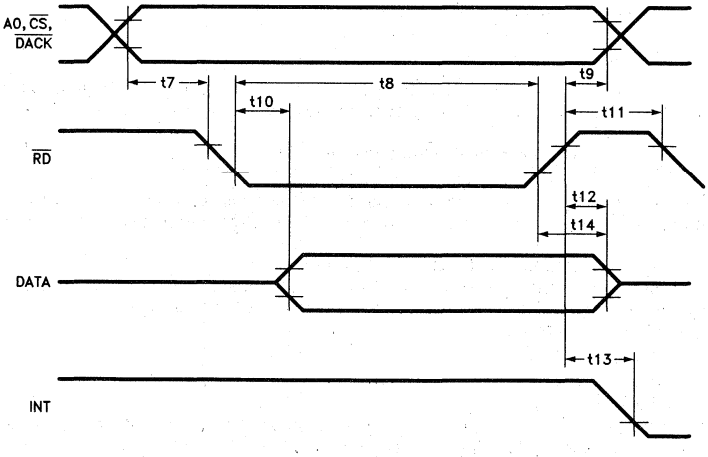
Part Specification	Supported Feature	
	Tape Drive Mode	Perpendicular Drive Support
82077AA-1	X	X
82077AA		X
82077AA-5		

CLOCK TIMINGS



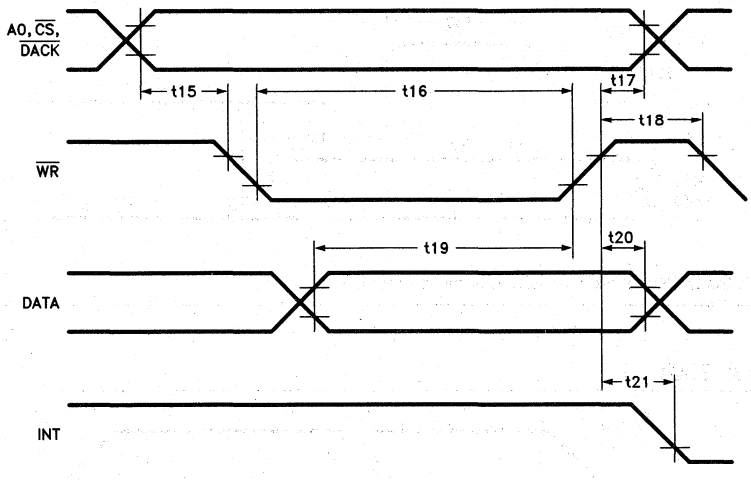
290166-9

HOST READ CYCLES

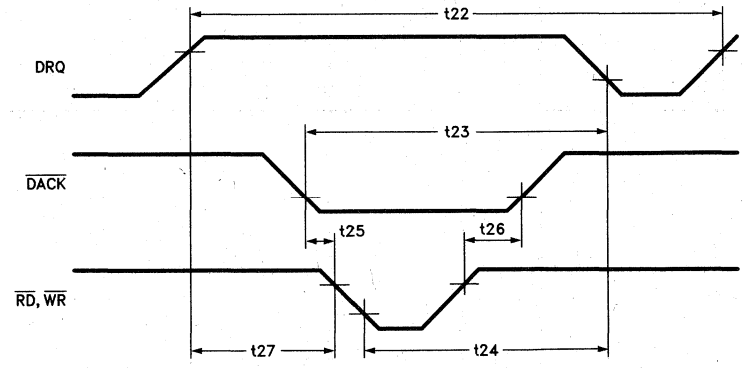


290166-10

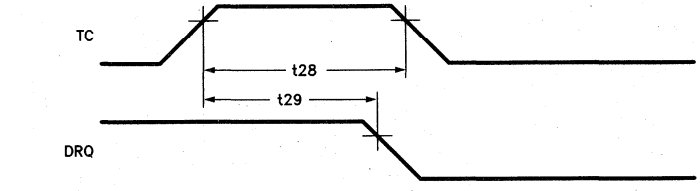
HOST WRITE CYCLES



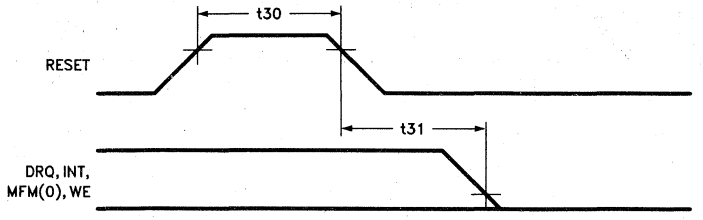
DMA CYCLES



TERMINAL COUNT



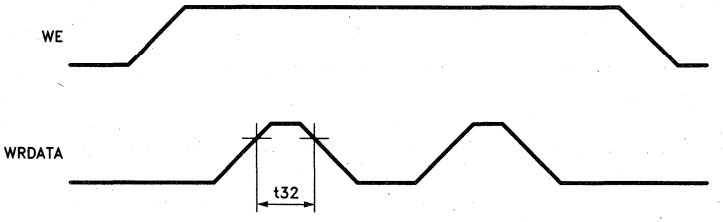
RESET



290166-14

NOTE:
MFM(0) refers to the MFM pin left open during hardware reset.

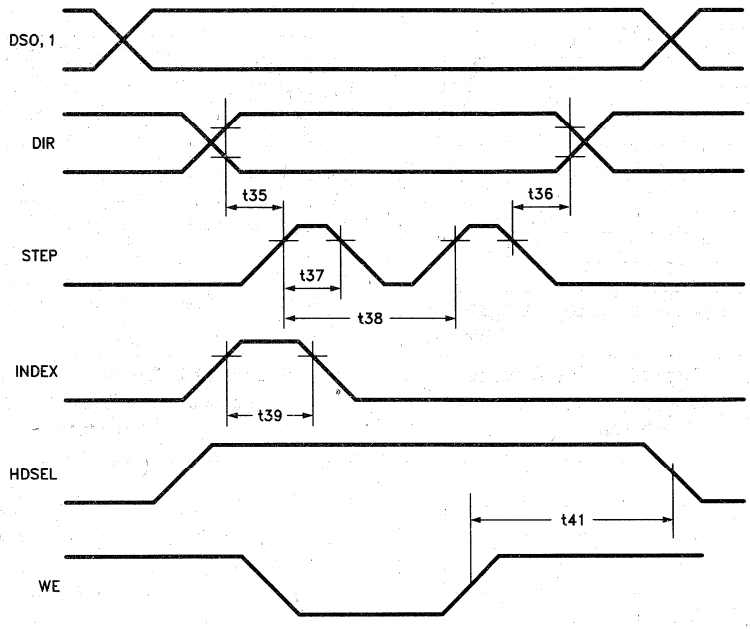
WRITE DATA TIMING



290166-15

NOTE:
Invert high.

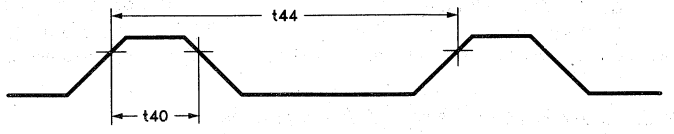
DRIVE CONTROL



290166-16

NOTE:
For overlapped seeks, only one step pulse per drive selection is issued. Non-overlapped seeks will issue all programmed step pulses. Invert high.

INTERNAL PLL



290166-17

NOTE:
Invert high.

12.0 DATA SEPARATOR CHARACTERISTICS FOR FLOPPY DISK MODE

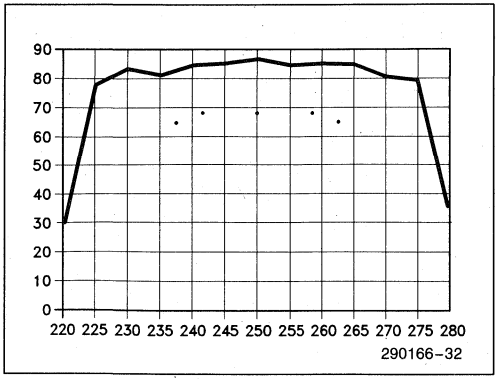


Figure 12-1. Typical Jitter Tolerance vs Data Rate (Capture Range) (250 Kbps)

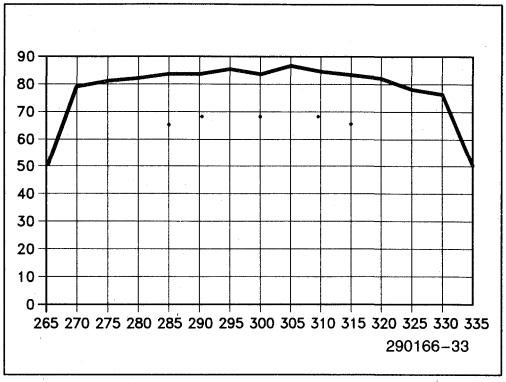


Figure 12-2. Typical Jitter Tolerance vs Data Rate (Capture Range) (300 Kbps)

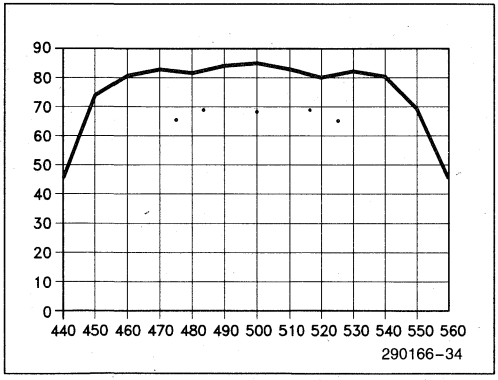


Figure 12-3. Typical Jitter Tolerance vs Data Rate (Capture Range) (500 Kbps)

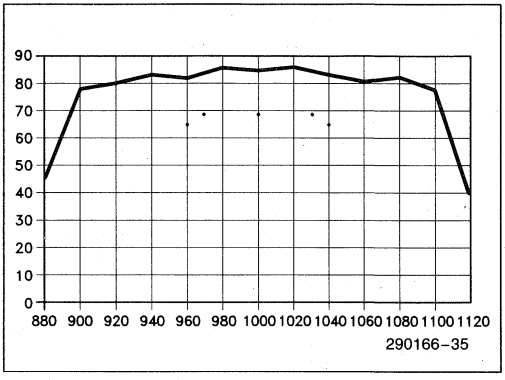


Figure 12-4. Typical Jitter Tolerance vs Data Rate (Capture Range) (1 Mbps), 82077AA-1

Jitter Tolerance measured in percent. See datasheet — Section 3.2.1 capture range expressed as a percent of data rate, i.e., ±3%.

• = Test Points:
250, 300, 500 Kbps are center, ±3% @ 68% jitter, ±5% @ 65% jitter
1 Mbps are center, ±3% @ 68% jitter, ±4% @ 63% jitter

Test points are tested at temperature and V_{CC} limits. Refer to the datasheet. Typical conditions are: room temperature, nominal V_{CC}.

13.0 DATA SEPARATOR CHARACTERISTICS FOR TAPE DRIVE MODE

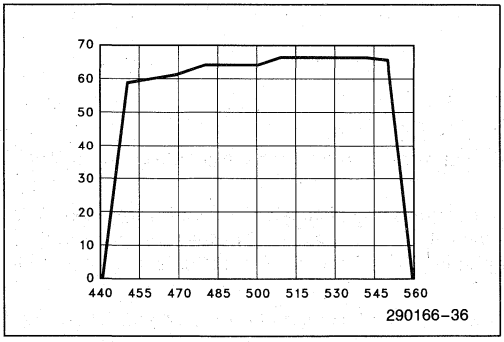


Figure 13-1. Jitter Tolerance vs Data Rate (Capture Range) (±0% ISV, 500 Kbps)

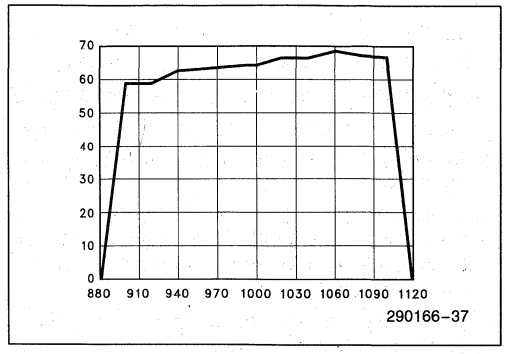


Figure 13-2. Jitter Tolerance vs Data Rate (Capture Range) (±0% ISV, 1 Mbps)

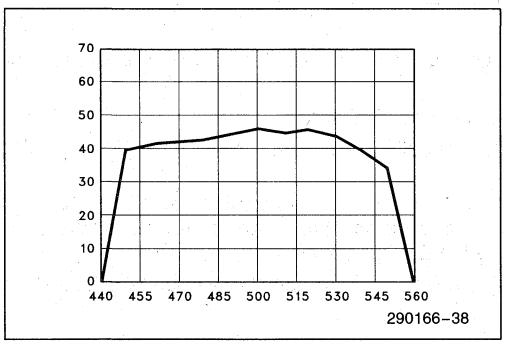


Figure 13-3. Jitter Tolerance vs Data Rate (Capture Range) (±3% ISV, 500 Kbps)

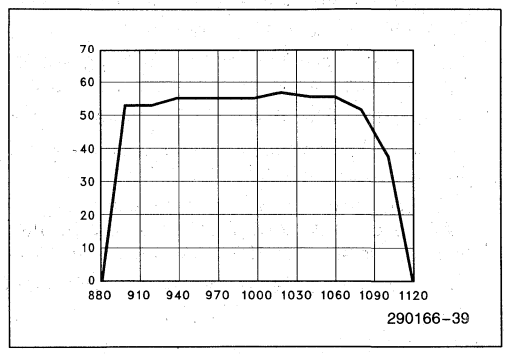


Figure 13-4. Jitter Tolerance vs Data Rate (Capture Range) (±3% ISV, 1 Mbps)

NOTES:
1. Jitter Tolerance measured in percent. See datasheet—Section 3.2.1 capture range expressed as a percent of data rate, i.e., ±5%.
2. Typical conditions are: room temperature, nominal V_{CC}.

14.0 82077AA 68-Lead PLCC Package Thermal Characteristics

T_A Ambient Temp. (°C)	Typical Values				θ_{ja} (°C/W)	θ_{jc} (°C/W)
	T_c (°C)	T_j (°C)	I_{cc} (mA)	V_{cc} (V)		
70	75	75	30	5.0	36	5

NOTES:
Case Temperature Formula:
$T_c = T_a + P\ [\theta_{ja} - \theta_{jc}]$
Junction Temperature Formula:
$T_j = T_c + p\ [\theta_{jc}]$
P = Power dissipated
θ_{jc} = thermal resistance from the junction to the case.
θ_{ja} = thermal resistance from the junction to the ambient.

REVISION HISTORY

DOCUMENT: 82077AA

NEW REVISION NUMBER: 290166-003

PREVIOUS REVISION NUMBER: 290166-002 OCTOBER 1989

1. Figure 1-1; Arrow direction for signals DRQ and $\overline{DACK}$ were corrected.
2. Section 2.1.1b; Bit polarity for bits 4 (TRK0) and 3 ($\overline{HDSEL}$) were corrected for the SRA register in Model 30 Mode.
3. Section 2.3 and 8.6; Because of the possibility of the FIFO not being disabled by a "Software Reset" (ie: New LOCK Command) DMA verifies will be possible with the FIFO enabled. The second paragraph of Section **2.3 DMA Transfers** was deleted and the first paragraph of Section **8.6 Verifies** was corrected to reflect this change.
4. Section 5.0; The explanation for how the 82077AA handles an invalid command was incorrect. A corrected explanation has been added.
5. Table 5-1; The following has been modified to reflect 8207AA enhancements:

   A) The eighth result byte of the DUMPREG command.

   B) Enhancement to the PERPENDICULAR MODE command.

   C) Addition of new LOCK command.
6. Parameter Abbreviations; The following symbol descriptions have been added to reflect 82077AA enhancements: D0, D1, D2, D3, LOCK, and OW.
7. Table 5-11; WGATE and GAP headers were swapped to correct the typo.
8. Section 5-3; This section was added to reflect the following 82077AA enhancements: PERPENDICULAR MODE, LOCK, and DUMPREG commands.
9. A.C. Specifications; t_{35} minimum value was changed from 0.5 μs to 1.0 μs to support new PERPENDICULAR floppy drive requirement, also Note 14 was added to the specification. f_{44} was corrected to reflect the different versions of 82077AAs, also Note 15 was added to this specification.
10. Section 13; Data Separator Characteristics for Tape Drive Mode: PLL graphs were added for tape drive mode.

Hard Disk Controller 5

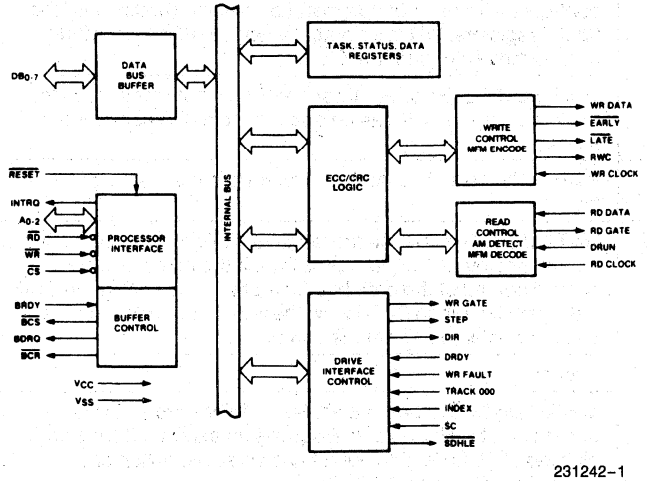
82064
CHMOS WINCHESTER DISK CONTROLLER WITH ON-CHIP ERROR DETECTION AND CORRECTION

- Controls ST506/ST412 Interface Winchester Disk Drives
- 5 Mbit/sec Data Transfer Rate
- Compatible with All Intel and Most Other Microprocessors
- High Speed Operation
  - "Zero Wait State" Operation with 8 MHz 80286 and 10 MHz 80186/188
  - "One Wait State" Operation with 10 MHz 80286
- Eight High-Level Commands: Restore, Seek, Read Sector, Write Sector, Scan ID, Write Format, Compute Correction, Set Parameter
- Low Power CHMOS III
- On-Chip ECC Unit Automatically Corrects Errors
- 5 or 11-Bit Correction—Span Software Selectable
- Implied Seeks with Read/Write Commands
- Multiple Sector Transfer Capability
- 128, 256, 512 and 1024 Byte Sector Lengths
- Available in 40-Lead Plastic Dual In-Line

  (See Packaging Spec., Order #231369)

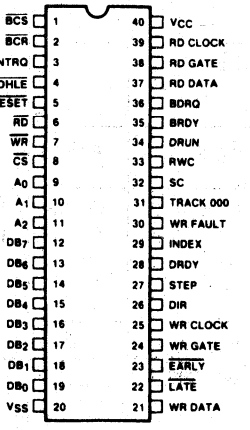
The 82064 Winchester Disk Controller (WDC) with on-chip error detection and correction circuitry interfaces microprocessor systems to 5¼" Winchester disk drives. The 82064 is a CHMOS version of the Western Digital WD2010. It is an upgrade to the Western Digital WD1010A-05 Winchester Disk Controller, and includes on-chip ECC, support for drives with up to 2k tracks, and has an additional control signal which eliminates an external decoder.

The 82064 is fabricated on Intel's advanced CHMOS III technology and is available in 40-lead plastic DIP.



231242-1

Figure 1. 82064 Block Diagram



231242-2

Figure 2. 82064 Pinout

5

August 1990
Order Number: 231242-006

Table 1. Pin Description

Symbol	Pin No.		Type	Name and Function
	DIP	PLCC		
$\overline{BCS}$	1	1	O	**BUFFER CHIP SELECT:** Output used to enable reading or writing of the external sector buffer by the 82064. When low, the host should not be able to drive the 82064 data bus, $\overline{RD}$, or $\overline{WR}$ lines.
$\overline{BCR}$	2	2	O	**BUFFER COUNTER RESET:** Output that is asserted by the 82064 prior to read/write operation. This pin is asserted whenever $\overline{BCS}$ changes state. Used to reset the address counter of the buffer memory.
INTRQ	3	3	O	**INTERRUPT REQUEST:** Interrupt generated by the 82064 upon command termination. It is reset when the STATUS register is read, or a new command is written to the COMMAND register. Optionally signifies when a data transfer is required on Read Sector commands.
$\overline{SDHLE}$	4	4	O	$\overline{SDHLE}$ is asserted when the SDH register is written by the host.
$\overline{RESET}$	5	7	I	**RESET:** Initializes the controller and clears all status flags. Does not clear the Task Register File.
$\overline{RD}$	6	8	I/O	**READ:** Tri-state, bi-directional signal. As an input, $\overline{RD}$ controls the transfer of information from the 82064 registers to the host. $\overline{RD}$ is an output when the 82064 is reading data from the sector buffer ($\overline{BCS}$ low).
$\overline{WR}$	7	9	I/O	**WRITE:** Tri-state, bi-directional signal. As an input, $\overline{WR}$ controls the transfer of command or task information into the 82064 registers. $\overline{WR}$ is an output when the 82064 is writing data to the sector buffer ($\overline{BCS}$ low).
$\overline{CS}$	8	10	I	**CHIP SELECT:** Enables $\overline{RD}$ and $\overline{WR}$ as inputs for access to the Task Registers. It has no effect once a disk command starts.
A_{0-2}	9–11	11–13	I	**ADDRESS:** Used to select a register from the task register file.
DB_{0-7}	12–19	14–16 18–22	I/O	**DATA BUS:** Tri-state, bi-directional 8-bit Data Bus with control determined by BCS. When BCS is high the microprocessor has full control of the data bus for reading and writing the Task Register File. When BCS is low the 82064 controls the data bus to transfer to or from the buffer.
V_{SS}	20	23		Ground
WR DATA	21	24	O	**WRITE DATA:** Output that shifts out MFM data at a rate determined by Write Clock. Requires an external D flip-flop clocked at 10 MHz. The output has an active pullup and pulldown that can sink 4.8 mA.
$\overline{LATE}$	22	25	O	**LATE:** Output used to derive a delay value for write precompensation. Valid when WR GATE is high. Active on all cylinders.
$\overline{EARLY}$	23	26	O	**EARLY:** Output used to derive a delay value for write precompensation. Valid when WR GATE is high. Active on all cylinders.

Table 1. Pin Description (Continued)

Symbol	Pin No.		Type	Name and Function
	DIP	PLCC		
WR GATE	24	27	O	**WRITE GATE:** High when write data is valid. WR GATE goes low if the WR FAULT input is active. This output is used by the drive to enable head write current.
WR CLOCK	25	29	I	**WRITE CLOCK:** Clock input used to derive the write data rate. Frequency = 5 MHz for the ST506 interface.
DIR	26	30	O	**DIRECTION:** High level on this output tells the drive to move the head inward (increasing cylinder number). The state of this signal is determined by the 82064's internal comparison of actual cylinder location vs. desired cylinder.
STEP	27	31	O	**STEP:** This signal is used to move the drive head to another cylinder at a programmable frequency. Pulse width = 1.6 μs for a step rate of 3.2 μs/step, and 8.4 μs for all other step rates.
DRDY	28	32	I	**DRIVE READY:** If DRDY from the drive goes low, the command will be terminated.
INDEX	29	33	I	**INDEX:** Signal from the drive indicating the beginning of a track. It is used by the 82064 during formatting, and for counting retries. Index is edge triggered. Only the rising edge is valid.
WR FAULT	30	34	I	**WRITE FAULT:** An error input to the 82064 which indicates a fault condition at the drive. If WR FAULT from the drive goes high, the command will be terminated.
TRACK 000	31	35	I	**TRACK ZERO:** Signal from the drive which indicates that the head is at the outermost cylinder. Used to verify proper completion of a RESTORE command.
SC	32	36	I	**SEEK COMPLETE:** Signal from the drive indicating to the 82064 that the drive head has settled and that reads or writes can be made. SC is edge triggered. Only the rising edge is valid.
RWC	33	37	O	**REDUCED WRITE CURRENT:** Signal goes high for all cylinder numbers above the value programmed in the Write Precomp Cylinder register. It is used by the precompensation logic and by the drive to reduce the effects of bit shifting.
DRUN	34	38	I	**DATA RUN:** This signal informs the 82064 when a field of all ones or all zeroes has been detected in the read data stream by an external one-shot. This indicates the beginning of an ID field. RD GATE is brought high when DRUN is sampled high for 16 clock periods.
BRDY	35	39	I	**BUFFER READY:** Input used to signal the controller that the buffer is ready for reading (full), or writing (empty), by the host μP. Only the rising edge indicates the condition.

Table 1. Pin Description (Continued)

Symbol	Pin No.		Type	Name and Function
	DIP	PLCC		
BDRQ	36	40	O	**BUFFER DATA REQUEST:** Activated during Read or Write commands when a data transfer between the host and the 82064's sector buffer is required. Typically used as a DMA request line.
RD DATA	37	41	I	**READ DATA:** Single ended input that accepts MFM data from the drive.
RD GATE	38	42	O	**READ GATE:** Output that is asserted when a search for an address mark is initiated. It remains asserted until the end of the ID or data field.
RD CLOCK	39	43	I	**READ CLOCK:** Clock input derived from the external data recovery circuits.
V_{CC}	40	44	I	**D.C. POWER:** +5V.
NC	—	5, 6 17, 28		**No Connects**

FUNCTIONAL DESCRIPTION

The Intel 82064 CHMOS Winchester Disk Controller (WDC) interfaces microprocessor systems to Winchester disk drives that use the Seagate Technology ST506/ST412 interface. The device translates parallel data from the microprocessor to a 5 Mbit/sec, MFM-encoded serial bit stream. It provides all of the drive control logic and control signals which simplify the design of external data separation and write precompensation circuitry. The 82064 is designed to interface to the host processor through an external sector buffer.

On-chip error detection algorithms include the CRC/CCITT and a 32-bit computer generated ECC polynomial. If the ECC code is selected, the 82064 provides three possible error handling techniques if an error is detected during a read operation:

1. Automatically correct the data in the sector buffer, providing the host with good information.
2. Provide the host with the error location and pattern, allowing the host to correct the error.
3. Take no action other than setting the error flag.

The Intel 82064 is an enhanced version of the Western Digital WD2010 Winchester Disk Controller. The 82064 has been completely redesigned for Intel's advanced CHMOS III fabrication process, allowing Intel to offer a high quality, low power device while at the same time maintaining complete compatibility with the WD2010.

Enhancements to the basic design include:

Conversion to a CHMOS III fabrication process for low power consumption.

Improvements to the processor interface to provide high-speed "zero wait state" operation with 10 MHz 80186/188 and 8 MHz 80286. High-speed "one wait state" operation with 10 MHz 80286.

The 82064 is completely socket and software compatible with the WD2010 Winchester Disk Controller. As with the WD2010, the 82064 is also socket and software compatible with existing WD1010A-05 designs that do not include external ECC.

INTERNAL ARCHITECTURE

The internal architecture of the 82064 is shown in more detail in Figure 3. It is made up of seven major blocks as described below.

PLA Controller

The PLA interprets commands and provides all control functions. It is synchronized with WR CLOCK.

Magnitude Comparator

An 11-bit magnitude comparator is used to calculate the direction and number of steps needed to move the heads from the present to the desired cylinder position. It compares the cylinder number in the task file to the internal "present position" cylinder number.

A separate high-speed equivelance comparator is used to compare ID field bytes when searching for a sector ID field.

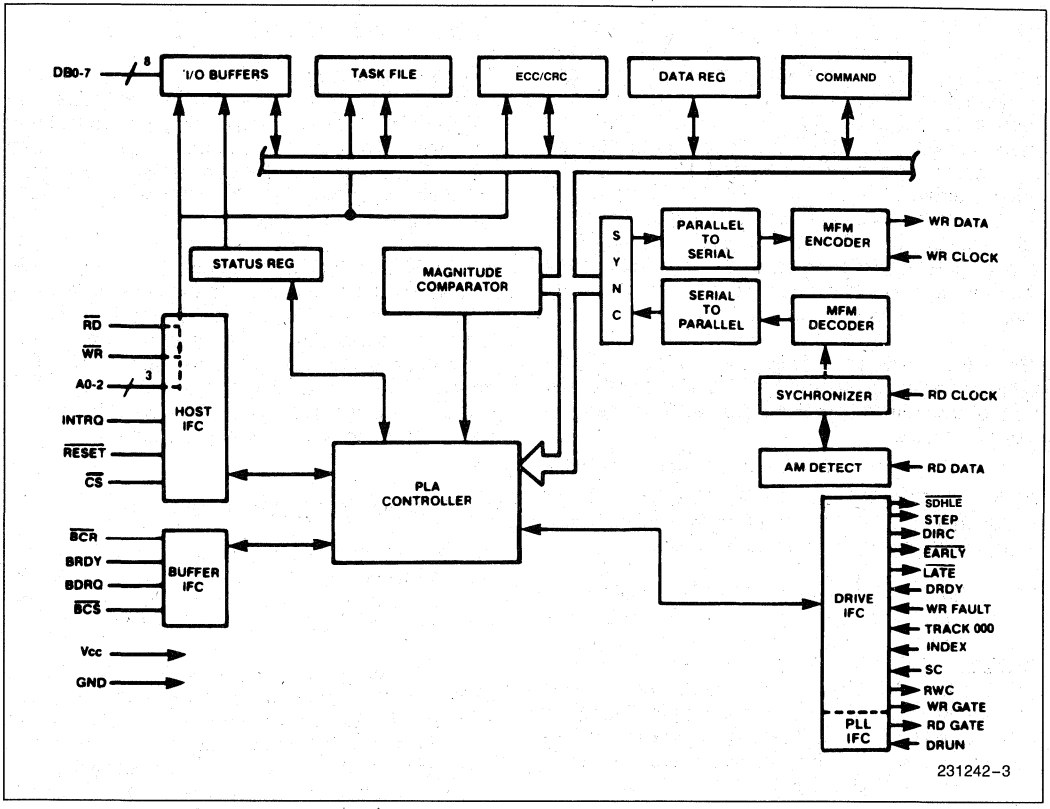


Figure 3. 82064 Detailed Block Diagram

CRC and ECC Generator and Checker

The 82064 provides two options for protecting the integrity of the data field. The data field may have either a CRC (SDH register, bit 7 = 0), or a 32-bit ECC (SDH register, bit 7 = 1) appended to it. The ID field is always protected by a CRC.

The CRC mode provides a means of verifying the accuracy of the data read from the disk, but does not attempt to correct it. The CRC generator computes and checks cyclic redundancy check characters that are written and read from the disk after ID and data fields. The polynomial used is:

$X^{16}+X^{12}+X^{5}+1$

The CRC register is preset to all one's before computation starts.

If the CRC character generated while reading the data does not equal the one previously written an error exists. If an ID field CRC error occurs the "ID not found" bit in the error register will be set. If a data field CRC error occurs the "ECC/CRC" bit in the error register will be set.

The ECC mode is only applicable to the data field. It provides the user with the ability to detect and correct errors in the data field automatically. The commands and registers which must be considered when ECC is used are:

1. SDH Register, bit 7 (CRC/ECC)
2. READ SECTOR Command, bit 0 (T)
3. READ SECTOR and WRITE SECTOR Commands, bit 1 (L)
4. COMPUTE CORRECTION Command
5. SET PARAMETER Command
6. STATUS Register, bit 2 - error correction successful
7. STATUS Register, bit 0 - error occurred
8. ERROR Register, bit 6 - uncorrectable error

To enable the ECC mode, bit 7 of the SDH register must be set to one.

Bit 0 (T) of the READ Command controls whether or not error correction is attempted. When T = 0 and an error is detected, the 82064 tries up to 10 times to correct the error. If the error is successfully corrected, bit 2 of the STATUS Register is set. The host can interrogate the status register and detect that an error occurred and was corrected. If the error was not correctable, bit 6 of the ERROR Register is set. If the correction span was set to 5 bits, the host may now execute the SET PARAMETER Command to change the correction span to 11 bits, and attempt the read again. If the error persists, the host can read the data, but it will contain errors.

When T = 1 and an error is detected, no attempt is made to correct it. Bit 0 of the STATUS Register and bit 6 of the ERROR Register are set. The user now has two choices:

1. Ignore the error and make no attempt to correct it.
2. Use the COMPUTE CORRECTION Command to determine the location and pattern of the error, and correct it within the user's program.

When the COMPUTE CORRECTION Command is implemented, it must be done before executing any command which can alter the contents of the ECC Register. The READ SECTOR, WRITE SECTOR, SCAN ID, and FORMAT Commands will alter this register and correction will be impossible. The COMPUTE CORRECTION Command may determine that the error is uncorrectable, at which point the error bits in the STATUS and ERROR Registers are set.

Although ECC generation starts with the first bit of the F8H byte in the data ID field, the actual ECC bytes written will be the same as if the A1H byte was included. The ECC polynomial used is:

$$X^{32}+X^{28}+X^{26}+X^{19}+X^{17}+X^{10}+X^{6}+X^{2}+1$$

For automatic error correction, the external sector buffer must be implemented with a static RAM and counter, not with a FIFO.

The SET PARAMETER Command is used to select a 5-bit or 11-bit correction span.

When the L Bit (bit 1) of the READ SECTOR and WRITE SECTOR commands is set to one, they are referred to as READ LONG and WRITE LONG commands. For these commands, no CRC or ECC characters are generated or checked by the 82064. In effect, the data field is extended by 4 bytes which are passed to/from the sector buffer.

With proper use of the WRITE SECTOR, READ LONG, WRITE LONG, and READ SECTOR Commands, a diagnostic routine may be developed to test the accuracy of the error correction process.

MFM ENCODER/DECODER

Encodes and decodes MFM data to be written/read from the drive. The MFM encoder operates from WR CLOCK, a clock having a frequency equal to the bit rate. The MFM decoder operates from RD CLOCK, a bit rate clock generated by the external data separator. RD CLOCK and WR CLOCK need not be synchronous.

The MFM encoder also generates the write precompensation control signals. Depending on the bit pattern of the data, $\overline{\text{EARLY}}$ or $\overline{\text{LATE}}$ may be asserted. External circuitry uses these signals to compensate for drift caused by the influence one bit has over another. More information on the use of the $\overline{\text{EARLY}}$ and $\overline{\text{LATE}}$ control signals can be found in the section which describes the drive interface.

Address Mark (AM) Detection

An address mark is comprised of two unique bytes preceeding both the ID field and the data field. The first byte is used for resynchronization. The second byte indicates whether it is an ID field or a data field.

The first byte, A1H, normally has a clock pattern of 0EH; however, one clock pulse has been suppressed, making it 0AH. With this pattern, the AM detector knows it is looking at an address mark. It now examines the next byte to determine if it is an ID or data field. If this byte is 111101XX or 111111XX it is an ID field. Bits 3, 1, and 0 are the high order cylinder number bits. If the second byte is F8H, it is a data field.

Host/Buffer Interface Control

The primary interface between the host processor and the 82064 is an 8-bit bi-directional bus. This bus is used to transmit and receive data for both the 82064 and the sector buffer. The sector buffer consists of a static RAM and counter. Since the 82064 makes the bus active when accessing the sector buffer, a transceiver must be used to isolate the host during this time. Figure 4 illustrates a typical interface with a sector buffer. Whenever the 82064 is not using the sector buffer, the BUFFER CHIP SELECT ($\overline{\text{BCS}}$) is high (disabled). This allows the host access to the 82064's Task Register File and to the sector buffer. A decoder is used to generate $\overline{\text{BCS}}$ when A_{0-2} is '000', an unused address in the 82064. A binary counter is enabled whenever $\overline{\text{RD}}$ or $\overline{\text{WR}}$ go active. The location within the sector buffer which is addressed by the counter will be accessed. The counter will be incremented by the trailing edge of the $\overline{\text{RD}}$ or $\overline{\text{WR}}$. This allows the host to access se-

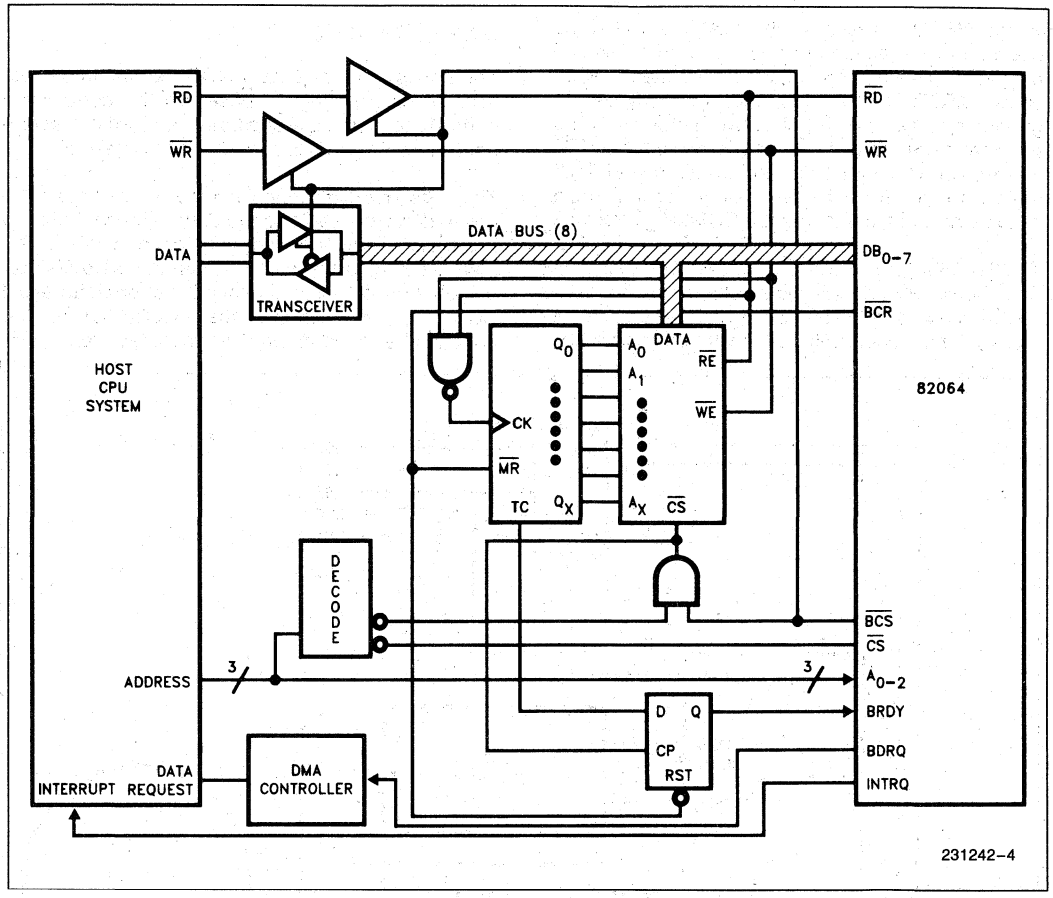


Figure 4. Host Interface Block Diagram

quential bytes within the sector buffer. The decoder also generates a $\overline{CS}$ for the 82064 whenever A_{0-2} does not equal '000', allowing access to the 82064's internal Task Register File while keeping the sector buffer tri-stated.

During a WRITE SECTOR Command, the host processor sets up data in the Task Register File and then issues the command. The 82064 asserts BUFFER COUNTER RESET ($\overline{BCR}$) to reset the counter. It then generates a status to inform the host that it can load the sector buffer with data to be written. When the counter reaches its maximum count, the BUFFER READY (BRDY) signal is asserted by the carry out of the counter, informing the 82064 that the sector buffer is full. (BRDY is a rising edge triggered signal which will be ignored if asserted before the 82064 asserts $\overline{BCR}$.) $\overline{BCS}$ is then asserted, disconnecting the host through the transceivers, and the $\overline{RD}$ and $\overline{WR}$ lines become outputs from the 82064 to allow access to the sector buffer. When the 82064 is done using the buffer, it deasserts $\overline{BCS}$ which again allows the host to access the local bus. The READ SECTOR command operates in a similar manner, except the buffer is loaded by the 82064 instead of the host.

Another control signal, BUFFER DATA REQUEST (BDRQ), can be used with a DMA controller to indicate that the 82064 is ready to send or receive data. When data transfer is via a programmed I/O environment, it is the responsibility of the host to interrogate the DRQ status bit to determine if the 82064 is ready (bit 3 of the status register). For further explanation, refer to the individual command descriptions and the A.C. Characteristics.

When INTRQ is asserted, the host is signaled that execution of a command has terminated (either a normal termination or an aborted command). For the READ SECTOR command, interrupts may be programmed to be asserted either at the termination of the command, or when BDRQ is asserted. INTRQ will remain active until the host reads the STATUS register to determine the cause of the termination, or writes a new command into the COMMAND register.

The 82064 asserts $\overline{\text{SDHLE}}$ whenever the SDH register is being written. This signal can be used to latch the drive and head select information in an external register for decoding. Figure 5 illustrates one method.

Drive Interface

The drive side of the 82064 WDC requires three sections of external logic. These are the control line buffer/receivers, data separator, and write precompensation. Figure 5 illustrates a drive interface.

The buffer/receivers condition the control lines to be driven down the cable to the drive. The control lines are typically single-ended, resistor terminated, TTL levels. The data lines to and from the drive also require buffering. This is typically done with differential RS-422 drivers. The interface specification for the drive will be found in the drive manufacturer's OEM

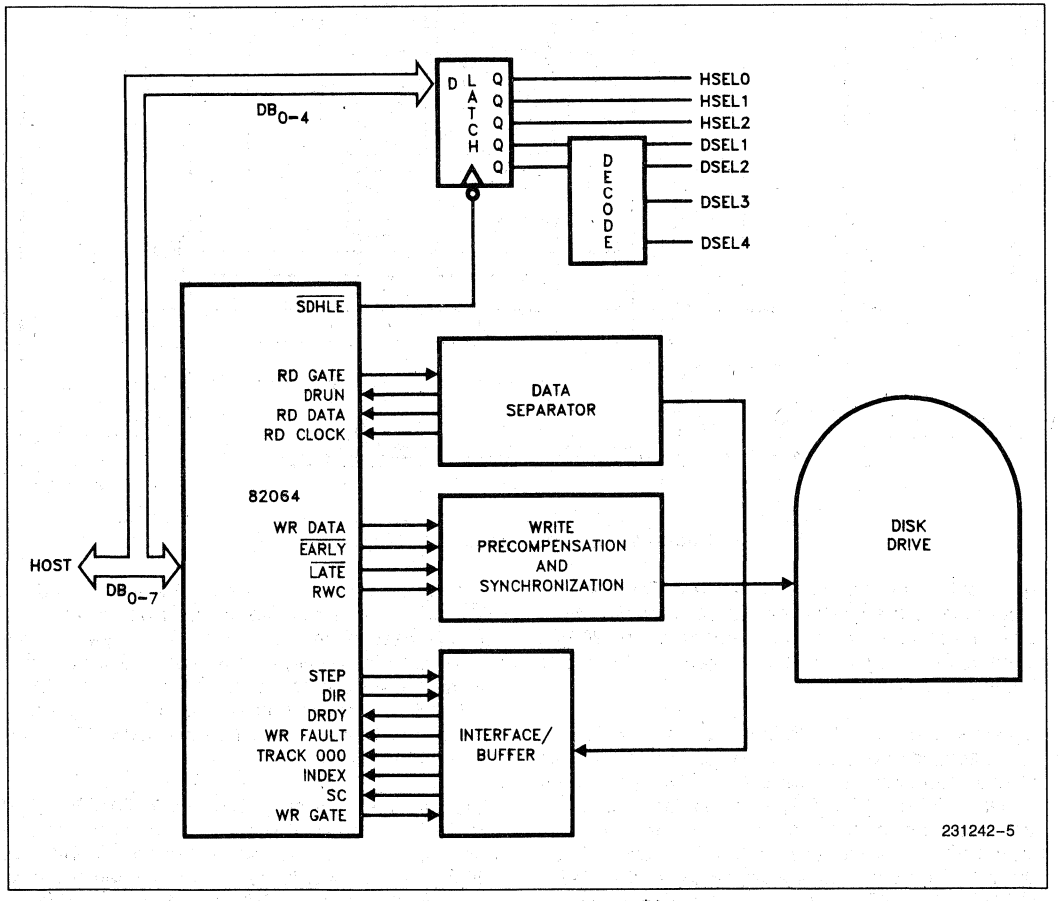


Figure 5. Drive Interface Block Diagram

manual. The 82064 supplies TTL compatible signals, and will interface to most buffer/driver devices.

The data recovery circuits consist of a phase locked loop, data separator, and associated components. The 82064 interacts with the data separator through the DATA RUN (DRUN) and RD GATE signals. A block diagram of a typical data separator circuit is shown in Figure 6. Read data from the drive is presented to the RD DATA input of the 82064, the reference multiplexor, and a retriggerable one shot. The RD GATE output will be deasserted when the 82064 is not inspecting data. The PLL should remain locked to the reference clock.

When any READ or WRITE command is initiated and a search for an address mark begins, the DRUN input is examined. The DRUN one-shot is set for slightly longer than one bit time, allowing it to retrigger constantly on a field of all ones or all zeroes. An internal counter times out to see that DRUN is asserted for two byte times. RD GATE is asserted by the 82064, switching the data separator to lock on to the incoming data stream. If DRUN is deasserted prior to an additional seven byte times, RD GATE is deasserted and the process is repeated. RD GATE will remain asserted until a non-zero, non-address mark byte is detected. The 82064 will then deassert RD GATE for two byte times to allow the PLL to lock back on the reference clock, and start the DRUN search again. If an address mark is detected, RD GATE remains asserted and the command will continue searching for the proper ID field. This sequence is shown in the flow chart in Figure 7.

The write precompensation circuitry is designed to reduce the drift in the data caused by interaction between bits. It is divided into two parts, REDUCED WRITE CURRENT (RWC) and $\overline{\text{EARLY}}$/$\overline{\text{LATE}}$ writing of bits. A block diagram of a typical write precompensation circuit is shown in Figure 8.

The cylinder in which the RWC line becomes active is controlled by the REDUCE WRITE CURRENT register in the Task Register File. When a cylinder is written which has a cylinder number greater than or equal to the contents of this register, the write current will be reduced. This will decrease the interaction between the bits.

Drift may also be caused by the bit pattern. With certain combinations of ones and zeroes some of the bits can drift far enough apart to be difficult to read without error. This phenomenon can be minimized by using $\overline{\text{EARLY}}$ and $\overline{\text{LATE}}$ as described below. The 82064 examines three bits, the last one written, the one being written, and the next one to be written. From this, it determines whether to assert $\overline{\text{EARLY}}$ or $\overline{\text{LATE}}$. Since the bit leaving the 82064 has already been written, it is too late to make it early. Therefore, the external delay circuit must be as follows:

$\overline{\text{EARLY}}$ asserted and $\overline{\text{LATE}}$ deasserted = no delay

$\overline{\text{EARLY}}$ deasserted and $\overline{\text{LATE}}$ deasserted = one unit delay (typically 12–15 ns)

$\overline{\text{EARLY}}$ deasserted and $\overline{\text{LATE}}$ asserted = two units delay (typically 24–30 ns)

$\overline{\text{EARLY}}$ and $\overline{\text{LATE}}$ are always active, and should be gated externally by the RWC signal. Figure 8 illustrates one method of using these signals.

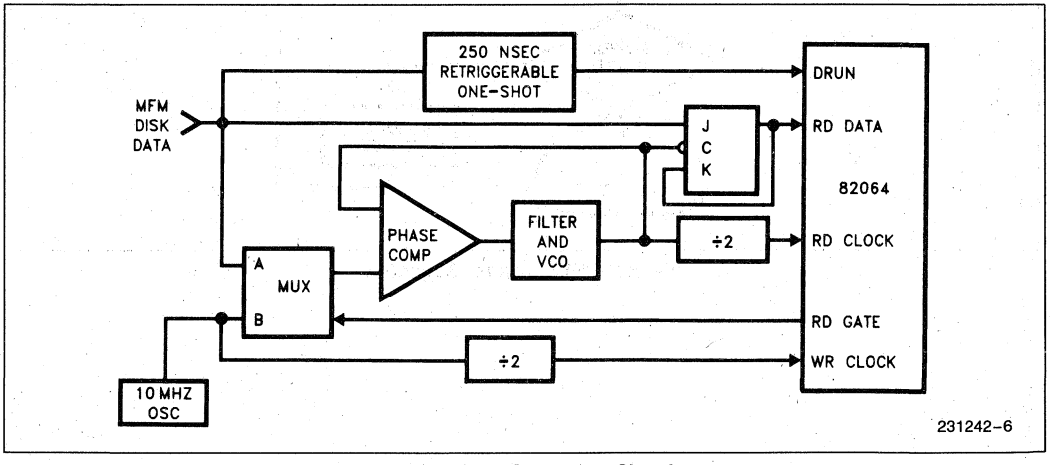


Figure 6. Data Separator Circuit

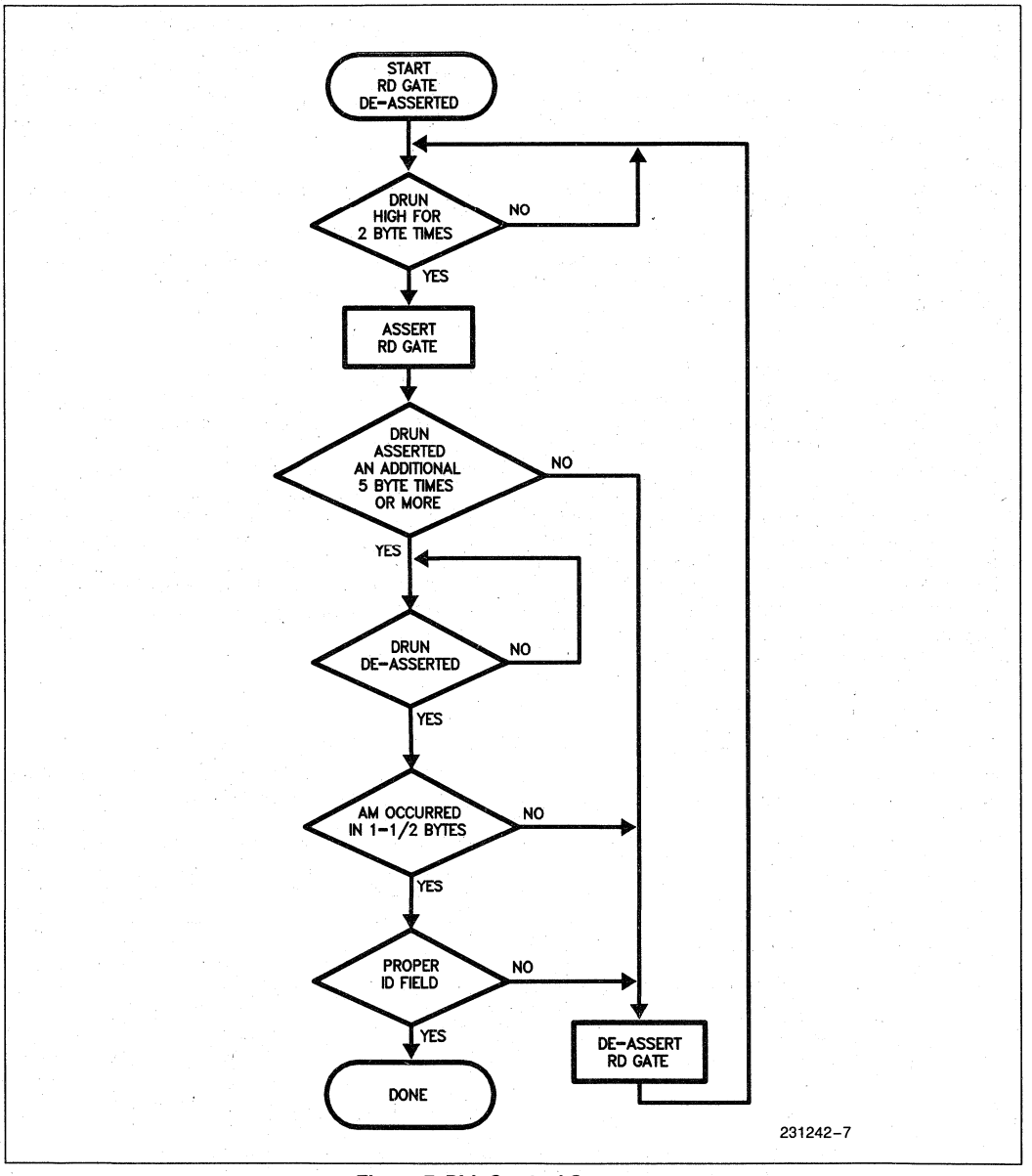


Figure 7. PLL Control Sequence

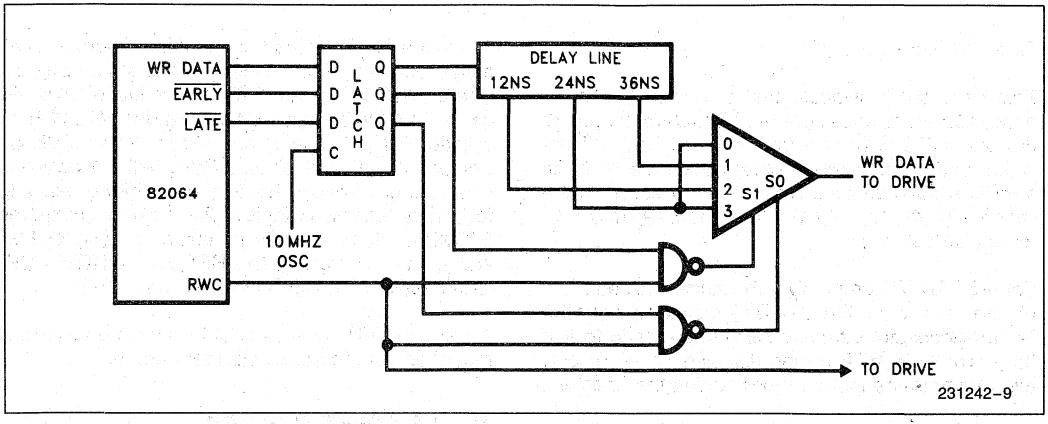


Figure 8. Write Precompensation Circuit

TASK REGISTER FILE

The Task Register File is a bank of nine registers used to hold parameter information pertaining to each command, status information, and the command itself. These registers and their addresses are:

A2	A1	A0	READ	WRITE
0	0	0	BUS TRI-STATED	BUS TRI-STATED
0	0	1	ERROR REGISTER	REDUCE WRITE CURRENT
0	1	0	SECTOR COUNT	SECTOR COUNT
0	1	1	SECTOR NUMBER	SECTOR NUMBER
1	0	0	CYLINDER LOW	CYLINDER LOW
1	0	1	CYLINDER HIGH	CYLINDER HIGH
1	1	0	SDH	SDH
1	1	1	STATUS	COMMAND

NOTE:
These registers are not cleared by $\overline{\text{RESET}}$ being asserted.

ERROR REGISTER

This read only register contains specific error information after the termination of a command. The bits are defined as follows:

7	6	5	4	3	2	1	0
BB	CRC/ECC	0	ID	0	AC	TK000	DAM

Bit 7 - Bad Block Detect (BB)

This bit is set when an ID field has been encountered that contains a bad block mark. It is used for bad sector mapping.

Bit 6 - CRC/ECC Data Field Error (CRC/ECC)

When in the CRC mode (SDH register, bit 7 = 0), this bit is set when a CRC error occurs in the data field. When retries are enabled, ten more attempts are made to read the sector correctly. If none of these attempts are successful bit 0 in the STATUS register is also set. If one of the attempts is successful, the CRC/ECC error bit remains set to inform the host that a marginal condition exists; however, bit 0 in the STATUS register is not set.

When in the ECC mode (SDH register, bit 7 = 1), this bit is set when the first non-zero syndrome is detected. When retries are enabled, up to ten attempts are made to correct the error. If the error is successfully corrected, this bit remains set; however, bit 2 of the STATUS register is also set to inform the host that the error has been corrected. If the error is not correctable, the CRC/ECC error bit remains set and bit 0 of the STATUS register is also set.

The data may be read even if uncorrectable errors exist.

NOTE: If the long mode (L) bit is set in the READ or WRITE command, no error checking is performed.

Bit 5 - Reserved

Not used. Forced to zero.

Bit 4 - ID Not Found (ID)

This bit is set to indicate that the correct cylinder, head, sector, or size parameter could not be found, or that a CRC error occurred in the ID field. This bit is set on the first failure and remains set even if the error is recovered on a retry. When recovery is unsuccessful, the Error bit (bit 0) of the STATUS register is also set.

For a SCAN ID command with retries enabled (T = 0), the Error bit in the STATUS register is set after ten unsuccessful attempts have been made to find the correct ID. With retries disabled (T = 1), only two attempts are made before setting the Error bit.

For a READ or WRITE command with retries enabled (T = 0), ten attempts are made to find the correct ID field. If there is still an error on the tenth try, an auto-scan and auto-seek are performed. Then ten more retries are made before setting the Error bit. When retries are disabled (T = 1), only two tries are made. No auto-scan or auto-seek operations are performed.

Bit 3 - Reserved

Not used. Forced to zero.

Bit 2 - Aborted Command (AC)

Command execution is aborted and this bit is set if a command was issued while DRDY is deasserted or WR FAULT is asserted. This bit will also be set if an undefined command is written to the COMMAND register; however, an implied seek will be executed.

Bit 1 - Track 000 Error (TK000)

This bit is set during the execution of a RESTORE command if the TRACK 000 pin has not gone active after the issuance of 2047 step pulses.

Bit 0 - Data Address Mark (DAM) Not Found

This bit is set during the execution of a READ SECTOR command if the DAM is not found following the proper sector ID.

REDUCE WRITE CURRENT REGISTER

This register is used to define the cylinder number where the RWC output (Pin 33) is asserted.

7	6	5	4	3	2	1	0
CYLINDER NUMBER ÷ 4							

The value (00-FFH) loaded into this cylinder is internally multiplied by four to specify the actual cylinder where RWC is asserted. Thus a value of 01H will cause RWC to be asserted on cylinder 04H, 02H on cylinder 08H, . . . , 9CH on cylinder 270H, 9DH on cylinder 274H, and so on. RWC will be asserted when the present cylinder is greater than or equal to four times the value of this register. For example, the ST506 interface requires precomp on cylinder 80H and above. Therefore, the REDUCE WRITE CURRENT register should be loaded with 20H.

A value of FFH causes RWC to remain deasserted, regardless of the actual cylinder number.

SECTOR COUNT REGISTER

This register is used to define the number of sectors that need to be transferred to the buffer during a READ MULTIPLE SECTOR or WRITE MULTIPLE SECTOR command.

7	6	5	4	3	2	1	0
NUMBER OF SECTORS							

The value contained in the register is decremented after each sector is transferred to/from the sector buffer. A zero represents a 256 sector transfer, a one a one sector transfer, etc. This register is a "don't care" when single sector commands are specified.

SECTOR NUMBER REGISTER

This register holds the sector number of the desired sector.

7	6	5	4	3	2	1	0
SECTOR NUMBER							

For a multiple sector command, it specifies the first sector to be transferred. It is incremented after each sector is transferred to/from the sector buffer. The SECTOR NUMBER register may contain any value from 0 to 255.

The SECTOR NUMBER register is also used to program the Gap 1 and Gap 3 lengths to be used when formating a disk. See the WRITE FORMAT command description for further explanation.

CYLINDER NUMBER LOW REGISTER

This register holds the lower byte of the desired cylinder number.

7	6	5	4	3	2	1	0
LS BYTE OF CYL. NUMBER							

It is used with the CYLINDER NUMBER HIGH register to specify the desired cylinder number over a range of 0 to 2047.

CYLINDER NUMBER HIGH REGISTER

This register holds the three most significant bits of the desired cylinder number.

7	6	5	4	3	2	1	0
x	x	x	x	x	#	#	#

The CYLINDER NUMBER LOW/HIGH register pair determine where the R/W heads are to be positioned. The host writes the desired cylinder number into these registers. Internal to the 82064 is another pair of registers that hold the present head location. When any command other than a RESTORE is executed, the internal head location registers are compared to the CYLINDER NUMBER registers to determine how many cylinders to move the heads and in what direction.

The internal head location registers are updated to equal the CYLINDER NUMBER registers after the completion of the seek.

When a RESTORE command is executed, the internal head location registers are reset to zero while DIR and STEP move the heads to track zero.

SECTOR/DRIVE/HEAD (SDH) REGISTER

The SDH register contains the desired sector size, drive number, and head parameters. The format is shown in Figure 9. The EXT bit (bit 7) is used to select between the CRC or ECC mode. When bit 7 = 1 the ECC mode is selected for the data field. When bit 7 = 0 the CRC mode is selected.

The SDH byte written in the ID field of the disk by the FORMAT command is different than the SDH register contents. The recorded SDH byte does not have the drive number recorded, but does have the bad block mark written. The format of the SDH byte written on the disk is:

7	6	5	4	3	2	1	0
BAD B.	SIZE		0	0	HEAD		

STATUS REGISTER

The status register is used to inform the host of certain events performed by the 82064, as well as reporting status from the drive control lines. Reading the STATUS register deasserts INTRQ. The format is:

7	6	5	4	3	2	1	0
BUSY	READY	WF	SC	DRQ	DWC	CIP	ERR

Bit 7 - Busy

This bit is asserted when a command is written into the COMMAND register and, except for the READ command, is deasserted at the end of the command. When executing a READ command, Busy will be deasserted when the sector buffer is full. Commands should not be loaded into the COMMAND register when Busy is set. When the Busy bit is set, no other bits in the STATUS or ERROR registers are valid.

Bit 6 - Ready

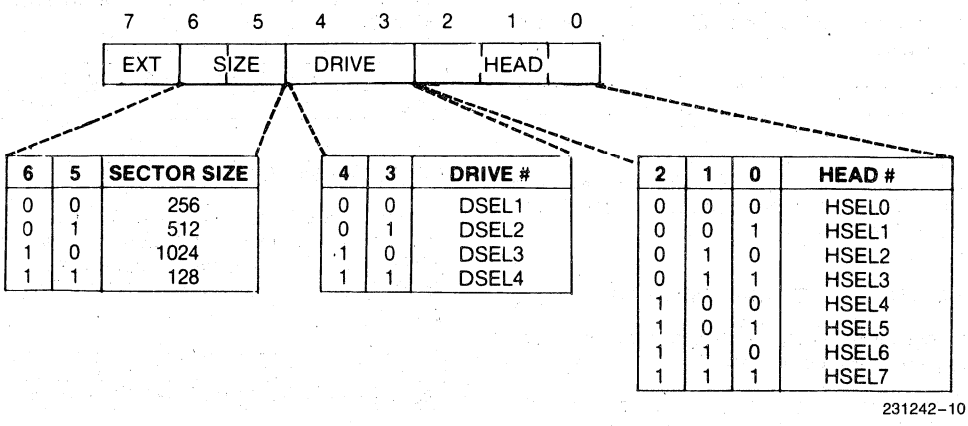
This bit reflects the status of DRDY (pin 28). When this bit equals zero, the command is aborted and the status of this bit is latched.

Bit 5 - Write Fault (WF)

This bit reflects the status of WR FAULT (pin 30). When this bit equals one the command is aborted, INTRQ is asserted, and the status of this bit is latched.

Bit 4 - Seek Complete (SC)

This bit reflects the status of SC (pin 32). When a seek or implied seek has been initiated by a command, execution of the command pauses until the seek is complete. This bit is latched after an aborted command error.



6	5	SECTOR SIZE
0	0	256
0	1	512
1	0	1024
1	1	128

4	3	DRIVE #
0	0	DSEL1
0	1	DSEL2
1	0	DSEL3
1	1	DSEL4

2	1	0	HEAD #
0	0	0	HSEL0
0	0	1	HSEL1
0	1	0	HSEL2
0	1	1	HSEL3
1	0	0	HSEL4
1	0	1	HSEL5
1	1	0	HSEL6
1	1	1	HSEL7

NOTE:
Drive select and head select lines must be generated externally. Figure 3 represents one method of achieving this.

Figure 9. SDH Register Format

Bit 3 - Data Request (DRQ)

The DRQ bit reflects the status of BDRQ (pin 36). It is asserted when the sector buffer must be written into or read from. DRQ and BDRQ remain asserted until BRDY indicates that the sector buffer has been filled or emptied, depending upon the command. BDRQ can be used for DMA interfacing, while DRQ is used in a programmed I/O environment.

Bit 2 - Data Was Corrected (DWC)

When set, this bit indicates that an ECC error has been detected during a read operation, and that the data in the sector buffer has been corrected. This provides the user with an indication that there may be a marginal condition within the drive before the errors become uncorrectable. This bit is forced to zero when not in the ECC mode.

Bit 1 - Command In Progress (CIP)

When this bit is set a command is being executed and a new command should not be loaded. Although a command is being executed, the sector buffer is still available for access by the host. When the 82064 is no longer Busy (bit 7 = 0) the STATUS register can be read. If other registers are read while CIP is set the contents of the STATUS register will be returned.

Bit 0 - Error

This bit is set whenever any bits in the ERROR register are set. It is the logical 'or' of the bits in the ERROR register and may be used by the host processor to quickly check for nonrecoverable errors. The host must read the ERROR register to determine what type of error occurred. This bit is reset when a new command is written into the COMMAND register.

COMMAND REGISTER

The command to be executed is written into this write-only register:

7	6	5	4	3	2	1	0
COMMAND							

The command sets Busy and CIP, and begins to execute as soon as it is written into this register. Therefore, all necessary information should be loaded into the Task Register File prior to entering the command. Any attempt to write a register will be ignored until command execution has terminated, as indicated by the CIP bit being cleared. INTRQ is deasserted when the COMMAND register is written.

COMMAND	7	6	5	4	3	2	1	0
RESTORE	0	0	0	1	R3	R2	R1	R0
SEEK	0	1	1	1	R3	R2	R1	R0
READ SECTOR	0	0	1	0	I	M	L	T
WRITE SECTOR	0	0	1	1	0	M	L	T
SCAN ID	0	1	0	0	0	0	0	T
WRITE FORMAT	0	1	0	1	0	G	0	0
COMPUTE CORRECTION	0	0	0	0	1	0	0	0
SET PARAMETER	0	0	0	0	0	0	0	S

R_{3-0} = Stepping Rate Field

For 5 MHz WR CLOCK:

R_{3-0} = 0000 35 μs
0001 0.5 ms
0010 1.0 ms
0011 1.5 ms
0100 2.0 ms
0101 2.5 ms
0110 3.0 ms
0111 3.5 ms
1000 4.0 ms
1001 4.5 ms
1010 5.0 ms
1011 5.5 ms
1100 6.0 ms
1101 6.5 ms
1110 3.2 μs
1111 16 μs

I = Interrupt Control

I = 0 INTRQ occurs with BDRQ/DRQ indicating the sector buffer is full. Valid only when M = 0.

I = 1 INTRQ occurs when the command is completed and the host has read the sector buffer.

M = Multiple Sector Flag

M = 0 Transfer one sector. Ignore the SECTOR COUNT register.

M = 1 Transfer multiple sectors.

L = Long Mode

L = 0 Normal mode. Normal CRC or ECC functions are performed.

L = 1 Long mode. No CRC or ECC bytes are developed or error checking performed on the data field. The 82064 appends the four additional bytes supplied by the host or disk to the data field.

T = Retry Enable

T = 0 Enable retries.

T = 1 Disable retries.

G = Gap Filler Byte.

G = 0. Gaps 1, 3 and pad bytes "4E".

G = 1. Gaps 1, 3 and pad bytes "AA".

S = Error Correction Span

S = 0 5-bit span.

S = 1 11-bit span.

RESTORE COMMAND

The RESTORE command is used to position the R/W heads over track zero. It is usually issued by the host when a drive has just been turned on. The 82064 forces an auto-restore when a FORMAT command has been issued following a drive number change.

The actual step rate used for the RESTORE command is determined by the seek complete time. A step pulse is issued and the 82064 waits for a rising edge on the SC line before issuing the next pulse. If the rising edge of SC has not occurred within ten revolutions (INDEX pulses) the 82064 switches to sensing the level of SC. If after 2047 step pulses the TRACK 000 line does not go active the 82064 will set the TRACK 000 bit in the ERROR register, assert INTRQ, and terminate execution of the command. An interrupt will also occur if WR FAULT is asserted on DRDY is deasserted at any time during execution.

The rate field specified (R_{3-0}) is stored in an internal register for future use in commands with implied seeks.

A flowchart of the RESTORE command is shown in Figure 10.

SEEK COMMAND

The SEEK command can be used for overlapping seeks on multiple drives. The step rate used is taken from the Rate Field of the command, and is stored in an internal register for future use by those commands with implied seek capability.

The direction and number of step pulses needed are calculated by comparing the contents of the CYLINDER NUMBER registers in the Task Register File to the present cylinder position stored internally. After all the step pulses have been issued the present cylinder position is updated, INTRQ is asserted, and the command terminated.

If DRDY is deasserted or WR FAULT is asserted during the execution of the command, INTRQ is asserted and the command aborts setting the AC bit in the ERROR register.

If an implied seek is performed, the step rate indicated by the rate field is used for all but the last step pulse. On the last pulse, the command execution continues until the rising edge of SC is detected. If 10 INDEX pulses are received without a rising edge of SC, the 82064 will switch to sensing the level of SC.

A flowchart of the SEEK command flow is shown in Figure 11.

READ SECTOR

The READ SECTOR command is used to transfer one or more sectors of data from the disk to the sector buffer. Upon receipt of the command, the 82064 checks the CYLINDER NUMBER LOW/HIGH register pair against the internal cylinder position register to see if they are equal. If not, the direction and number of steps calculation takes place, and a seek is initiated. As stated in the description of the SEEK command, if an implied seek occurs, the step rate specified by the rate field is used for all but the last step pulse. On the last step pulse the seek continues until the rising edge of SC is detected.

If the 82064 detects a change in the drive number since the last command, an auto-scan ID is performed. This updates the internal cylinder position register to reflect the current drive before the seek begins.

After the 82064 senses SC (with or without an implied seek) it must find an ID field with the correct cylinder number, head, sector size, and CRC. If retries are enabled (T = 0), ten attempts are made to find the correct ID field. If there is still an error on the tenth try, an auto-scan ID and auto-seek are performed. Then ten more retries are attempted before setting the ID Not Found error bit. When retries are disabled (T = 1) only two tries are made. No auto-scan or auto-seek operations are performed.

When the data address mark (DAM) is found, the 82064 is ready to transfer data into the sector buffer. When the disk has filled the sector buffer, the 82064 asserts BDRQ and DRQ and then checks the I flag. If I = 0, INTRQ is asserted, signaling the host to read the contents of the sector buffer. If I = 1, INTRQ occurs after the host has read the sector buffer and the command has terminated. If after successfully reading the ID field, the DAM is not found the DAM Not Found bit in the ERROR register is set.

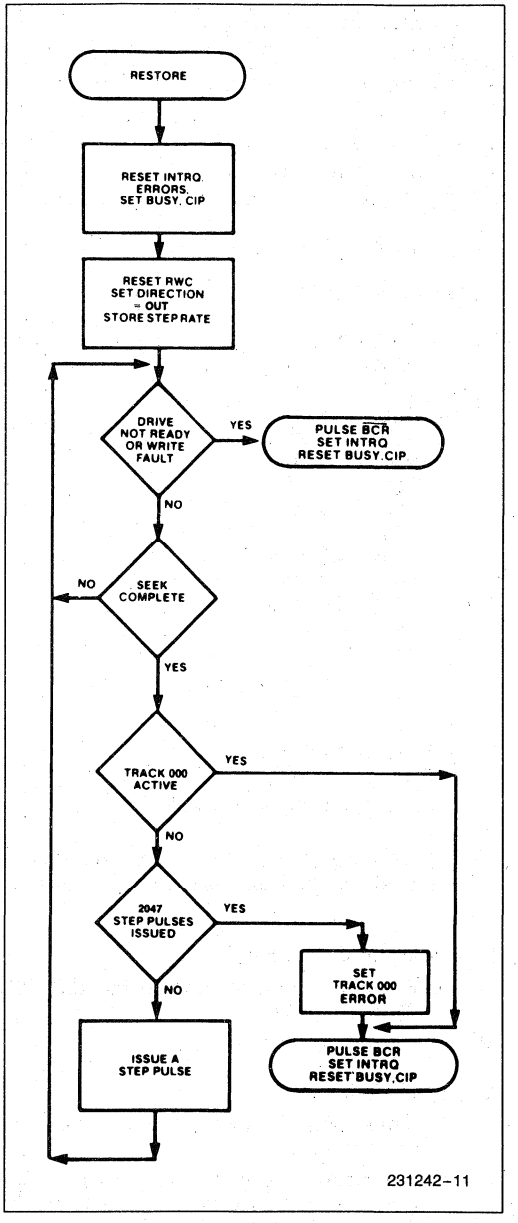


Figure 10. Restore Command Flow

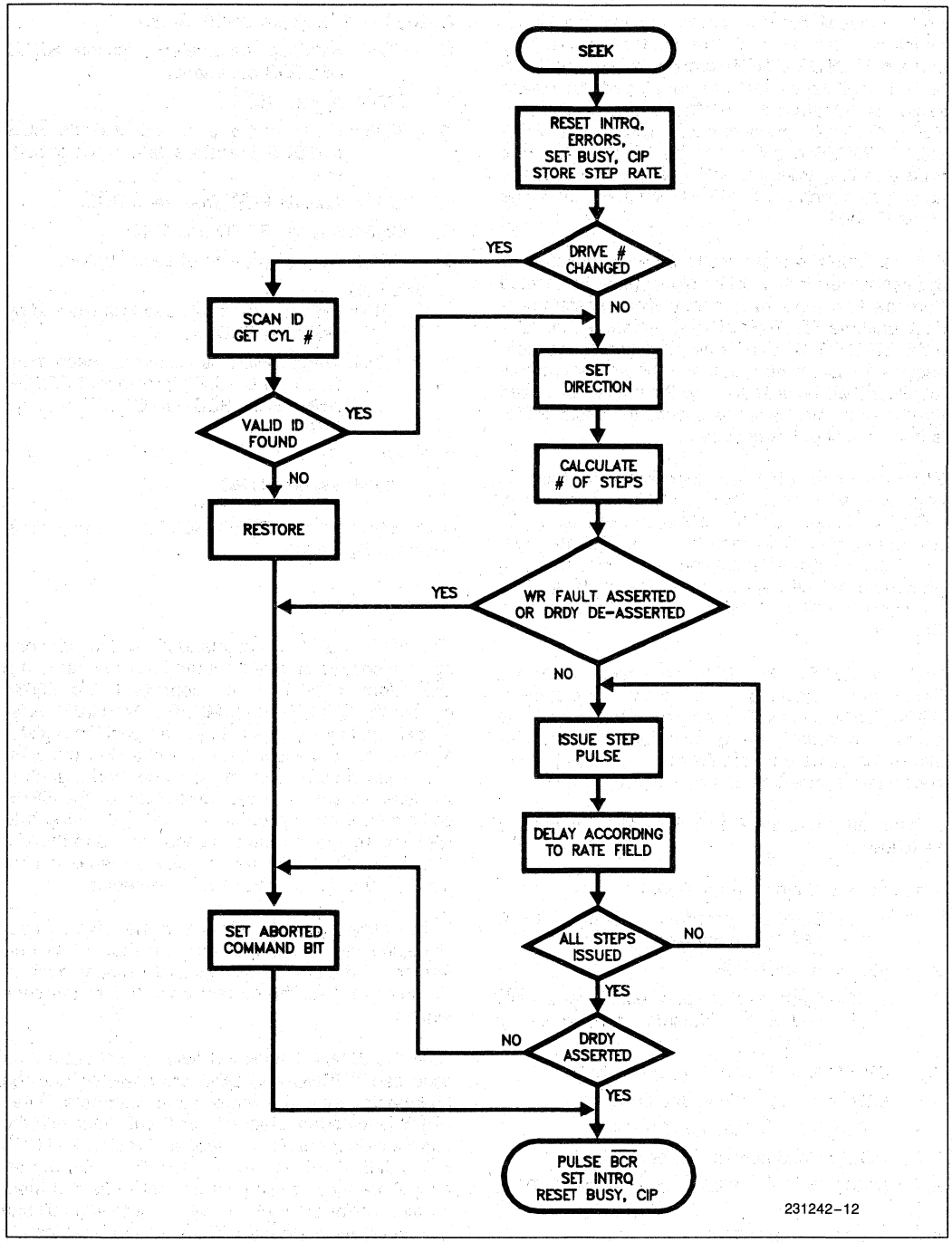


Figure 11. Seek Command Flow

An optional M flag can be set for multiple sector transfers. When M = 0, one sector is transferred and the SECTOR COUNT register is ignored. When M = 1, multiple sectors are transferred. After each sector is transferred, the 82064 decrements the SECTOR COUNT register and increments the SECTOR NUMBER register. The next logical sector is transferred regardless of any interleave. Sectors are numbered during the FORMAT command by a byte in the ID field.

For the 82064 to make multiple sector transfers to the sector buffer, the BRDY signal must be toggled from low to high for each sector. The transfers continue until the SECTOR COUNT register equal zero. If the SECTOR COUNT is not zero (indicating more sectors remain to be read), and the sector buffer is full, BDRQ will be asserted and the host must unload the sector buffer. Once this occurs, the sector buffer is free to accept the next sector.

WR FAULT and DRDY are monitored throughout the command execution. If WR FAULT is asserted or DRDY is deasserted, the command will terminate and the Aborted Command bit in the ERROR register will be set. For a description of the error checking procedure on the data field see the explanation in the section entitled "CRC and ECC Generator and Checker."

Both the READ and WRITE commands feature a "simulated completion" to ease programming. BDRQ, DRQ, and INTRQ are generated in a normal manner upon detection of an error condition. This allows the same program flow for successful or unsuccessful completion of a command.

In summary then, the READ SECTOR operation is as follows:

When M = 0 (Single Sector Read)

1. HOST: Sets up parameters. Issues READ SECTOR command.
2. 82064: Asserts $\overline{BCR}$.
3. 82064: Finds sector specified. Asserts $\overline{BCR}$ and $\overline{BCS}$. Transfers data to sector buffer.
4. 82064: Asserts $\overline{BCR}$. Deasserts $\overline{BCS}$.
5. 82064: Asserts BDRQ and DRQ.
6. 82064: If I = 1 then go to 9.
7. HOST: Read contents of sector buffer.
8. 82064: Wait for BRDY, then assert INTRQ. End.
9. 82064: Assert INTRQ.
10. HOST: Read contents of sector buffer. End.

When M = 1 (Multiple Sector Read)

1. HOST: Sets up parameters. Issues READ SECTOR command.
2. 82064: Asserts $\overline{BCR}$.
3. 82064: Finds sector specified. Asserts $\overline{BCR}$ and $\overline{BCS}$. Transfers data to sector buffer.
4. 82064: Asserts $\overline{BCR}$. Deasserts $\overline{BCS}$.
5. 82064: Asserts BDRQ and DRQ.
6. HOST: Reads contents of sector buffer.
7. SECTOR BUFFER: Indicates data has been transferred by asserting BRDY.
8. 82064: When BRDY is asserted, decrement SECTOR COUNT, increment SECTOR NUMBER. If SECTOR COUNT = 0, go to 10.
9. 82064: Go to 2.
10. 82064: Assert INTRQ.

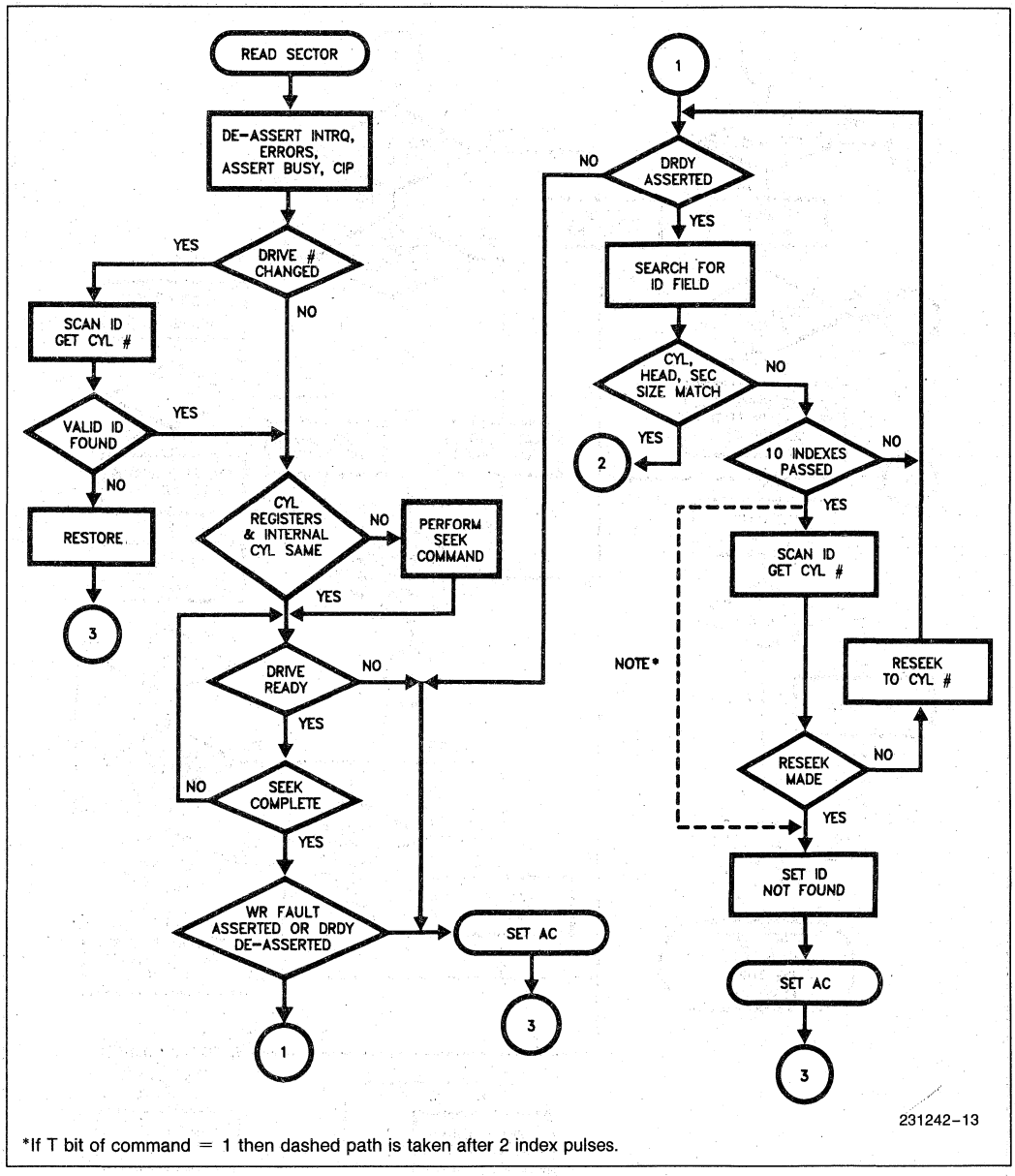
A flowchart of the READ SECTOR command is shown in Figure 12.

WRITE SECTOR

The WRITE SECTOR command is used to write one or more sectors of data from the sector buffer to the disk. Upon receipt of the command, the 82064 checks the CYLINDER NUMBER LOW/HIGH register pair against the internal cylinder position register to see if they are equal. If not, the direction and number of steps calculation takes place, and a seek is initiated. As stated in the description of the SEEK command, if an implied seek occurs, the step rate specified by the rate field is used for all but the last step pulse. On the last step pulse the seek continues until the rising edge of SC is detected.

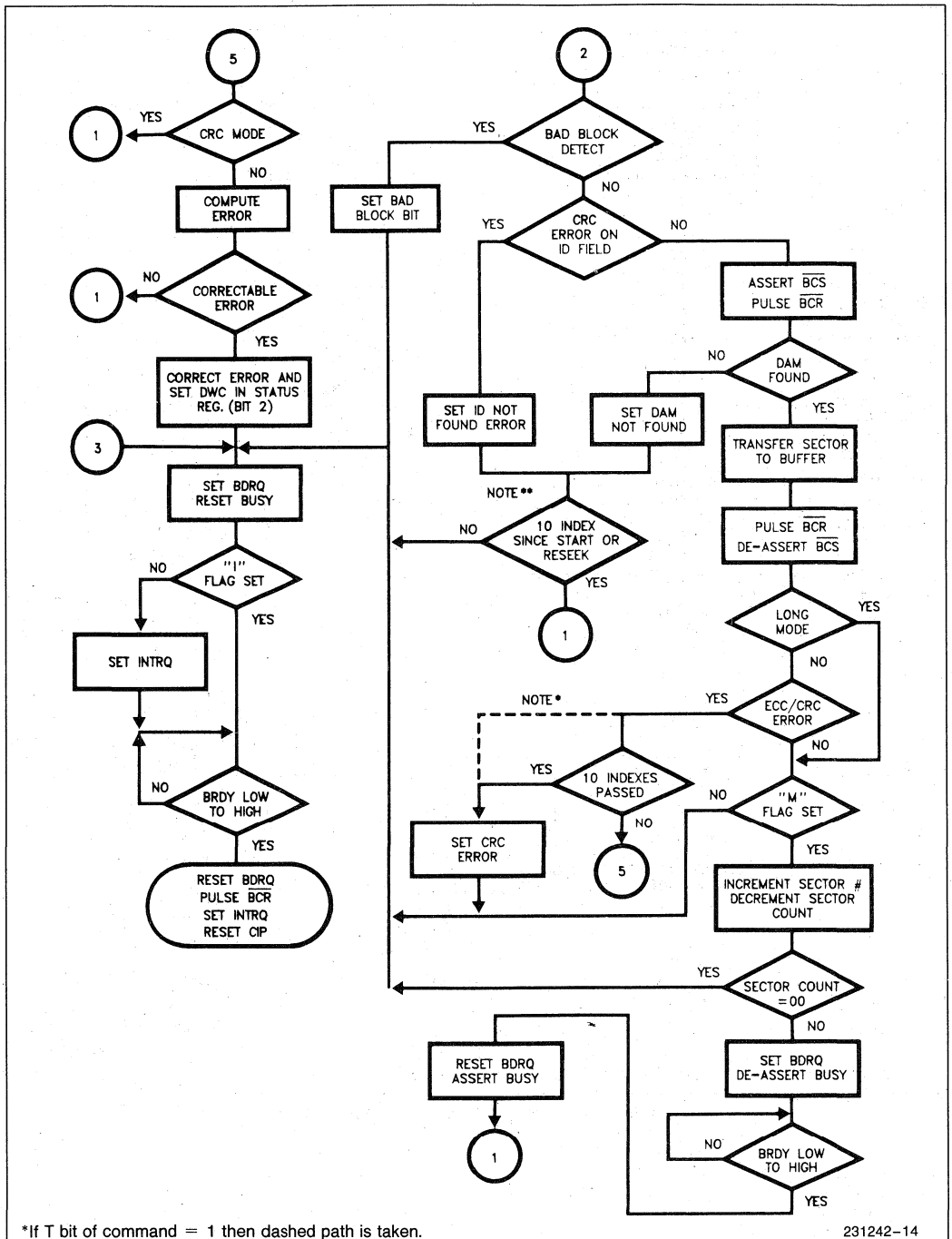
If the 82064 detects a change in the drive number since the last command, an auto-scan ID is performed. This updates the internal cylinder position register to reflect the current drive before the seek begins.

After the 82064 senses SC (with or without an implied seek) BDRQ and DRQ are asserted and the host begins filling the sector buffer with data. When BRDY is asserted, a search for the ID field with the correct cylinder number, head, sector size, and CRC is initiated. If retries are enabled (T = 0), ten attempts are made to find the correct ID field. If there is still an error on the tenth try, an auto-scan ID and auto-seek are performed. Then ten more retries are attempted before setting the ID Not Found error bit. When retries are disabled (T = 1) only two tries are made. No auto-scan or auto-seek operations are performed.



*If T bit of command = 1 then dashed path is taken after 2 index pulses.

Figure 12a. Read Sector Command Flow



*If T bit of command = 1 then dashed path is taken.
**If T bit of command = 1 then test is for 2 index pulses.

231242-14

Figure 12b. Read Sector Command Flow (Continued)

When the correct ID is found, WR GATE is asserted and data is written to the disk. When the CRC/ECC bit (SDH Register, bit 7) is zero, the 82064 generates a two byte CRC character to be appended to the data. When the CRC/ECC bit is one, four ECC bytes replace the CRC character. When L = 1, the polynomial generator is inhibited and neither CRC or ECC bytes are generated. Instead four bytes of data supplied by the host are written.

During a WRITE MULTIPLE SECTOR command (M = 1), the SECTOR NUMBER register is incremented and the SECTOR COUNT register is decremented. If BRDY is asserted after the first sector is read from the sector buffer, the 82064 continues to read data from the sector buffer for the next sector. If BRDY is deasserted, the 82064 asserts BDRQ and waits for the host to place more data in the sector buffer.

In summary then, the WRITE SECTOR operation is as follows:

When M = 0,1

1. HOST: Sets up parameters. Issues WRITE SECTOR command.
2. 82064: Asserts BDRQ and DRQ.
3. HOST: Loads sector buffer with data.
4. 82064: Waits for rising edge of BRDY.
5. 82064: Finds specified ID field. Writes sector to disk.
6. 82064: If M = 0, asserts INTRQ. End.
7. 82064: Increments SECTOR NUMBER. Decrements SECTOR COUNT.
8. 82064: IF SECTOR COUNT = 0, assert INTRQ. End.
9. 82064: Go to 2.

A flowchart of the WRITE SECTOR command is shown in Figure 13.

SCAN ID

The SCAN ID command is used to update the SDH, SECTOR NUMBER, and CYLINDER NUMBER LOW/HIGH registers.

After the command is loaded, the SC line is sampled until it is valid. The DRDY and WR FAULT lines are also monitored throughout execution of the command. If a fault occurs the command is aborted and the appropriate error bits are set. When the first ID field is found, the ID information is loaded into the SDH, SECTOR NUMBER, and CYLINDER NUMBER registers. The internal cylinder position register is also updated. If this is an auto-scan caused by a change in drive numbers, only the internal position register is updated. If a bad block is detected, the BAD BLOCK bit will also be set.

If an ID field is not found, or if a CRC error occurs, and if retries are enabled (T = 0), ten attempts are made to read it. If retries are disabled (T = 1), only two tries are made. There is no auto-seek in this command and the sector buffer is not disturbed.

A flowchart of the SCAN ID command is shown in Figure 14.

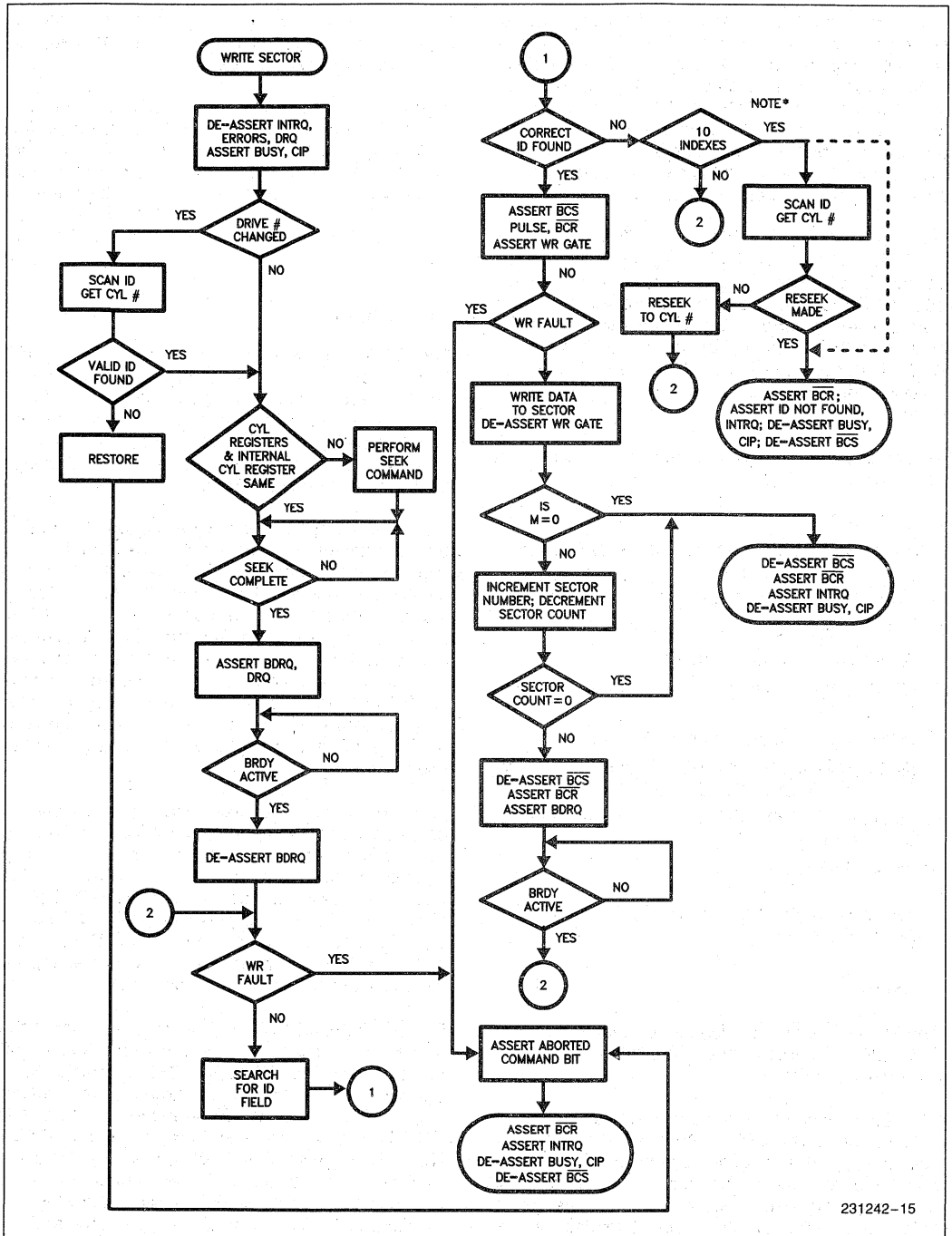
WRITE FORMAT

The WRITE FORMAT command is used to format one track using information in the Task Register File and the sector buffer. During execution of this command, the sector buffer is used for additional parameter information instead of data. Shown in Figure 15 is the contents of a sector buffer for a 32 sector track with an interleave factor of two.

Each sector requires a two byte sequence. The first byte designates whether a bad block mark is to be recorded in the sector's ID field. An 00H is normal; an 80H indicates a bad block mark for that sector. In the example of Figure 15, sector 04 will get a bad block mark recorded. The second byte indicates the logical sector number to be recorded. This allows sectors to be recorded with any interleave factor desired. The remaining memory in the sector buffer may be filled with any value; its only purpose is to generate a BRDY to tell the 82064 to begin formatting the track.

If the drive number has been changed since the last command, an auto-restore is initiated, positioning the heads to track 000. The internal cylinder position register is set to zero and the heads seek to the track specified in the Task Register File CYLINDER NUMBER register. This prevents an ID Not Found error from occuring due to an incompatible format, or the track having been erased. A normal implied seek is also in effect for this command.

The SECTOR COUNT register is used to hold the total number of sectors to be formatted (FFH = 255 sectors), while the SECTOR NUMBER register holds the number of bytes, minus three, to be used for Gap 1 and Gap 3. If, for example, the SECTOR COUNT register value is 02H and the SECTOR NUMBER register value is 00H, then 2 sectors are formatted and 3 bytes of 4EH are written for Gap 1 and Gap 3. The data fields are filled with FFH and the CRC or ECC is automatically generated and appended. After the last sector is written the track is filled with 4EH.



*If retries disabled then dashed path is taken after 2 index pulses.

Figure 13. Write Sector Command Flow

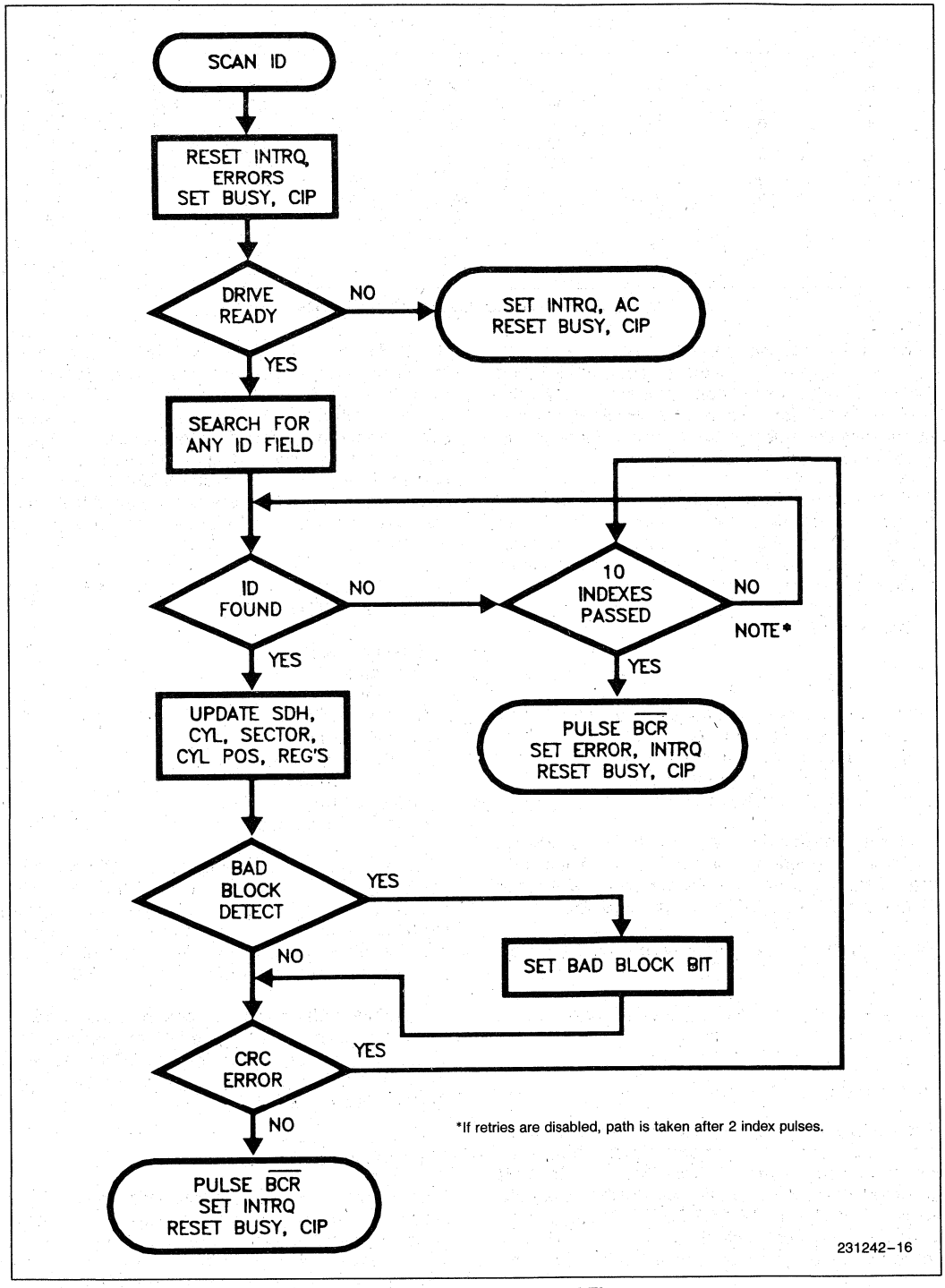


Figure 14. Scan ID Command Flow

ADDR	DATA 0	1	2	3	4	5	6	7
00	00	00	00	10	00	01	00	11
08	00	02	00	12	00	03	00	13
10	80	04	00	14	00	05	00	15
18	00	06	00	16	00	07	00	17
20	00	08	00	18	00	09	00	19
28	00	0A	00	1A	00	0B	00	1B
30	00	0C	00	1C	00	0D	00	1D
38	00	0E	00	1E	00	0F	00	1F
40	FF	FF	FF	FF	FF	FF	FF	FF
				:				
F0	FF	FF	FF	FF	FF	FF	FF	FF

Figure 15. Format Command Buffer Contents

The user may select a value of 4EH or AAH for Gaps 1, 3 and pad bytes. This is done by setting bit 2 (Gap Filler Byte) of the format command to "0" for a value of 4EH or "1" for a value of AAH. AAH provides better frequency discrimination with MFM decoding, allowing for simpler circuitry.

The Gap 3 value is determined by the drive motor speed variation, data sector length, and the interleave factor. The interleave factor is only important when 1:1 interleave is used. The formula for determining the minimum Gap 3 length is:

$$\text{Gap 3} = (2*M*S)+K+E$$

where:

M = motor speed variation (e.g., 0.03 for + 3%)
S = sector length in bytes
K = 18 for an interleave factor of 1
    0 for any other interleave factor
E = 2 if ECC is enabled (SDH register, bit 7 = 1)

As for all commands, if WR FAULT is asserted or DRDY is deasserted during execution of the command, the command terminates and the Aborted Command bit in the ERROR register is set.

Figure 16 shows the format which the 82064 will write on the disk.

A flowchart of the WRITE FORMAT command is shown in Figure 17.

COMPUTE CORRECTION

The COMPUTE CORRECTION command determines the location and pattern of a single burst error, but does not correct it. The host, using the data provided by the 82064, must perform the actual correction. The COMPUTE CORRECTION command is used following a data field ECC error. The command initiating the read must specify no retries (T = 1).

The COMPUTE CORRECTION command first writes the four syndrome bytes from the internal ECC register to the sector buffer. Then the ECC register is clocked. With each clock, a counter is incremented and the pattern examined. If the pattern is correctable, the procedure is stopped and the count and pattern are written to the sector buffer, following the syndrome. The process is also stopped if the count exceeds the sector size before a correctable pattern is found.

When the command terminates the sector buffer contains the following data:

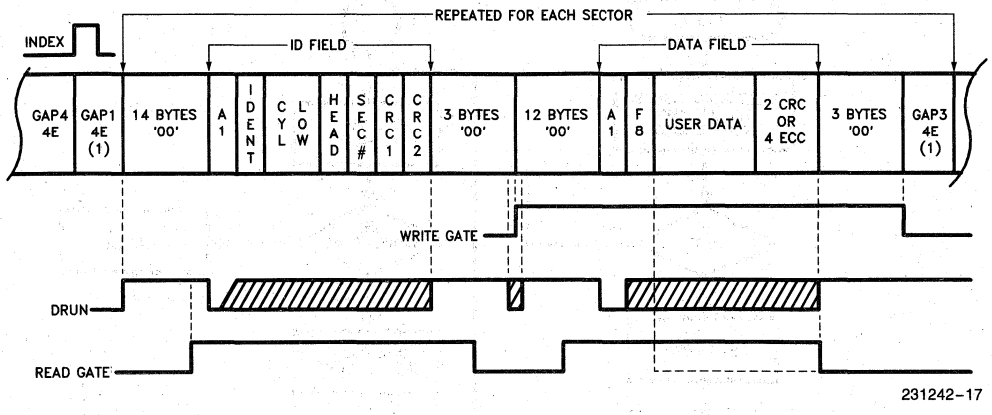
Syndrome MSB
Syndrome
Syndrome
Syndrome LSB
Error Pattern Offset
Error Pattern Offset
Error Pattern MSB
Error Pattern
Error Pattern LSB

As an example, when the Error Pattern Offset is zero the following procedure may correct the error. The first data byte of the sector is exclusive OR'd with the MSB of the Error Pattern, the second data byte with the second byte of the Error Pattern, and the third data byte with the LSB of the Error Pattern.

If the sector buffer count exceeds the sector size, or if the error burst length is greater than that selected by the Set Parameter command, the ECC/CRC error in the ERROR register and the Error bit in the STATUS register is set.

SET PARAMETER

This command selects the correction span to be used for the error correction process. A 5-bit span is selected when bit zero of the command equals 0, and an 11-bit span when bit zero equals 1. The 82064 defaults to a 5-bit span after a RESET.



ID FIELD		
A1	=	A1H with 0AH Clock
IDENT	=	Bits 3, 1, 0 = Cylinder High FE = 0-255 Cylinders FF = 256-511 Cylinders FC = 512-767 Cylinders FD = 768-1023 Cylinders F6 = 1024-1279 Cylinders F7 = 1280-1535 Cylinders F4 = 1536-1791 Cylinders F5 = 1792-2047 Cylinders
HEAD	=	Bits 0, 1, 2 = Head Number Bits 3, 4 = 0 Bits 5, 6 = Sector Size Bit 7 = Bad Block Mark
Sec #	=	Logical Sector Number
DATA FIELD		
A1	=	A1H with 0AH clock
F8	=	Data Address Mark; Normal Clock
USER	=	Data Field 128 to 1024 Bytes

NOTE:
1. GAP 1 and 3 length determined by Sector Number Register contents during formatting.

Figure 16. Track Format

5

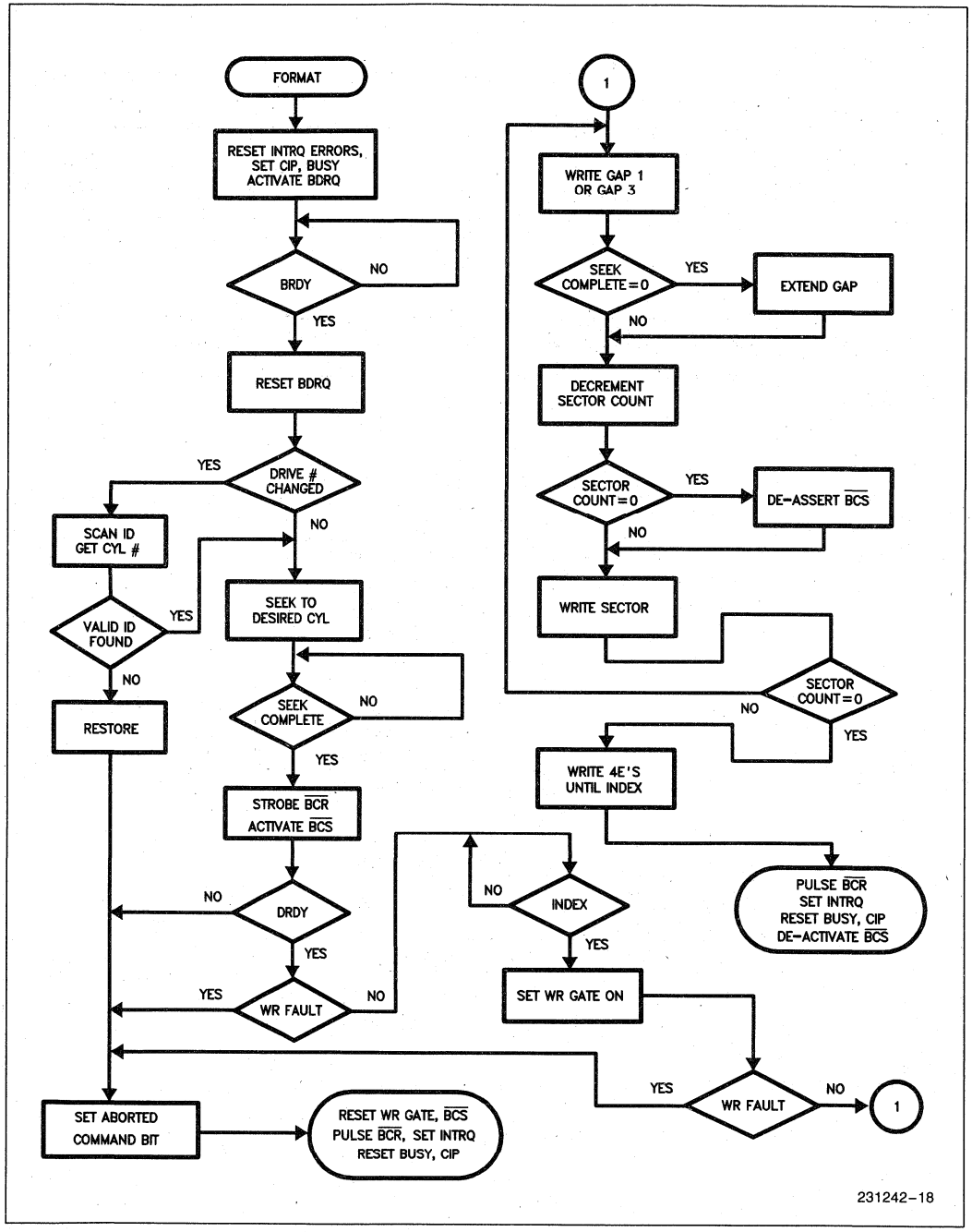


Figure 17. Write Format Command Flow

ELECTRICAL CHARACTERISTICS
ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Supply Voltage .....−0.5V to +8V
Voltage on Any Input .....GND − 2V to +6.5V
Voltage on Any Output .GND − 0.5V to V_{CC} + 0.5V
Power Dissipation .....1 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C; V_{CC} = +5V ±10%; GND = 0V

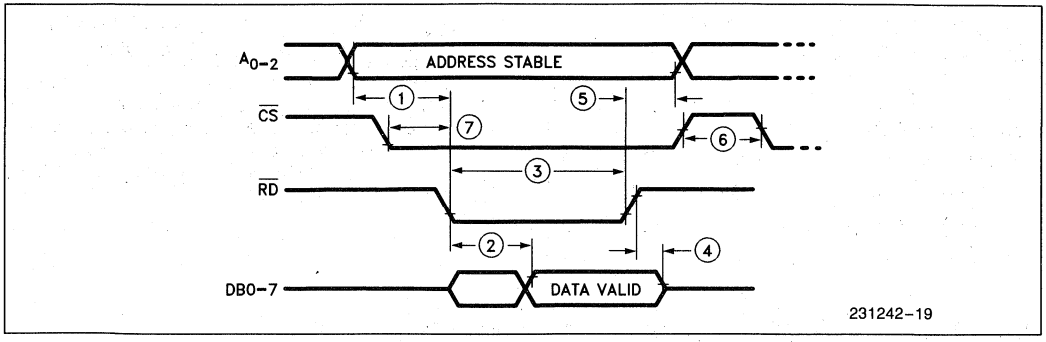
Symbol	Parameter	Min	Max	Units	Test Conditions
I_{IL}	Input Leakage Current		±10	µA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Leakage Current		±10	µA	V_{OUT} = V_{CC} to 0.45V
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{OH}	Output High Voltage	V_{CC} − 0.4 3.0		V	I_{OH} = −100 µA I_{OH} = −2.5 mA
V_{OL}	Output Low Voltage		0.4 0.45	V	I_{OL} = 2.5 mA 6.0 mA P21, 22, 23
I_{CC}	Supply Current		20 45	mA	See Note 10 See Note 11
I_{CCSB}	Standby Supply Current		2	mA	See Note 12
C_{IN}	Input Capacitance		10	pF	f_c = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to GND
For Pins 25, 34, 37, 39 (WR CLOCK, DRUN, READ DATA, READ CLOCK)					
TRS	Rise Time		30	ns	0.9V to 4.2V

A.C. CHARACTERISTICS T_A = 0°C to 70°C; V_{CC} = +5V ± 10%; GND = 0V

HOST READ TIMING WR CLOCK = 5.0 MHz

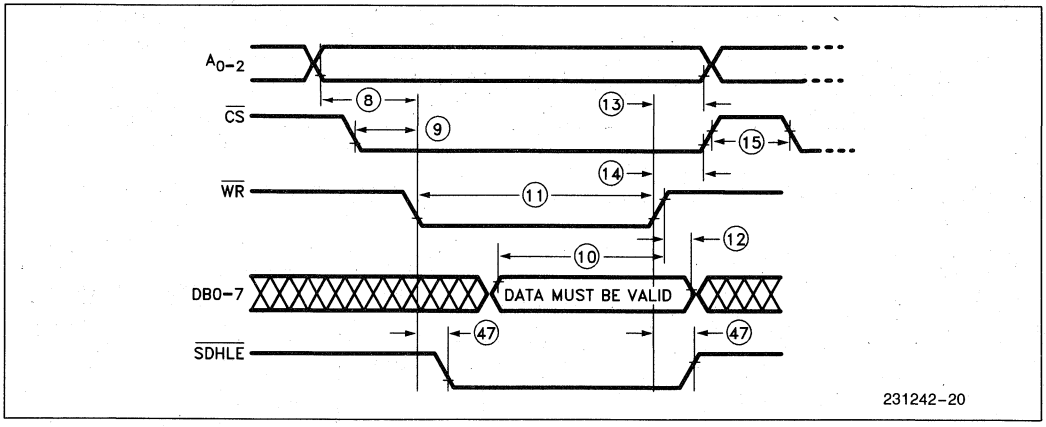
Symbol	Parameter	Min	Max	Units	Test Conditions
1	Address Stable Before $\overline{RD}$ ↓	0		ns	
2	Data Delay from $\overline{RD}$ ↓		150	ns	
3	$\overline{RD}$ Pulse Width	100		ns	
4	Data Valid after RD ↑	10	100	ns	
5	Address Hold Time after $\overline{RD}$ ↑	0		ns	
6	Read Recovery Time	300		ns	
7	$\overline{CS}$ Stable before $\overline{RD}$ ↓	0		ns	See Note 6

5



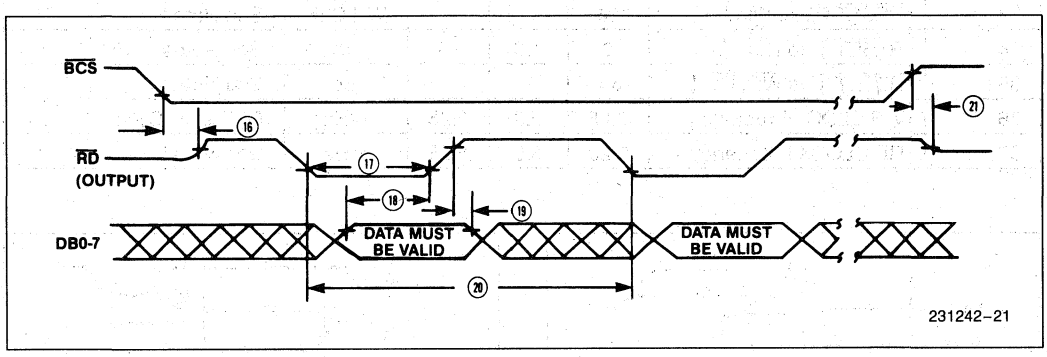
HOST WRITE TIMING WR CLOCK = 5.0 MHz

Symbol	Parameter	Min	Max	Units	Test Conditions
8	Address Stable Before $\overline{WR}\downarrow$	0		ns	
9	$\overline{CS}$ Stable Before $\overline{WR}\downarrow$	0		ns	
10	Data Setup Time Before $\overline{WR}\uparrow$	75		ns	
11	$\overline{WR}$ Pulse Width	100	10000	ns	
12	Data Hold Time After $\overline{WR}\uparrow$	0		ns	
13	Address Hold Time After $\overline{WR}\uparrow$	0		ns	
14	$\overline{CS}$ Hold Time After $\overline{WR}\uparrow$	0		ns	See Note 7
15	Write Recovery Time	300		ns	
47	$\overline{SDHLE}$ Propagation Delay	20	150	ns	



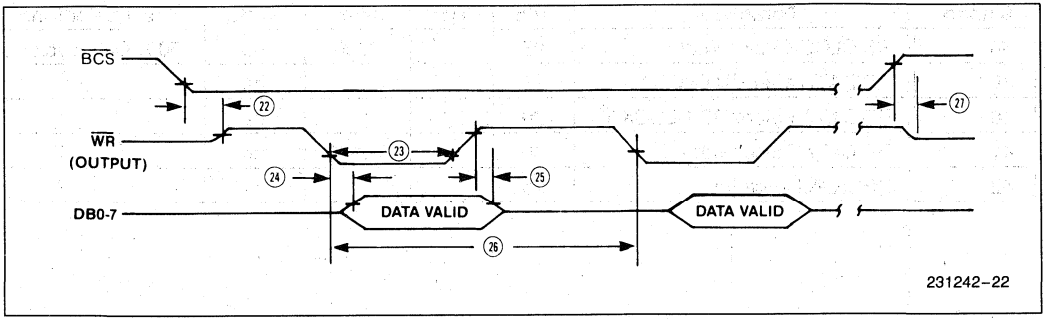
BUFFER READ TIMING (WRITE SECTOR COMMAND) WR CLOCK = 5.0 MHz

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
16	$\overline{BCS}$ ↓ to $\overline{RD}$ Valid	0		100	ns	
17	$\overline{RD}$ Output Pulse Width	300	400	500	ns	See Note 3
18	Data Setup to $\overline{RD}$ ↑	140			ns	
19	Data Hold from $\overline{RD}$ ↑	0			ns	
20	$\overline{RD}$ Repetition Rate	1.2	1.6	2.0	μs	See Note 8
21	$\overline{RD}$ Float from $\overline{BCS}$ ↑	0		100	ns	



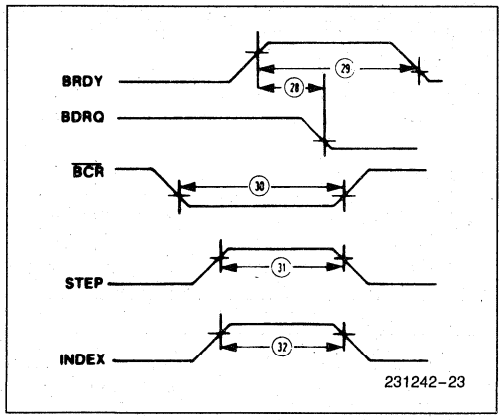
BUFFER WRITE TIMING (READ SECTOR COMMAND) WR CLOCK = 5.0 MHz

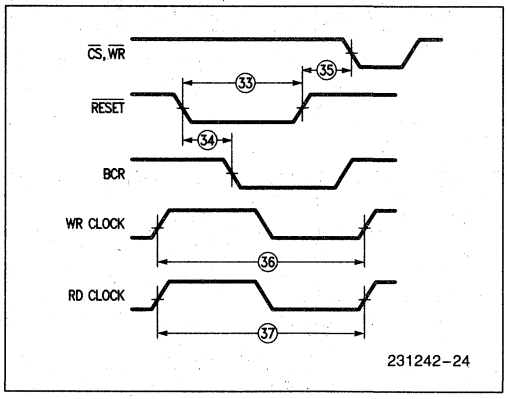
Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
22	$\overline{BCS}$ ↓ to $\overline{WR}$ Valid	0		100	ns	
23	$\overline{WR}$ Output Pulse Width	300	400	500	ns	See Note 3
24	Data Valid from $\overline{WR}$ ↓			150	ns	
25	Data Hold from $\overline{WR}$ ↑	60		200	ns	
26	$\overline{WR}$ Repetition Rate	1.2	1.6	2.0	μs	See Note 8
27	$\overline{WR}$ Float from $\overline{BCS}$ ↑	0		100	ns	



MISCELLANEOUS TIMING

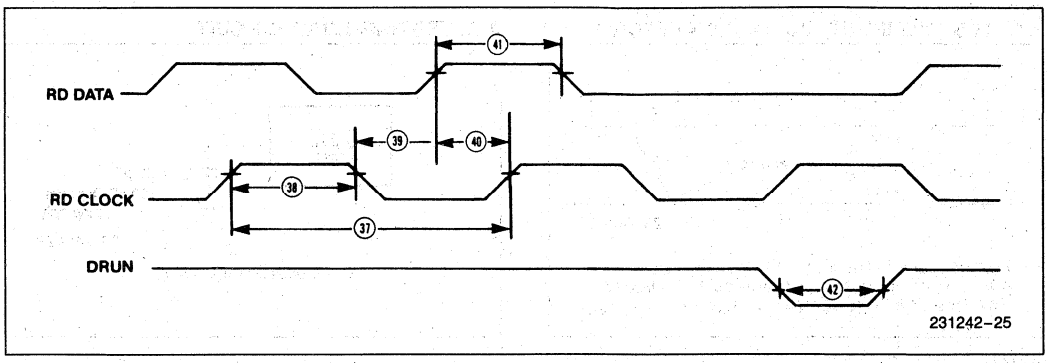
Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
28	BDRQ Reset from BRDY	20		200	ns	
29	BRDY Pulse Width	400			ns	See Note 4
30	$\overline{BCR}$ Pulse Width	1.4	1.6	1.8	μs	See Notes 9, 13, 15
31	STEP Pulse Width	1.5	1.6	1.7	μs	Step Rate = 3.2 μs/step
		7.6	8.0	8.4	μs	All other step rates, See Notes 14, 15
32	INDEX Pulse Width	500			ns	
33	$\overline{RESET}$ Pulse Width	24			WR CLK	See Note 2
34	$\overline{RESET}$ ↓ to $\overline{BCR}$ ↓	0	1.6	3.2	μs	See Notes 1, 15
35	$\overline{RESET}$ ↑ to $\overline{WR}$, $\overline{CS}$ ↓	6.4			μs	See Note 1
36	WR CLOCK Frequency	0.25	5.0	5.25	MHz	50% Duty Cycle
37	RD CLOCK Frequency	0.25	5.0	5.25	MHz	See Note 5





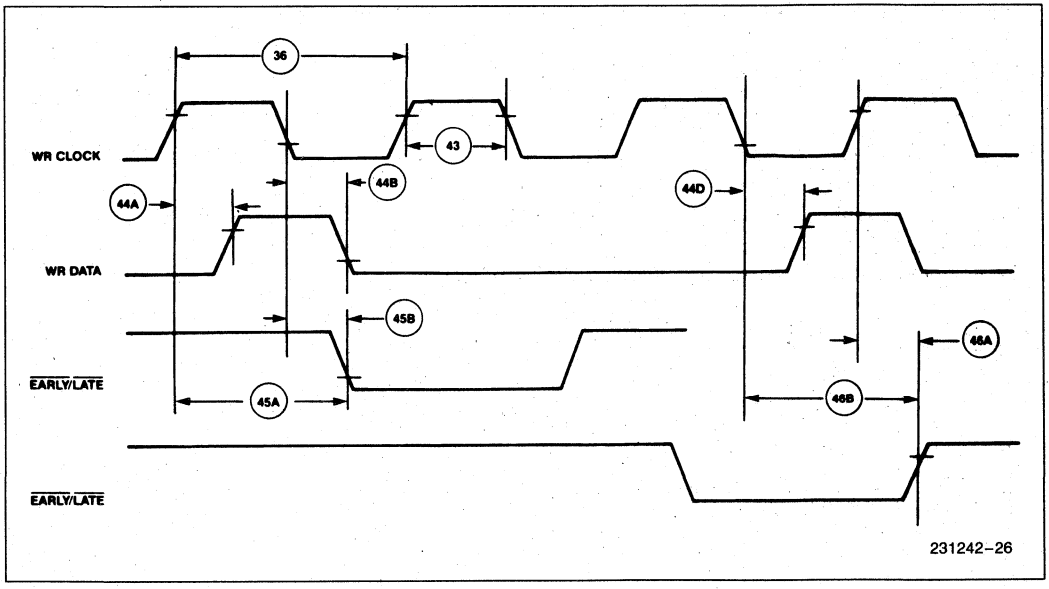
READ DATA TIMING WR CLOCK = 5.0 MHZ

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
38	RD CLOCK Pulse Width	95		2000	ns	50% Duty Cycle
39	RD DATA after RD CLOCK ↓	10			ns	
40	RD DATA before RD CLOCK ↑	20			ns	
41	RD DATA Pulse Width	40		T38/2	ns	
42	DRUN Pulse Width	30			ns	

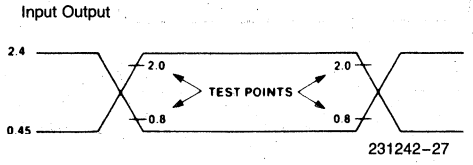


WRITE DATA TIMING WR CLOCK = 5.0 MHZ

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
43	WR CLOCK Pulse Width	95		2000	ns	50% Duty Cycle
	Propagation Delay					
44A	WR CLOCK ↑ to WR DATA ↑	10		65	ns	
44B	WR CLOCK ↓ to WR DATA ↓					
44D	WR CLOCK ↓ to WR DATA ↑					
45A	WR CLOCK ↑ to $\overline{\text{EARLY}}/\overline{\text{LATE}}$ ↓	10		65	ns	
45B	WR CLOCK ↓ to $\overline{\text{EARLY}}/\overline{\text{LATE}}$ ↓					
46A	WR CLOCK ↑ to $\overline{\text{EARLY}}/\overline{\text{LATE}}$ ↑	10		65	ns	
46B	WR CLOCK ↓ to $\overline{\text{EARLY}}/\overline{\text{LATE}}$ ↑					

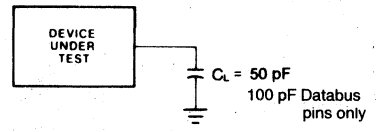


A.C. TESTING INPUT, OUTPUT WAVEFORM



AC Testing: Inputs Are Driven At 2.4V For A Logic .1, And 0.45V For A Logic .0. Timing Measurements Are Made At 2.0V For a Logic .1, And 0.8V For A Logic .0.

A.C. TESTING LOAD CIRCUIT



C_L Includes Jig Capacitance

NOTES

1. Based on WR CLOCK = 5.0 MHz
2. 24 WR CLOCK periods = 4.8 μs at 5.0 MHz.
3. 2 WR CLOCK periods ± 100 ns.
4. Previous restrictions on BRDY no longer apply. There are no restrictions on when BRDY may come. BRDY may be connected directly to BDRQ.
5. WR CLOCK Frequency = RD CLOCK Frequency ± 15%.
6. $\overline{RD}$ may be asserted before $\overline{CS}$ as long as it remains active for at least the minimum T3 pulse width after $\overline{CS}$ is asserted.
7. $\overline{WR}$ may be asserted before $\overline{CS}$ as long as it remains active for at least the minimum T11 pulse width after $\overline{CS}$ is asserted.
8. 8 WR CLOCK periods ±2 WR CLOCK periods.
9. 8 WR CLOCK periods ±1 WR CLOCK period.
10. V_{IL} = GND, V_{IH} = V_{CC}, Outputs Open.
11. V_{IL} = 0.8V, V_{IH} = 2.0V, Outputs Open.
12. WR CLOCK & RD CLOCK = DC, V_{IL} = 0V, V_{IH} = V_{CC}, all output open, $\overline{CS}$ inactive.
13. This specification is for $\overline{BCR}$ pulse width during command execution. $\overline{BCR}$ is also triggered by $\overline{RESET}$. In this case, $\overline{BCR}$ pulse width is greater than $\overline{RESET}$ pulse width.
14. 40 WR Clocks ±2.
15. Specification represents actual functionality of 82064 and WD2010. Previous datasheets contain typographical errors.

September 1989

Multimodule™ Winchester Controller Using the CHMOS 82064

Order Number: 231927-001

MULTIMODULE™ WINCHESTER CONTROLLER USING THE 82064

CONTENTS

PAGE

1.0 INTRODUCTION 5-36
1.1 ST506 Winchester Drive Overview ... 5-36

2.0 82064 WINCHESTER DISK CONTROLLER 5-37
2.1 Clock Inputs 5-37
2.2 Microprocessor Interface 5-37
2.3 Sector Buffer Control 5-39
2.4 Data Transfer Logic 5-39
2.5 CRC/ECC Generator and Checker ... 5-41
2.6 Drive Interface 5-42
2.7 Microprocessor Interfaces 5-44

3.0 PIN DESCRIPTIONS 5-46

4.0 TASK REGISTER FILE 5-48
4.1 Error Register 5-48
4.2 Reduce Write Current Register 5-49
4.3 Sector Count Register 5-49
4.4 Sector Number 5-49
4.5 Cylinder Number Low Register 5-49
4.6 Cylinder Number High Register 5-49
4.7 Sector/Drive/Head Register 5-49
4.8 Status Register 5-50
4.9 Command Register 5-50

5.0 PROGRAMMING THE 82064 5-51
5.1 Commands 5-51
5.2 Software Section: General Programming 5-59

6.0 APPLICATION EXAMPLE 5-65
6.1 iSBX Bus Multimodule Boards 5-65
6.2 The SBX82064 Design Example 5-67
6.3 Software Driver Overview 5-70

CONTENTS

PAGE

APPENDIX A: ST506 INTERFACE 5-72
Data Transfer Rate 5-72
ID Fields 5-73
Sector Interleaving 5-73
Electrical Interface 5-74

CONTENTS

PAGE

APPENDIX B: SCHEMATICS 5-76

APPENDIX C: PAL SCHEMATICS 5-80

1.0 INTRODUCTION

The 82064 Winchester Disk Controller (WDC) was developed to ease the complex task of interfacing Winchester disk drives to microprocessor systems. Specifically, the 82064 WDC interfaces to drives that conform to the ST506 specification, which is the dominant interface for 5¼ inch drives. This Application Note provides some background on the 82064 WDC, the drive interfaces and general software routines. It concludes with a design example using the 82064 WDC interfaced to the SBX™ bus. Appendix B contains the listing of the software necessary to operate this controller board.

1.1 ST506 Winchester Drive Overview

Since the 82064 WDC interfaces only to drives conforming to the ST506 specification, this overview will limit itself to those drives. A summary of the ST506 specification is shown in Appendix A for those who are not familiar with it. The ST506 Winchester Disk contains from 1 to 8 hard disks (or platters) with the average being 2 to 3 disks. These disks are made from aluminum (hence the term hard disk) and are coated with some type of recording media. The recording media is typically made of magnetic-oxide, which is similar to the material used on floppy disks and cassette tapes. Each side of a hard disk is coated with recording media and each side can store data. Each surface of a disk has its own read/write head.

Hard disk drives are sealed units because the R/W heads actually fly above the disk surface at about 8 to 20 microinches. A piece of dust or dirt, which appears as a boulder to the gap between the heads and the disk surface, will wreak havoc upon the disk media.

The R/W heads are mechanically connected together and move as a single unit across the surface of the disk. There are 2 basic methods for positioning the heads. The first is with stepper motors, which is the most common method and is also used on most floppy disk drives. These positioners are used mainly because of their low cost.

The second method of positioning the heads is to use a voice-coil mechanism. These units do not move in steps but swing across the disk. These mechanisms generally permit greater track density than steppers, but also require complex feedback electronics which increases the cost of the drive. Generally, voice-coil head positioners use closed loop servo positioning, as compared to the open loop positioning used with stepper motors.

The surface of a disk is divided logically into concentric circles radiating from the center as shown in Figure 1. Each concentric circle is called a track.

The group of tracks, all in the same position, on all of the disks (platters) in the drive is collectively called a cylinder. The number of tracks on a surface (which affects storage density) is determined by the head positioners. Typically, stepper head positioners have fewer tracks than drives that use a voice coil positioner. Which type of positioner is used is irrelevant to the 82064 as positioners are part of the drive electronics. The 82064 can access up to 2048 tracks per surface.

Once the surface is divided into cylinders it is further divided radially (as with a pie). The area between the radial spokes is referred to as a sector. The number of sectors per track is determined by many variables, but is basically determined by the number of data bytes and the length of the ID field (which locates a sector). Figure 2 shows one manufacturer's specifications for their drive. The manufacturer formats the drive with 32–256 byte sectors per track. Alternatively, the drive could be reformatted to contain 17–512 byte sectors per track. This second option has fewer sectors per track but stores more data. Determining how many bytes each sector contains is done by extensive analysis of the hardware and operating system. The 82064 WDC is programmable for sector size during formatting.

The order in which sectors are logically numbered on the track is called interleaving. An interleave factor of four would have three sectors separating logically sequential sectors. Starting at the index pulse, an example of four way interleaving is:

Sector 1, Sector X, Sector Y, Sector Z, Sector 2, Sector . . .

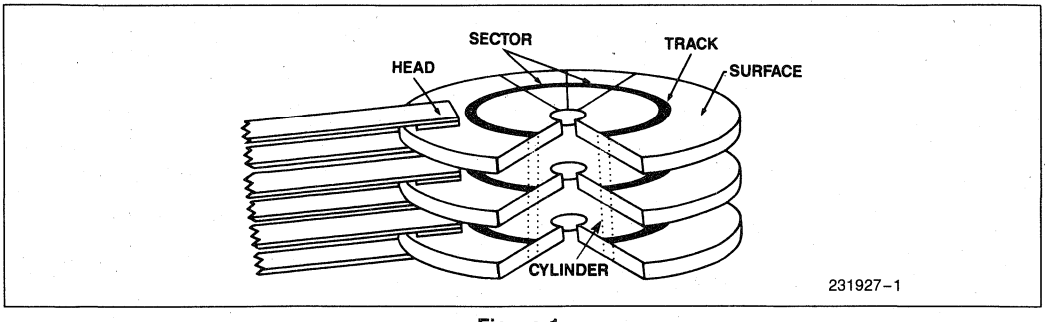


Figure 1

Capacity	
Unformatted	
Per Drive	6.38 Megabytes
Per Surface	1.59 Megabytes
Per Track	10416 Bytes
Formatted	
Per Drive	5.0 Megabytes
Per Surface	1.25 Megabytes
Per Track	8192 Bytes
Per Sector	256 Bytes
Sectors per Track	32
Transfer Rate	5.0 Megabits per second
Access Time	
Track to Track	3 ms
Average (Inc. Settle)	170 ms
Maximum (Inc. Settle)	500 ms
Settling Time	15 ms
Average Latency	8.33 ms
Functional Specifications	
Rotational speed	3600 rpm ± 1%
Recording density	7690 bpi max
Flux density	7690 fci
Track density	255 tpi
Cylinders	153
Tracks	612
R/W Heads	4
Disks	2

Figure 2. A Typical Drive Specification

Interleaving is used primarily because one sector at a time is transferred from disk to sector buffer to system RAM. This transferring of data takes time, and causes a delay between the first sector transferred and sectors that follow it. Without interleaving, the delay in transferring data would result in sectors on the disk rotating past the heads before they could be read. The operating system would then have to wait one disk revolution to get to the next sector (a 16.7 msec delay). With interleaved sectors, the next logical sector would be positioned beneath the heads after the previous sector of data had been transferred to the system RAM. Interleaving unfortunately slows down the overall transfer rate from the disk. A 5 Mbit/second transfer rate averages out to a 1.25 Mbit/second transfer rate when many sectors are transferred with four way interleaving. Again, how much interleaving to use is determined by extensive hardware/software benchmarking.

Whenever data is stored on a multiple platter disk drive, the same track on all surfaces whould be used before repositioning the heads to another track. Repositioning the heads generates a longer delay due to the mechanical delay of moving the heads. Switching to another head incurs no mechanical positioning delay. Only one head can be selected at a time.

Hard disk drives tend to be faster than floppies for two reasons. The speed at which the disk spins is about 10 times faster than the floppy (a floppy spins at 360 rpm for the popular double density disk drives). This yields an immediate one-tenth reduction in access times for the same size drive. While both ST506 drives and floppies use stepper motors, the steppers utilized by the hard disk drives are approximately twice as fast as those used by floppies.

2.0 82064 WINCHESTER DISK CONTROLLER

The 82064 WDC provides most of the functions necessary to interface between a microprocessor and an ST506 compatible disk drive. The 82064 converts the high level commands and parallel data of a microprocessor bus into ST506 compatible disk control signals and serial MFM encoded data. This section presents a detailed description of the 82064 and a discussion of various techniques which can be used to interface the 82064 to a microprocessor.

The internal structure of the 82064 is divided into several sections as shown in Figure 3. They are:

1. the microprocessor interface which includes the status and task registers;
2. sector buffer control;
3. the drive interface;
4. the data transfer section, which includes the MFM encoding/decoding of microprocessor data;
5. and CRC/ECC generation and checker.

5

2.1 Clock Inputs

The 82064 has two clock inputs: read clock (RD CLOCK) and write clock (WR CLOCK). The PLA controller, the processor interface, buffer control and MFM encoding sections operate off the WR CLOCK input. The RD CLOCK input is used only for decoding the MFM data stream. The clocks may be asynchronous to one another. Both clocks have non-TTL compatible inputs. The easiest method to interface to TTL requires a pull-up resistor to satisfy their input voltage needs. The resistor's value must be compatible with the VIL specification of these pins. See the Pin Descriptions Section for more specific information.

2.2 Microprocessor Interface

The microprocessor interface of the 82064 contains the control logic which permits commands and data to be transferred between the host and the 82064. The interface consists of an 8 bit, tri-state, bidirectional data bus; the task registers; a 3 to 8 address decoder for selecting one of the seven registers; and the general read, write, and chip select logic. Externally, the 82064 expects a buffer equal in size to a sector on the disk, and tri-state

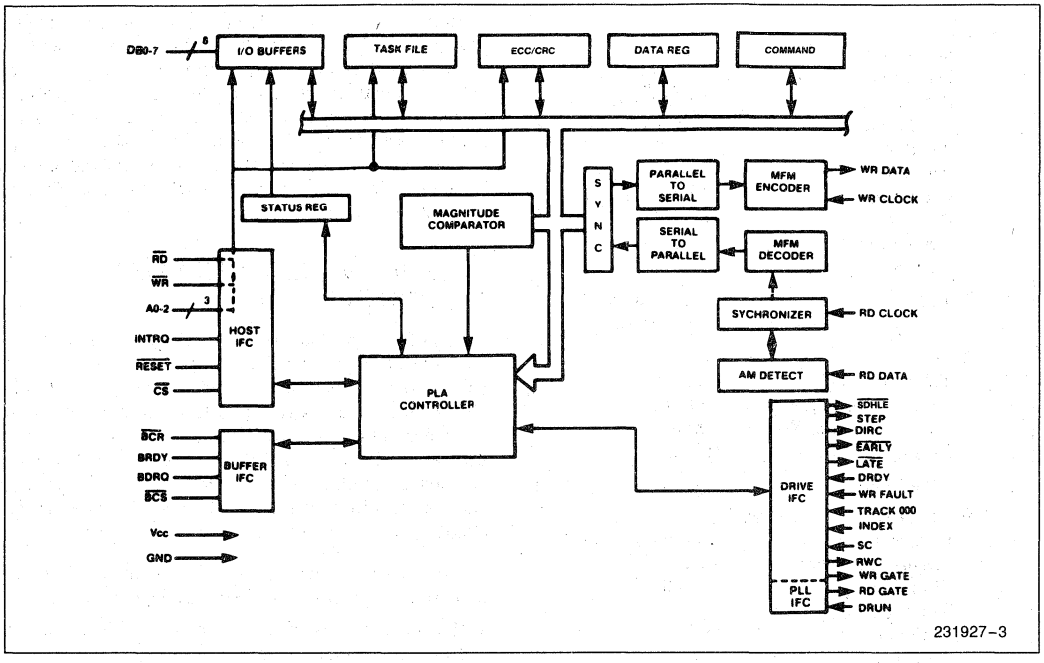


Figure 3. 82064 Internal Block Diagram

transceivers between the sector buffer and the microprocessors data bus in order to isolate itself from the microprocessor during disk data transfers.

A0–A2, Data Bus

These three address lines are active high signals and select one of the seven register locations in the 82064. They are not latched internally. If the three addresses are equal to 0 and the 82064 is selected, the data bus is kept tri-stated to ease interfacing to a sector buffer. The 82064's data bus is controlled by both the microprocessor and the 82064. The microprocessor has control for loading the registers and command. During disk reads or writes, control switches to the 82064 so that it may access the local sector buffer when transferring data between the disk and the buffer.

$\overline{RD}$, $\overline{WR}$, $\overline{CS}$

The chip select ($\overline{CS}$) is typically decoded from the higher order address lines. $\overline{CS}$ only permits data to be placed into, or read from, the 82064's task registers. Once a disk operation starts, $\overline{CS}$ no longer efffects the 82064. $\overline{RD}$ and $\overline{WR}$ are bidirectional lines and are used to read or write the 82064's registers by the host microprocessor and are valid only if $\overline{CS}$ is present. The 82064 will drive $\overline{RD}$ and $\overline{WR}$ when transferring data between the sector buffer and the disk. A signal is provided to tri-state the $\overline{RD}$ and $\overline{WR}$ lines from the host during a buffer access. This is covered in the Sector Buffer Control Section.

Interrupts

An interrupt is issued at the end of all commands, and the interrupt is cleared by reading any register. For the Read Sector command only, the 82064 allows the user the option of an interrupt either at the termination of the command, as is the case with all other commands, or when data needs to be transferred to the host from the sector buffer. This is discussed further in the Interrupt Mode Section. When selecting the data transfer option, the interrupt line will go active at the same time as the BDRQ line and the interrupt will be removed only when the proper handshake occurs with the sector buffer.

Task Registers

The Task Register File contains the command, status, track number, sector number, and other information necessary to properly execute a command. These registers are accessed with A0–A2, $\overline{RD}$ (or $\overline{WR}$), and $\overline{CS}$ being valid and are not cleared by a reset. The registers are covered in detail in the Task Register File Section.

2.3 Sector Buffer Control

The 82064 was designed to operate with an external buffer equal in size to one sector. To ease the design-in of this buffer, the 82064 provides all of the control signals it needs to operate the buffer. This buffer must be isolated from the system bus, using tri-state buffers, during disk transfers to prevent contention during the period that the 82064 is accessing the buffer. A sector buffer is generally used to ease interfacing to the system due to the high disk data rates (625 kbytes/sec), although it is not required.

$\overline{BCS}$

The Buffer Chip Select ($\overline{BCS}$) line goes active whenever the 82064 is accessing the sector buffer. This signal should remove the microprocessors ability to access the 82064 and sector buffer and must enable the sector buffer for use by the 82064.

At a 5 Mbit/sec disk data rate, the 82064 will access the buffer every 1.6 microseconds (8 bits × 200 ns/bit). $\overline{BCS}$ will remain low the entire time the 82064 is accessing the buffer. The 82064 will pulse the appropriate $\overline{RD}$ or $\overline{WR}$ line for each byte transferred.

$\overline{BCR}$

Buffer Counter Reset ($\overline{BCR}$) goes active each time that $\overline{BCS}$ changes state. Its purpose is to reset the address counter of the sector buffer back to zero before and after the 82064 uses the sector buffer. Its function is optimized for single sector transfers. Multiple sector transfers should use a software controlled buffer counter reset and not use $\overline{BCR}$ as the sector buffer will be reset to the beginning after each sector is transferred.

BDRQ, BRDY

Buffer Data Request (BDRQ) and Buffer Ready (BRDY) provide the handshake needed to transfer data between the sector buffer and the host. BDRQ signals that data must be moved to/from the sector buffer and the host. BRDY has two functions. Once the transfer signaled by BDRQ is finished, asserting BRDY will inform the 82064 that the transfer is completed and that it may finish executing the command. BRDY is also used in multiple sector commands. BRDY going high during a multiple sector transfer indicates that the buffer is full (or empty—depending upon the command) and the transfer should wait until the buffer is serviced. The sector that was being transferred will finish and the 82064 will deactivate $\overline{BCS}$ and activate BDRQ. The host microprocessor must then transfer the data between the buffer and system memory. When this transfer is finished, asserting BRDY will cause the 82064 to resume the command.

The handshaking between BDRQ and BRDY occurs only in full sector increments—not on a byte basis. A high on BDRQ indicates a full sector's worth of data is required; BRDY going high indicates a full sector of data is available to the 82064 without interruption.

Only the rising edge of BRDY is valid. A falling edge may occur at any time without effect. $\overline{BCR}$ will pulse and $\overline{BCS}$ will go active eight byte times (8 bytes × 8 bits/byte × 200 ns/bit = 12.8 microseconds) before the first data byte is transferred from the sector buffer to the disk.

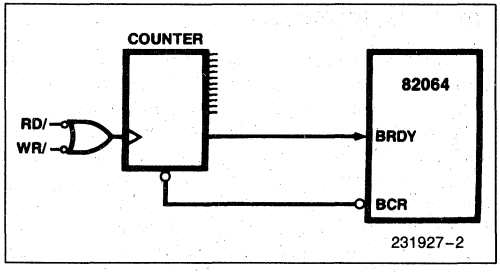


Figure 4. BRDY Generation Logic

2.4 Data Transfer Logic

This section of the 82064 is responsible for conversion of serial disk data to parallel data (and vice versa); encoding/decoding of the disk's MFM serial bit stream; and detecting the address mark.

Polled Interface

Since the 82064 isolates itself from the host during several commands, the host cannot read the status register during some periods to determine what course should be taken. In Figure 10, trying to read the status register when $\overline{BCS}$ is active will return indeterminate data. To prevent the microprocessor from reading this indeterminate data, a hardware generated "Busy" pattern should be driven onto the data bus if $\overline{BCS}$ is active. This is shown in Figure 11. The status register contains a data request (DRQ) bit whose timing is equal to the BDRQ output signal, thus making a polled operation possible. DRQ will stay set in the status register until a BRDY is generated.

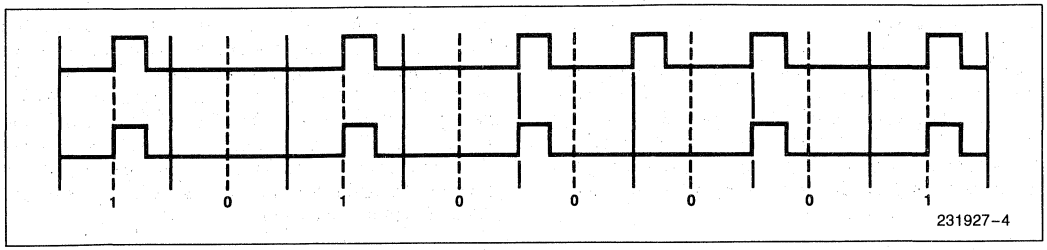


Figure 5. Data Address Mark

One design issue with the polled interface occurs when the microprocessor is polling the status and the 82064 deactivates $\overline{BCS}$. The microprocessor would normally read the hardware busy pattern. If $\overline{BCS}$ is deasserted, the hardware pattern is disabled and the microprocessor will start to read the real status register. The read cycle may almost be finished, and the read access period of the 82064 will not be satisfied. The data returned to the microprocessor will be invalid.

Interrupt Interface

There are cases where the designer does not want to tie up the microprocessor with polling. The typical 82064 design will need two interrupts per command. One for a data transfer and one for the completion of the command. The 82064 has an output to issue an interrupt when the command has finished. However for data transfers an interrupt must be generated from the BDRQ line as shown in Figure 12 (whether a DMA controller is used or not). When a data transfer is needed, the 82064 will activate the BDRQ line. The microprocessor will be interrupted and do the data transfer function. BDRQ will stay active until BRDY is generated, so the system must either use edge triggered interrupts or must not write the end-of-interrupt byte until BDRQ is removed (this is true of Intel's 8259A).

MFM Encoding/Decoding

The MFM encoding section will receive 8 bit parallel data when a valid command has been recognized and BRDY has gone high. The parallel data is first serialized and converted to an intermediate, NRZ encoded, data stream. The serial NRZ data is sent to the MFM encoding section and then transferred to the disk. Decoding of the MFM bit stream (during disk reads) happens in reverse order.

The control logic operates off the write clock (WR CLOCK) running at a frequency of the desired tranfer rate. The MFM decoding portion operates off of the read clock (RD CLOCK) input, which is supplied by an external phase lock loop. The two clocks need not be synchronized to each other. Data is written (and hence read) with the most significant bit first.

Address Mark Detector

The address mark is a unique 2 byte code written at the beginning of each ID field and data field. This address mark serves two purposes. It tells the controller what type of data is about to be received so that internal computations can be performed, and to ensure that ID fields are not sent to the host. The second purpose is to align the serial data back to the original 8 bit boundaries that existed when data was written (there are no byte boundaries on a disk).

An address mark is always preceded by the all zeros synchronization field. The 82064 starts comparing the incoming data stream when the synchronization field ends. A high speed comparator is used since the 82064 does not yet know where the proper byte boundaries are. When a proper comparison of the address mark is made the controller starts assembling bytes, starting with the second byte of the address mark.

The first byte of the address mark is an "A1" Hex, but purposely violates the MFM encoding rules by removing a clock pulse. In Figure 5, the first example is of a normal MFM encoded A1H. The second example is of the address mark and shows the missing clock pulse. The non-MFM compatible A1 is to prevent the host from issuing a similar data byte and possibly confusing detection logic.

The second byte specifies either an ID or data field and is encoded according to normal MFM rules. It is either an "F8" Hex for a data field, or "FC" through "FF" for an ID field. The different values correspond to a range of cylinders on the drive in increments of 256 tracks. The 82064 makes no use of this information, but writes it for compatibility with the ST506 specification during formatting.

PLA Control

The PLA Controller interprets command sent by the microprocessor. Its operation is synchronized to the WR CLOCK input. The PLA controller is started when a command is written into the command register. It generates control signals and operates in a handshake mode when communicating with the MFM decoding block.

Magnitude Comparator

A 10 bit magnitude comparator is used to calculate the direction and number of step pulses needed to move the head from the present cylinder position to the desired position. A separate high speed equivalence comparator is used to compare ID field bytes when searching for a sector ID field.

2.5 CRC/ECC Generator and Checker

The 82064 provides two options for protecting the integrity of the data field. The data field may have either a CRC (SDH register, bit 7 = 0), or a 32-bit ECC (SDH register, bit 7 = 1) appended to it. The ID field is always protected by a CRC.

CRC Generation/Checking

The CRC generator computes and checks the cyclic redundancy check bytes that are appended to the ID and data fields. CRC generation/checking is always done on ID fields. Data fields have a choice between 82064 CRC, internal ECC or externally supplied ECC. The CRC mode is chosen by setting bit 7 of the SDH register low. The CRC mode provides a means of verifying the accuracy of the data read from the disk, but does not attempt to correct it. The CRC generator computes and checks cyclic redundancy check characters that are written and read from the disk after the ID and data fields.

The generator polynomial for the CRC-CCITT (CRC-16) code is:

$$x16 + x12 + x5 + 1 = (x + 1)(x15 + x14 + x13 + x12 + x4 + x3 + x2 + x + 1)$$

The code's capability is as follows:

a) Detects all occurrences of an odd number of bits in error.

b) Detects all single, double, and triple bit errors if the record length (including check bits) is less than 32,767 bits.

c) Detects all single-burst errors of sixteen bits or less.

d) Detects 99.99695% of all possible 17 bit burst errors, and 99.99847% of all possible longer burst, assuming all errors are possible and equally probable.

The CRC code has some double-burst capability when used with short records (sectors). For a 256 byte sector the code will detect double-bursts as long as the total number of bits in error does not exceed 7.

If the CRC character generated while reading the data does not equal the one previously written, an error exists. If an ID field CRC error occurs the "ID not found" bit in the error register will be set. If a data field CRC error occurs the "ECC/CRC" bit in the error register will be set.

ECC Generation/Checking

The ECC mode is only applicable to the data field. It provides the user with the ability to detect and correct errors in the data field automatically. The commands and registers which must be considered when ECC is used are:

1. SDH Register, bit 7 (CRC/ECC)
2. READ SECTOR Command, bit 0 (T)
3. READ SECTOR and WRITE SECTOR Commands, bit 1 (L)
4. COMPUTE CORRECTION Command
5. SET PARAMETER Command
6. STATUS Register, bit 2—error correction successful
7. STATUS Register, bit 0—error occurred
8. ERROR Register, bit 6—uncorrectable error

To enable the ECC mode, bit 7 of the SDH register must be set to one.

Bit 0 (T) of the READ Command controls whether or not error correction is attempted. When T = 0 and an error is detected, the 82064 tries up to 10 times to correct the error. If the error is successfully corrected, bit 2 of the STATUS Register is set. The host can interrogate the status register and detect that an error occurred and was corrected. If the error was not correctable, bit 6 of the ERROR Register is set. If the correction span was set to 5 bits, the host may now execute the SET PARAMETER Command to change the correction span to 11 bits, and attempt the read again. If the error persists, the host can read the data, but it will contain errors.

When T = 1 and an error is detected, no attempt is made to correct it. Bit 0 of the STATUS Register and bit 6 of the ERROR Register are set. The user now has two choices:

1. Ignore the error and make no attempt to correct it.
2. Use the COMPUTER CORRECTION Command to determine the location and pattern of the error, and correct it within the user's program.

When the COMPUTE CORRECTION Command is implemented, it must be done before executing any command which can alter the contents of the ECC Register. The READ SECTOR, WRITE SECTOR,

SCAN ID, and FORMAT Commands will alter this register and correction will be impossible. The COMPUTE CORRECTION Command may determine that the error is uncorrectable, at which point the error bits in the STATUS and ERROR Registers are set.

Although ECC generation starts with the first bit of the F8H byte in the data ID field, the actual ECC bytes written will be the same as if the A1H byte was included. The ECC polynomial used is:

$$X^{32}+X^{28}+X^{26}+X^{19}+X^{17}+X^{10}+X^{6}+X^{2}+1$$

For automatic error correction, the external sector buffer must be implemented with a static RAM and counter, not with a FIFO.

The SET PARAMETER Command is used to select a 5-bit or 11-bit correction span.

2.6 Drive Interface

The drive interface of the 82064 contains the logic that makes possible the storage and reliable recovery of data. This interface consists of the drive and head select logic, the disk control signals, and read and write data logic as shown in Figure 6. This section describes the external circuitry which is required to complete the 82064's drive interface.

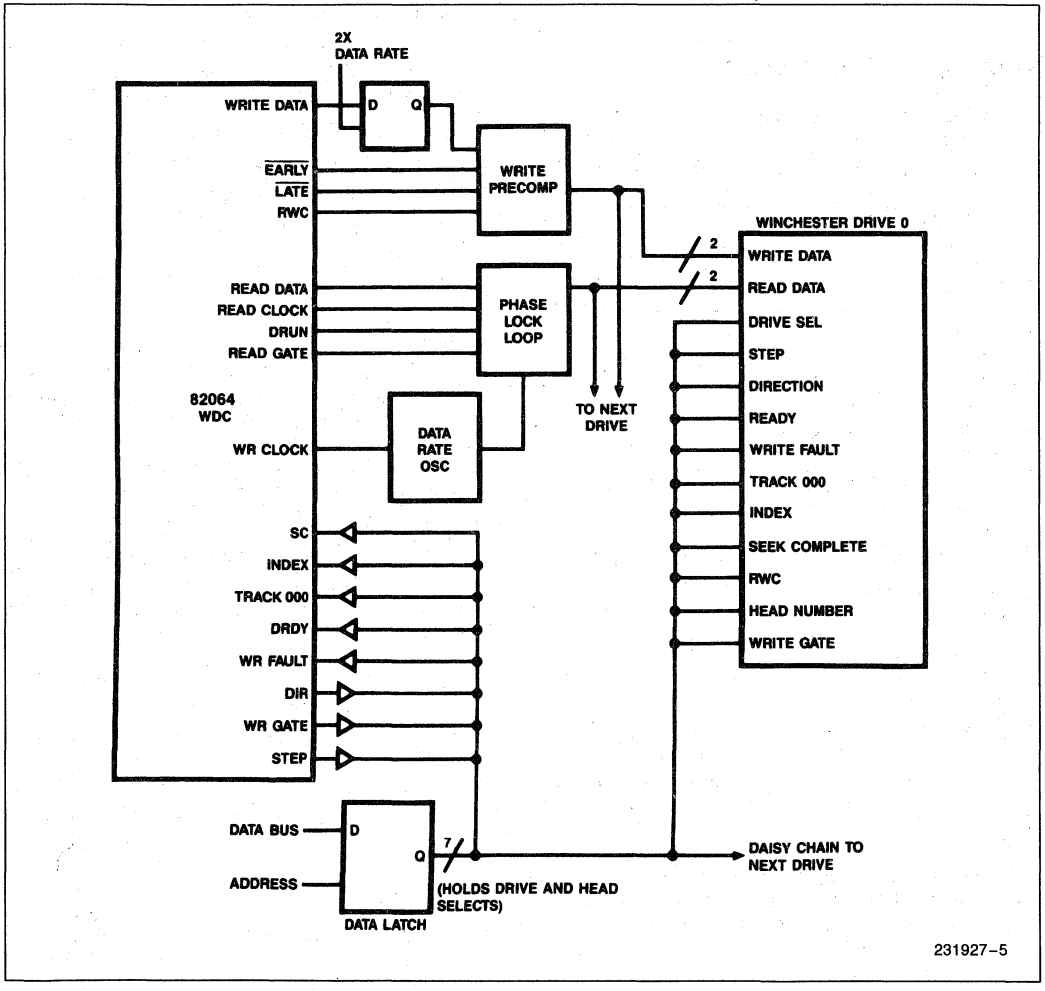


Figure 6. Drive Interface

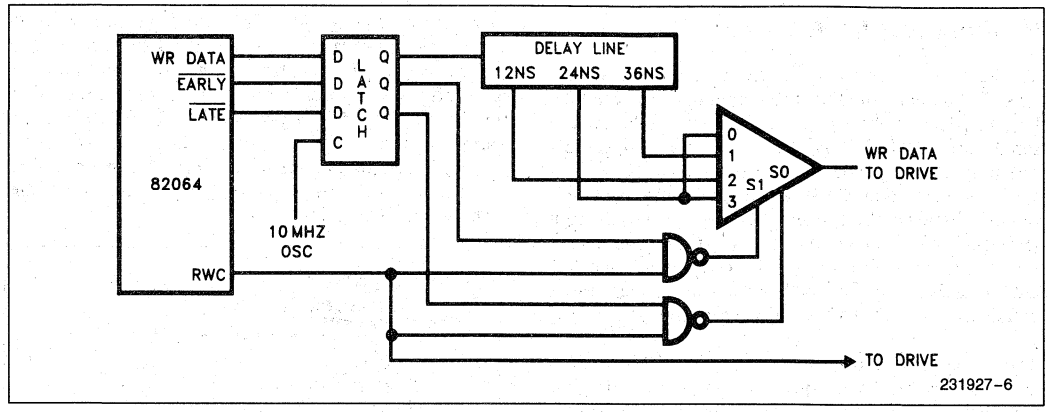


Figure 7. Write Precompensation Logic

Drive/Head Select

The 82064 has no outputs for selecting the head or drive. Therefore these signals must be generated by the user as shown in Figure 6. Data bits 0–4 should be latched whenever the SDH register is written. Bits 0–2 would then be driven onto the drive cable with open collector buffers. Bits 3 and 4 would be decoded after being latched, then buffered for the cable. The head information written to the 82064's SDH register is used to write the proper ID fields during formatting. Changing the drive bits in the SDH register will cause a Scan ID to be performed by the 82064 to update non user accessible registers.

Drive Control

The drive control (STEP, DIR, WR FAULT, TRACK 000, INDEX, SC, RWC, and WR GATE) signals are merely conditioned for transmission over the drive cable. The purpose of each pin can be found in the section on Pin Descriptions and their use in the Command Section.

WR DATA, $\overline{\text{EARLY}}$, $\overline{\text{LATE}}$

Figure 7 is a diagram of the interface required on the write data line. The final stage of the MFM encoding requires applying the WR DATA to an external flip-flop clocked at 10 MHz. The 82064 monitors the serial write data output for particular bit patterns that require precompensation to prevent bit shifting. $\overline{\text{EARLY}}$ and $\overline{\text{LATE}}$ are active on all cylinders and will normally require that RWC be factored into them to activate the data precompensation on the proper cylinder.

A delay line is required to generate the delayed data for precompensation since the actual delay varies between drive manufacturers. $\overline{\text{EARLY}}$ and $\overline{\text{LATE}}$ go active in the same clock period that generates the data bit to be shifted.

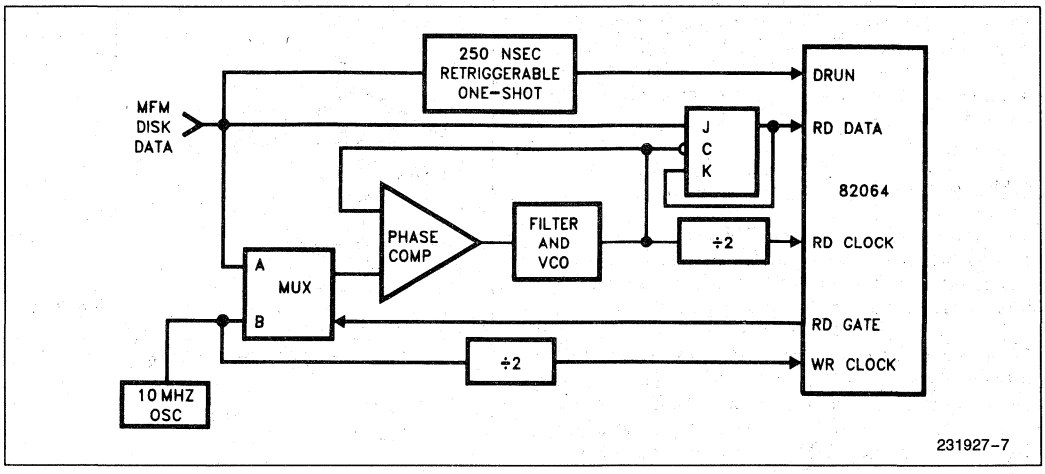


Figure 8. Data Separator Circuit

RD Data, DRUN, RD Gate

The read data interface is shown in Figure 8, and consists of the data run (DRUN) signal and a phase lock loop to generate the RD CLOCK input to decode the serial data. DRUN is generated from a retriggerable one-shot with a period just exceeding one bit cell. A sync field consisting of a string of clock pulses will continually retrigger the one-shot producing a steady high level on DRUN. The 82064 counts off 16 clock pulses internally, and if DRUN is still active, will make RD GATE active. Any byte other than an address mark will deactivate RD GATE and the sequence starts over.

The phase lock loop generates RD CLOCK which is used to decode the incoming serial data. Until RD GATE is activated by the 82064, the phase lock loop (PLL) should be locked onto a local 10 MHz clock to minimize PLL lock-up times. When RD GATE is activated, the PLL starts locking onto the incoming data stream, which should consist of the all zeros sync field. Once the PLL locks onto this synch field, the 82064 will start examining the serial data for a non-zero byte. A non-zero byte will be indicated by DRUN dropping since the address mark follows the sync field and is an "A1" Hex. This sequence is shown in Figure 9. If the address mark is detected, and if it was preceded by at least 9 bytes of zeros, RD GATE will stay active. The 82064 will then assemble bytes of data, and ensure the proper ID field is found. If a non-zero or non-address mark byte was detected, RD GATE will go inactive for a minimum of 2 byte times. If a data field or the wrong ID field is detected, or the ID field was not preceded by 8 bytes of zeros, then RD GATE goes inactive and the sequence starts over with the 82064 examining the DRUN input.

2.7 Microprocessor Interfaces

This section shows the general 82064 interfaces to a microprocessor system. There are essentially four interfaces which consist of a combination of polled, DMA, and interrupts. While the 82064 was designed to interface directly to one type, it accommodates all with minor additional logic.

DMA Interface

The 82064 is designed to use a DMA controller for data transfer between its sector buffer and the host system, and to interrupt the host when the command has finished. This interface is shown in Figure 10.

When the 82064 determines that a transfer is needed between the sector buffer and the host (either at the beginning of a command or through BRDY going active in a multiple sector transfer), it will assert BDRQ. BDRQ will initiate a DMA transfer via the DMA request input. The DMA controller will generate reads or writes which will increment an address counter. BRDY indicates that the data transfer has finished and is issued off the carry-out line (or high order address line) of the counter. The 82064 will assert BDRQ at this point and activate $\overline{BCS}$ to prevent the host from interfering with disk/buffer transfers. There can be no polling for a data transfer or a register read without an interrupt in this scheme.

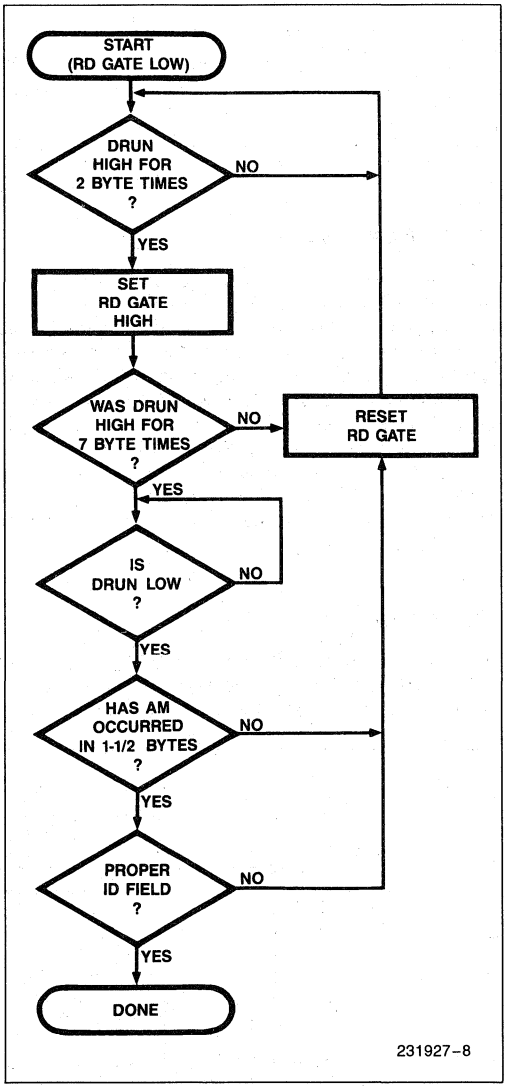


Figure 9. PLL Control Sequence

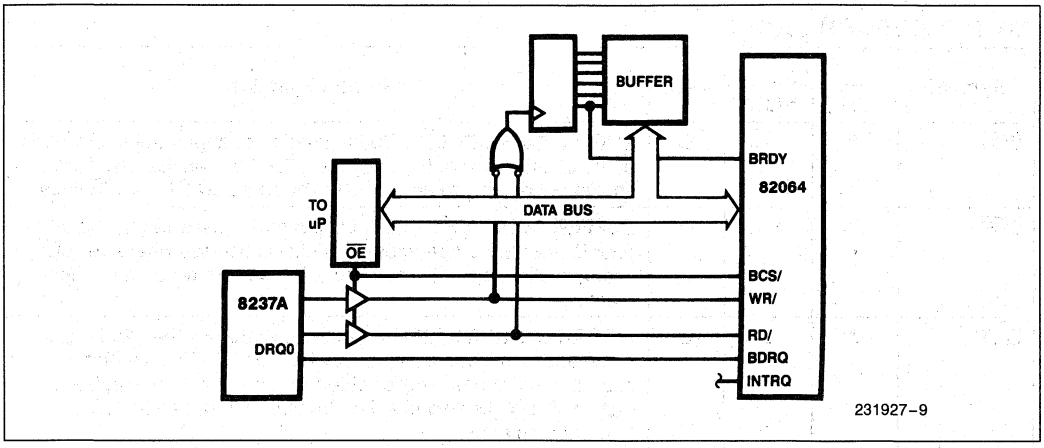


Figure 10. 82064 DMA Interface

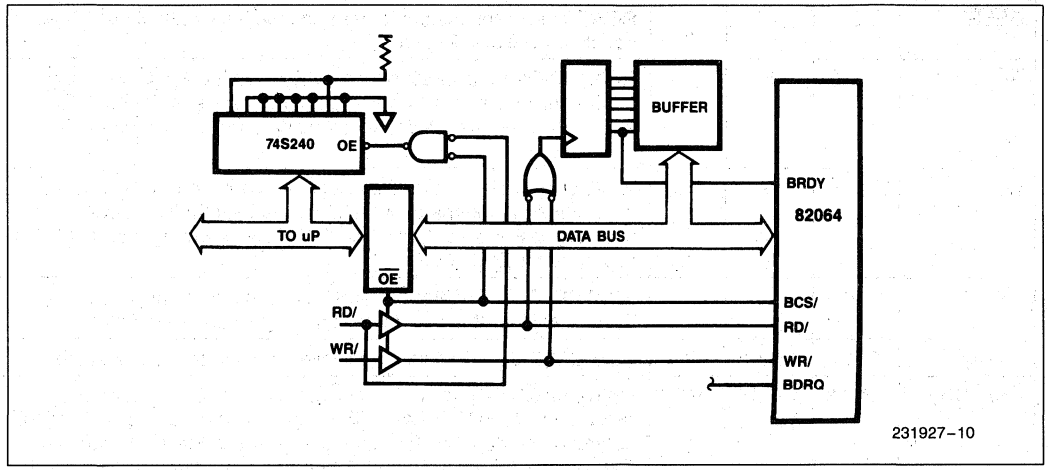


Figure 11. 82064 Polled Interface

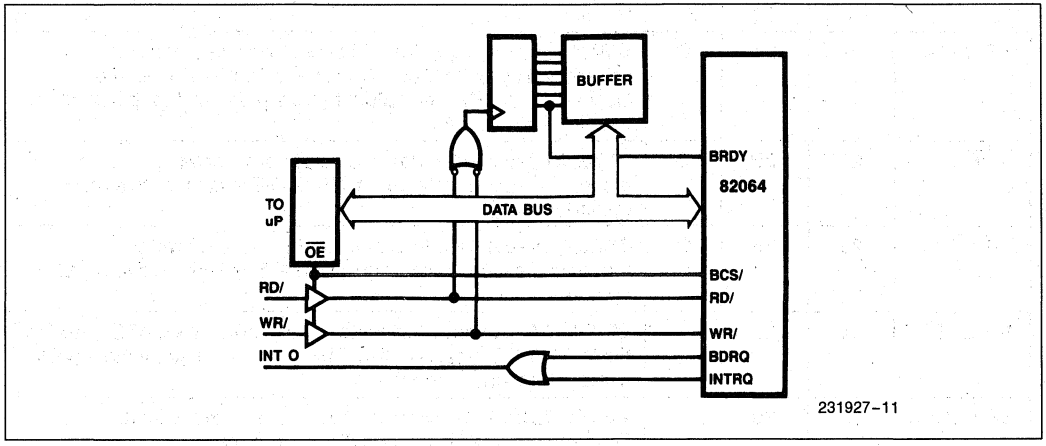


Figure 12. 82064 Interrupt Interface

3.0 PIN DESCRIPTIONS

Symbol	Pin No.		Type	Name and Function
	DIP	PLCC		
$\overline{BCS}$	1	1	O	**BUFFER CHIP SELECT:** Output used to enable reading or writing of the external sector buffer by the 82064. When low, the host should not be able to drive the 82064 data bus, $\overline{RD}$, or $\overline{WR}$ lines.
$\overline{BCR}$	2	2	O	**BUFFER COUNTER RESET:** Output that is asserted by the 82064 prior to read/write operation. This pin is asserted whenever $\overline{BCS}$ changes state. Used to reset the address counter of the buffer memory.
INTRQ	3	3	O	**INTERRUPT REQUEST:** Interrupt generated by the 82064 upon command termination. It is reset when the STATUS register is read, or a new command is written to the COMMAND register. Optionally signifies when a data transfer is required on Read Sector commands.
$\overline{SDHLE}$	4	4	O	$\overline{SDHLE}$ is asserted when the SDH register is written by the host.
$\overline{RESET}$	5	7	I	**RESET:** Initializes the controller and clears all status flags. Does not clear the Task Register File.
$\overline{RD}$	6	8	I/O	**READ:** Tri-state, bi-directional signal. As an input, $\overline{RD}$ controls the transfer of information from the 82064 registers to the host. $\overline{RD}$ is an output when the 82064 is reading data from the sector buffer ($\overline{BCS}$ low).
$\overline{WR}$	7	9	I/O	**WRITE:** Tri-state, bi-directional signal. As an input, $\overline{WR}$ controls the transfer of command or task information into the 82064 registers. $\overline{WR}$ is an output when the 82064 is writing data to the sector buffer ($\overline{BCS}$ low).
$\overline{CS}$	8	10	I	**CHIP SELECT:** Enables $\overline{RD}$ and $\overline{WR}$ as inputs for access to the Task Registers. It has no effect once a disk command starts.
A_0–A_2	9–11	11–13	I	**ADDRESS:** Used to select a register from the task register file.
DB_0–DB_7	12–19	14–16 18–22	I/O	**DATA BUS:** Tri-state, bi-directional 8-bit Data Bus with control determined by BCS. When BCS is high the microprocessor has full control of the data bus for reading and writing the Task Register File. When BCS is low the 82064 controls the data bus to transfer data to or from the buffer.
V_{SS}	20	23		Ground
WR DATA	21	24	O	**WRITE DATA:** Output that shifts out MFM data at a rate determined by Write Clock. Requires an external D flip-flop clocked at 10 MHz. The output has an active pullup and pulldown that can sink 4.8 mA.
$\overline{LATE}$	22	25	O	**LATE:** Output used to derive a delay value for write precompensation. Valid when WR GATE is high. Active on all cylinders.
$\overline{EARLY}$	23	26	O	**EARLY:** Output used to derive a delay value for write precompensation. Valid when WR GATE is high. Active on all cylinders.
WR GATE	24	27	O	**WRITE GATE:** High when write data is valid. WR GATE goes low if the WR FAULT input is active. This output is used by the drive to enable head write current.
WR CLOCK	25	29	I	**WRITE CLOCK:** Clock input used to derive the write data rate. Frequency = 5 MHz for the ST506 interface.

3.0 PIN DESCRIPTIONS (Continued)

Symbol	Pin No.		Type	Name and Function
	DIP	PLCC		
DIR	26	30	O	**DIRECTION:** High level on this output tells the drive to move the head inward (increasing cylinder number). The state of this signal is determined by the 82064's internal comparison of actual cylinder location vs. desired cylinder.
STEP	27	31	O	**STEP:** This signal is used to move the drive head to another cylinder at a programmable frequency. Pulse width = 1.6 μs for a step rate of 3.2 μs/step, and 8.4 μs for all other step rates.
DRDY	28	32	I	**DRIVE READY:** If DRDY from the drive goes low, the command will be terminated.
INDEX	29	33	I	**INDEX:** Signal from the drive indicating the beginning of a track. It is used by the 82064 during formatting, and for counting retries. Index is edge triggered. Only the rising edge is valid.
WR FAULT	30	34	I	**WRITE FAULT:** An error input to the 82064 which indicates a fault condition at the drive. If WR FAULT from the drive goes high, the command will be terminated.
TRACK 000	31	35	I	**TRACK ZERO:** Signal from the drive which indicates that the head is at the outermost cylinder. Used to verify proper completion of a RESTORE command.
SC	32	36	I	**SEEK COMPLETE:** Signal from the drive indicating to the 82064 that the drive head has settled and that reads or writes can be made. SC is edge triggered. Only the rising edge is valid.
RWC	33	37	O	**REDUCED WRITE CURRENT:** Signal goes high for all cylinder numbers above the value programmed in the Write Precomp Cylinder register. It is used by the precompensation logic and by the drive to reduce the effects of bit shifting.
DRUN	34	38	I	**DATA RUN:** This signal informs the 82064 when a field of all ones or all zeroes has been detected in the read data stream by an external one-shot. This indicates the beginning of an ID field. RD GATE is brought high when DRUN is sampled high for 16 clock periods.
BRDY	35	39	I	**BUFFER READY:** Input used to signal the controller that the buffer is ready for reading (full), or writing (empty), by the host μP. Only the rising edge indicates the condition.
BDRQ	36	40	O	**BUFFER DATA REQUEST:** Activated during Read or Write commands when a data transfer between the host and the 82064's sector buffer is required. Typically used as a DMA request line.
RD DATA	37	41	I	**READ DATA:** Single ended input that accepts MFM data from the drive.
RD GATE	38	42	O	**READ GATE:** Output that is asserted when a search for an address mark is initiated. It remains asserted until the end of the ID or data field.
RD CLOCK	39	43	I	**READ CLOCK:** Clock input derived from the external data recovery circuits.
V_{CC}	40	44	I	**D.C. POWER:** +5V.
NC	—	5, 6, 17, 28		**NO CONNECTS**

5

4.0 TASK REGISTER FILE

The Task Register File is a bank of registers used to hold parameter information pertaining to each command. These registers and their addresses are:

A2	A1	A0	READ	WRITE
0	0	0	(Bus Tri-Stated)	(Bus Tri-Stated)
0	0	1	Error Flags	Reduce Write Current
0	1	0	Sector Count	Sector Count
0	1	1	Sector Number	Sector Number
1	0	0	Cylinder Low	Cylinder Low
1	0	1	Cylinder High	Cylinder High
1	1	0	SDH	SDH
1	1	1	Status Register	Command Register

NOTE:
Registers are not cleared by $\overline{\text{RESET}}$

4.1 Error Register

This read-only register contains specific error status after the completion of a command. If any bit in this register is set, then the Error bit in the Status Register will also be set. The bits are defined as follows:

7	6	5	4	3	2	1	0
BBD	CRC/ECC	O	ID	O	AC	TK000	DM

Bit 7 - Bad Block Detect (BBD)

This bit is set when an ID field has been encountered that contains a bad block mark. The bad block bit is set only during formatting. The 82064 will terminate a command if an attempt is made to read a sector that contains this bit.

Bit 6 - CRC/ECC Data Field Error (CRC/ECC)

When in the CRC mode (SDH register, bit 7 = 0), this bit is set when a CRC error occurs in the data field. When retries are enabled, ten more attempts are made to read the sector correctly. If none of these attempts are successful bit 0 in the STATUS register is also set. If one of the attempts is successful, the CRC/ECC error bit remains set to inform the host that a marginal condition exists; however, bit 0 in the STATUS register is not set.

When in the ECC mode (SDH register, bit 7 = 1), this bit is set when the first non-zero syndrome is detected. When retries are enabled, up to ten attempts are made to correct the error. If the error is successfully corrected, this bit remains set; however, bit 2 of the STATUS register is also set to inform the host that the error has been corrected. If the error is not correctable, the CRC/ECC error bit remains set and bit 0 of the STATUS register is also set.

The data may be read even if uncorrectable errorS exist.

NOTE:
If the long mode (L) bit is set in the READ or WRITE command, no error checking is performed.

Bit 5 - Reserved.

Not used. Set to zero.

Bit 4 - ID Not Found

This bit is set to indicate that the correct cylinder, head, sector, or size parameter could not be found, or that a CRC error occurred in the ID field. Ths bit is set on the first failure and remains set even if the error is recovered on a retry. When recovery is unsuccessful, the Error bit (bit 0) of the STATUS register is also set.

For a SCAN ID command with retries enabled (T = 0), the Error bit in the STATUS register is set after ten unsuccessful attempts have been made to find the correct ID. With retries disabled (T = 1), only two attempts are made before setting the Error bit.

For a READ or WRITE command with retries enabled (T = 0), ten attempts are made to find the correct ID field. If there is still an error on the tenth try, an autoscan and auto-seek are performed. Then ten more retries are made before setting the Error bit. When retries are disabled (T = 1), only two tries are made. No autoscan or auto-seek operations are performed.

Bit 3 - Reserved.

Not used. Set to zero.

Bit 2 - Aborted Command

This bit is set if a command was issued or in progress while DRDY (Pin 28) was deasserted or WR FAULT (Pin 30) was asserted. The Aborted Command bit will also be set if an undefined command is written into the COMMAND register, but an implied seek will be executed.

Bit 1 - TRACK 000 Error (TK000)

This bit is set only by the RESTORE command. It indicates that TRACK 000 (Pin 31) has not gone active after the issuance of 2048 stepping pulses.

Bit 0 - Data Address Mark

This bit is set during a READ SECTOR command if the Data Address Mark is not found after the proper Sector ID is read.

4.2 Reduce Write Current Register

This register is used to define the cylinder number where RWC (Pin 33) is asserted:

7	6	5	4	3	2	1	0
CYLINDER NUMBER / 4							

The value (0–255) written into this register is internally multiplied by 4 to specify the actual cylinder where RWC is asserted. Thus a value of 01H will cause RWC to activate on cylinder 4, 02H on cylinder 8 and so on. RWC will be asserted when the present cylinder is greater than or equal to the cylinder indicated by this register. For example, one ST506 compatible drive requires precompensation on cylinder 128 (80H) and above. Therefore the REDUCE WRITE CURRENT register should be loaded with 32 (20H). A value of FFH will keep the RWC output inactive regardless of the actual cylinder number.

4.3 Sector Count Register

This register is used to define the number of sectors that need to be transferred to the buffer during a READ MULTIPLE SECTOR or WRITE MULTIPLE SECTOR command.

7	6	5	4	3	2	1	0
# OF SECTORS							

The value contained in the register is decremented after each sector is transferred to/from the sector buffer. A zero represents a 256 sector transfer, a one a 1 sector transfer, etc. This register is ignored when single sector commands are specified in the Command register.

4.4 Sector Number

This register holds the sector number of the desired sector:

7	6	5	4	3	2	1	0
SECTOR NUMBER							

For a multiple sector command it specifies the first sector to be transferred. It is decremented after each sector is transferred to/from the sector buffer. The SECTOR NUMBER register may contain any value from 0 to 255. The ID Not Found bit will be set if the desired sector cannot be located on the track.

The SECTOR NUMBER register is also used to program the Gap 1 and Gap 3 lengths to be used when formatting a disk. See the WRITE FORMAT command description for further explanation.

4.5 Cylinder Number Low Register

This register holds the lower byte of the desired cylinder number:

7	6	5	4	3	2	1	0
LS BYTE OF CYLINDER NUMBER							

It is used in conjunction with the CYLINDER NUMBER HIGH register to specify a range of 0 to 2048 tracks.

4.6 Cylinder Number High Register

This register holds the three most significant bits of the desired cylinder number:

7	6	5	4	3	2	1	0
x	x	x	x	x	(10)	(9)	(8)

x = ignored

The 82064 contains a pair of registers that store the actual position where the R/W head are located. The CYLINDER NUMBER HIGH and LOW registers are considered the cylinder destination registers for seeks and other commands. The 82064 compares its internal registers to the destination registers and issues the number of steps in the right direction to make both sets of registers equal. After a command is executed, the internal cylinder position registers' contents are equal to the cylinder high/low registers. If a drive number change is detected on a new command, the 82064 automatically reads an ID field to update its internal cylinder position registers. This affects all commands except a RESTORE.

When a RESTORE command is executed, the internal head location registers are reset to zero while DIR and STEP move the heads to track zero.

4.7 Sector/Drive/Head (SDH) Register

The SDH register contains the desired sector size, drive number, and head number parameters. The format is shown below.

7	6	5	4	3	2	1	0
EXT	SECT SIZE		DRIVE		HEAD #		

Both head number and sector size are compared against the disk's ID field. Head select and drive select lines are not available as outputs from the 82064 and must be generated externally.

Bit 7, the extension bit (EXT), is used to select between the CRC or ECC mode. When bit 7 = 1, the ECC mode is selected for the data field. When bit 7 = 0, the CRC mode is selected. The CRC is checked on the ID field regardless of the state of EXT. The SDH byte written into the ID field is different than the SDH Register contents. The recorded SDH byte does not have the drive number (DRIVE) written but does have the BAD BLOCK mark written.

Note that use of the extension bit requires the gap lengths to be modified as described in the WRITE FORMAT command description.

4.8 Status Register

The status register is a read-only register which informs the host of certain events. This register is a flow-through latch until the microprocessor reads it at which point the drive status lines are latched. The INTRQ line will be reset when this register is read. The format is:

7	6	5	4	3	2	1	0
BUSY	READY	WF	SC	DRQ	DWC	CIP	ERROR

Bit 7 - Busy

This bit is asserted when a command is written into the COMMAND register and, except for the READ command, is deasserted at the end of the command. When executing a READ command, Busy will be deasserted when the sector buffer is full. Commands should not be loaded into the COMMAND register when Busy is set. When the Busy bit is set, no other bits in the STATUS or ERROR registers are valid.

During other non-data transfer commands, Busy should be ignored as it will go active for short periods.

Bit 6 - Ready

This bit reflects the state of the DRDY (Pin 28) line at the time the microprocessor reads the status register. Transitions on the DRDY line will abort a command and set the aborted command bit in the error register.

Bit 5 - Write Fault

This bit reflects the state of the WR FAULT (Pin 30) line. Transitions on this line will abort a command and set the aborted command bit in the error register.

Bit 4 - Seek Complete

This bit reflects the state of the SC (Pin 32) line. Commands which initiate a seek will pause until Seek Complete is set. This bit is latched after an aborted command error.

Bit 3 - Data Request

The Data request bit (DRQ) reflects the state of the BDRQ (Pin 36) line. It is set when the sector buffer should be loaded with data or read by the host processor, depending upon the command. The DRQ bit and the BDRQ line remain high until BRDY indicates that the sector buffer has been filled or emptied, depending upon the command. BRDQ can be used for DMA.

Bit 2 - Data Was Corrected (DWC)

When set, this bit indicates that an ECC error has been detected during a read operation, and that the data in the sector buffer has been corrected. This provides the user with an indication that there may be a marginal condition within the drive before the errors become uncorrectable. This bit is forced to zero when not in the ECC mode.

Bit 1 - Command in Progress

When this bit is set, a command is being executed and a new command should not be loaded until it is cleared. Although a command may be executing, the sector buffer is still available for access by the host processor. When the 82064 is no longer busy (bit 7 = 0) the status register can be read. If CIP is set, only the status register can be read regardless of which register is selected.

Bit 0 - Error

This bit is a logical OR of the contents of the error register. Any bit being set in the error register sets this bit. The host must read the ERROR register to determine what type of error occurred. This bit is cleared when a new command is loaded.

4.9 Command Register

This write-only register is loaded with the desired command:

7	6	5	4	3	2	1	0
COMMAND							

The 82064 begins to execute immediately upon loading any value into this register. This register should not be written while the Busy or Command in Progress bits are set in the STATUS register. The INTRQ line (Pin 3) if set, will be cleared by a write to the COMMAND register.

Instruction Set

The 82064 WDC instruction set contains six commands. Prior to loading the command register, the host processor must first set up the Task Register File with the information needed for the command. Except for

the COMMAND register, the registers may be loaded in any order. If a command is in progress, a subsequent write to the COMMAND register will be ignored. A command is finished when the command in progress (CIP) bit in the STATUS register is cleared. See the Command Section for an explanation of each command.

COMMAND	7	6	5	4	3	2	1	
RESTORE	0	0	0	1	R3	R2	R1	R0
SEEK	0	1	1	1	R3	R2	R1	R0
READ SECTOR	0	0	1	0	I	M	0	T
WRITE SECTOR	0	0	1	1	0	M	0	T
SCAN ID	0	1	0	0	0	0	0	T
WRITE FORMAT	0	1	0	1	0	0	0	0
COMPUTE CORRECTION	0	0	0	0	1	0	0	0
SET PARAMETER	0	0	0	0	0	0	0	S

R 3–0 = Rate Field

For 5 MHz WR Clock:

0000 — ≈35 μs
0001 — 0.5 ms
0010 — 1.0 ms
0011 — 1.5 ms
0100 — 2.0 ms
0101 — 2.5 ms
0110 — 3.0 ms
0111 — 3.5 ms
1000 — 4.0 ms
1001 — 4.5 ms
1010 — 5.0 ms
1011 — 5.5 ms
1100 — 6.0 ms
1101 — 6.5 ms
1110 — 3.2 μs
1111 — 16 μs

COMMAND	7	6	5	4	3	2	1
T = Retry Enable							
T = 0 Enable Retries T = 1 Disable Retries							
M = Multiple Sector Flag							
M = 0 Transfer 1 Sector M = 1 Transfer Multiple Sectors							
I = Interrupt Enable							
I = 0 Interrupt at BDRQ time I = 1 Interrupt at end of command							
S = Error Correction Span							
S = 0 5-bit Span S = 1 11-bit Span							

5.0 PROGRAMMING THE 82064

This section consists of two parts. The first part gives an explanation of each command, a flowchart showing the 82064's sequence of events, and the commands' sequence of events as seen by the host microprocessor. The second section shows flowcharts of general software routines and their PLM equivalent, for both polled and interrupt driven software.

The designer must remember that the 82064 expects a full sector buffer that can be isolated from the host during data transfers between the 82064 and the disk. Since the 82064 assumes a full sector buffer is available, it does not check for data overrun or underrun error conditions. If such a condition occurs, corruption of data will happen and the host will have no indication of an error. The design must guarantee against over-run and under-run conditions when not using the sector buffer approach.

5.1 Commands

A command is placed into the command register only after the Task Registers have been written with proper values. The Task Registers may be loaded in any order. A command, once started, can only be terminated by a hardware reset to the 82064. This may corrupt data on the disk by removing necessary control signals out of sequence.

The general sequence of a command is as follows:

— The host loads the Task Registers
— The host loads the Command Register
— The 82064 locates the correct cylinder
— Data transfer takes place
— The 82064 issues an interrupt

Restore Command – 0 0 0 1 R3 R2 R1 R0

The Restore command is used to position the heads to track 0. This command is usually issued to the 82064 on power-up to initialize internal registers. The user specified rate field (R3–R0) is stored internally for FUTURE use in commands with implied seeks.

The step rate value is not used with this command. The actual stepping rate used is dependent upon the handshake delay between the 82064 issuing a step pulse and the drive returning a seek complete for each track. After each step pulse is issued, the 82064 waits for a rising edge on the Seek Complete (SC) line before issuing the next pulse. If 8 index pulses are received without a rising edge on SC, the 82064 will switch to sampling the level of the SC line. If after 2048 step pulses the Track 00 signal has not gone active, the 82064 will terminate

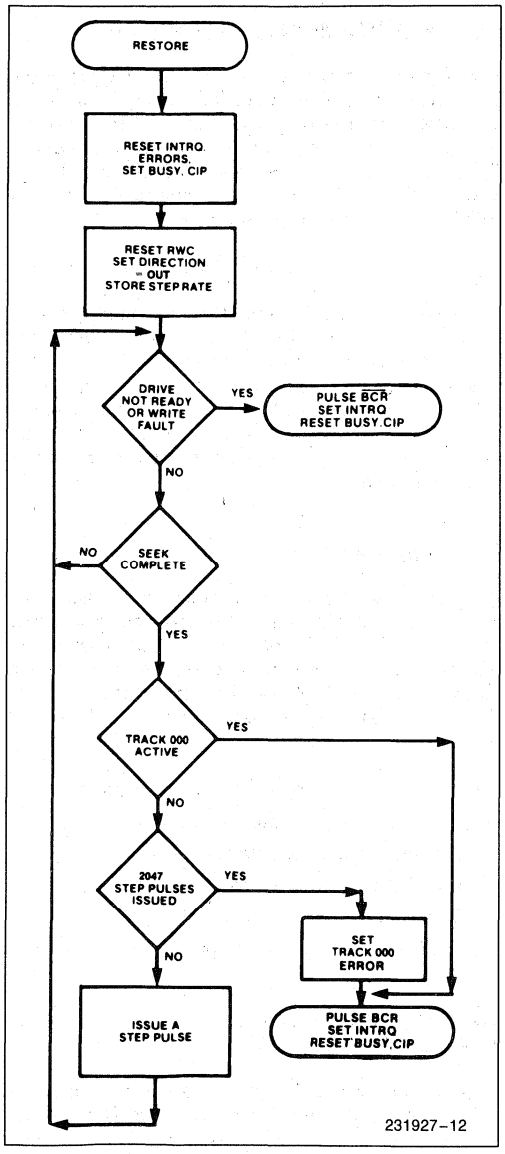


Figure 13. Restore Command Flow

the command, assert INTRQ and set the TRACK 000 bit in the Error Register. The command will terminate if WR Fault goes active or DRDY goes inactive at any time. Figure 13 is a flow chart of the command.

Seek Command – 0 1 1 1 R3 R2 R1 R0

The Seek command positions the heads to the cylinder specified in the Task Registers. The direction and number of step pulses issued is calculated by comparing the cylinder high/low registers to an internal "present position" cylinder register. The present position register is updated after all step pulses are issued and the command is terminated.

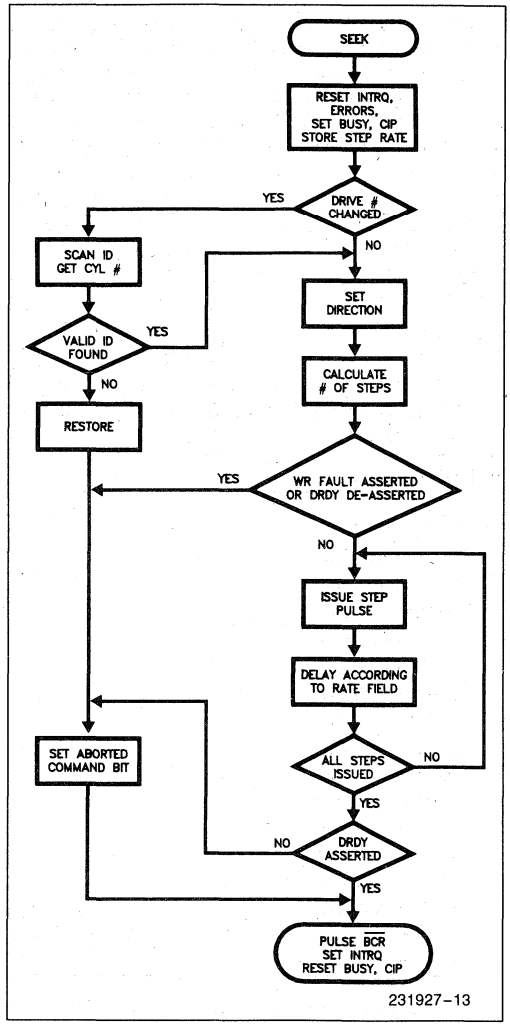


Figure 14. Seek Command Flow

The actual stepping rate is taken from the rate field bits (R3–R0) and stored for future use. The command terminates at once if WR FAULT goes active or DRDY goes inactive at any time. Figure 14 is a flowchart of the command.

Since the data transfer commands feature implied seeks, this command is of use mainly to those using multiple drives and software that can take advantage of overlapped seeks.

Scan ID Command – 0 1 0 0 0 0 0 T

The Scan ID command is used by both the 82064 and the host to update the SDH, the Sector Number, Cylinder and internal present position registers. Once the command is issued, the Seek Complete line is sampled until valid. The first ID field found, as indicated by the address mark, is loaded into the previously mentioned registers. The Bad Block bit will be set if detected, and the command will terminate. ID CRC errors will start the search sequence over for a maximum of 10 index pulses, but the registers will be loaded with whatever data the 82064 had perceived as ID information. Improper states on WR Fault on DRDY will terminate the command. Figure 15 is the flow chart of the command.

The main use for this command is to determine where the heads are currently located and what size the sectors are (i.e. 256, 512 etc.). Without this command, it would be necessary to recall the heads to track zero and then step out to the desired cylinder each time a drive was changed. Specifying the wrong sector size would yield an ID not found error. This command enables the system to read the disk drive to determine what size sectors were recorded.

Read Sector Command – 0 0 1 0 I M 0 T

The READ SECTOR command is used to transfer one or more sectors of data from the disk to the sector buffer. Upon receipt of the READ SECTOR command, the 82064 checks the CYLINDER NUMBER LOW/HIGH register pair against an internal cylinder position register to see if they are equal. If not, the direction and number of steps are calculated and a seek takes place. If an implied seek is performed, the 82064 will search until a rising edge of SC is received. The WR FAULT and DRDY lines are monitored throughout the command.

Once the Seek Complete (SC) line is high (with or without an implied seek having occurred), the search for an ID field begins. If T = 0 (retries enabled), the 82064 must find an ID with the correct cylinder number, head, sector size, and CRC within 10 revolutions, or a Scan ID and re-Seek will be performed. The search for the proper ID will again be tried for up to 10 revolutions. If the correct sector is still not found, the appropriate error bits will be set and the command terminated. Data CRC errors will also be retried for up to 10 revolutions (if T = 0).

If T = 1 (retries disabled), the ID search must find the correct sector within 2 revolutions or the appropriate error bits will be set and the command terminated.

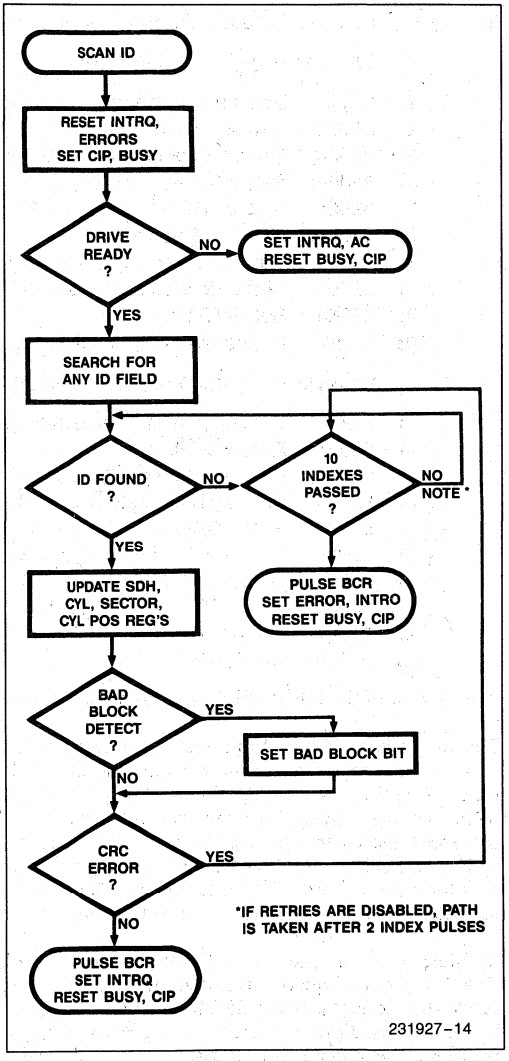


Figure 15. Scan ID Command Flow

Both the READ SECTOR and WRITE SECTOR commands feature a "simulated completion" to ease programming. DRQ/BDRQ will be generated upon detecting an error condition. This allows the same program flow for successful or unsuccessful completion of a command.

When the data address mark is found, the 82064 is ready to tranfer data to the sector buffer. After the data has been transferred, the I bit is checked. If I = 0, INTRQ is made active coincident with BDRQ, indicating that a transfer of data from the buffer to the host processor is required. If I = 1, INTRQ will occur at the end of the command, i.e. after the buffer is unloaded by the host.

5

In summary then, READ SECTOR operation is as follows:

When M = 0 (READ SECTOR)

(1) Host: Sets up parameters; issues READ SECTOR command.
(2) 82064: Strobes $\overline{BCR}$.
(3) 82064: Finds sector specified; asserts $\overline{BCR}$ and $\overline{BCS}$; transfers data to buffer.
(4) 82064: Sets $\overline{BCR}$ = 1, $\overline{BCS}$ = 0.
(5) 82064: Sets BDRQ = 1; DRQ = 1.
(6) 82064: If I bit = 1 go to (9).
(7) Host: Reads contents of sector buffer.
(8) 82064: Waits for BRDY, then sets INTRQ = 1: END.
(9) 82064: Sets INTRQ = 1.
(10) Host Reads out contents of buffer; END.

When M = 1 (READ MULTIPLE SECTOR)

(1) Host: Sets up parameters; issues READ SECTOR command.
(2) 82064: Assert $\overline{BCR}$.
(3) 82064: Finds sector specified; asserts $\overline{BCR}$ and $\overline{BCS}$; transfers data to buffer.
(4) 82064: Strobes $\overline{BCR}$; sets $\overline{BCS}$ = 0.
(5) 82064: Sets BDRQ = 1; DRQ = 1.
(6) Host: Reads out contents of buffer.
(7) 82064: Waits for BRDY; Decrements SECTOR COUNT; increments SECTOR NUMBER.
(8) 82064: When BRDY = 1, if Sector Count = 0 then go to (10).
(9) 82064: Go to (2).
(10) 82064: Set INTRQ = 1; End.

A flowchart of the READ SECTOR command is shown in Figures 16A and 16B.

The M bit is set for multiple sector transfers. When M = 0, one sector is transferred and the SECTOR COUNT register is ignored. When M = 1, multiple sectors are transferred. After each sector is transferred, the 82064 decrements the SECTOR COUNT register and increments the SECTOR NUMBER register. The next logical sector will be transferred regardless of any interleave. Sectors are numbered at format time.

Multiple sector transfers continue until the SECTOR COUNT register equals zero, or the BRDY line goes active (low to high). If the SECTOR COUNT register is non-zero (indicating more sectors are to be transferred but the buffer is full), BDRQ will be made active and the host must unload the buffer. After this occurs, the buffer will again be free to accept the remaining sectors from the 82064. This scheme enables the user to transfer more sectors than the buffer memory has capacity for.

Write Sector Command – 0 1 1 1 0 M 0 T

The WRITE SECTOR command is used to write one or more sectors of data to the disk from the sector buffer. Upon receipt of a WRITE SECTOR command the 82064 checks the CYLINDER NUMBER LOW/HIGH register pair against the internal cylinder position register to see if they are equal. If not, the direction and number of steps calculation is performed and a seek takes place. The WR FAULT and DRDY lines are checked throughout the command.

When the Seek Complete (SC) line is found to be true (with or without an implied seek having occurred), the BDRQ signal is made active and the host proceeds to load the buffer. Once BRDY goes high, the ID field with the specified cylinder number, head, and sector size is searched for. Once found, WR GATE is made active and the data is written to the disk. If retries are enabled (T = 0), and if the ID field cannot be found within 10 revolutions, a Scan ID and re-Seek are performed. If the correct ID field is not found within 10 additional revolutions, the ID Not Found error bit is set and the command is terminated. If retries are disabled, (T = 1) and if the ID field cannot be found within 2 revolutions, the ID Not Found error bit is set and the command is terminated.

During a WRITE MULTIPLE SECTOR command (M = 1), the SECTOR NUMBER register is decremented and the SECTOR COUNT register is incremented after the transfer to the disk takes place. During multiple sector transfers if BRDY is asserted after the first sector is transferred from the buffer, the 82064 will transfer the next sector before issuing BDRQ. The 82064 will set BDRQ and wait for the host processor to place more data in the buffer.

In summary then, the WRITE SECTOR operation is as follows:

When M = 0, 1 (WRITE SECTOR)

(1)	Host:	Sets up parameters; issues WRITE SECTOR command.
(2)	82064:	Sets BDRQ = 1, DRQ = 1.
(3)	Host:	Loads sector buffer with data.
(4)	82064:	Waits for BRDY = 0 to 1.
(5)	82064:	Finds specified ID field; writes sector to disk.
(6)	82064:	If M = 0, then set INTRQ = 1; END.
(7)	82064:	Increment SECTOR NUMBER register; decrement SECTOR COUNT register.
(8)	82064:	If SECTOR = 0, then set INTRQ = 1; END.
(9)	82064:	Go to (2).

A flowchart of the WRITE SECTOR command is shown in Figure 17.

Write Format Command 01010000

The WRITE FORMAT command is used to format one track using the Task Register File and the sector buffer. During execution of this command, the sector buffer is used for additional parameter information instead of sector data. Shown in Figure 18 is the contents of the sector buffer for a 32 sector/track format with an interleave factor of two. Each sector requires a two byte sequence. The first byte designates whether a bad block mark is to be recorded in the sector's ID field. A 00 Hex is normal; an 80H indicates a bad block mark for the sector. In the example of Figure 18, sector 04 will get a back block mark recorded. Any attempt to access sector 4 in the future will terminate the command.

The second byte indicates the logical sector number to be recorded. This allows sectors to be recorded with any interleave factor desired. The remaining memory in the sector buffer may contain any value. Its only purpose is to generate a BRDY to tell the 82064 to begin formatting the track. An implied seek is in effect on this command. As for other commands, if the drive number has been changed an ID field will be scanned for cylinder position information before the implied seek is performed. If no ID field can be read (because the track had been erased or because an incomplete format had been used), an ID Not Found error will result and the WRITE FORMAT command will be aborted. This can be avoided by issuing a RESTORE command before formatting.

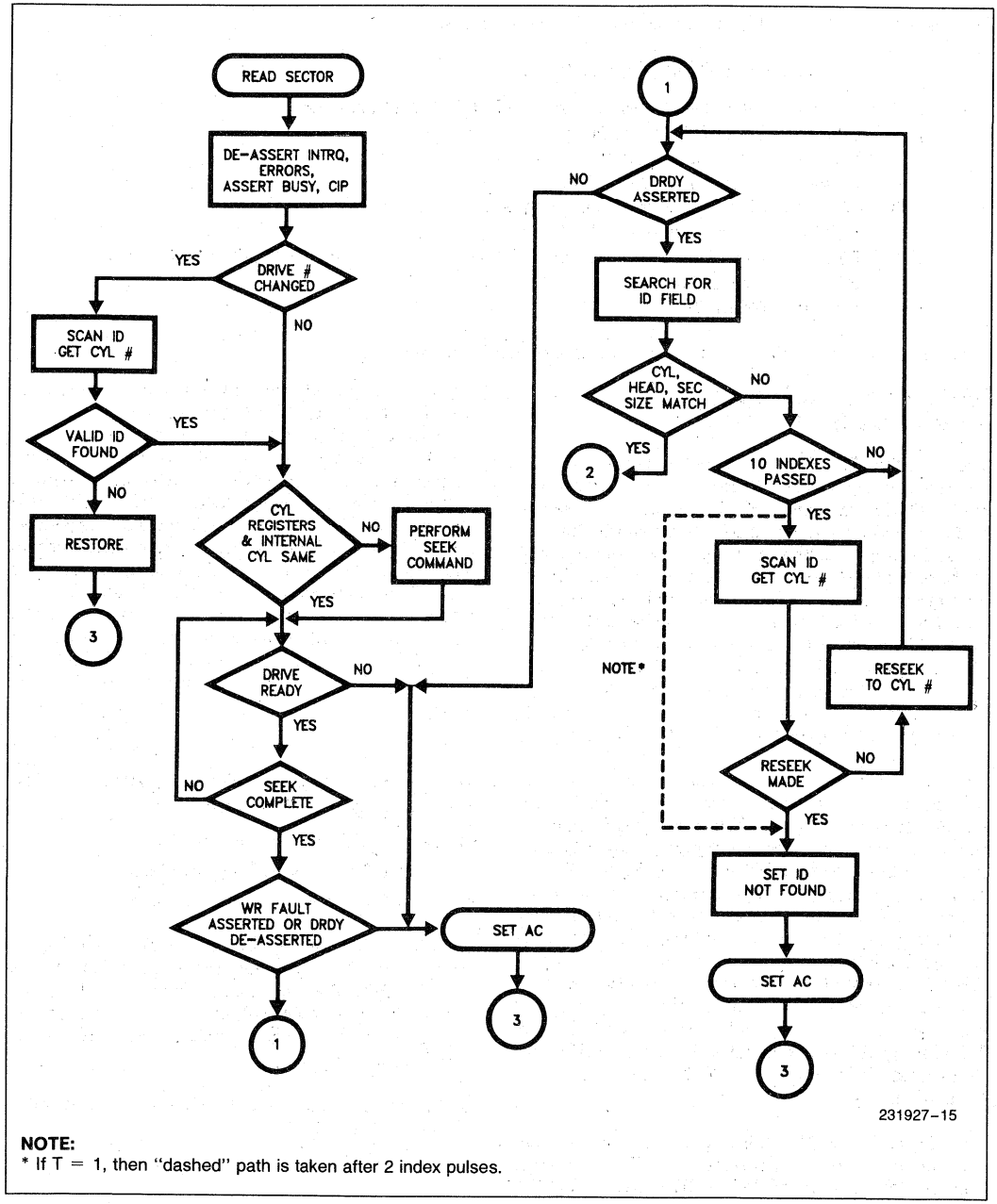
The SECTOR COUNT register is used to hold the total number of sectors to be formatted (01H = 1 sector; 00H = 256 sectors), while the SECTOR NUMBER register holds the number of bytes (minus three) to be used for Gap 1 and Gap 3. For instance, if the SECTOR COUNT register value is 02H and the SECTOR NUMBER register value is 00H, then 2 sectors are written on a track and 3 bytes of 4EH are written for Gap 1 and Gap 3. The data fields are filled with FFH and the CRC is automatically generated and appended. All gaps are filled with 4EH. After the last sector is written, the track is filled with 4EH until the index pulse terminates the write. The Gap 3 value is determined by the drive motor speed variation, data sector length, and the interleave factor. The interleave factor is only important when 1:1 (no) interleave is used. The formula for determining the minimum Gap 3 length value is:

$$\text{Gap 3} = (2 * M * S) + K + E$$

M	= motor speed variation (e.g., 0.03 for ±3%)
S	= sector length in bytes
K	= 25 for interleave factor of 1
K	= 0 for any other interleave factor
E	= 7 if the sector is to be extended

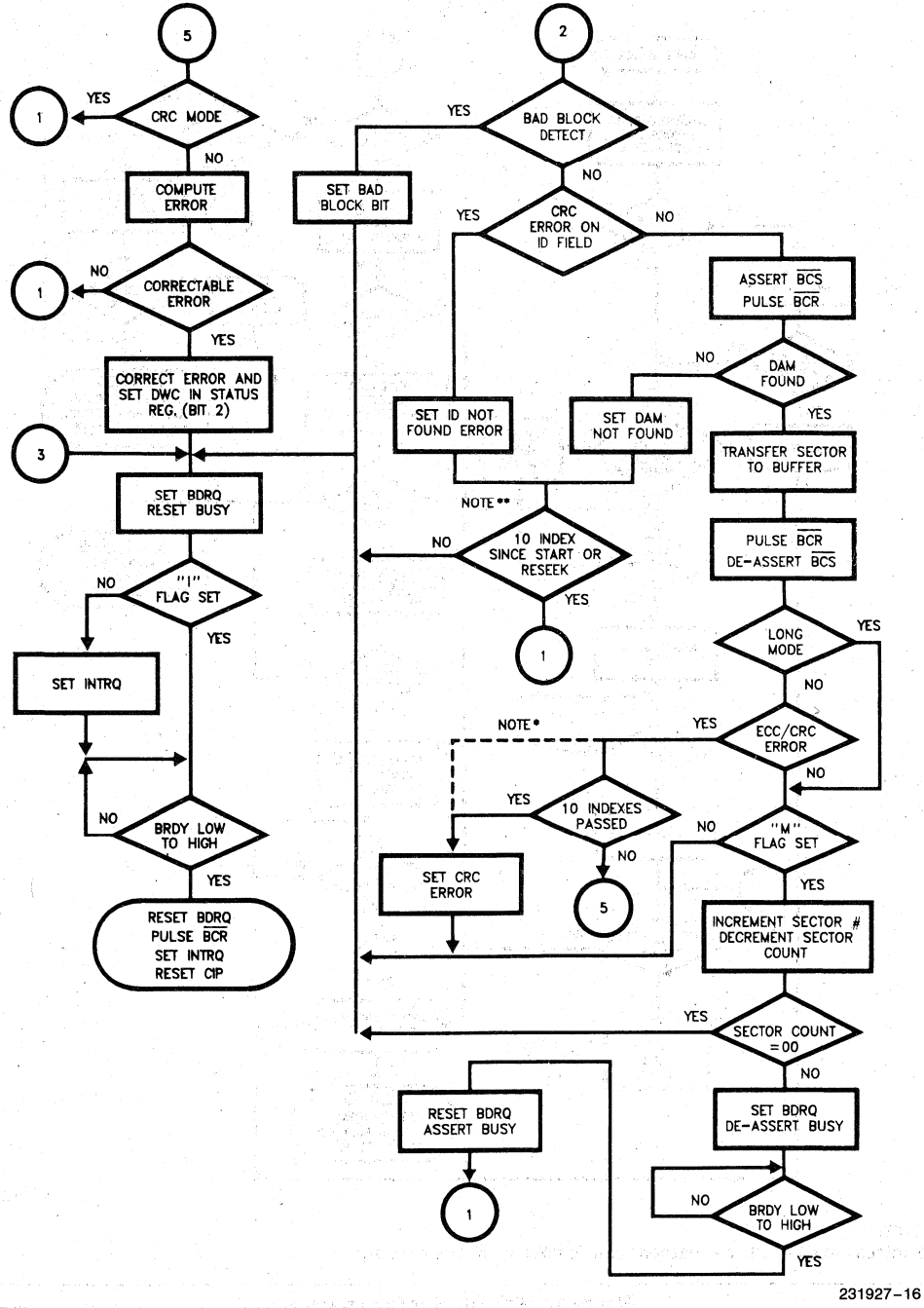
As with all commands, a WR FAULT or drive not ready condition, will terminate execution of the WRITE FORMAT command. Figure 19 shows the format that the 82064 will write on the disk. The extend bit in the SDH register must not be set during the Format command.

A flowchart of the WRITE FORMAT command is shown in Figure 20.



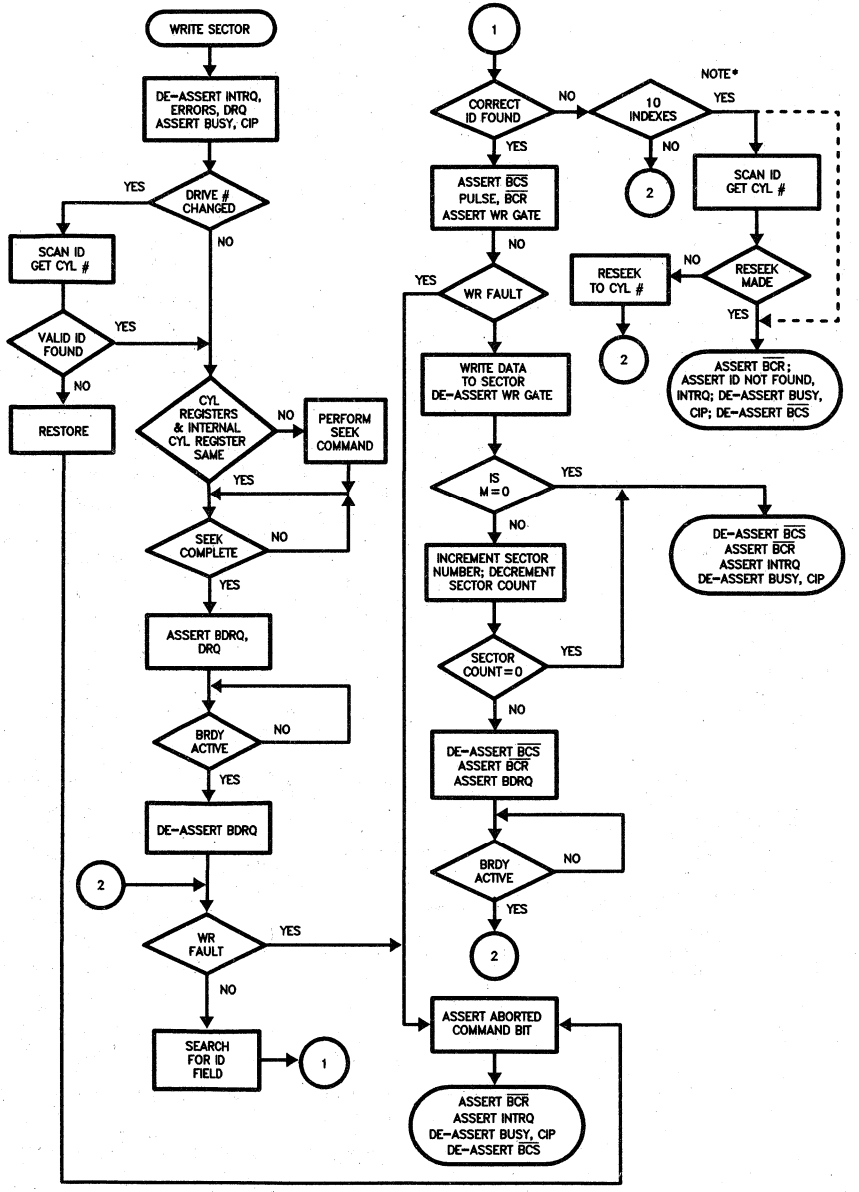
NOTE:
* If T = 1, then "dashed" path is taken after 2 index pulses.

Figure 16A. Read Sector Command Flow



* If T bit of command = 1, then dashed path is taken.
** If T bit of command = 1, then test is for 2 index pulses.

Figure 16B. Read Sector Command Flow (Continued)



231927-17

NOTE:
*If retries are disabled, the "dashed" path is taken after 2 index pulses.

Figure 17. Write Sector Command Flow

00	00	00	10	00	01	00	11	00	02	00	12	00	03	00	13
80	04	00	14	00	05	00	15	00	06	00	15	00	07	00	17
00	08	00	18	00	09	00	19	00	0A	00	19	00	0B	00	1B
00	0C	00	1C	00	0D	00	1D	00	0E	00	1E	00	0F	00	1F
FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 18. Sector Buffer Contents For Format

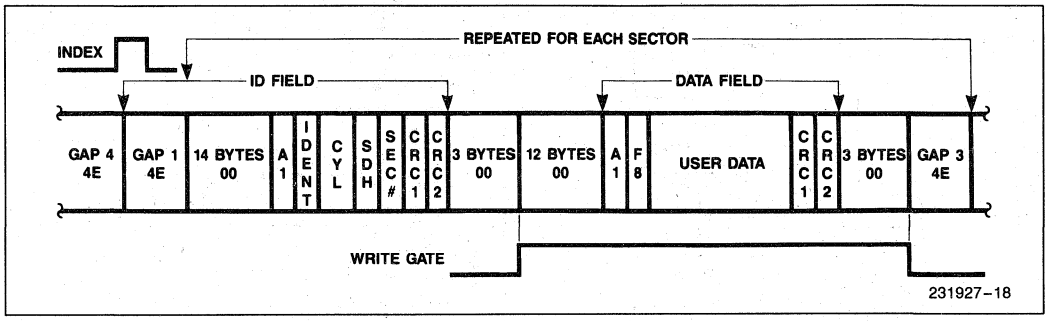


Figure 19. 82064 Sector Format

5.2 Software Section: General Programming

This section describes the software needed to communicate with the 82064 in order to store and retrieve data. This chapter describes the software in a general manner and Appendix B contains the actual implementation used to exercise the 82064 SBX board.

Polled Mode

As discussed in the Polled Interface Section, the 82064 does not directly support polled operation for data transfers without the addition of hardware. This section is based upon the polled interface as described in the Polled Interface Section.

The six 82064 commands can be divided into two groups, those with data transfers and those without. The commands that do not use the sector buffer are: Restore, Seek and Scan ID. The functions of each command are explained in the Commands Section. Figure 21 is a flowchart of a polled operation and a PLM example.

The last status that was read will contain any error conditions that might have occurred during the command.

For commands that do make use of the sector buffer, the size of the sector buffer will affect the software. If the sector buffer is equal in size to one sector, then a carry out of an address counter (for the sector buffer) as the buffer is being filled will indicate to the 82064 that the command should continue. If the sector buffer size is equal to two or more disk sectors, and only one sector is being transferred, then the carry out signal would not go active, and the 82064 will be forever waiting for BRDY. In this case an I/O port would have to be used to generate this signal for the 82064 so that command execution can finish. Figure 22 is a flowchart of the READ SECTOR command, and its PLM representation. The WRITE SECTOR and FORMAT TRACK commands are equivalent in terms of software interfacing. Their flowcharts and their PLM equivalents are shown in Figure 23.

Once the command register is written the 82064 requests a data transfer before locating the proper track. Once the buffer is filled and BRDY is asserted, the 82064 will locate the target track and sector. If the ID is not located before the selected number of retries have occurred, the 82064 will terminate the command. The data transferred to the sector buffer will not have been used. Once the command has finished (i.e., CIP = 0), the status and error registers will inform the host of an error.

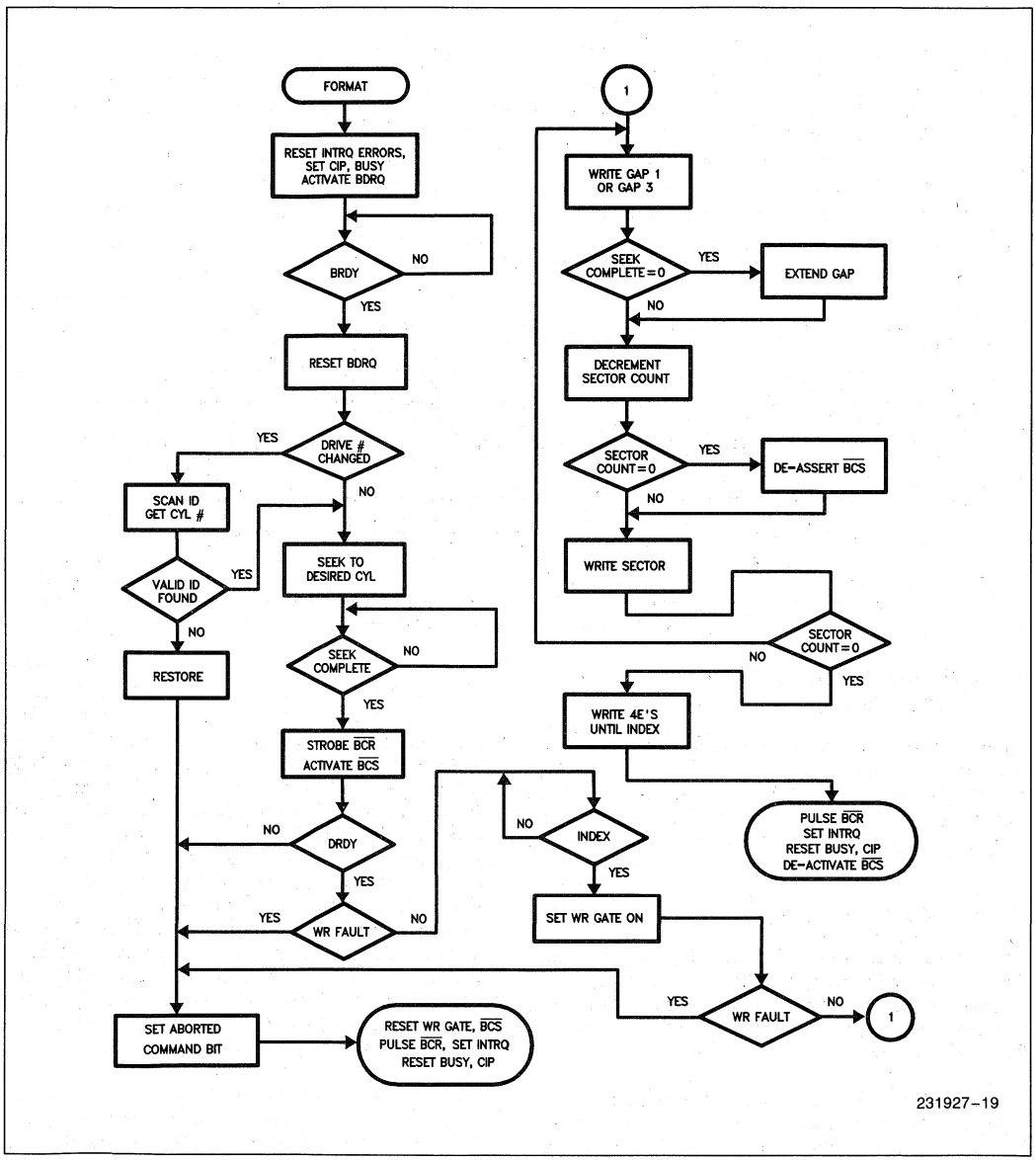
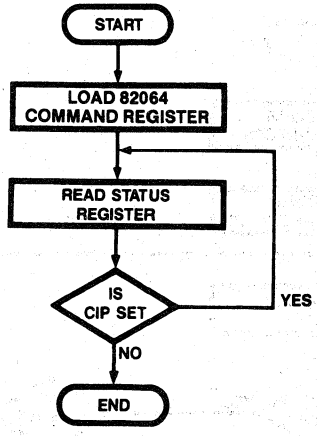


Figure 20. Write Format Command Flow

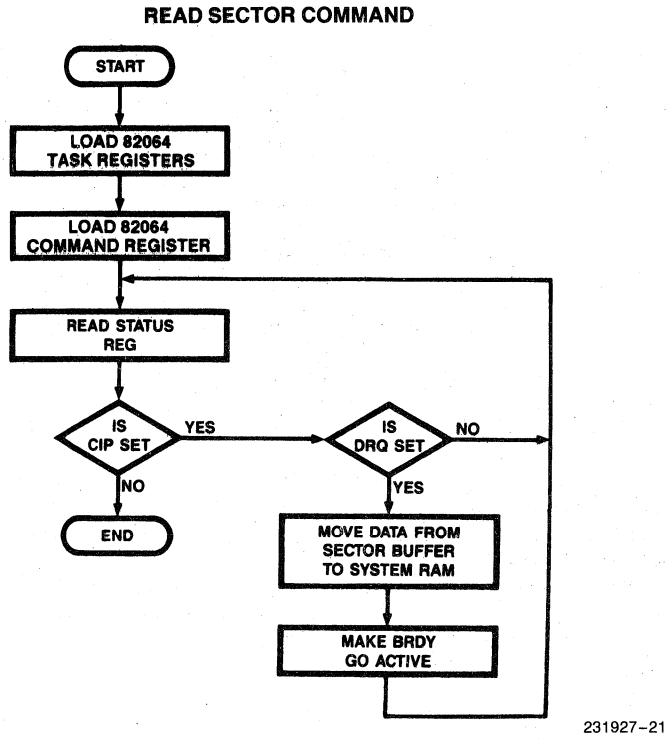


231927-20

```
Disk$Operation: Procedure;
   Call Write$82064$Task$Reg's; /* Write Task Registers */
   Output (Command$Reg) = Command;
   Status = Input (Status$Reg); /* Read Status Reg */
   Do while Status and CIP = CIP; /* Wait until command finishes */
      Status = Input (Status$Reg);
   End;
End Disk$Operation;
```

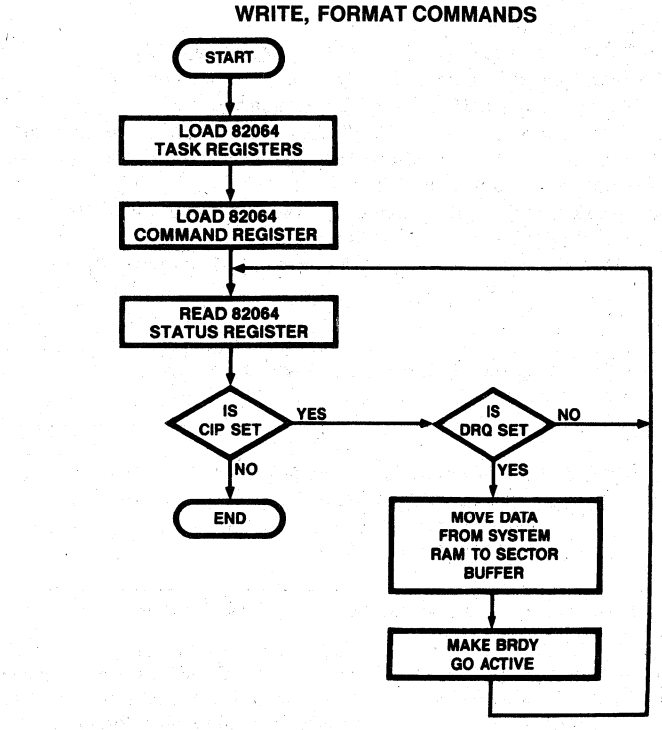
Figure 21. Polling Status

5




```
Disk$Operation: Procedure;
    Call Write$82064$Task$Regs;
    Output (Command $ Reg) = Command;
    Status = Input (Status$Reg);
    Do while Status and CIP = CIP;
        If Status and DRQ = DRQ then Do;
          Call Read$Data$From$Buffer;
          Output (BRDY$PORT) = 01;
        End;
        Status = Input (Status$Port)
    End;
End Disk$Operation;
```

Figure 22. Polling For Read Data




```
Disk$Operation: Procedure;
  Call Write$82064$Task$Regs;
  Output (Command$Reg) = Command;
  Status = Input (Status$Reg);
  Do while status and CIP = CIP;
    If status and DRQ = DRW then do;
      Call Write$Data$to$Buffer;
      Output (BRDY$Port) = 01; /* Make BRDY go active */
    End;
    Status = Input (Status$Reg)
  End;
End Disk$Operation;
```

Figure 23. Polling For Write Data

```
Disk$Operation: Procedure;
   Call Write$82064$Task$Regs; /* Write registers */
   Output (Command$Reg) = Command; /* Start command */
   Status = Input (Status$Reg); /* Read status */
   Do while status and CIP = CIP; /* Is a command in progress */
     If status and DRQ = DRQ then do; /* Data transfer? = yes */
       If command = Read$Sector then
        Call Read$Data$From$Buffer; /* Remove data */
       Else Call Write$Data$to$Buffer; /* Send data */
       Output (BRDY$PORT) = 01; /* Toggle BRDY 0 to 1 */
     End;
End Disk$Operation;
```

Figure 24. Complete Polled Flow

```
Start$Disk$Operation: Procedure;
    Call Write$82064$Task$Reg's;
    Output (Command $ Reg) = Command;
End Start$Disk$Operation;
```

Figure 25. Interrupt Mode; Starting a Disk Transfer

Figure 24 is the PLM routine that allows for all six of the commands. It differs from the READ and WRITE routines in that the direction that data is to be transferred is determined by the command.

Figure 24 also works for multiple sector transfers. However, the BRDY signal must be generated in hardware (the carry-out of an address counter).

Interrupt Mode

Interrupt driven software is chosen when the microprocessor must execute other tasks and cannot sit waiting for the disk to reposition its heads, as in a polled environment. The delay in repositioning heads can be anything from a couple of milliseconds to a second or more.

The 82064's interrupt (INTRQ) pin goes active to indicate that the command has finished. The READ SECTOR command provides the programmable choice of having the interrupt occur at the end of the data transfer or the normal end of the command. The reason for this option is that when the 82064 signals that a data transfer is required (via BDRQ, DRQ) the disk has been read and the data has been placed in the buffer. The host would remove the data and issue BRDY. The 82064 would then issue an interrupt indicating that the command has finished. The interrupt procedure would only have to read the status register. If the interrupt is issued at BDRQ the host would remove the buffer data and generate BRDY. At this point the status and error registers contain valid information. Generating an interrupt at BDRQ time may save some systems some software overhead.

The WRITE SECTOR and FORMAT commands do not have this option because the sector buffer is filled before the track and sector are located. Hence, there can be significant delays between asking for data and the command terminating.

In an interrupt driven environment, the 82064 can interface to a DMA controller for data transfers between the sector buffer and the host's RAM. If a DMA controller is not available an interrupt must be generated via the BDRQ line. However, BDRQ can stay active for long periods of time (until BRDY is generated). The interrupt sensing logic must take this into account to avoid being retriggered constantly. Intel's 8259A Interrupt Controller 8259A provides that capability. It should be programmed for edge triggered interrupts or the end of interrupt byte must not be issued until BDRQ is removed to prevent retriggering.

Figure 25 is a PLM example of starting a disk operation in an interrupt driven environment. The command starts, and some indefinite amount of time later an interrupt would be generated, indicating service is required.

```
End$of$Transfer: Procedure Interrupt;
    Status = Input (Status$Register);
    Output (8259A PIC) = End$of$Interrupt;
End End$of$Transfer;
```

Figure 26. Checking Status via Interrupt

```
Service$Disk$Controller: Procedure Interrupt;
    Status = Input (Status$Port);
    If Status and DRQ = DRQ then
        Call Transfer$Data$To/From$Buffer;/* Enable DMAC */
    Output (8259A PIC) = End$of$Interrupt;
End Service$Disk$Controller;
```

Figure 27. Complete Interrupt Procedure

If a DMA controller is used, it would have to be programmed and initialized before the command is issued to the 82064. Recall that once a data transfer between the microprocessor and 82064 has finished, BRDY must be set high. As long as BRDY is generated from hardware, no microprocessor intervention is needed. If BRDY is generated by an I/O port the microprocessor will have to perform this function (this will be the case with any system that has a sector buffer larger than one sector). (One option could be to generate an interrupt from the terminal count pin of the DMA controller. The microprocessor would then issue a BRDY.) Data transfers between host RAM and the sector buffer would be handled without microprocessor intervention. The interrupt would then signal that the command has finished as shown in Figure 26. The only operation the host processor would perform is to check the status register of the 82064 for any error conditions.

If BDRQ is used to generate an interrupt in addition to the normal interrupt, then the routines shown in Figure 27 will check the status register to see if a data transfer should be executed or if the command is finished. If DRQ is not set, the command has finished and any error conditions would be in the status register.

Another possibility would be to have separate interrupt routines for the two possible sources of interrupts (INTRQ, BRDQ). There would then be no need to test the status to see which interrupt had occurred.

6.0 APPLICATION EXAMPLE

This section shows an application using the 82064 interfaced to the SBX bus. A quick overview of the SBX bus is provided (pin descriptions, general wave forms) as a background for the application. Designing the 82064 onto an SBX Multimodule board was chosen to highlight the size and complexity differences between earlier TTL, MSI, LSI-based disk controller boards and what is possible using the 82064. Both the hardware and software sections will be applicable to most other designs using the 82064. This design example is called SBX82064 and does not represent a real product offered by Intel Corporation. Appendix C contains the schematic of the SBX board.

The advantage of the SBX Multimodule is that it permits the system to be tailored for specific needs with a minimum of effort. The advantage of an SBX based disk controller is that a current system can make use of the capacity, reliability and speed of a hard disk with no (or minimal) hardware redesign.

6.1 iSBX Bus Multimodule Boards

The iSBX Multimodule boards are small, specialized, I/O mapped boards which plug onto base boards. The iSBX boards connect to the iSBX bus connector and convert the iSBX bus signals to a defined I/O interface.

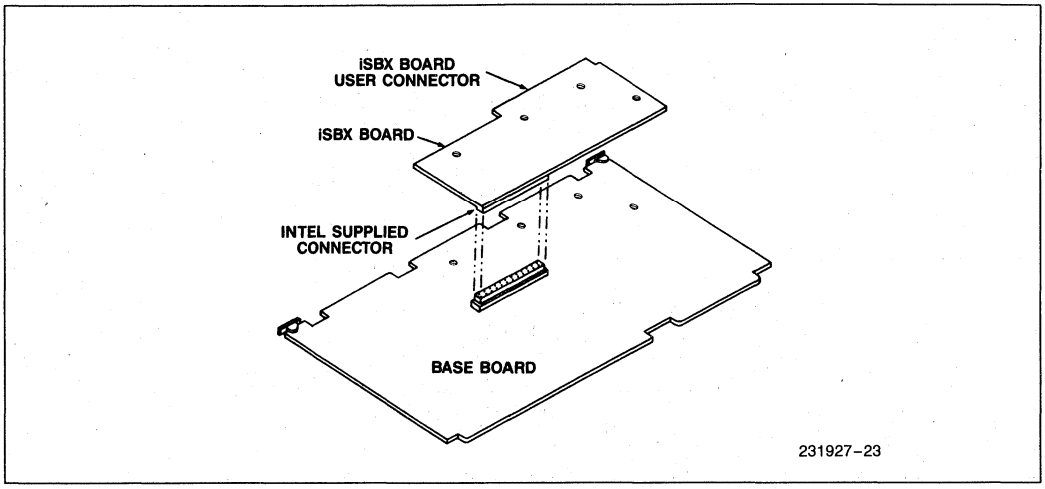


Figure 28. iSBX Multimodule Board Concept (Double Wide)

Base Boards

The base board decodes I/O addresses and generates the chip selects for the iSBX Multimodule boards. In 8-bit systems, the base board decodes all but the lower three addresses in generating the iSBX Multimodule board chip selects. In 16-bit systems, the base board decodes all but the lower order four addresses in generating the iSBX Multimodule board chip selects. Thus, a base board would normally reserve two blocks of 8 I/O ports for each iSBX socket it provides.

There are two classes of base boards, those with Direct Memory Access (DMA) support and those without. Base boards with DMA support are boards with DMA controllers on them. These boards, in conjunction with an iSBX Multimodule board (with DMA capability), can perform direct I/O to memory or memory to I/O operations.

iSBX Bus Interface

The iSBX bus interface can be grouped into six functional classes:

1. Control Lines
2. Address and Chip Select Lines
3. Data Lines
4. Interrupt Lines
5. Option Lines
6. Power Lines

Control Lines

The following signals are classified as control lines:

COMMANDS:
- $\overline{\text{IORD}}$ (I/O Read)
- $\overline{\text{IOWRT}}$ (I/O Write)

DMA:
- MDRQT (DMA Request)
- $\overline{\text{MDACK}}$ (DMA Acknowledge)
- TDMA (Terminate DMA)

INITIALIZE:
- RESET

CLOCK:
- MCLK (iSBX Multimodule Clock)

SYSTEM CONTROL:
- $\overline{\text{MWAIT}}$
- $\overline{\text{MPST}}$ (iSBX Multimodule Board Present)

Command Lines ($\overline{\text{IORD}}$, $\overline{\text{IOWRT}}$)

The command lines are active low signals which provide the communication link between the base board and the iSBX Multimodule board. An active command line, conditioned by chip select, indicates to the iSBX Multimodule board that the address lines are valid and the iSBX Multimodule board should perform the specified operation.

DMA Lines (MDRQT, $\overline{\text{MDACK}}$, TDMA)

The DMA lines are the communication link between the DMA controller device on the base board and the iSBX Multimodule board. MDRQT is an active high output signal from the iSBX Multimodule board to the base board's DMA device requesting a DMA cycle. $\overline{\text{MDACK}}$ is an active low input signal to the iSBX Multimodule board from the base board DMA device acknowledging that the requested DMA cycle has been granted. TDMA is an active high output signal from the iSBX Multimodule board to the base board. TDMA is used by the iSBX Multimodule board to terminate DMA activity. The use of the DMA lines is optional as not all base boards will provide DMA channels and not all iSBX Multimodule boards will be capable of supporting a DMA channel.

Initialize Lines (Reset)

This input line to the iSBX Multimodule board is generated by the base board to put the iSBX Multimodule board into a known internal state.

Clock Lines (MCLK)

This input to the iSBX Multimodule board is a timing signal. The 10 MHz (+0%, −10%) frequency can vary from base board to base board. This clock is asynchronous from all other iSBX bus signals.

System Control Lines ($\overline{\text{MWAIT}}$, $\overline{\text{MPST}}$)

These output signals from the iSBX Multimodule board control the state of the system.

An active $\overline{\text{MWAIT}}$ (Active Low) will put the CPU on the board into wait states providing additional time for the iSBX Multimodule board to perform the requested operation. $\overline{\text{MWAIT}}$ must be generated from address (address plus chip select) information only. If $\overline{\text{MWAIT}}$ is driven active due to a glitch on the CS line during address transitions, $\overline{\text{MWAIT}}$ must be driven inactive in less than 75 ns.

The iSBX Multimodule board present ($\overline{\text{MPST}}$) is an active low signal (tied to signal ground) that informs the base board I/O decode logic that an iSBX Multimodule board has been installed.

Address and Chip Select Lines

The address and chip select lines are made up of two groups of signals.

Address Lines: MA0–MA2

Chip Select Lines: $\overline{\text{MCS0}}$–$\overline{\text{MCS1}}$

The base board decodes I/O addresses and generates the chip selects for the iSBX Multimodule boards. The base board decodes all but the lower order three addresses in generating the iSBX Multimodule board chip selects.

Address Lines (MA0–MA2)

These positive true input lines to the iSBX Multimodule boards are generally the least three significant bits of the I/O address. In conjunction with the command and chip select lines, they establish the I/O port address being accessed. In 16-bit systems, MA0–MA2 may be connected to ADR1–ADR3 of the base board address lines.

Chip Select Lines (MCS0–MCS1/)

In an 8-bit system, these input lines to the iSBX Multimodule board are the result of the base board I/O decode logic. $\overline{\text{MCS}}$ is an active low signal which conditions the I/O command signals and thus enables communication with the iSBX Multimodule boards.

6.2 The SBX82064 Design Example

The SBX82064 Multimodule board will interface an ST506 compatible drive to any host board having an SBX connector. Two restrictions on the disk drive are that there is a maximum of 2048 cylinders and/or 8 heads. The SBX connector cannot supply the power-up current requirements of the drive. The drive must be connected directly to the power supply. The SBX82064 in Appendix C does not support DMA transfers. The version in Appendix D does support DMA transfers. Since this multimodule has a 2 kbyte sector buffer, the host microprocessor must generate a BRDY by accessing an I/O port during data transfers.

The software for communicating to the SBX board is intended to be interrupt driven. Polling for data transfers is not supported. Reading the status without an interrupt is not recommended. During the times the 82064 is accessing the sector buffer, the SBX82064 will isolate itself from the host. To support polling, a hardware generated busy pattern should be driven onto the host's data bus as is shown in the Polled Interface section. The sector buffer stores up to 2 kbytes of disk data, for multiple sector transfers. The SBX board only interfaces to one drive (for space reasons), but four drives could be used with the addition of a read data multiplexor (one IC) and the drive data cables.

Microprocessor Interface

Figure 29 is a block diagram of the SBX82064's microprocessor interface. The I/O port assignments are listed in Table 1. The functional blocks of the interface are:

Sector Buffer Isolation Logic

Wait State Logic

Sector Buffer

Sector/Drive/Head Register Logic

Table 6-1. I/O Port Assignments

Port Address	Read	Write
80H	Sector Buffer	Sector Buffer
82H	Error Reg	RWC Reg
84H	Sector Count	Sector Count
86H	Sector Number	Sector Number
88H	Cylinder Low	Cylinder Low
8AH	Cylinder High	Cylinder High
8CH	SDH Reg	SDH Reg
8EH	Status Reg	Command Reg
90H	None	None
92H	None	Asserts $\overline{BCR}$
94H	None	Asserts BRDY

NOTE:
Address assignments are determined by the host board.

Sector Buffer Isolation Logic

The host will be isolated from the SBX board whenever the 82064 is accessing its sector buffer which is enabled by $\overline{BCS}$. The host's control signals, $\overline{RD}$, $\overline{WR}$, $\overline{MCSO}$, and $\overline{MCS1}$ and data bus are also disabled at the same time to prevent any data in the sector buffer from being corrupted. The host should wait for an interrupt before reading the 82064's Status register. Attempting to read the SBX board while $\overline{BCS}$ is active will return invalid data, since the SBX board will have the data bus tri-stated.

Wait State Logic

The wait state logic drives the 'not ready' line, $\overline{MWAIT}$, active whenever the host reads the SBX board. $\overline{MWAIT}$ does not go active for buffer or 82064 register writes. This logic was required for two reasons. First, a delayed read is generated, because the address setup to $\overline{RD}$ margin of the SBX bus is less than the 82064's needs (50 ns vs 100 ns). Second, the $\overline{RD}$ to data valid access period of the 82064 (375 ns), is greater than the SBX bus' full speed read cycle (275 ns) permits.

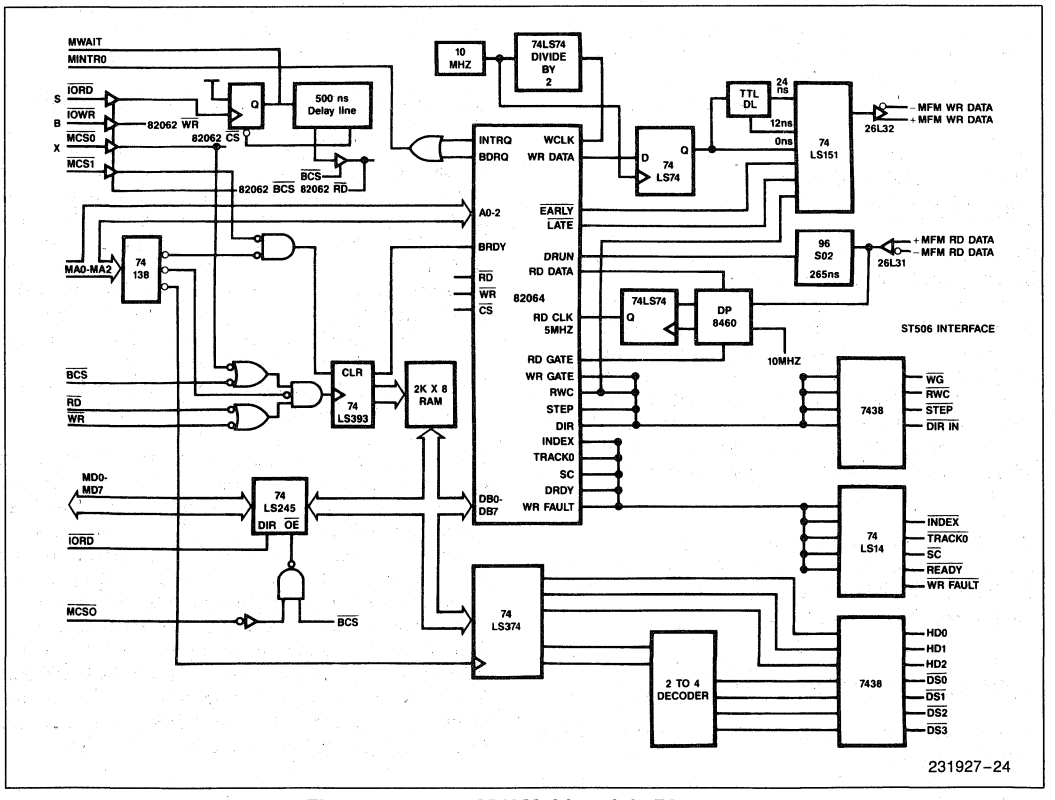


Figure 29. 82064 SBX Multimodule Diagram

$\overline{MWAIT}$ is deactivated after allowing for the delayed $\overline{RD}$ and the access period of the 82064. This delay is accomplished with a 500 ns delay line. The first tap at 100 ns generates the read request to allow for the address setup margin. The next tap 400 ns later removes $\overline{MWAIT}$ to allow the host to continue.

Sector Buffer

The sector buffer consists of an address counter (using '1s393's) and a 2 kbyte static RAM. The address counter is incremented on the trailing edge of a valid $\overline{RD}$ or $\overline{WR}$ cycle, either host microprocessor or 82064 initiated. The counter is reset by a hardware reset, the 82064 buffer reset $\overline{BCR}$, or by accessing an I/O port to provide software control. The 82064 will issue $\overline{BCR}$ each time $\overline{BCS}$ changes state (i.e. twice per sector). Resetting the buffer counter can be put under software control for multiple sector transfers. BRDY going high tells the 82064 that the buffer is available for its use. BRDY is generated by the address counter, by filling or emptying the entire buffer in multiple sector transfers, or from an I/O port when single sector transfers are done (since single sectors won't use all 2 kbytes of the buffer, the hardware signal will not be generated). When the 82064 is using the buffer, $\overline{BCS}$ will be low, and the $\overline{RD}$ or $\overline{WR}$ line will be pulsed every 1.6 microseconds.

When the 82064 is using the buffer it prevents access by the host by tristating the read, write, select and data lines with a low on $\overline{BCS}$.

SDH Register Logic

The drive and head select bits must be latched externally to the 82064, since these outputs are not provided. An 8 bit latch is strobed on the trailing edge of the $\overline{WR}$ pulse when the SDH register is selected. The two drive select bits are then demultiplexed to provide a one of four drive select line. If multiple drives are used then these outputs would also be used to select which disk's read data line would be gated into the PLL.

Interrupts

While the interrupt line is programmable (to notify of an end of command or data transfer request for the Read Sector command only), software will ensure that the interrupt from the 82064 signifies command termination. The BDRQ line is OR'ed with the 82064's INTRQ line or BDRQ can generate its own interrupt. BDRQ is also gated off-board for a DMA controller.

Disk Interface

Figure 30 is a block diagram of the interface between the 82064 and the disk drive. The functional blocks are:

Write Data Logic

Read Data Logic (PLL)

Drive Control

Write Data Logic

The WR DATA output requires a D flip-flop clocked at 10 MHz to complete the conversion of data to MFM. The output of this D flip-flop is true MFM and is sent to a delay line. A delay line determines the amount of delay for precompensation. No delay corresponds to shifting the data bit early; the first tap is approximately 12 ns of delay and is the "normal", or no delay and the second tap provides 12 ns of delay, referenced to the "normal" write data. Which output is selected is determined by the states on RWC, $\overline{Early}$ and $\overline{Late}$. This function was generated with a 74s151 multiplexer. When RWC is inactive $\overline{EARLY}$ and $\overline{LATE}$ only select "normal" data since they are always active. The precompensated write data is then driven onto the data cable by an RS-422 driver.

Read Data Logic

The PLL generates the RD CLOCK that is used to decode the serial MFM data from the drive. A selected drive issues read data, unless WR GATE is active. A one-shot generates a pulse of 220–270 ns to provide the DRUN input. Only during an all zero's or one's field will the DRUN input stay high, as it will be retriggered every 200 ns (the minimum distance that separates continuous clock and data bits). As soon as DRUN is determined to be valid, the RD GATE output will go active, switching the PLL from the 10 MHz local clock input to disk data. The PLL will synchronize to the incoming serial data and generate a Read Clock of the proper timing and phase. The 82064 will then start to search for the address mark which is indicated by DRUN going low at the address mark.

No detail is provided herein on PLL design, as it is beyond the scope of this document. PLL design should be left to experienced designers, since minute changes in temperature and component values will drastically affect the soft error rate. As an alternative, several companies manufacture very high speed PLL chips for MFM encoded disk drives. Besides being fairly easy to design in, they reduce the number of components and board area needed for the sophisticated PLL.

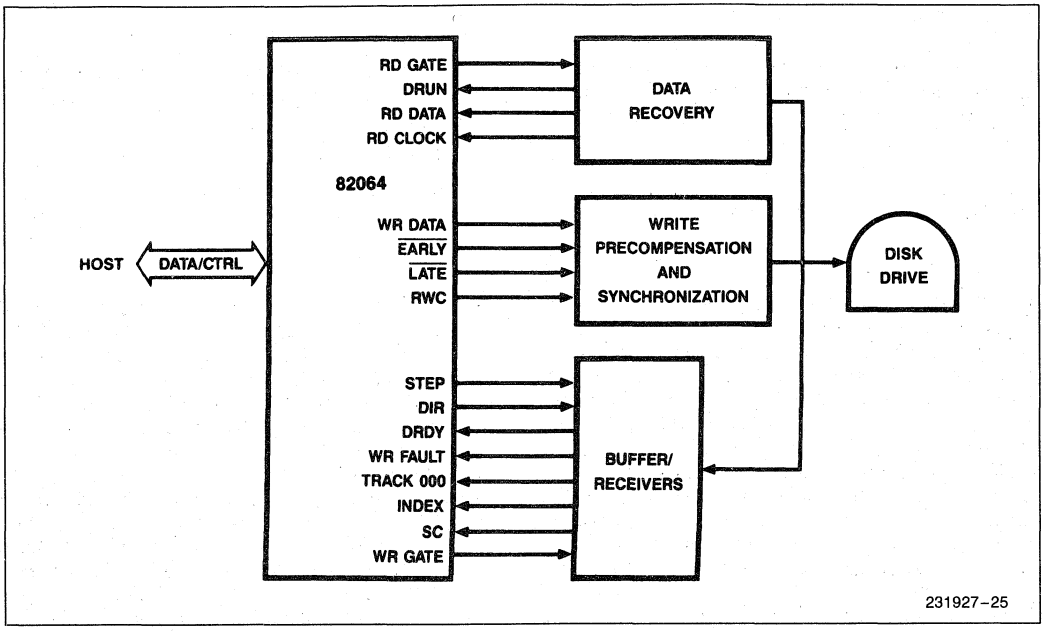


Figure 30. 82064 Disk Interface Block Diagram

6.3 Software Driver Overview

Presented in Appendix B is a listing of the software used to exercise the SBX 82064 board. Communication between the host software and the SBX driver routine is done through a structure located in system RAM. The host routine fills in required parameters, then passes the address of this communication block to the SBX driver routine. The driver routine pulls necessary values from this command block (CBL), executes a disk operation, then fills the CBL with the 82064's register contents, plus status and error information. The command block structure is shown in Figure 31.

Command	Byte
Rwc Reg	Byte
Sector Cnt.	Byte
Sector Num.	Byte
Cyl Low	Byte
Cyl High	Byte
SDH Reg	Byte
Status Reg	Byte
Error Reg	Byte
Host Buffer	Pointer

Figure 31

The host board did not have a DMA controller available, so an interrupt is issued from the BDRQ line and OR'ed with the 82064's interrupt line as interrupt sources were limited by the host. When an interrupt occurs, the interrupt procedure checks for either a data transfer, and executes it, or the completion of the command. If the interrupt signifies command completion, the interrupt procedure fills the command block with the 82064's task, status and error registers.

In this example, the host software examines one byte in the command block and until this byte is changed to a 00, no other command blocks will be passed to the disk driver routine. An alternative would be to issue a software interrupt to notify the microprocessor that the disk operation has finished and the command block contains parameters from the last operation and that a new disk command could start.

The driver for this example allows polling for non-data transfer commands, and must use interrupts for data transfers. As mentioned earlier, microprocessor intervention is required since the sector buffer is much larger than one sector and will not generate a BRDY. The microprocessor must write to an I/O port, which sets BRDY, after each host to sector buffer transfer. An

actual software implementation would not include the polling and interrupt routines together, as only one method would generally be used.

The calling routine, which would normally be a directory program, places the values for which sector, number of sectors, etc., in the CBL. The disk routine is called and the address of this structure is passed on the stack. The disk driver places these parameters in the 82064's Task registers and initiates a command.

If the interrupt driven method was chosen, the disk driver routine returns to the calling routine. This permits other processing to be performed while the disk is executing a command. At some point, an interrupt will be generated, either from BRDY or INTRQ. Control will pass to the driver and the status register will be checked. If a data transfer is needed, either the microprocessor can transfer data or a DMA controller can perform the function. Once the transfer of data to the buffer is finished the microprocessor must set BRDY through an I/O port.

APPENDIX A
ST506 INTERFACE

THE ST506 INTERFACE

The ST506 interface is a modified version of Shugarts floppy disk drive interface and has been promoted by Seagate Technology. This interface is intended to be easy and low in cost to implement, yet provide a medium level of performance. The interface rigidly defines several areas: the hardware interconnects, the data transfer rate, the data encoding method, and how the disk is formatted.

Data Transfer Rate

The data transfer rate depends upon the linear bit density of the disk media and the speed at which the disk spins. ST506 specifies a 5 Mbit/second transfer rate. The typical ST506 drive has a nominal linear density of 10,416 bytes and a disk speed of 3600 rpm, which yields a 5 Mbit/second data transfer rate. No deviation from 5 M/bits second is allowed.

Increasing the linear density to increase storage capacity would require a decrease in disk speed. Otherwise, the data rate would increase. This decrease in disk speed would cause access times to increase, which many would deem unacceptable. To increase storage capacity, and remain ST506 compatible, either the number of cylinders and/or the number of platters can increase.

Data Encoding

ST506 requires that the serial data, sent between the drive and the controller, be encoded according to MFM rules. The basic unit of storage is a bit cell, which stores one bit information. This bit cell is divided into two halves, consisting of a clock bit and a data bit (see Figure A-1).

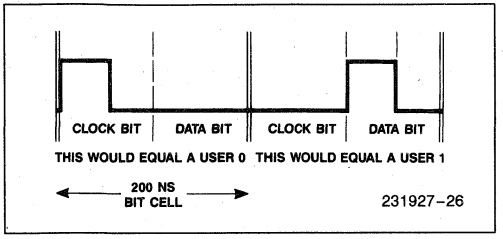


Figure A-1

The encoding rules for MFM are fairly simple:

1. A clock bit is written when the previous and the current bit cell does not contain a data bit.
2. A data bit is written whenever there is a "one" from the user.

Sync fields are composed of zeroes which generates a series of clock bits in the bit cell's. A phase lock loop locks on to the data stream during this period and generates a signal of the proper phase and frequency which is used to decode the combined clock and data serial data stream.

Disk Format

All disk media must be written with a specified format so that data may be reliably stored and retrieved. The smallest unit of controller accessible data is the sector which typically contains sync fields, ID fields, and a data field, and buffer fields.

The format of the disk required by ST506 is shown in Figure A-2. It should be noted that this format is fixed in the 82064. The user has options only for GAP1 and 3 length (when changing sector size or ECC) and whether to have 82064 CRC checking or user supplied ECC syndrome bits.

Gap 1 – Index Gap

Gap 1 serves two purposes. The first is to allow for variations in the index pulse timing due to motor speed variations. The second purpose is to allow a small delay to permit a different head to be selected without missing a sector. This is more of a data transfer optimization function and requires the disk controller to know which head is to be selected, when the last sector of a track has been read, and the next logical sector in the file exists on another platter. The 82064 does not switch heads automatically. Whether this scheme can be used or not depends upon the μP being able to alter one register in the 82064, before the next sector passes beneath the heads.

This gap is typically 12 bytes long and is written by the 82064 as 4E Hex.

Gap 2 – Write Splice Gap

This gap follows the CRC bytes of the ID field and continues up to the data field address mark. When updating a previously written sector, motor speed variations could turn on the write coil, as the head was passing over the ID field. This gap prevents this from occurring. The value written is OOH and also serves as the PLL sync field for the data field. The minimum value is determined by the "lock up" performance of the PLL. The 82064 writes sixteen bytes for this field once WG is activated. The user has no control over this field.

Gap 3 – Post Data Field Gap

Gap 3 is very similar to Gap 2 as it is used as a speed tolerance buffer also. Without this gap, and with the motor speed varying slightly, it would be possible for the upcoming sector's sync field and ID field to be overwritten. This value is '4E' H and is typically 15 bytes long. The 82064's Gap 3 length is programmable. The exact value is dependent upon several factors. Refer to 82064 Format command, Software Section: General Programming Section.

Gap 4 – Track Buffer Gap

This gap follows the last sector on a track and is written until an index pulse is received. Its purpose is to prevent the last sector from overflowing past the index gap, and absorb track length variations when ECC is used (ECC uses more bytes than CRC). The value is '4E' H and is about 320 bytes when CRC and 256 byte sectors are used. The 82064 writes this field only during formatting. The user has no control over the number of bytes written with the 82064.

ID Fields

The controller uses ID fields to locate any individual sector. An address mark of two bytes precedes the ID field and the data field in a sector. An address mark tells the controller the nature of the upcoming information. ID fields are used by the disk controller and are not passed to the host.

Sector Interleaving

Sector interleaving occurs when logical sectors are in a non-sequential order, which is determined during formatting. An advantage is that there is a delay between logically sequential sectors. This delay can be used for data processing and then deciding if the next sector should be read. Without interleaving, the next sector could slip by, imposing a one revolution delay (approx. 16.7 ms). An additional benefit to this delay is that bus utilization is reduced by spreading the data transfer over a greater amount of time. The delay between sectors can be determined as follows:

$$\frac{\text{1 Revolution Period}}{\text{Sectors/Track}} \times (\text{Interleave factor} - 1) = \text{Delay}$$

For the typical ST506 drive with four-way interleaving this yields 1.57 ms of delay.

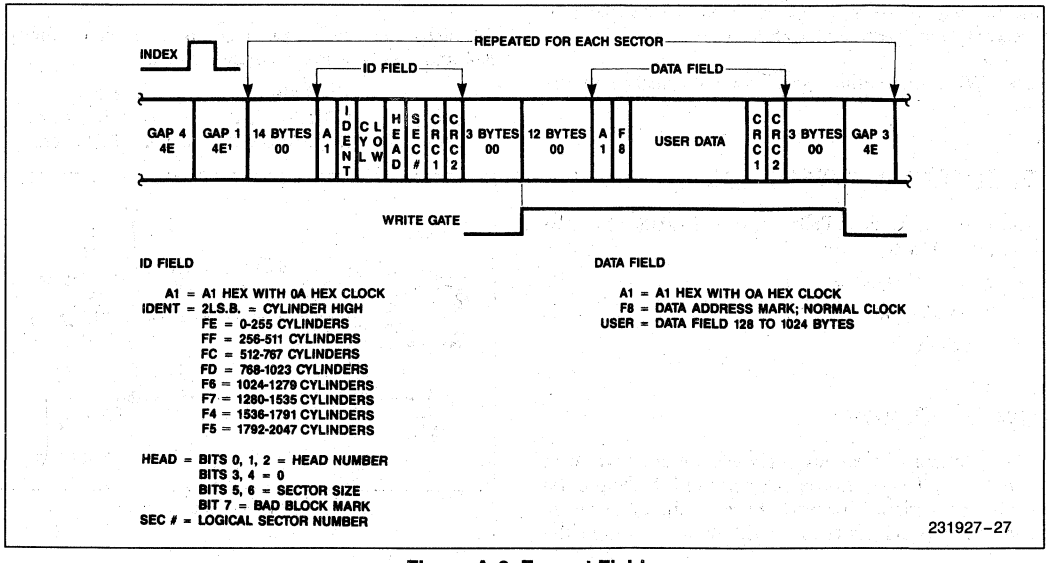


Figure A-2. Format Field

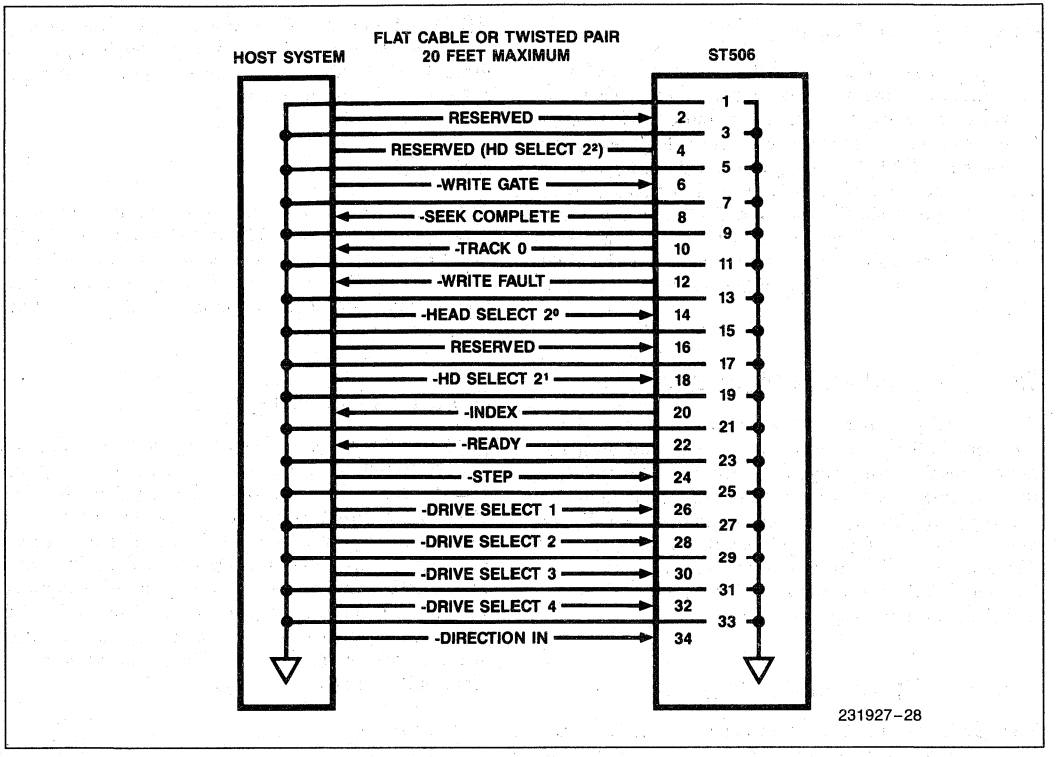


Figure A-3

The disadvantage to interleaving is that file transfers take longer, which may slow down the overall system. A four-way interleaved disk will have the transfer rate reduced to an average of 1.25 Mbit/sec.

The 82064 leaves the logical sector sequence to the user.

ELECTRICAL INTERFACE

The interface to the ST506 drive is divided into three categories and they are:

1. control signals,
2. data signals,
3. power.

Control Signals

The functions of the control signals are not covered in detail here. Their purpose can be found in the pin descriptions section. All control lines are digital in nature and either provide signals to the drive or inform the host of certain conditions. A diagram of the 34 pin control connector is shown in Figure A-3.

The driver/receiver logic diagram is shown in Figure A-4 and the electrical characteristics are:

	Voltage	Current
True	0.0 VDC to 0.4 VDC	−40 mA (IOL max.)
False	2,5 VDC to 5.25 VDC	250 μA (IOH open)

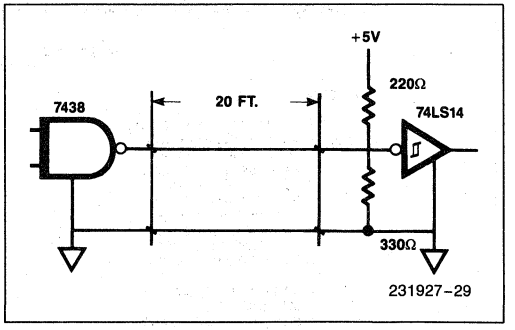


Figure A-4

Data Signals

The lines associated with the transfer of read/write data between the host and the drive are differential in nature and may not be multiplexed between drives. There is one pair of balanced lines for each read and write data line per drive and must conform to the RS-422 specification. Figure A-5 shows the receiver/transmitter combination.

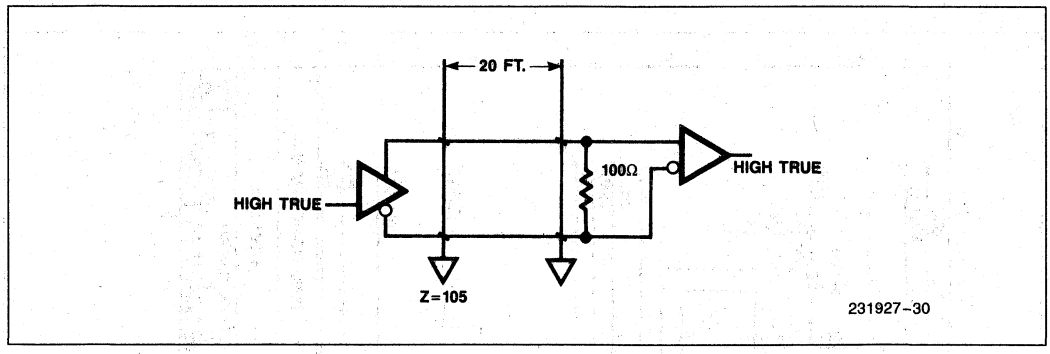
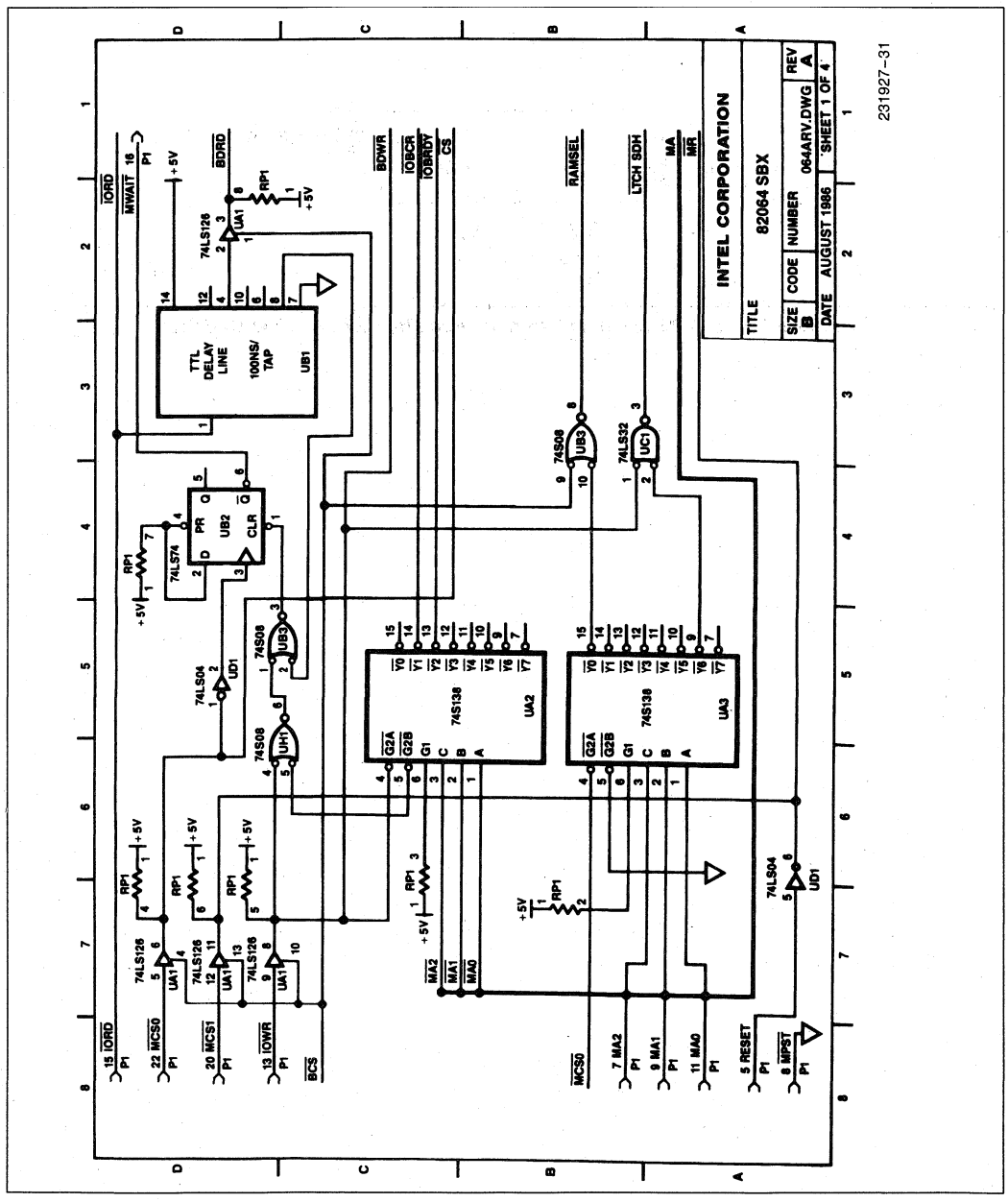
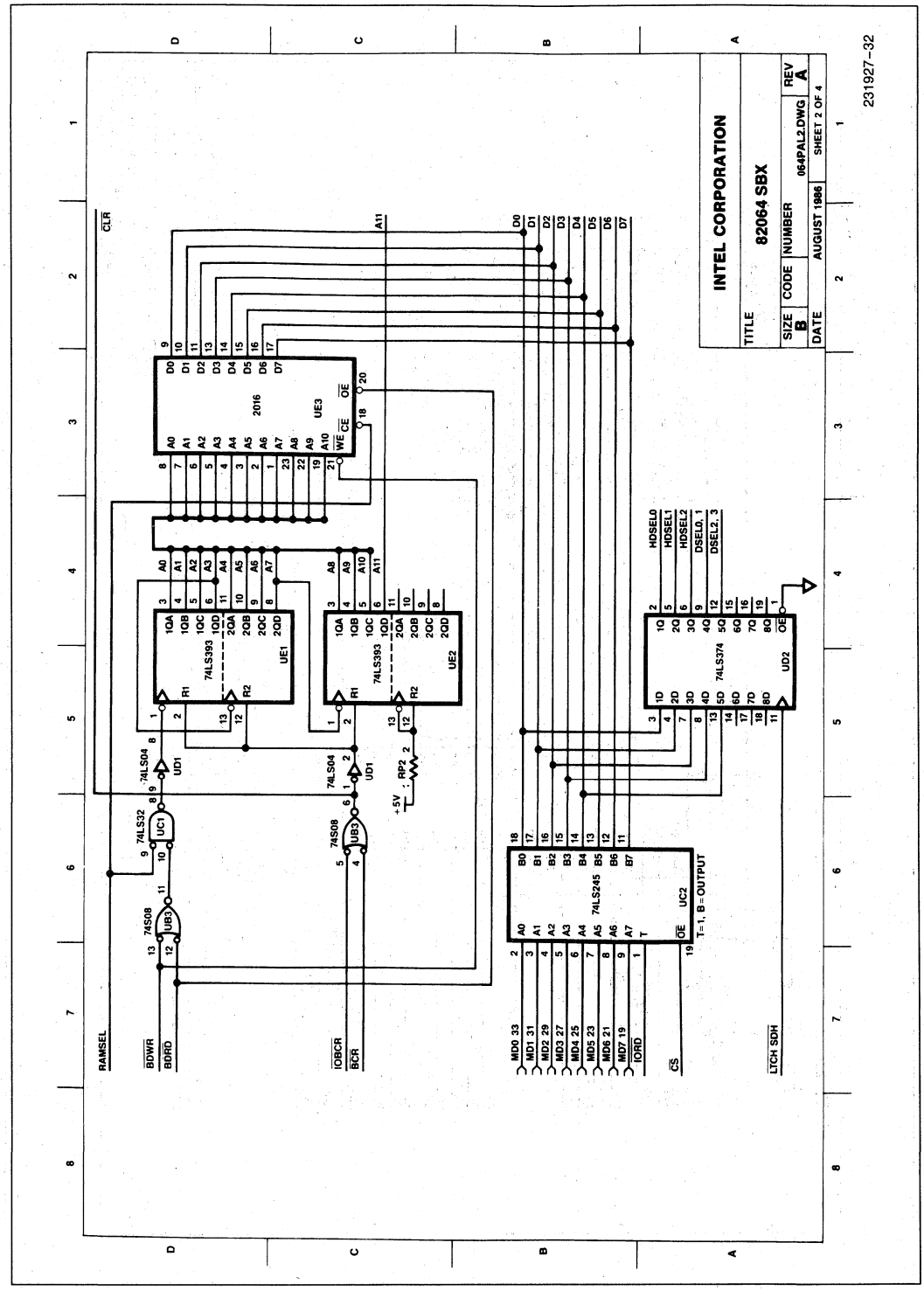


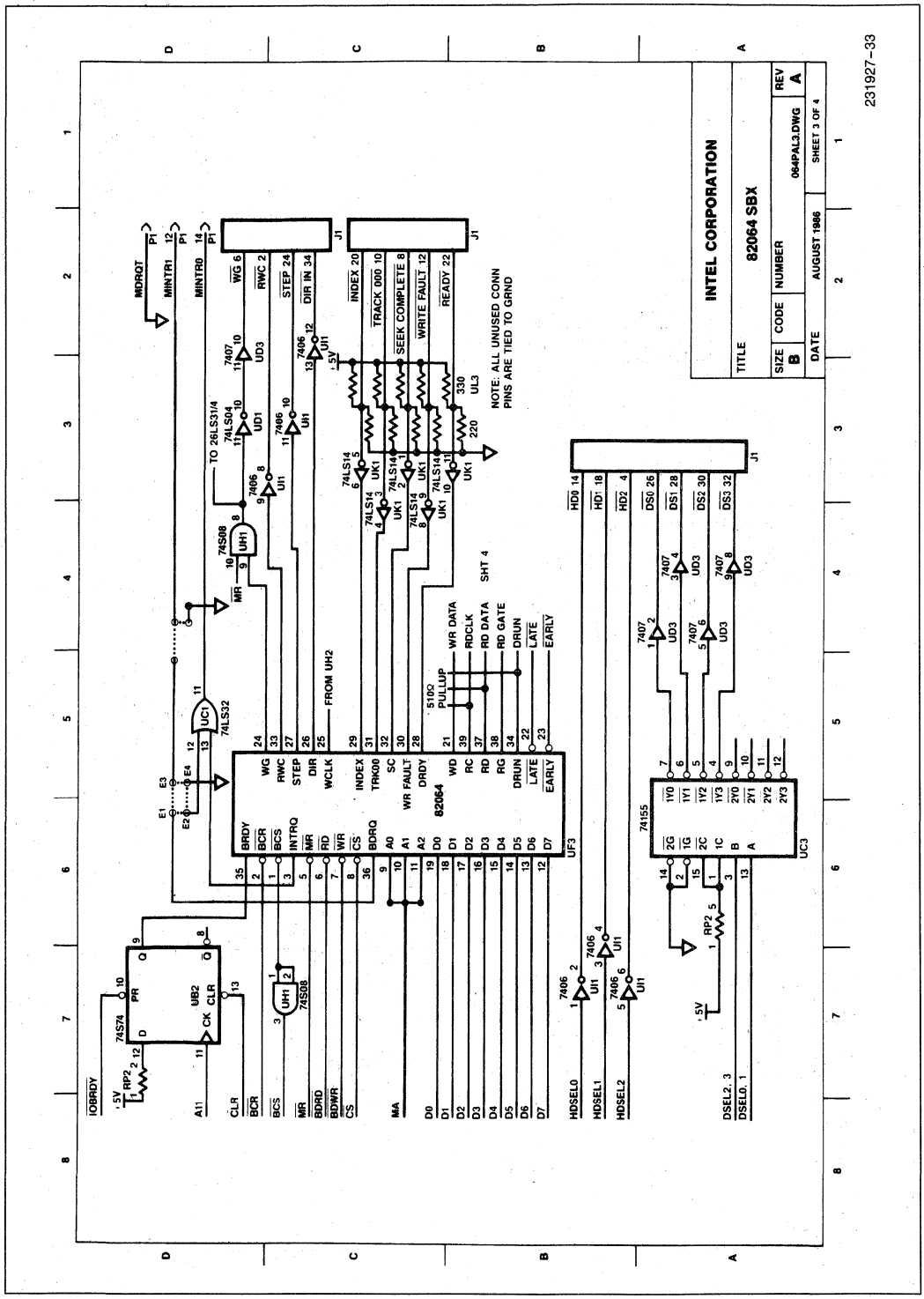
Figure A-5. E1A RS22 Driver/Receiver Pair Flat Ribbon or Twisted Pair

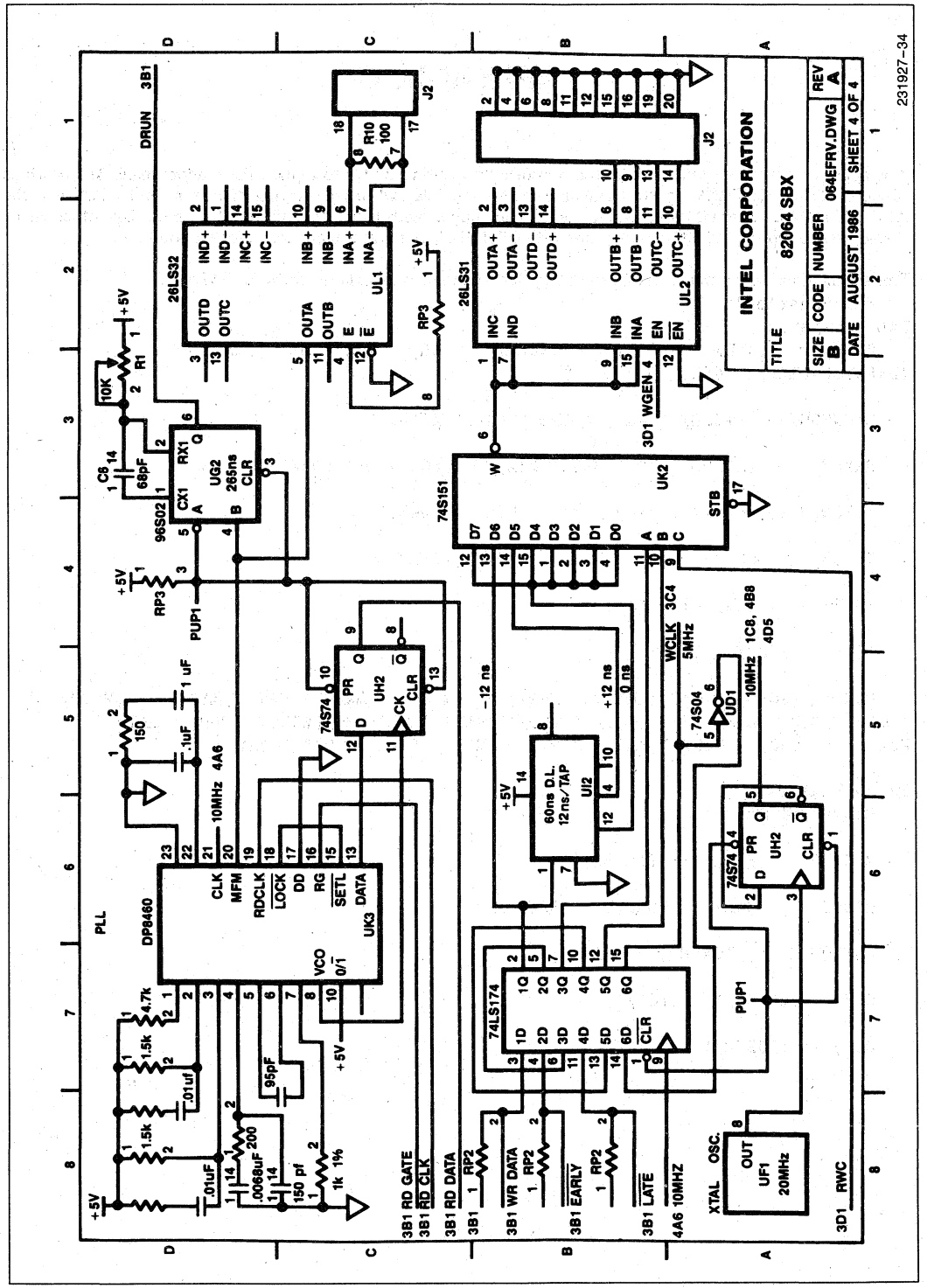
APPENDIX B
SCHEMATICS




INTEL CORPORATION
TITLE
82064 SBX
SIZE B
CODE
NUMBER
064PAL2.DWG
REV A
DATE
AUGUST 1986
SHEET 2 OF 4
231927-32
RAMSEL
CLR
BDWR
BDRD
IOBCR
BCR
LTCH SDH
CS
MD0 33
MD1 31
MD2 29
MD3 27
MD4 25
MD5 23
MD6 21
MD7 19
IORD
+5V
RP2
UE1
UE2
UE3
UC1
UC2
UD1
UD2
UB3
74LS393
74LS04
74LS32
74S08
74LS245
74LS374
2016
T=1, B=OUTPUT
HDSEL0
HDSEL1
HDSEL2
DSEL0, 1
DSEL2, 3

5


INTEL CORPORATION
TITLE 82064 SBX
SIZE B
CODE
NUMBER 064PAL3.DWG
REV A
DATE AUGUST 1986
SHEET 3 OF 4
231927-33
NOTE: ALL UNUSED CONN PINS ARE TIED TO GRND
82064
WD
WG
RWC
STEP
DIR
WCLK
INDEX
TRK00
SC
WR FAULT
DRDY
RC
RD
RG
DRUN
LATE
EARLY
WR DATA
RDCLK
RD DATA
RD GATE
BRDY
BCR
BCS
INTRQ
MR
RD
WR
CS
BDRQ
A0
A1
A2
D0
D1
D2
D3
D4
D5
D6
D7
MDRQT
MINTR1
MINTR0
INDEX
TRACK 000
SEEK COMPLETE
WRITE FAULT
READY
WG
RWC
STEP
DIR IN
HDSEL0
HDSEL1
HDSEL2
DSEL2, 3
DSEL0, 1
IOBRDY
CLR
BCR
BCS
MR
BDRD
BDWR
CS
MA
D0
D1
D2
D3
D4
D5
D6
D7
74S74
74S08
74LS32
74LS04
74LS14
74S06
74155
7406
7407
74LS31/4
TO 26LS31/4
FROM UH2
SHT 4
5V
5100 PULLUP
330
220

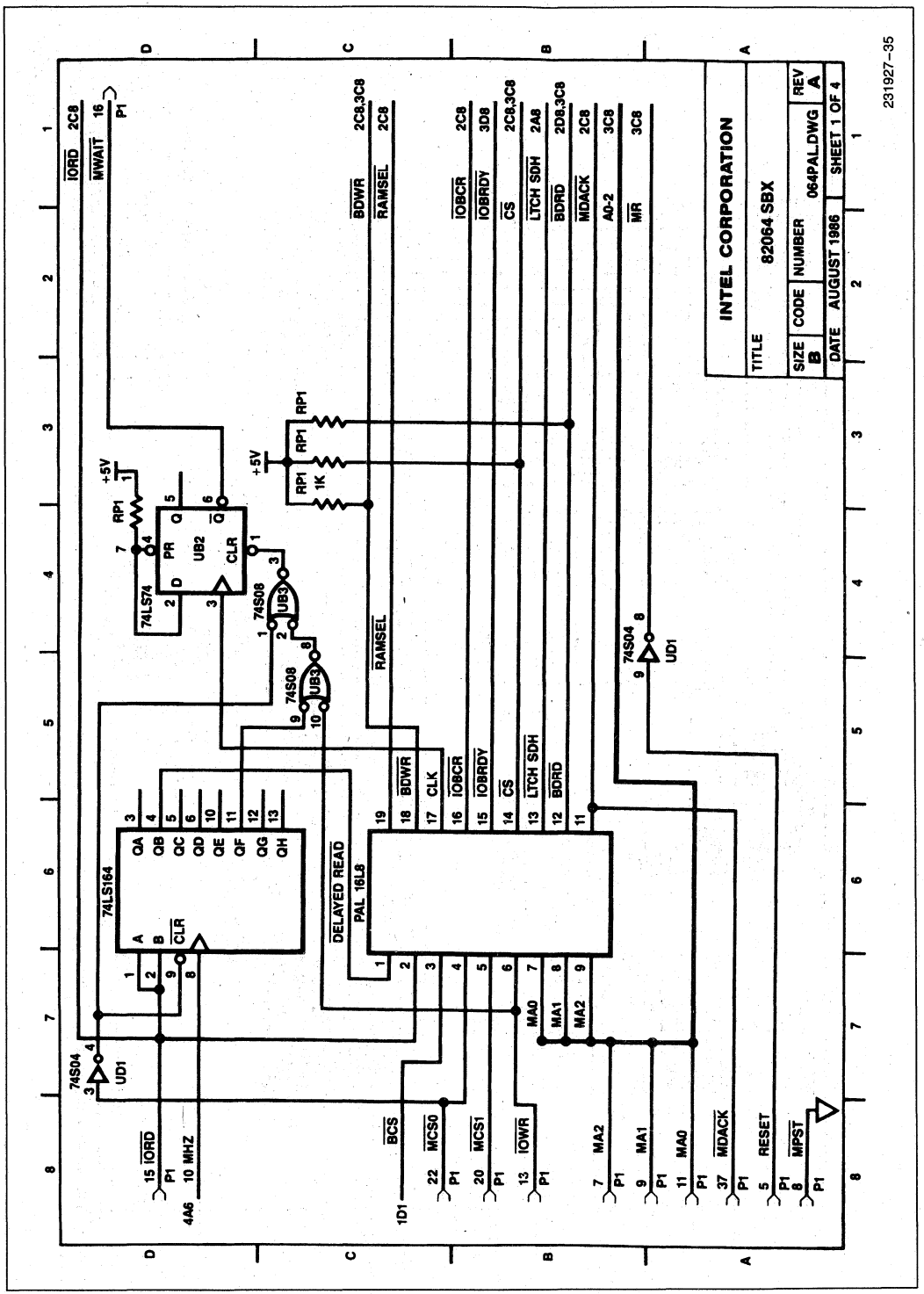

INTEL CORPORATION
TITLE 82064 SBX
SIZE B | CODE | NUMBER 064EFRV.DWG | REV A
DATE AUGUST 1986 | SHEET 4 OF 4
231927-34
PLL
DP8460
UK3
74LS174
UI2
74S74
UH2
74S151
UK2
26LS32
UL1
26LS31
UL2
UG2
96S02
74S04
UD1
WGEN 4
3D1
DRUN 3B1
WCLK 3C4
5MHz
10MHz
1C8, 4B8
4D5
4A6
10MHz
4A6 10MHZ
XTAL OSC.
UF1
20MHz
3D1 RWC
3B1 RD GATE
3B1 RD CLK
3B1 RD DATA
3B1 WR DATA
3B1 EARLY
3B1 LATE
PUP1
+5V
10K
R1
68pF
C6
RP3
RP2
R10
100
150
.1uF
1 uF
4.7k
1.5k
.01uf
95pF
.0068uF
200
150 pf
1k 1%
.01uF
60ns D.L.
12 ns/TAP
-12 ns
+12 ns
0 ns
J2

APPENDIX C

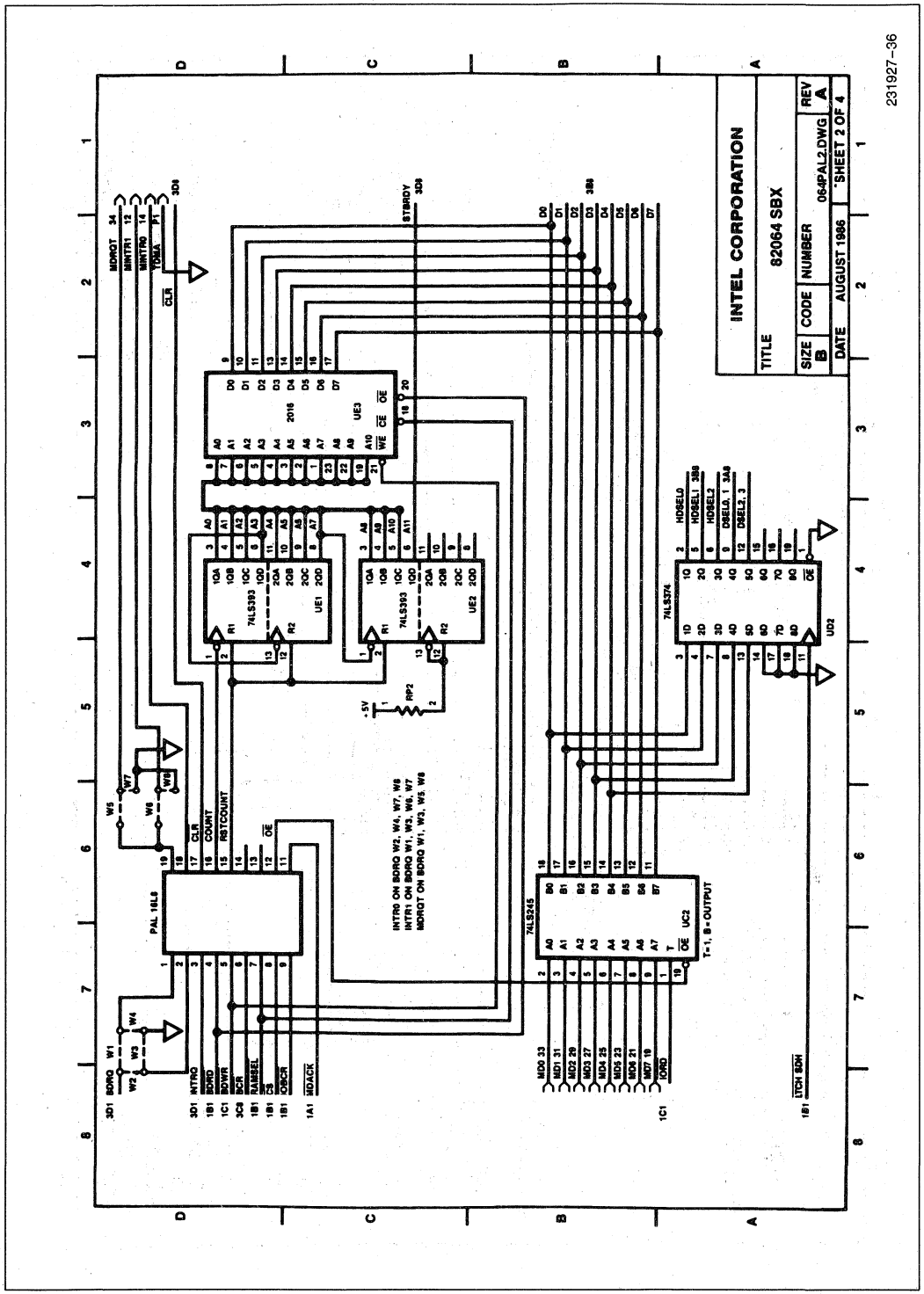
This appendix contains a schematic of the previous design using PAL's to replace the random logic. The previous design could not do DMA transfers and inserted a large delay when transferring data from buffer RAM to the system. The PAL version does do DMA transfers and buffer reads happen at full SBX bus speed. One other minor change was to replace the 500 ns delay line with a 74LS164, which is a more cost effective solution.

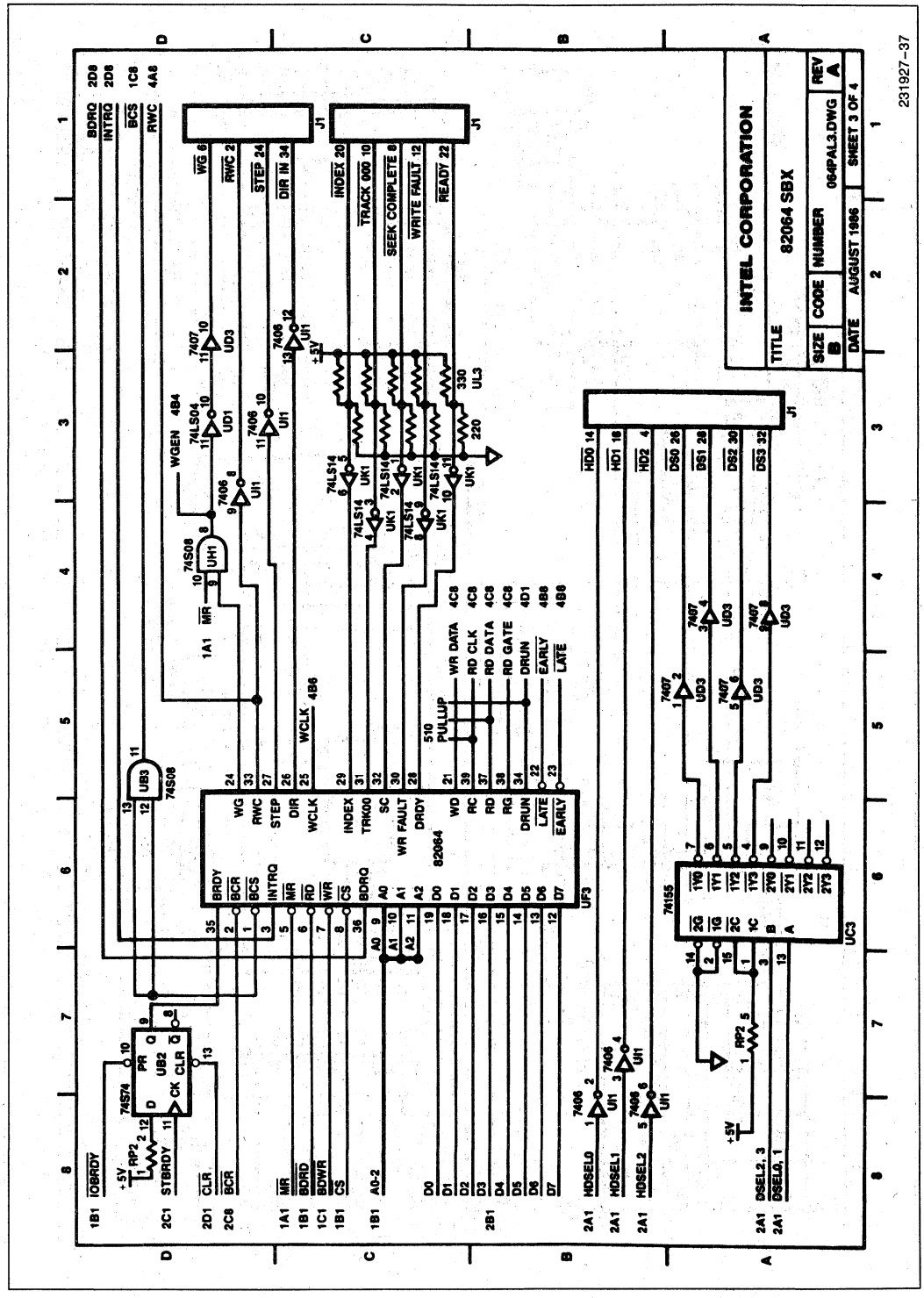
This schematic is only a paper design since only random logic was replaced with the PAL's.

```
   PAL Equation's
PAL - Page 1:
   BDRD/ = (IORD/ * MDACK/) + (IORD/ * MCS0/ * MA0 * MA1 * MA2) +
(DELAYED-READ/ * CLK) IF BCS

   LTCHSDH/ = (MCS0/ * MA0/ * MA1 * MA2 * IOWR/)

   RAMSEL/ = (MCS0 * MA0 * MA1 * MA2) + (BCS/) + (MDACK/)

   IOBRDY/ = (MCS1/ * MA0/ * MA1 * MA2/ * IOWR/)

   IOBCR/ = (MCS1/ * MA0 * MA1/ * MA2/ * IOWR/)

   BDWR/ = (IOWR/) IF BCS

   CS/ = (MCS0/) IF BCS

   CLK = (MCS0/ * MA0 * MA1/ * MA2/) + (MCS0/ * MA0/ * MA1 * MA2/) + (MCS0/
* MA0 * MA1 * MA2/) + (MCS0/ * MA0/ * MA1/ * MA2) + (MCS0/ * MA0 * MA1/ *
MA2) + (MCS0/ * MA0/ * MA1 * MA2) + (MCS0/ * MA0 * MA1 * MA2)

PAL - Page 2:
   MINTR1/MDRQT = (PIN1)

   MINTR0 = (PIN2) + (INTRQ)

   COUNT = (BDWR/ + BDRD/) * (RAMSEL/)

   RSTCOUNT = (IOBCR/) + (BCR/)

   OE/ = (MDACK/) + (CS/)

   CLR/ = (IOBCR/) + (BCR/)
```

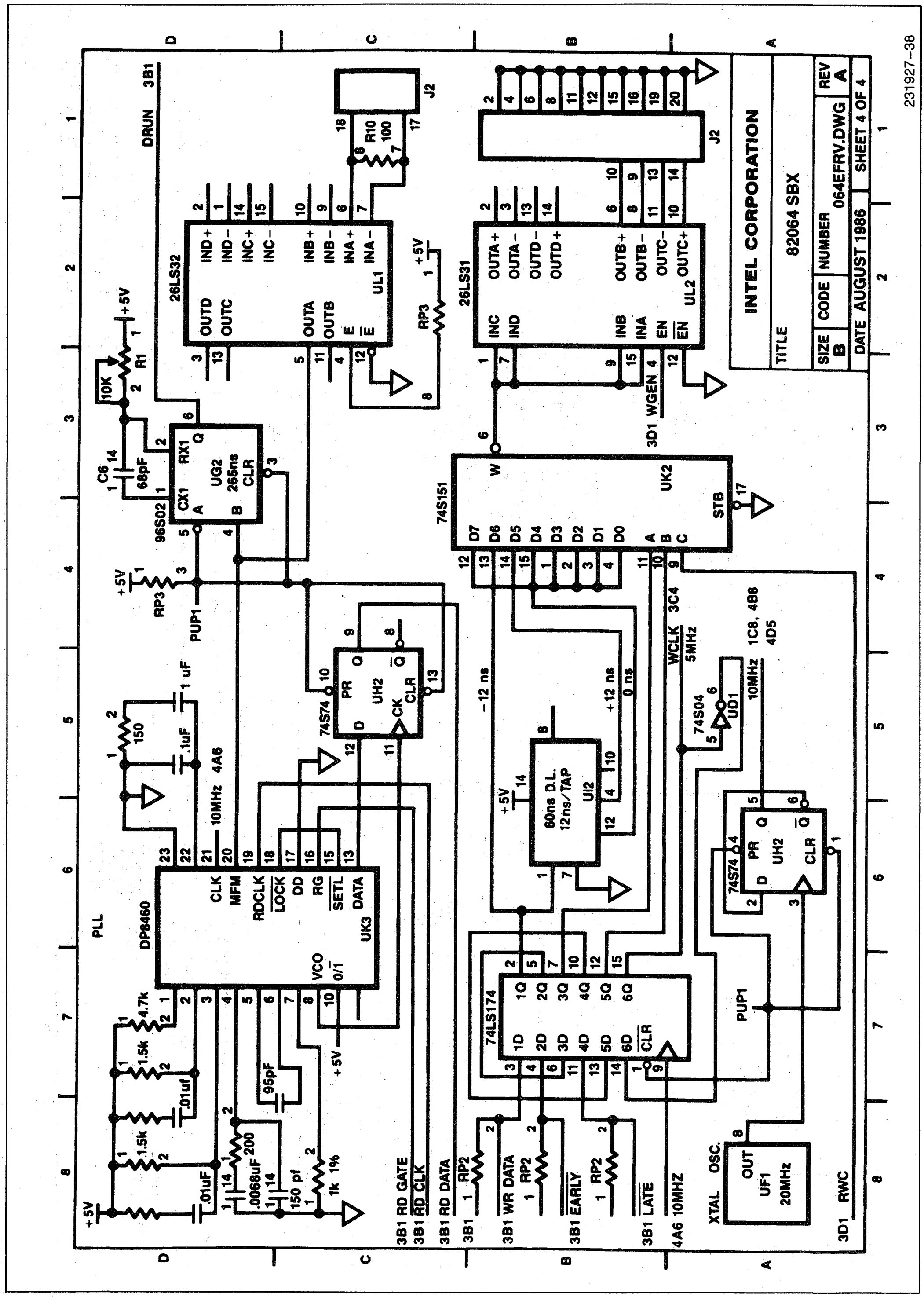

INTEL CORPORATION
TITLE 82064 SBX
SIZE B | CODE | NUMBER 064PAL.DWG | REV A
DATE AUGUST 1986 | SHEET 1 OF 4
231927-35
74LS164
PAL 16L8
74S04
74S08
74LS74
RP1
1K
+5V
DELAYED READ
RAMSEL
BDWR
IOBCR
IOBRDY
CS
LTCH SDH
BDRD
MDACK
A0-2
MR
MA0
MA1
MA2
BCS
MCS0
MCS1
IOWR
RESET
MPST
IORD
MWAIT
4A6
10 MHZ

5


INTEL CORPORATION
TITLE 82064 SBX
SIZE B CODE NUMBER 064PAL2.DWG REV A
DATE AUGUST 1986 SHEET 2 OF 4
231927-36
INTR0 ON BDRQ W2, W4, W7, W8
INTR1 ON BDRQ W1, W3, W6, W7
MDRQT ON BDRQ W1, W3, W5, W8
T=1, B=OUTPUT


INTEL CORPORATION
TITLE 82064 SBX
SIZE B
CODE NUMBER 064PAL3.DWG
REV A
DATE AUGUST 1986
SHEET 3 OF 4
231927-37
82064
74S74
74S04
74S08
74S06
74LS14
74LS04
74155
7406
7407
WGEN
WR DATA
RD CLK
RD DATA
RD GATE
DRUN
EARLY
LATE
PULLUP
WCLK
INDEX
TRK00
WR FAULT
DRDY
SEEK COMPLETE
WRITE FAULT
READY
TRACK 000
IOBRDY
STBRDY
HDSEL0
HDSEL1
HDSEL2
DSEL2, 3
DSEL0, 1

5


INTEL CORPORATION
TITLE 82064 SBX
SIZE B
CODE
NUMBER 064EFRV.DWG
REV A
DATE AUGUST 1986
SHEET 4 OF 4
26LS32
UL1
26LS31
UL2
J2
96S02
UG2
265ns
74S151
UK2
74S74
UH2
DP8460
UK3
PLL
74LS174
74S04
UD1
60ns D.L.
12ns/TAP
UI2
XTAL OSC.
UF1
20MHz
DRUN 3B1
PUP1
WCLK 3C4
5MHz
3D1 WGEN
10MHz
3B1 RD GATE
3B1 RD CLK
3B1 RD DATA
3B1 WR DATA
3B1 EARLY
3B1 LATE
4A6 10MHZ
3D1 RWC
231927-38

Universal Peripheral Interface Slave Microcontrollers

6

intel®

September 1990

Microprocessor Peripherals UPI™-41A/41AH/42/42AH User's Manual

6

Order Number: 231318-004

UPI USER'S MANUAL

CONTENTS

PAGE

CHAPTER 1
INTRODUCTION 6-4
Interface Registers for Multiprocessor Configurations 6-6
Powerful 8-Bit Processor 6-6
Special Instruction Set Features 6-7
Preprogrammed UPI's 6-8
Development Support 6-8
UPI Development Support 6-8

CHAPTER 2
FUNCTIONAL DESCRIPTION 6-9
Pin Description 6-9
CPU Section 6-12
Program Memory 6-13
Interrupt Vectors 6-13
Data Memory 6-13
Program Counter 6-14
Program Counter Stack 6-14
Program Status Word 6-15
Conditional Branch Logic 6-15
Oscillator and Timing Circuits 6-16
Interval Timer/Event Counter 6-18
Test Inputs 6-19
Interrupts 6-20
Reset 6-21
Data Bus Buffer 6-22
System Interface 6-23
Input/Output Interface 6-24
Ports 1 and 2 6-24
Ports 4, 5, 6, and 7 6-25

CHAPTER 3
INSTRUCTION SET 6-28
Instruction Set Description 6-30
Alphabetic Listing 6-32

CHAPTER 4
SINGLE-STEP, PROGRAMMING, AND POWER-DOWN MODES 6-55
Single Step 6-55
External Access 6-57
Power Down Mode (8041AH/8042 ROM Only) 6-57

CONTENTS

PAGE

CHAPTER 5
SYSTEM OPERATION 6-58
Bus Interface 6-58
Design Examples 6-59
General Handshaking Protocol 6-62

CONTENTS

PAGE

CHAPTER 6
APPLICATIONS 6-64
Abstracts 6-64

CHAPTER 1
INTRODUCTION

Accompanying the introduction of microprocessors such as the 8088, 8086, 80186 and 80286 there has been a rapid proliferation of intelligent peripheral devices. These special purpose peripherals extend CPU performance and flexibility in a number of important ways.

Table 1-1. Intelligent Peripheral Devices

8255 (GPIO)	Programmable Peripheral Interface
8251A(USART)	Programmable Communication Interface
8253 (TIMER)	Programmable Interval Timer
8257 (DMA)	Programmable DMA Controller
8259	Programmable Interrupt Controller
82077AA	Programmable Floppy Disk Controller
8273 (SDLC)	Programmable Synchronous Data Link Controller
8274	Programmable Multiprotocol-Serial Communications Controller
8275/8276 (CRT)	Programmable CRT Controllers
8279 (PKD)	Programmable Keyboard/Display Controller
8291A, 8292, 8293	Programmable GPIB System Talker, Listener, Controller

Intelligent devices like the 82077AA floppy disk controller and 8273 synchronous data link controller (see Table 1-1) can preprocess serial data and perform control tasks which off-load the main system processor. Higher overall system throughput is achieved and software complexity is greatly reduced. The intelligent peripheral chips simplify master processor control tasks by performing many functions externally in peripheral hardware rather than internally in main processor software.

Intelligent peripherals also provide system flexibility. They contain on-chip mode registers which are programmed by the master processor during system initialization. These control registers allow the peripheral to be configured into many different operation modes. The user-defined program for the peripheral is stored in main system memory and is transferred to the peripheral's registers whenever a mode change is required. Of course, this type of flexibility requires software overhead in the master system which tends to limit the benefit derived from the peripheral chip.

In the past, intelligent peripherals were designed to handle very specialized tasks. Separate chips were designed for communication disciplines, parallel I/O, keyboard encoding, interval timing, CRT control, etc. Yet, in spite of the large number of devices available and the increased flexibility built into these chips, there is still a large number of microcomputer peripheral control tasks which are not satisfied.

With the introduction of the Universal Peripheral Interface (UPI) microcomputer, Intel has taken the intelligent peripheral concept a step further by providing an intelligent controller that is fully user programmable. It is a complete single-chip microcomputer which can connect directly to a master processor data bus. It has the same advantages of intelligence and flexibility which previous peripheral chips offered. In addition, UPIs are user-programmable: it has 1K/2K bytes of ROM or EPROM memory for program storage plus 64/128/256 bytes of RAM memory UPI-41A, 41AH/42, 42AH respectively for data storage or initialization from the master processor. The UPI device allows a designer to fully specify his control algorithm in the peripheral chip without relying on the master processor. Devices like printer controllers and keyboard scanners can be completely self-contained, relying on the master processor only for data transfer.

The UPI family currently consists of seven components:

- 8741A microcomputer with 1K EPROM memory
- 8741AH microcomputer with 1K OTP™ EPROM memory
- 8041AH microcomputer with 1K ROM memory
- 8742 microcomputer with 2K EPROM memory
- 8742AH microcomputer with 2K "OTP" EPROM memory
- 8042AH microcomputer with 2K ROM memory
- 8243 I/O expander device

The UPI-41A/41AH/42/42AH family of microcomputers are functionally equivalent except for the type and amount of program memory available with each. In addition, the UPI-41AH/42AH family has a Signature Row outside the EPROM Array. The UPI-41AH/42AH family also has a Security Feature which renders the EPROM Array unreadable when set.

All UPI's have the following main features:

- 8-bit CPU
- 8-bit data bus interface registers
- Interval timer/event counter
- Two 8-bit TTL compatible I/O ports
- Resident clock oscillator circuits

The UPI family has the following differences:

Table 1-2

UPI-41A	UPI-42	UPI-41AH	UPI-42AH
1K x 8 EPROM	2K x 8 EPROM	1K x 8 ROM or 1K x 8 OTP	2K x 8 ROM or 2K x 8 OTP
64 x 8 RAM	128 x 8 RAM	128 x 8 RAM	256 x 8 RAM
		*Set Security Feature **Signature Row Feature 32 Bytes with: 1. Test Code/Checksum 2. Intel Signature 3. Security Byte 4. User Signature	
PROGRAMMING			
UPI-41A	**UPI-42**	**UPI-41AH/UPI-42AH**	
V_{DD} = 25V	21V	12.5V	
I_{DD} = 50 ms	50 mA	30 mA	
EA = 21.5V–24.5V	18V	12.5V	
V_{PH} = 21.5V–24.5V	18V	20.V–5.5V	
TPW = 50 ms	50 ms	1 ms	
PIN DESCRIPTION			
UPI-41A/UPI-42		**UPI-41AH/UPI-42AH**	
(T1) T1 functions as a test input which can be directly tested using conditional branching instructions. It functions as the event timer input under software control.		T1 functions as a test input that can be directly tested using conditional branching instructions. It works as the event timer input under software control. It is used during sync mode to reset the instruction state to S1 and synchronize the internal clock to phase 1.	
(SS) Single step input used with the sync output to step the program through each instruction.		Single step input used with the sync output to step the program through each instruction. This pin is used to put the device in sync mode by applying +12.5V to it.	
Port 1 (P10–P17): 8-bit, Quasi-Bidirectional I/O Lines.		Port 1 (P10–P17): 8-bit, Quasi-Bidirectional I/O Lines. P10–P17 access the Signature Row and Security Bit.	

NOTES:
*For a complete description of the Security Feature, refer to the UPI-41AH/42AH Datasheet.
**For a complete description of the Signature Row, refer to the UPI-41AH/42AH Datasheet.

6

HMOS processing has been applied to the UPI family to allow for additional performance and memory capability while reducing costs. The UPI-41A/41AH/42/42AH are all pin and software compatible. This allows growth in present designs to incorporate new features and add additional performance. For new designs, the additional memory and performance of the UPI-41A/41AH/42/42AH extends the UPI 'grow your own solution' concept to more complex motor control tasks, 80-column printers and process control applications as examples.

The 8243 device is an I/O multiplexer which allows expansion of I/O to over 100 lines (if seven devices are used). All three parts are fabricated with N-channel MOS technology and require a single, 5V supply for operation.

INTERFACE REGISTERS FOR MULTI-PROCESSOR CONFIGURATIONS

In the normal configuration, the UPI-41A/41AH/42/42AH interfaces to the system bus, just like any intelligent peripheral device (see Figure 1-1). The host processor and the UPI-41A/41AH/42/42AH form a loosely coupled multi-processor system, that is, communications between the two processors are direct. Common resources are three addressable registers located physically on the UPI-41A/41AH/42/42AH. These registers are the Data Bus Buffer Input (DBBIN), Data Bus Buffer Output (DBBOUT), and Status (STATUS) registers. The host processor may read data from DBBOUT or write commands and data into DBBIN. The status of DBBOUT and DBBIN plus user-defined status is supplied in STATUS. The host may read STATUS at any time. An interrupt to the UPI processor is automatically generated (if enabled) when DBBIN is loaded.

Because the UPI contains a complete microcomputer with program memory, data memory, and CPU it can function as a "Universal" controller. A designer can program the UPI to control printers, tape transports, or multiple serial communication channels. The UPI can also handle off-line arithmetic processing, or any number of other low speed control tasks.

POWERFUL 8-BIT PROCESSOR

The UPI contains a powerful, 8-bit CPU with as fast as 1.2 μsec cycle time and two single-level interrupts. Its instruction set includes over 90 instructions for easy software development. Most instructions are single byte and single cycle and none are more than two bytes long. The instruction set is optimized for bit manipulation and I/O operations. Special instructions are included to allow binary or BCD arithmetic operations, table look-up routines, loop counters, and N-way branch routines.

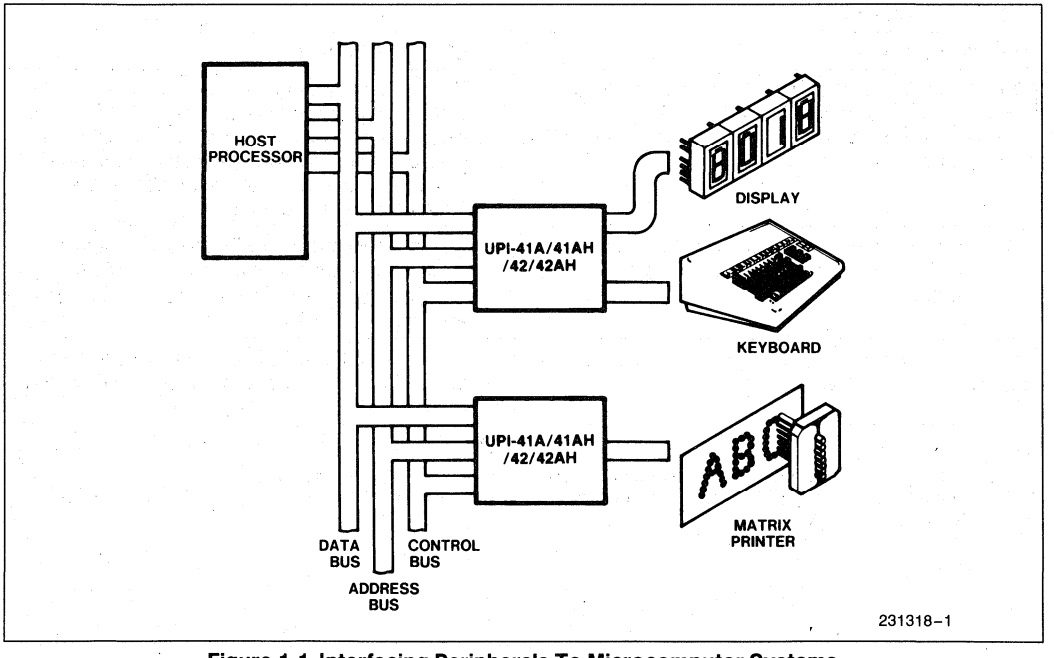


Figure 1-1. Interfacing Peripherals To Microcomputer Systems

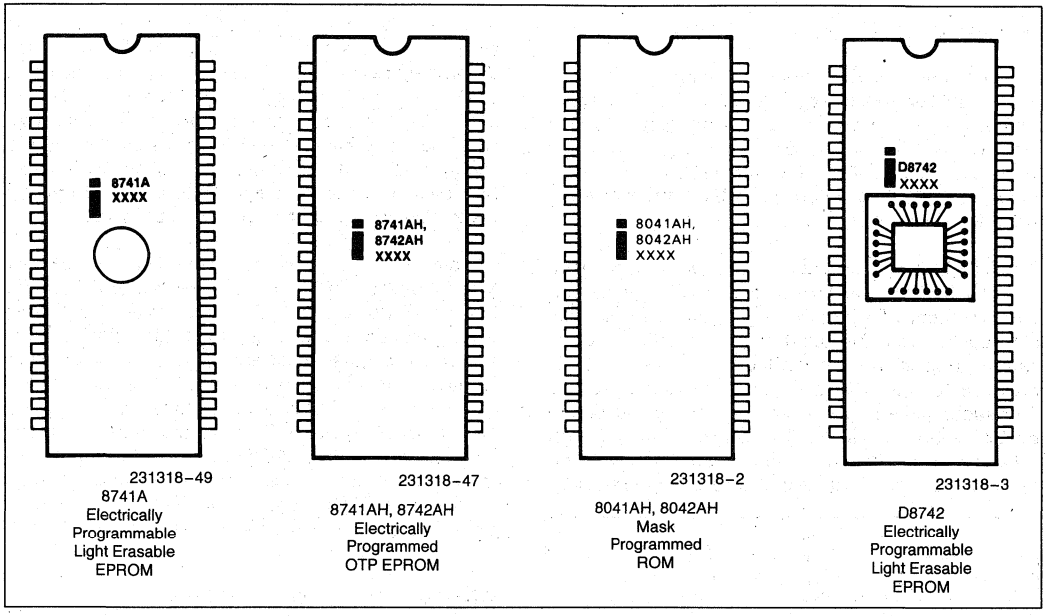


Figure 1-2. Pin Compatible ROM/EPROM Versions

SPECIAL INSTRUCTION SET FEATURES

- For Loop Counters:
  Decrement Register and Jump if not zero.
- For Bit Manipulation:
  AND to A (immediate data or Register)
  OR to A (immediate data or Register)
  XOR to A (immediate data or Register)
  AND to Output Ports (Accumulator)
  OR to Output Ports (Accumulator)
  Jump Conditionally on any bit in A
- For BDC Arithmetic:
  Decimal Adjust A
  Swap 4-bit Nibbles of A
  Exchange lower nibbles of A and Register
  Rotate A left or right with or without Carry
- For Lookup Tables:
  Load A from Page of ROM (Address in A)
  Load A from Current Page of ROM (Address in A)

6

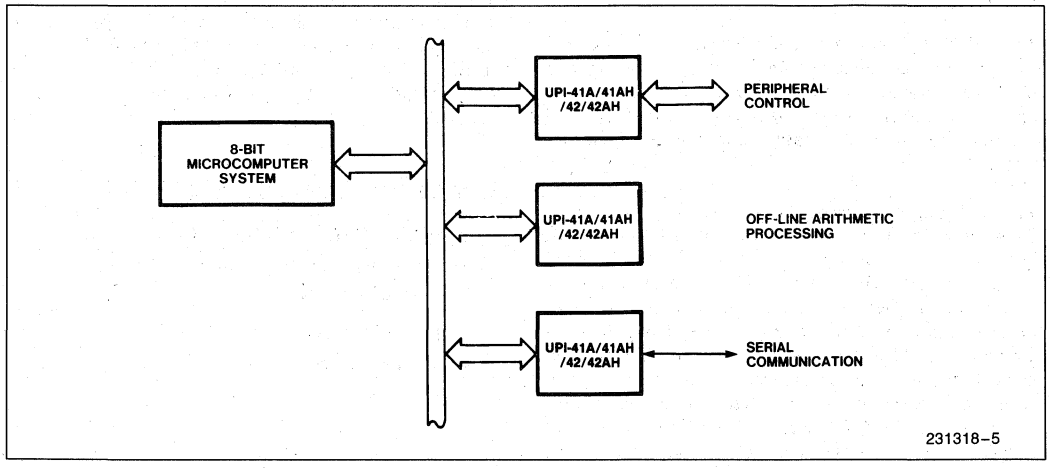


Figure 1-3. Interfaces and Protocols for Multiprocessor Systems

Features for Peripheral Control

The UPI 8-bit interval timer/event counter can be used to generate complex timing sequences for control applications or it can count external events such as switch closures and position encoder pulses. Software timing loops can be simplified or eliminated by the interval timer. If enabled, an interrupt to the CPU will occur when the timer overflows.

The UPI I/O complement contains two TTL-compatible 8-bit bidirectional I/O ports and two general-purpose test inputs. Each of the 16 port lines can individually function as either input or output under software control. Four of the port lines can also function as an interface for the 8243 I/O expander which provides four additional 4-bit ports that are directly addressable by UPI software. The 8243 expander allows low cost I/O expansion for large control applications while maintaining easy and efficient software port addressing.

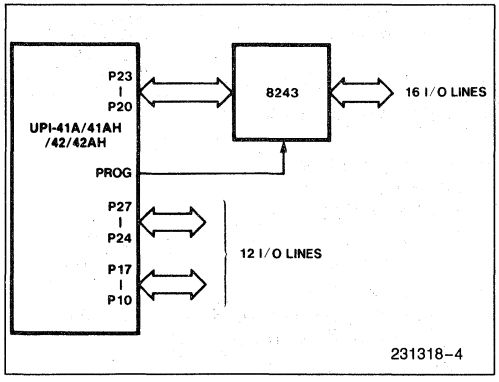


Figure 1-4. 8243 I/O Expander Interface

On-Chip Memory

The UPI's 64/128/256 bytes data memory include dual working register banks and an 8-level program counter stack. Switching between the register banks allows fast response to interrupts. The stack is used to store return addresses and processor status upon entering a subroutine.

The UPI program memory is available in three types to allow flexibility in moving from design to prototype to production with the same PC layout. The 8741A/8742 device with EPROM memory is very economical for initial system design and development. Its program memory can be electrically programmed using the Intel Universal PROM Programmer. When changes are needed, the entire program can be erased using UV lamp and reprogrammed in about 20 minutes. This means the 8741A/8742 can be used as a single chip "breadboard" for very complex interface and control problems. After the 8741A/8742 is programmed it can be tested in the actual production level PC board and the actual functional environment. Changes required during system debugging can be made in the 8741A/8742 program much more easily than they could be made in a random logic design. The system configuration and PC layout can remain fixed during the development process and the turn around time between changes can be reduced to a minimum.

At any point during the development cycle, the 8741A/8742 EPROM part can be replaced with the low cost UPI-41AH/42AH respectively with factory mask programmed memory or OTP EPROM. The transition from system development to mass production is made smoothly because the 8741A/8742, 8741AH and 8041AH, 8742AH and 8042AH parts are completely pin compatible. This feature allows extensive testing with the EPROM part, even into initial shipments to customers. Yet, the transition to low-cost ROMs or OTP EPROM is simplified to the point of being merely a package substitution.

PREPROGRAMMED UPI's

The 8242AH, 8292, and 8294 are 8042AH's that are programmed by Intel and sold as standard peripherals. The 8242AH is a keyboard controller for AT-compatible systems. The device contains Phoenix Technologies, Ltd. keyboard controller firmware for AT-compatible systems. The 8292 is a GPIB controller, part of a three chip GPIB system. The 8294 is a Data Encryption Unit that implements the National Bureau of Standards data encryption algorithm. These parts illustrate the great flexibility offered by the UPI family.

DEVELOPMENT SUPPORT

The UPI microcomputer is fully supported by Intel with development tools like the UPP PROM programmer already mentioned. The combination of device features and Intel development support make the UPI an ideal component for low-speed peripheral control applications.

UPI DEVELOPMENT SUPPORT

- 8048/UPI-41A/41AH/42/42AH Assembler
- Universal PROM Programmer UPP Series
- Insite User's Library
- Application Engineers
- Training Courses

CHAPTER 2
FUNCTIONAL DESCRIPTION

The UPI microcomputer is an intelligent peripheral controller designed to operate in iAPX-86, 88, MCS-85, MCS-80, MCS-51 and MCS-48 systems. The UPI's architecture, illustrated in Figure 2-1, is based on a low cost, single-chip microcomputer with program memory, data memory, CPU, I/O, event timer and clock oscillator in a single 40-pin package. Special interface registers are included which enable the UPI to function as a peripheral to an 8-bit master processor.

This chapter provides a basic description of the UPI microcomputer and its system interface registers. Unless otherwise noted the descriptions in this section apply to the 8741AH, 8742AH with OTP EPROM memory, the 8741A/8742 (with UV erasable program memory) and the 8041AH, 8042AH (with factory mask programmed memory). These devices are so similar that they can be considered identical under most circumstances. All functions described in this chapter apply to the UPI-41A/41AH/42/42AH.

PIN DESCRIPTION

The UPI-41A/41AH/42/42AH are packaged in 40-pin Dual In-Line (DIP) packages. The pin configuration for both devices is shown in Figure 2-2. Figure 2-3 illustrates the UPI Logic Symbol.

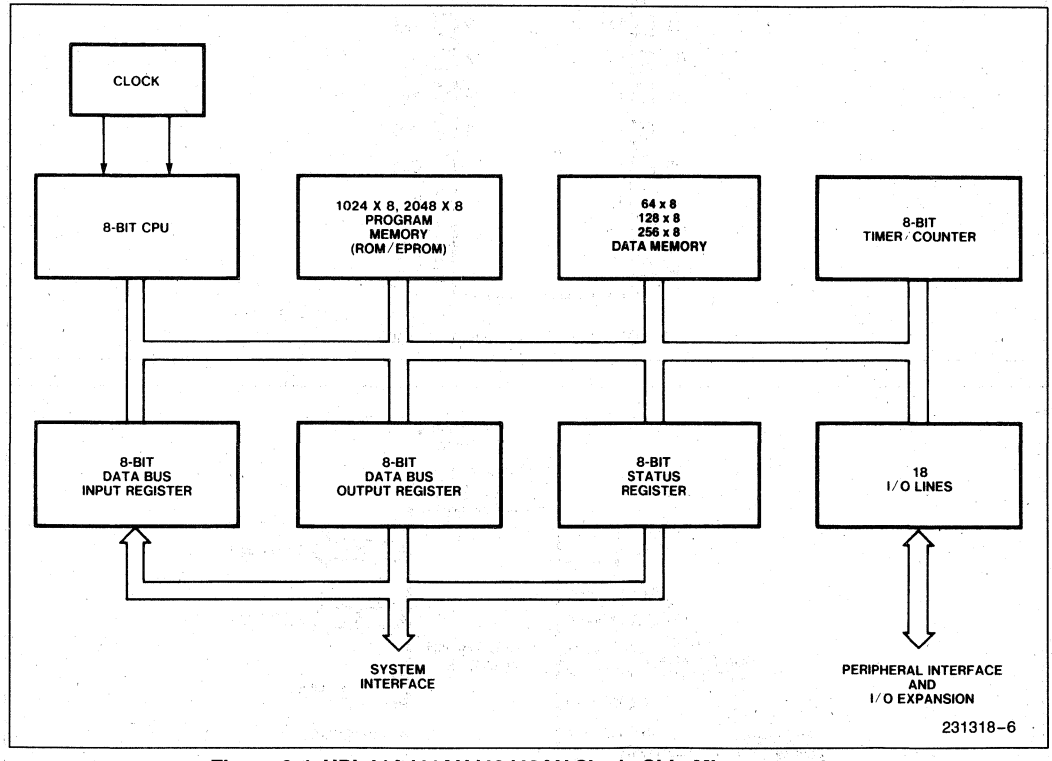


Figure 2-1. UPI-41A/41AH/42/42AH Single Chip Microcomputer

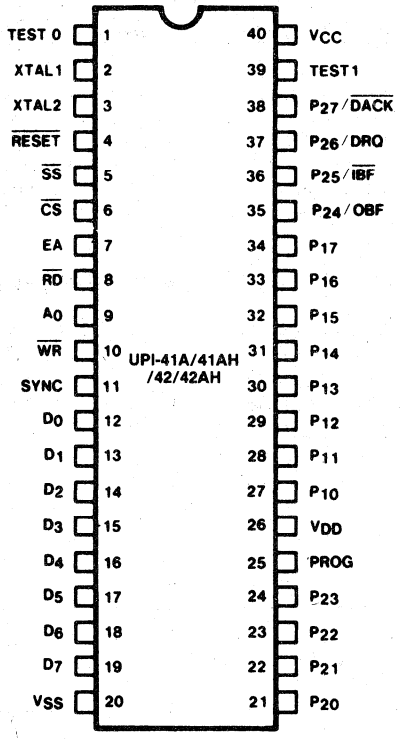


Figure 2-2. Pin Configuration

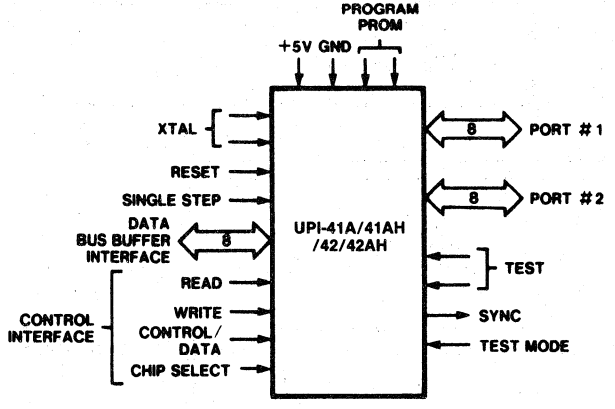


Figure 2-3. Logic Symbol

The following section summarizes the functions of each UPI pin. NOTE that several pins have two or more functions which are described in separate paragraphs.

Table 2-1. Pin Description

Symbol	Pin No.	Type	Name and Function
D_0–D_7 (BUS)	12–19	I/O	**DATA BUS:** Three-state, bidirectional DATA BUS BUFFER lines used to interface the UPI-41A/41AH/42/42AH microcomputer to an 8-bit master system data bus.
P_{10}–P_{17}	27-34	I/O	**PORT 1:** 8-bit, PORT 1 quasi-bidirectional I/O lines.
P_{20}–P_{27}	21-24 35–38	I/O	**PORT 2:** 8-bit, PORT 2 quasi-bidirectional I/O lines. The lower 4 bits (P_{20}–P_{23}) interface directly to the 8243 I/O expander device and contain address and data information during PORT 4–7 access. The upper 4 bits (P_{24}–P_{27}) can be programmed to provide interrupt Request and DMA Handshake capability. Software control can configure P_{24} as Output Buffer Full ($\overline{OBF}$) interrupt, P_{25} as Input Buffer Full ($\overline{IBF}$) interrupt, P_{26} as DMA Request ($\overline{DRQ}$), and P_{27} as DMA ACKnowledge ($\overline{DACK}$).
$\overline{WR}$	10	I	**WRITE:** I/O write input which enables the master CPU to write data and command words to the UPI INPUT DATA BUS BUFFER.
$\overline{RD}$	8	I	**READ:** I/O read input which enables the master CPU to read data and status words from the OUTPUT DATA BUS BUFFER or status register.
$\overline{CS}$	6	I	**CHIP SELECT:** Chip select input used to select one UPI-41A/41AH/42/42AH microcomputer out of several connected to a common data bus.
A_0	9	I	**COMMAND/DATA SELECT:** Address input used by the master processor to indicate whether byte transfer is data ($A_0 = 0$) or command ($A_0 = 1$).
TEST 0, TEST 1	1 39	I	**TEST INPUTS:** Input pins can be directly tested using conditional branch instructions. **FREQUENCY REFERENCE:** TEST 1 (T_1) also functions as the event timer input (under software control). TEST0 (T_0) is used during PROM programming and verification in the UPI-41A/41AH/42/42AH.
XTAL 1, XTAL 2	2 3	I	**INPUTS:** Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
SYNC	11	O	**OUTPUT CLOCK:** Output signal which occurs once per UPI instruction cycle. SYNC can be used as a strobe for external circuitry; it is also used to synchronize single step operation.
EA	7	I	**EXTERNAL ACCESS:** External access input which allows emulation, testing and PROM/ROM verification.
PROG	25	I/O	**PROGRAM:** Multifunction pin used as the program pulse input during PROM programming. During I/O expander access the PROG pin acts as an address/data strobe to the 8243.
$\overline{RESET}$	4	I	**RESET:** Input used to reset status flip-flops and to set the program counter to zero. $\overline{RESET}$ is also used during PROM programming and verification.
$\overline{SS}$	5	I	**SINGLE STEP:** Single step input used in conjunction with the SYNC output to step the program through each instruction.
V_{CC}	40		**POWER:** +5V main power supply pin.
V_{DD}	26		**POWER:** +5V during normal operation. +25V for UPI-41A, 21V for UPI-42 programming operation, +12V for programming, UPI-41AH/42AH. Low power standby pin in ROM version.
V_{SS}	20		**GROUND:** Circuit ground potential.

The following sections provide a detailed functional description of the UPI microcomputer. Figure 2-4 illustrates the functional blocks within the UPI device.

CPU SECTION

The CPU section of the UPI-41A/41AH/42/42AH microcomputer performs basic data manipulations and controls data flow throughout the single chip computer via the internal 8-bit data bus. The CPU section includes the following functional blocks shown in Figure 2-4:

- Arithmetic Logic Unit (ALU)
- Instruction Decoder
- Accumulator
- Flags

Arithmetic Logic Units (ALU)

The ALU is capable of performing the following operations:

- ADD with or without carry
- AND, OR, and EXCLUSIVE OR
- Increment, Decrement
- Bit complement
- Rotate left or right
- Swap
- BCD decimal adjust

In a typical operation data from the accumulator is combined in the ALU with data from some other source on the UPI-41A/41AH/42/42AH internal bus (such as a register or an I/O port). The result of an ALU operation can be transferred to the internal bus or back to the accumulator.

If an operation such as an ADD or ROTATE requires more than 8 bits, the CARRY flag is used as an indicator. Likewise, during decimal adjust and other BCD operations the AUXILIARY CARRY flag can be set and acted upon. These flags are part of the Program Status Word (PSW).

Instruction Decoder

During an instruction fetch, the operation code (opcode) portion of each program instruction is stored and decoded by the instruction decoder. The decoder generates outputs used along with various timing signals to control the functions performed in the ALU. Also, the instruction decoder controls the source and destination of ALU data.

Accumulator

The accumulator is the single most important register in the processor. It is the primary source of data to the ALU and is often the destination for results as well. Data to and from the I/O ports and memory normally passes through the accumulator.

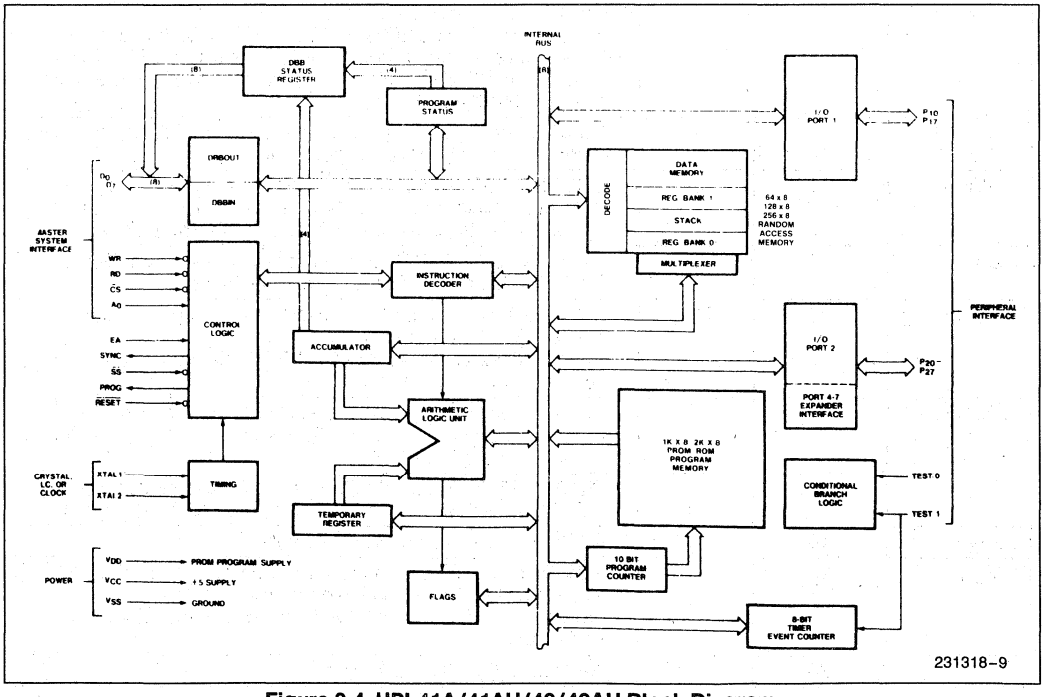


Figure 2-4. UPI-41A/41AH/42/42AH Block Diagram

PROGRAM MEMORY

The UPI-41A/41AH/42/42AH microcomputer has 1024, 2048 8-bit words of resident, read-only memory for program storage. Each of these memory locations is directly addressable by a 10-bit program counter. Depending on the type of application and the number of program changes anticipated, three types of program memory are available:

- 8041AH, 8042AH with mask programmed ROM Memory
- 8741AH, 8742AH with electrically programmable OTP EPROM Memory
- 8741A and 8742 with electrically programmable EPROM Memory

A program memory map is illustrated in Figure 2-5. Memory is divided into 256 location 'pages' and three locations are reserved for special use:

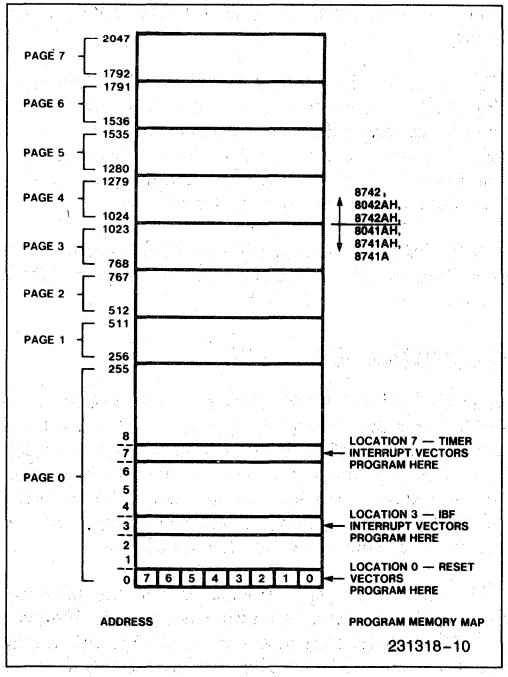


Figure 2-5. Program Memory Map

INTERRUPT VECTORS

1) **Location 0**
Following a $\overline{\text{RESET}}$ input to the processor, the next instruction is automatically fetched from location 0.

2) **Location 3**
An interrupt generated by an Input Buffer Full (IBF) condition (when the IBF interrupt is enabled) causes the next instruction to be fetched from location 3.

3) **Location 7**
A timer overflow interrupt (when enabled) will cause the next instruction to be fetched from location 7.

Following a system $\overline{\text{RESET}}$, program execution begins at location 0. Instructions in program memory are normally executed sequentially. Program control can be transferred out of the main line of code by an input buffer full (IBF) interrupt or a timer interrupt, or when a jump or call instruction is encountered. An IBF interrupt (if enabled) will automatically transfer control to location 3 while a timer interrupt will transfer control to location 7.

All conditional JUMP instructions and the indirect JUMP instruction are limited in range to the current 256-location page (that is, they alter PC bits 0–7 only). If a conditional JUMP or indirect JUMP begins in location 255 of a page, it must reference a destination on the following page.

Program memory can be used to store constants as well as program instructions. The UPI-41AH, 42AH instruction set contains an instruction (MOVP3) designed specifically for efficient transfer of look-up table information from page 3 of memory.

6

DATA MEMORY

The UPI-41A has 64 8-bit words of Random Access Memory, the UPI-41AH has 128 8-bit words of Random Access Memory; the UPI-42 has 128 8-bit words of RAM; and the UPI-42AH has 256 8-bit words of RAM. This memory contains two working register banks, an 8-level program counter stack and a scratch pad memory, as shown in Figure 2-6. The amount of scratch pad memory available is variable depending on the number of addresses nested in the stack and the number of working registers being used.

Addressing Data Memory

The first eight locations in RAM are designated as working registers R_0–R_7. These locations (or registers) can be addressed directly by specifying a register number in the instruction. Since these locations are easily addressed, they are generally used to store frequently

accessed intermediate results. Other locations in data memory are addressed indirectly by using R_0 or R_1 to specify the desired address.

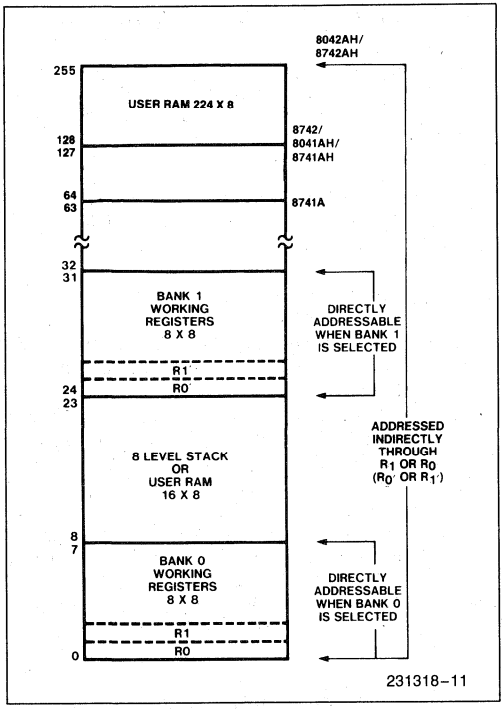


Figure 2-6. Data Memory Map

Working Registers

Dual banks of eight working registers are included in the UPI-41A/41AH/42/42AH data memory. Locations 0–7 make up register bank 0 and locations 24–13 form register bank 1. A $\overline{\text{RESET}}$ signal automatically selects register bank 0. When bank 0 is selected, references to R_0–R_7 in UPI-41A/41AH/42/42AH instructions operate on locations 0–7 in data memory. A "select register bank" instruction is used to selected between the banks during program execution. If the instruction SEL RB1 (Select Register Bank 1) is executed, then program references to R_0–R_7 will operate on locations 24–31. As stated previously, registers 0 and 1 in the active register bank are used as indirect address registers for all locations in data memory.

Register bank 1 is normally reserved for handling interrupt service routines, thereby preserving the contents of the main program registers. The SEL RB1 instruction can be issued at the beginning of an interrupt service routine. Then, upon return to the main program, an RETR (return & restore status) instruction will automatically restore the previously selected bank. During interrupt processing, registers in bank 0 can be accessed indirectly using $R_{0'}$ and $R_{1'}$.

If register bank 1 is not used, registers 24–31 can still serve as additional scratch pad memory.

Program Counter Stack

RAM locations 8–23 are used as an 8-level program counter stack. When program control is temporarily passed from the main program to a subroutine or interrupt service routine, the 10-bit program counter and bits 4–7 of the program status word (PSW) are stored in two stack locations. When control is returned to the main program via an RETR instruction, the program counter and PSW bits 4–7 are restored. Returning via an RET instruction does not restore the PSW bits, however. The program counter stack is addressed by three stack pointer bits in the PSW (bits 0–2). Operation of the program counter stack and the program status word is explained in detail in the following sections.

The stack allows up to eight levels of subroutine 'nesting'; that is, a subroutine may call a second subroutine, which may call a third, etc., up to eight levels. Unused stack locations can be used as scratch pad memory. Each unused level of subroutine nesting provides two additional RAM locations for general use.

The following sections provide a detailed description of the Program Counter Stack and the Program Status Word.

PROGRAM COUNTER

The UPI-41A/41AH/42/42AH microcomputer has a 10-bit program counter (PC) which can directly address any of the 1024, 2048 locations in program memory. The program counter always contains the address of the next instruction to be executed and is normally incremented sequentially for each instruction to be executed when each instruction fetches occurs.

When control is temporarily passed from the main program to a subroutine or an interrupt routine, however, the PC contents must be altered to point to the address of the desired routine. The stack is used to save the current PC contents so that, at the end of the routine, main program execution can continue. The program counter is initialized to zero by a $\overline{\text{RESET}}$ signal.

PROGRAM COUNTER STACK

The Program Counter Stack is composed of 16 locations in Data Memory as illustrated in Figure 2-7. These RAM locations (8 through 23) are used to store the 10-bit program counter and 4 bits of the program status word.

An interrupt or Call to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the program counter stack.

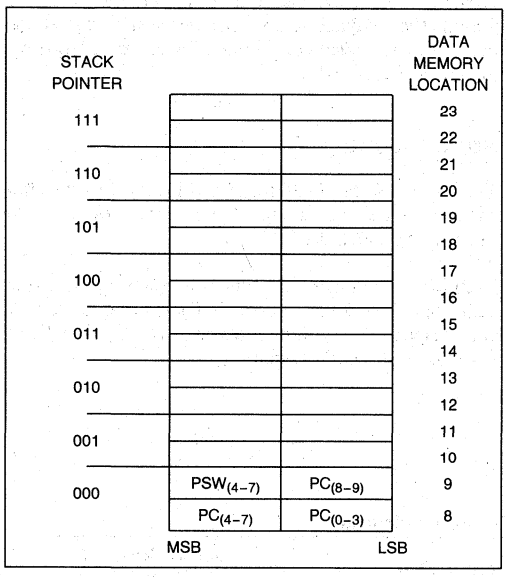


Figure 2-7. Program Counter Stack

A 3-bit Stack Pointer which is part of the Program Status Word (PSW) determines the stack pair to be used at a given time. The stack pointer is initialized by a RESET signal to 00H which corresponds to RAM locations 8 and 9.

The first call or interrupt results in the program counter and PSW contents being transferred to RAM locations 8 and 9 in the format shown in Figure 2-7. The stack pointer is automatically incremented by 1 to point to location is 10 and 11 in anticipation of another CALL.

Nesting of subroutines within subroutines can continue up to 8 levels without overflowing the stack. If overflow does occur the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 07H to 00H. Likewise, the stack pointer will underflow from 00H to 07H.

The end of a subroutine is signaled by a return instruction, either RET or RETR. Each instruction will automatically decrement the Stack Pointer and transfer the contents of the proper RAM register pair to the Program Counter.

PROGRAM STATUS WORD

The 8-bit program status word illustrated in Figure 2-8 is used to store general information about program execution. In addition to the 3-bit Stack Pointer discussed previously, the PSW includes the following flags:

- CY — Carry
- AC — Auxiliary Carry
- F_0 — Flag 0
- BS — Register Bank Select

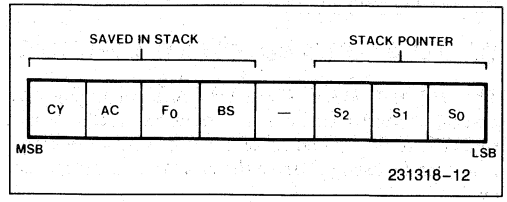


Figure 2-8. Program Status Word

The Program Status Word (PSW) is actually a collection of flip-flops located throughout the machine which are read or written as a whole. The PSW can be loaded to or from the accumulator by the MOV A, PSW or MOV PSW, A instructions. The ability to write directly to the PSW allows easy restoration of machine status after a power-down sequence.

The upper 4 bits of the PSW (bits 4, 5, 6, and 7) are stored in the PC Stack with every subroutine CALL or interrupt vector. Restoring the bits on a return is optional. The bits are restored if an RETR instruction is executed, but not if an RET is executed.

PSW bit definitions are as follows:

- Bits 0–2 Stack Pointer Bits S_0, S_1, S_2
- Bit 3 Not Used
- Bit 4 Working Register Bank
  0 = Bank 0
  1 = Bank 1
- Bit 5 Flag 0 bit (F_0)
  This is a general purpose flag which can be cleared or complemented and tested with conditional jump instructions. It may be used during data transfer to an external processor.
- Bit 6 Auxiliary Carry (AC)
  The flag status is determined by an ADD instruction and is used by the Decimal Adjustment instruction DAA
- Bit 7 Carry (CY)
  The flag indicates that a previous operation resulted in overflow of the accumulator.

CONDITIONAL BRANCH LOGIC

Conditional Branch Logic in the UPI-41AH, 42AH allows the status of various processor flags, inputs, and other hardware functions to directly affect program execution. The status is sampled in state 3 of the first cycle.

Table 2-2 lists the internal conditions which are testable and indicates the condition which will cause a jump. In all cases, the destination address must be within the page of program memory (256 locations) in which the jump instruction occurs.

OSCILLATOR AND TIMING CIRCUITS

The UPI-41A/41AH/42/42AH's internal timing generation is controlled by a self-contained oscillator and timing circuit. A choice of crystal, L-C or external clock can be used to derive the basic oscillator frequency.

The resident timing circuit consists of an oscillator, a state counter and a cycle counter as illustrated in Figure 2-9. Figure 2-10 shows instruction cycle timing.

Oscillator

The on-board oscillator is a series resonant circuit with a frequency range of 1 to 12.5 MHz depending on which UPI is used. Refer to Table 1.1. Pins XTAL 1 and XTAL 2 are input and output (respectively) of a high gain amplifier stage. A crystal or inductor and capacitor connected between XTAL 1 and XTAL 2 provide the feedback and proper phase shift for oscillation. Recommended connections for crystal or L-C are shown in Figure 2-11.

State Counter

The output of the oscillator is divided by 3 in the state counter to generate a signal which defines the state times of the machine.

Each instruction cycle consists of five states as illustrated in Figure 2-10 and Table 2-3. The overlap of address and execution operations illustrated in Figure 2-10 allows fast instruction execution.

Table 2-2. Conditional Branch Instructions

Device	Instruction Mnemonic		Jump Condition Jump if:
Accumulator	JZ	addr	All bits zero
	JNZ	addr	Any bit not zero
Accumulator bit	JBb	addr	Bit "b" = 1
Carry flag	JC	addr	Carry flag = 1
	JNC	addr	Carry flag = 0
User flag	JFO	addr	F_0 flag = 1
	JF1	addr	F_1 flag = 1
Timer flag	JTF	addr	Timer flag = 1
Test Input 0	JT0	addr	T_0 = 1
	JNT0	addr	T_0 = 0
Test Input 1	JT1	addr	T_1 = 1
	JNT1	addr	T_1 = 0
Input Buffer flag	JNIBF	addr	IBF flag = 0
Output Buffer flag	JOBF	addr	OBF flag = 1

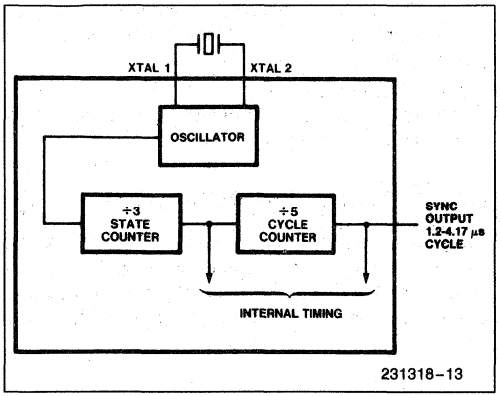


Figure 2-9. Oscillator Configuration

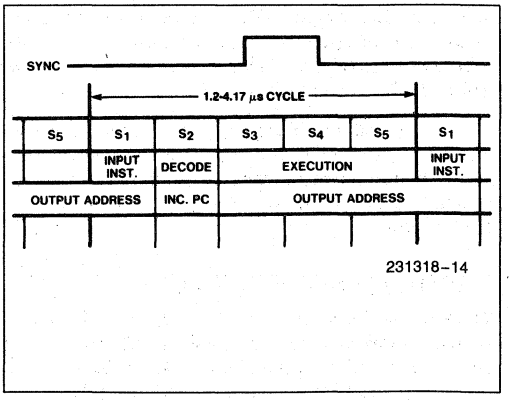


Figure 2-10. Instruction Cycle Timing

Table 2-3. Instruction Timing Diagram

Instruction	CYCLE 1					CYCLE 2				
	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,Pp	Fetch Instruction	Increment Program Counter	—	Increment Timer	—	—	Read Port	—	—	—
OUTL Pp,A	Fetch Instruction	Increment Program Counter	—	Increment Timer	Output To Port	—	—	—	—	—
ANL Pp, DATA	Fetch Instruction	Increment Program Counter	—	Increment Timer	Read Port	Fetch Immediate Data	—	Increment Program Counter	Output To Port	—
ORL Pp, DATA	Fetch Instruction	Increment Program Counter	—	Increment Timer	Read Port	Fetch Immediate Data	—	Increment Program Counter	Output To Port	—
MOVD A,Pp	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	—	—	Read P2 Lower	—	—	—
MOVD Pp, A	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	Output Data To P2 Lower	—	—	—	—	—
D Pp, A	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	Output Data	—	—	—	—	—
ORLD Pp, A	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	Output Data	—	—	—	—	—
J (Conditional)	Fetch Instruction	Increment Program Counter	Sample Condition	Increment Timer	—	Fetch Immediate Data	—	Update Program Counter	—	—
MOV STS, A	Fetch Instruction	Increment Program Counter	—	Increment Timer	Update Status Register					
IN A, DBB	Fetch Instruction	Increment Program Counter	—	Increment Timer	—					
OUT DBB, A	Fetch Instruction	Increment Program Counter	—	Increment Timer	Output To Port					
STRT T STRT CNT	Fetch Instruction	Increment Program Counter	—	—	Start Counter					
STOP TCNT	Fetch Instruction	Increment Program Counter	—	—	Stop Counter					
EN I	Fetch Instruction	Increment Program Counter	—	Enable Interrupt	—					
DIS I	Fetch Instruction	Increment Program Counter	—	Disable Interrupt	—					
EN DMA	Fetch Instruction	Increment Program Counter	—	DMA Enabled DRQ Cleared	—					
EN FLAGS	Fetch Instruction	Increment Program Counter	—	OBF, $\overline{IBF}$ Output Enabled	—					

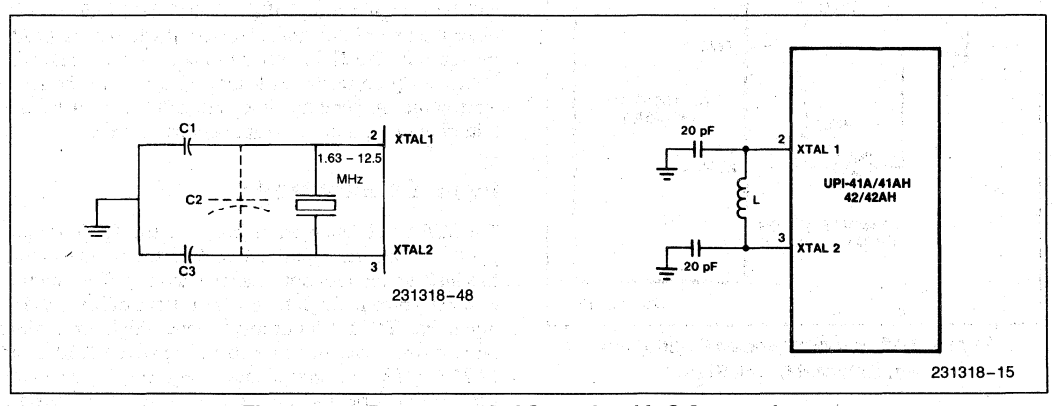


Figure 2-11. Recommended Crystal and L-C Connections

Cycle Counter

The output of the state counter is divided by 5 in the cycle counter to generate a signal which defines a machine cycle. This signal is call SYNC and is available continuously on the SYNC output pin. It can be used to synchronize external circuitry or as a general purpose clock output. It is also used for synchronizing single-step.

Frequency Reference

The external crystal provides high speed and accurate timing generation. A crystal frequency of 5.9904 MHz is useful for generation of standard communication frequencies by the UPI-41A/41AH/42/42AH. However, if an accurate frequency reference and maximum processor speed are not required, an inductor and capacitor may be used in place of the crystal as shown in Figure 2-11.

A recommended range of inductance and capacitance combinations is given below:

- L = 130 μH corresponds to 3 MHz
- L = 45 μH corresponds to 5 MHz

An external clock signal can also be used as a frequency reference to the UPI-41A/41AH/42/42AH; however, the levels are *not* TTL compatible. The signal must be in the 1–12.5 MHz frequency range depending on which UPI is used. Refer to Table 1-2. The signal must be connected to pins XTAL 1 and XTAL 2 by buffers with a suitable pull-up resistor to guarantee that a logic "1" is above 3.8 volts. The recommended connection is shown in Figure 2-12.

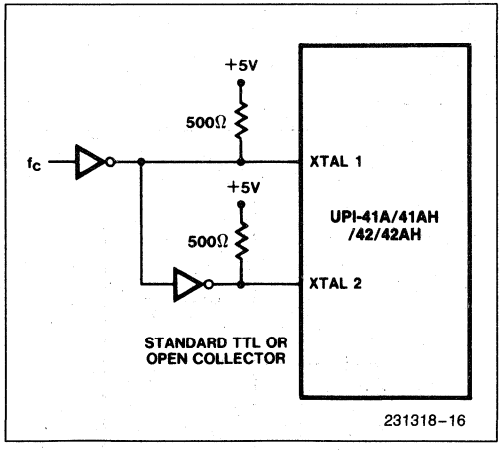


Figure 2-12. Recommended Connection For External Clock Signal

INTERVAL TIMER/EVENT COUNTER

The UPI-41A/41AH/42/42AH has a resident 8-bit timer/counter which has several software selectable modes of operation. As an interval timer, it can generate accurate delays from 80 microseconds to 20.48 milliseconds without placing undue burden on the processor. In the counter mode, external events such as switch closures or tachometer pulses can be counted and used to direct program flow.

Timer Configuration

Figure 2-13 illustrates the basic timer/counter configuration. An 8-bit register is used to count pulses from either the internal clock and prescaler or from an external source. The counter is presettable and readable with two MOV instructions which transfer the contents of the accumulator to the counter and vice-versa (i.e. MOV T, A and MOV A, T). The counter is stopped by a $\overline{\text{RESET}}$ or STOP TCNT instruction and remains stopped until restarted either as a timer (START T instruction) or as a counter (START CNT instruction). Once started, the counter will increment to its maximum count (FFH) and overflow to zero continuing its count until stopped by a STOP TCNT instruction or $\overline{\text{RESET}}$.

The increment from maximum count to zero (overflow) results in setting the Timer Flag (TF) and generating an interrupt request. The state of the overflow flag is testable with the conditional jump instruction, JTF. The flag is reset by executing a JTF or by a $\overline{\text{RESET}}$ signal.

The timer interrupt request is stored in a latch and ORed with the input buffer full interrupt request. The timer interrupt can be enabled or disabled independent of the IBF interrupt by the EN TCNTI and DIS TCTNI instructions. If enabled, the counter overflow will cause a subroutine call to location 7 where the timer service routine is stored. If the timer and Input Buffer Full interrupts occur simultaneously, the IBF source will be recognized and the call will be to location 3. Since the timer interrupt is latched, it will remain pending until the DBBIN register has been serviced and will immediately be recognized upon return from the service routine. A pending timer interrupt is reset by the initiation of a timer interrupt service routine.

Event Counter Mode

The STRT CNT instruction connects the TEST 1 input pin to the counter input and enables the counter. Note this instruction does not clear the counter. The counter is incremented on high to low transitions of the TEST 1 input. The TEST 1 input must remain high for a minimum of one state in order to be registered (250 ns at 12 MHz). The maximum count frequency is one count per three instruction cycles (267 kHz at 12 MHz). There is no minimum frequency limit.

Timer Mode

The STRT T instruction connects the internal clock to the counter input and enables the counter. The input clock is derived from the SYNC signal of the internal oscillator and the divide-by-32 prescaler. The configuration is illustrated in Figure 2-13. Note this instruction does not clear the timer register. Various delays and timing sequences between 40 μsec and 10.24 msec can easily be generated with a minimum of software timing loops (at 12 MHz).

Times longer than 10.24 msec can be accurately measured by accumulating multiple overflows in a register under software control. For time resolution less than 40 μsec, an external clock can be applied to the TEST 1 counter input (see Event Counter Mode). The minimum time resolution with an external clock is 3.75 μsec (267 kHz at 12 MHz).

TEST 1 Event Counter Input

The TEST 1 pin is multifunctional. It is automatically initialized as a test input by a $\overline{\text{RESET}}$ signal and can be tested using UPI-41A conditional branch instructions.

In the second mode of operation, illustrated in Figure 2-13, the TEST 1 pin is used as an input to the internal 8-bit event counter. The Start Counter (STRT CNT) instruction controls an internal switch which connects TEST 1 through an edge detector to the 8-bit internal counter. Note that this instruction does not inhibit the testing of TEST 1 via conditional Jump instructions.

In the counter mode the TEST 1 input is sampled once per instruction cycle. After a high level is detected, the next occurrence of a low level at TEST 1 will cause the counter to increment by one.

The event counter functions can be stopped by the Stop Timer/Counter (STOP TCNT) instruction. When this instruction is executed the TEST 1 pin becomes a test input and functions as previously described.

TEST INPUTS

There are two multifunction pins designated as Test Inputs, TEST 0 and TEST 1. In the normal mode of operation, status of each of these lines can be directly tested using the following conditional Jump instructions:

- JT0 Jump if TEST 0 = 1
- JNT0 Jump if TEST 0 = 0
- JT1 Jump if TEST 1 = 1
- JNT1 Jump if TEST 1 = 0

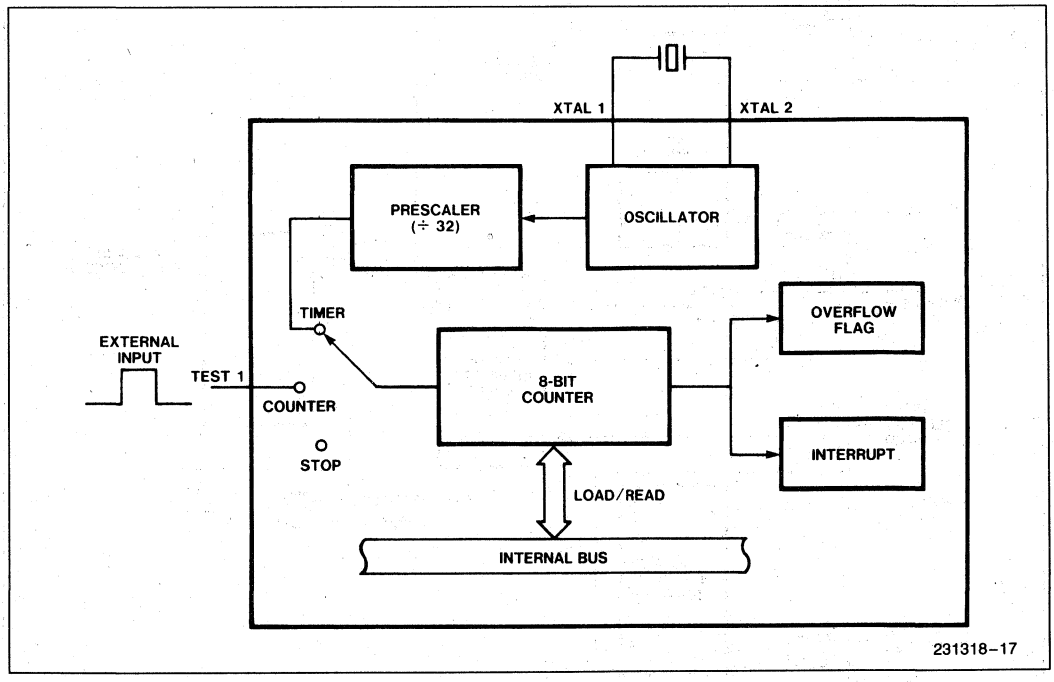


Figure 2-13. Timer Counter

The test imputs are TTL compatible. An external logic signal connected to one of the test inputs will be sampled at the time the appropriate conditional jump instruction is executed. The path of program execution will be altered depending on the state of the external signal when sampled.

INTERRUPTS

The UPI-41A/41AH/42/42AH has the following internal interrupts:

- Input Buffer Full (IBF) interrupt
- Timer Overflow interrupt

The IBF interrupt forces a CALL to location 3 in program memory; a timer-overflow interrupts forces a CALL to location 7. The IBF interrupt is enabled by the EN I instruction and disabled by the DIS I instruction. The timer-overflow interrupt is enabled and disabled by the EN TNCTI and DIS TCNTI instructions, respectively.

Figure 2-14 illustrates the internal interrupt logic. An IBF interrupt request is generated whenever $\overline{WR}$ and $\overline{CS}$ are both low, regardless of whether interrupts are enabled. The interrupt request is cleared upon entering the IBF service routine only. That is, the DIS I instruction does not clear a pending IBF interrupt.

Interrupt Timing Latency

When the IBF interrupt is enabled and an IBF interrupt request occurs, an interrupt sequence is intiated as soon as the currently executing instruction is completed. The following sequence occurs:

- A CALL to location 3 is forced.
- The program counter and bits 4–7 of the Program Status Word are stored in the stack.
- The stack pointer is incremented.

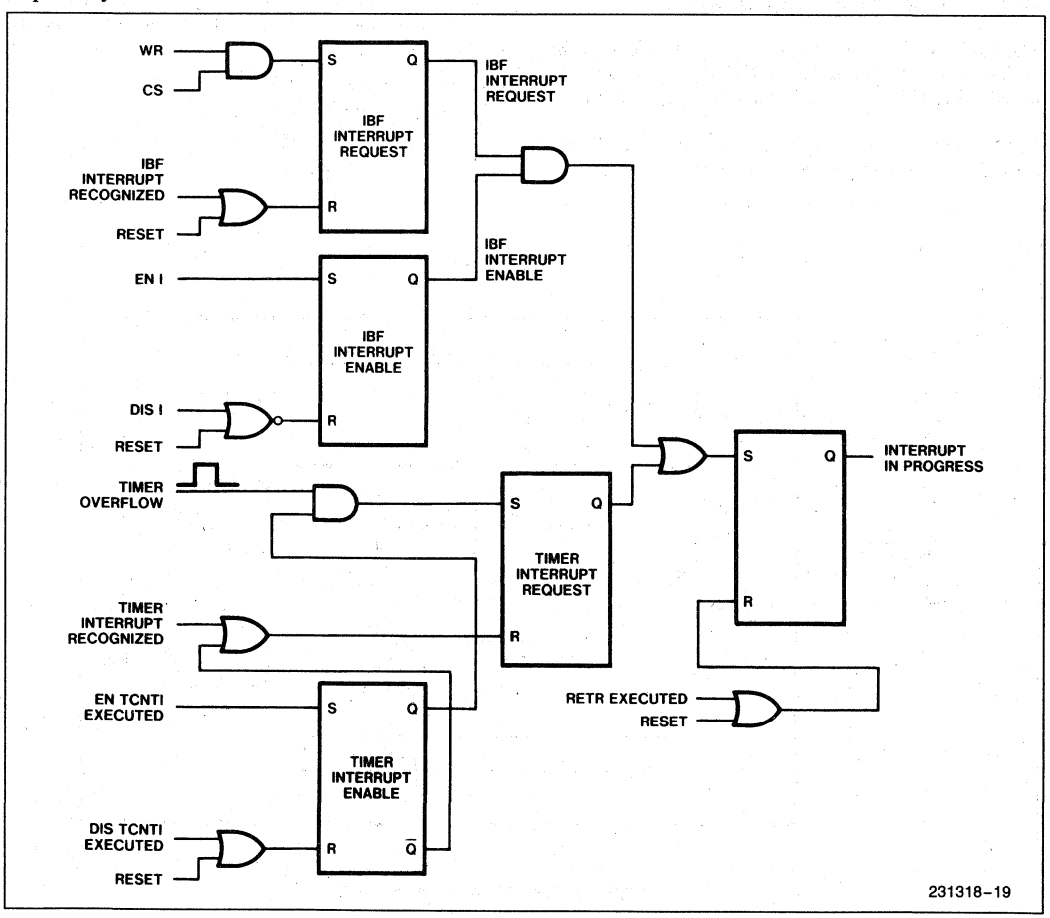


Figure 2-14. Interrupt Logic

Location 3 in program memory should contain an unconditional jump to the beginning of the IBF interrupt service routine elsewhere in program memory. At the end of the service routine, an RETR (Return and Restore Status) instruction is used to return control to the main program. This instruction will restore the program counter and PSW bits 4–7, providing automatic restoration of the previously active register bank as well. RETR also re-enables interrupts.

A timer-overflow interrupt is enabled by the EN TCNTI instruction and disabled by the DIS TCNTI instruction. If enabled, this interrupt occurs when the timer/counter register overflows. A CALL to location 7 is forced and the interrupt routine proceeds as described above.

The interrupt service latency is the sum of current instruction time, interrupt recognition time, and the internal call to the interrupt vector address. The worst case latency time for servicing an interrupt is 7 clock cycles. Best case latency is 4 clock cycles.

Interrupt Timing

Interrupt inputs may be enabled or disabled under program control using EN I, DIS I, EN TCNTI and DIS TCNTI instructions. Also, a $\overline{\text{RESET}}$ input will disable interrupts. An interrupt request must be removed before the RETR instruction is executed to return from the service routine, otherwise the processor will re-enter the service routine immediately. Thus, the $\overline{\text{WR}}$ and $\overline{\text{CS}}$ inputs should not be held low longer than the duration of the interrupt service routine.

The interrupt system is single level. Once an interrupt is detected, all further interrupt requests are latched but are not acted upon until execution of an RETR instruction re-enables the interrupt input logic. This occurs at the beginning of the second cycle of the RETR instruction. If an IBF interrupt and a timer-overflow interrupt occur simultaneously, the IBF interrupt will be recognized first and the timer-overflow interrupt will remain pending until the end of the interrupt service routine.

External Interrupts

An external interrupt can be created using the UPI-41A/41AH/42/42AH timer/counter in the event counter mode. The counter is first preset to FFH and the EN TCNTI instruction is executed. A timer-overflow interrupt is generated by the first high to low transition of the TEST 1 input pin. Also, if an IBF interrupt occurs during servicing of the timer/counter interrupt, it will remain pending until the end of the service routine.

Host Interrupts And DMA

If needed, two external interrupts to the host system can be created using the EN FLAGS instruction. This instruction allocates two I/O lines on PORT 2 (P_{24} and P_{25}). P_{24} is the Output Buffer Full interrupt request line to the host system; P_{25} is the Input Buffer empty interrupt request line. These interrupt outputs reflect the internal status of the OBF flag and the IBF inverted flag. Note, these outputs may be inhibited by writing a "0" to these pins. Reenabling interrupts is done by writing a "1" to these port pins. Interrupts are typically enabled after power on since the I/O ports are set in a "1" condition. The EN FLAG's effect is only cancelled by a device RESET.

DMA handshaking controls are available from two pins on PORT 2 of the UPI-41A/41AH/42/42AH microcomputer. These lines (P_{26} and P_{27}) are enabled by the EN DMA instruction. P_{26} becomes DMA request (DRQ) and P_{27} becomes DMA acknowledge ($\overline{\text{DACK}}$). The UPI program initiates a DMA request by writing a "1" to P_{26}. The DMA controller transfers the data into the DBBIN data register using DACK which acts as a chip select. The EN DMA instruction can only be cancelled by a chip RESET.

RESET

The $\overline{\text{RESET}}$ input provides a means for internal initialization of the processor. An automatic initialization pulse can be generated at power-on by simply connecting a 1 μfd capacitor between the $\overline{\text{RESET}}$ input and ground as shown in Figure 2-15. It has an internal pull-up resistor to charge the capacitor and a Schmitt-trigger circuit to generate a clean transition. A 2-stage synchronizer has been added to support reliable operation up to 12.5 MHz.

If automatic initialization is used, $\overline{\text{RESET}}$ should be held low for at least 10 milliseconds to allow the power supply to stabilize. If an external $\overline{\text{RESET}}$ signal is used, $\overline{\text{RESET}}$ may be held low for a minimum of 8 instruction cycles. Figure 2-15 illustrates a configuration using an external TTL gate to generate the $\overline{\text{RESET}}$ input. This configuration can be used to derive the $\overline{\text{RESET}}$ signal from the 8224 clock generator in an 8080 system.

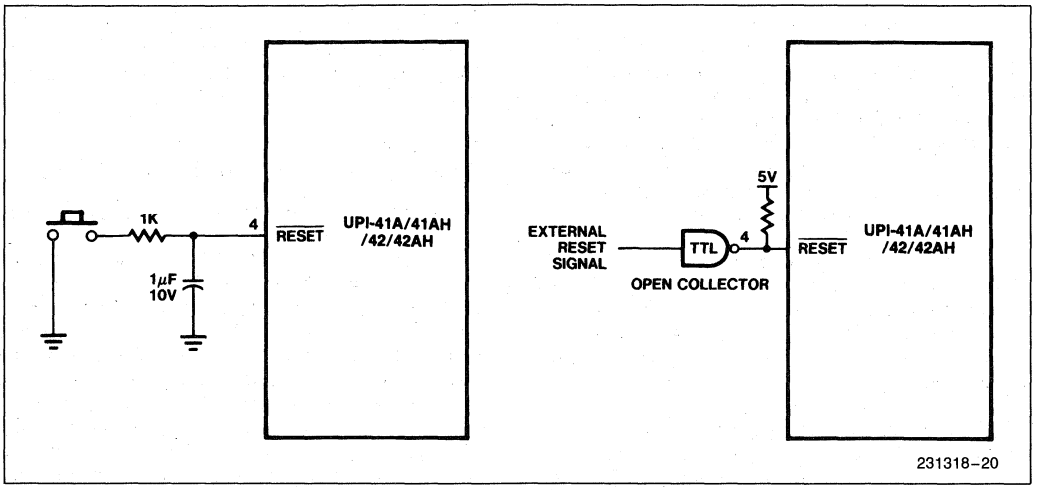


Figure 2-15. External Reset Configuration

The $\overline{\text{RESET}}$ input performs the following functions:

- Disables Interrupts
- Clears Program Counter to Zero
- Clears Stack Pointer
- Clears Status Register and Flags
- Clears Timer and Timer Flag
- Stops Timer
- Selects Register Bank 0
- Sets PORTS 1 and 2 to Input Mode

DATA BUS BUFFER

Two 8-bit data bus buffer registers, DBBIN and DBBOUT, serve as temporary buffers for commands and data flowing between it and the master processor. Externally, data is transmitted or received by the DBB registers upon execution of an INput or OUTput instruction by the master processor. Four control signals are used:

- A_0 Address input signifying control or data
- $\overline{\text{CS}}$ Chip Select
- $\overline{\text{RD}}$ Read Strobe
- $\overline{\text{WR}}$ Write Strobe

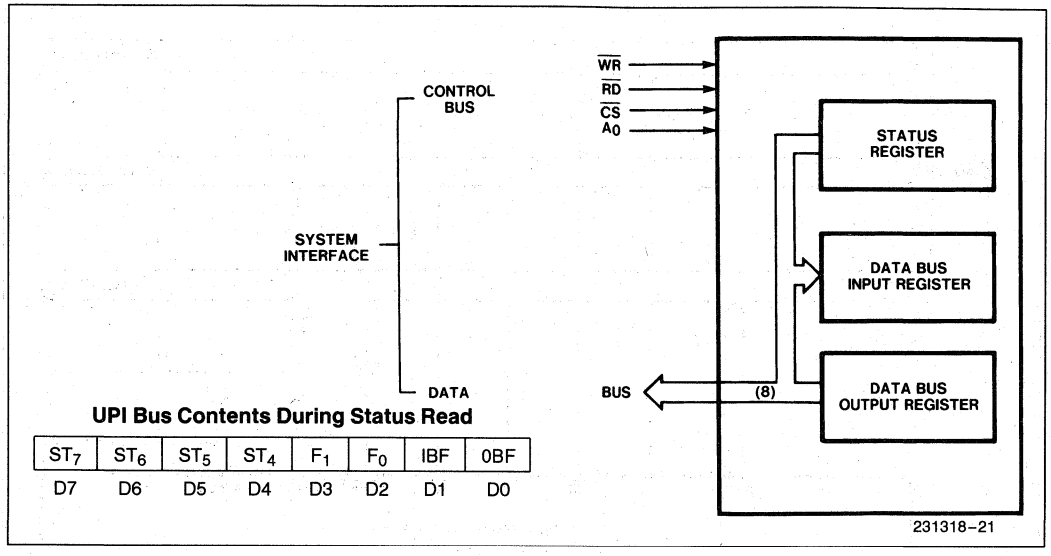
Transfer can be implemented with or without UPI program interference by enabling or disabling an internal UPI interrupt. Internally, data transfer between the DBB and the UPI accumulator is under software control and is completely asynchronous to the external processor timing. This allows the UPI software to handle peripheral control tasks independent of the main processor while still maintaining a data interface with the master system.

Configuration

Figure 2-16 illustrates the internal configuration of the DBB registers. Data is stored in two 8-bit buffer registers, DBBIN and DBBOUT. DBBIN and DBBOUT may be accessed by the external processor using the $\overline{\text{WR}}$ line and the $\overline{\text{RD}}$ line, respectively. The data bus is a bidirectional, three-state bus which can be connected directly to an 8-bit microprocessor system. Four control lines ($\overline{\text{WR}}$, $\overline{\text{RD}}$, $\overline{\text{CS}}$, A_0) are used by the external processor to transfer data to and from the DBBIN and DBBOUT registers.

An 8-bit register containing status flags is used to indicate the status of the DBB registers. The eight status flags are defined as follows:

- **OBF Output Buffer Full**
  This flag is automatically set when the UPI-Microcomputer loads the DBBOUT register and is cleared when the master processor reads the data register.
- **IBF Input Buffer Full**
  This flag is set when the master processor writes a character to the DBBIN register and is cleared when the UPI INputs the data register contents to its accumulator.



UPI Bus Contents During Status Read

ST_7	ST_6	ST_5	ST_4	F_1	F_0	IBF	OBF
D7	D6	D5	D4	D3	D2	D1	D0

Figure 2-16. Data Bus Buffer Configuration

- **F_0**
  This is a general purpose flag which can be cleared or toggled under UPI software control. The flag is used to transfer UPI status information to the master processor.
- **F_1 Command/Data**
  This flag is set to the condition of the A_0 input line when the master processor writes a character to the data register. The F_1 flag can also be cleared or toggled under UPI-Microcomputer program control.
- **ST4 through ST7**
  These bits are user defined status bits. They are defined by the MOV STS,A instruction.

SYSTEM INTERFACE

Figure 2-17 illustrates how a UPI-Microcomputer can be connected to a standard 8080-type bus system. Data lines D_0–D_7 form a three-state, bidirectional port which can be connected directly to the system data bus. The UPI bus interface has sufficient drive capability (400 μA) for small systems, however, a larger system may require buffers.

Four control signals are required to handle the data and status information transfer:

- $\overline{WR}$
  I/O WRITE signal used to transfer data from the system bus to the UPI DBBIN register and set the F_1 flag in the status register.
- $\overline{RD}$
  I/O READ signal used to transfer data from the DBBOUT register or status register to the system data bus.
- $\overline{CS}$
  CHIP SELECT signal used to enable one 8041AH out of several connected to a common bus.
- $\overline{A_0}$
  Address input used to select either the 8-bit status register or DBBOUT register during an I/O READ. Also, the signal is used to set the F_1 flag in the status register during an I/O WRITE.

The $\overline{WR}$ and $\overline{RD}$ signals are active low and are standard MCS-80 peripheral control signals used to synchronize data transfer between the system bus and peripheral devices.

The $\overline{CS}$ and A_0 signals are decoded from the address bus of the master system. In a system with few I/O devices a linear addressing configuration can be used where A_0 and A_1 lines are connected directly to A_0 and $\overline{CS}$ inputs (see Figure 2-17).

Data Read

Table 2-4 illustrates the relative timing of a DBBOUT Read. When $\overline{CS}$, A_0, and $\overline{RD}$ are low, the contents of the DBBOUT register is placed on the three-state Data lines D_0–D_7 and the OBF flag is cleared.

The master processor uses $\overline{CS}$, A_0, $\overline{WR}$, and $\overline{RD}$ to control data transfer between the DBBOUT register and the master system. The following operations are under master processor control:

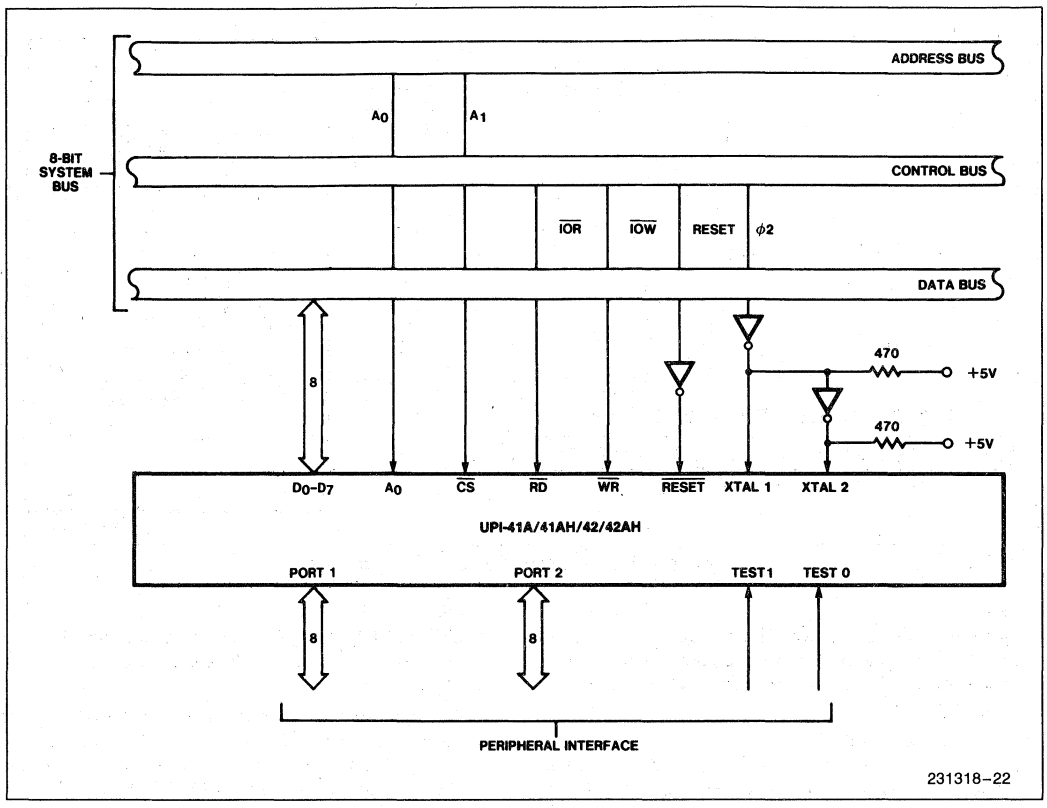


Figure 2-17. Interface to 8080 System Bus

Table 2-4. Data Transfer Controls

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_0	
0	0	1	0	Read DBBOUT register
0	0	1	1	Read STATUS register
0	1	0	0	Write DBBIN data register
0	1	0	1	Write DBBIN command register
1	x	x	x	Disable DBB

Status Read

Table 2-4 shows the logic sequence required for a STATUS register read. When $\overline{CS}$ and $\overline{RD}$ are low with A_0 high, the contents of the 8-bit status register appears on Data lines D_0–D_7.

Data Write

Table 2-4 shows the sequence for writing information to the DBBIN register. When $\overline{CS}$ and $\overline{WR}$ are low, the contents of the system data bus is latched into DBBIN. Also, the IBF flag is set and an interrupt is generated, if enabled.

Command Write

During any write (Table 2-4), the state of the A_0 input is latched into the status register in the F_1 (command/data) flag location. This additional bit is used to signal whether DBBIN contents are command (A_0 = 1) or data (A_0 = 0) information.

INPUT/OUTPUT INTERFACE

The UPI-41A/41AH/42/42AH has 16 lines for input and output functions. These I/O lines are grouped as two 8-bit TTL compatible ports: PORTS 1 and 2. The port lines can individually function as either inputs or outputs under software control. In addition, the lower 4 lines of PORT 2 can be used to interface to an 8243 I/O expander device to increase I/O capacity to 28 or more lines. The additional lines are grouped as 4-bit ports: PORTS 4, 5, 6, and 7.

PORTS 1 and 2

PORTS 1 and 2 are each 8 bits wide and have the same I/O characteristics. Data written to these ports by an

OUTL Pp,A instruction is latched and remains unchanged until it is rewritten. Input data is sampled at the time the IN, A, Pp instruction is executed. Therefore, input data must be present at the PORT until read by an INput instruction. PORT 1 and 2 inputs are fully TTL compatible and outputs will drive one standard TTL load.

Circuit Configuration

The PORT 1 and 2 lines have a special output structure (shown in Figure 2-18) that allows each line to serve as an input, an output, or both, even though outputs are statically latched.

Each line has a permanent high impedance pull-up (50 KΩ) which is sufficient to provide source current for a TTL high level, yet can be pulled low by a standard TTL gate drive. Whenever a "1" is written to a line, a low impedance pull-up (5K) is switched in momentarily (500 ns) to provide a fast transition from 0 to 1. When a "0" is written to the line, a low impedance pull-down (300Ω) is active to provide TTL current sinking capability.

To use a particular PORT pin as an input, a logic "1" must first be written to that pin.

NOTE:
A $\overline{\text{RESET}}$ initializes all PORT pins to the high impedance logic "1" state.

An external TTL device connected to the pin has sufficient current sinking capability to pull-down the pin to the low state. An IN A, Pp instruction will sample the status of PORT pin and will input the proper logic level. With no external input connected, the IN A,Pp instruction inputs the previous output status.

This structure allows input and output information on the same pin and also allows any mix of input and output lines on the same port. However, when inputs and outputs are mixed on one PORT, a PORT write will cause the strong internal pull-ups to turn on at all inputs. If a switch or other low impedance device is connected to an input, a PORT write ("1" to an input) could cause current limits on internal lines to be exceeded. Figure 2-19 illustrates the recommended connection when inputs and outputs are mixed on one PORT.

The bidirectional port structure in combination with the UPI-41A/41AH/42/42AH logical AND and OR instructions provide an efficient means for handling single line inputs and outputs within an 8-bit processor.

PORTS 4, 5, 6, and 7

By using an 8243 I/O expander, 16 additional I/O lines can be connected to the UPI-41AH, 42AH and directly addressed as 4-bit I/O ports using UPI-41AH, 42AH

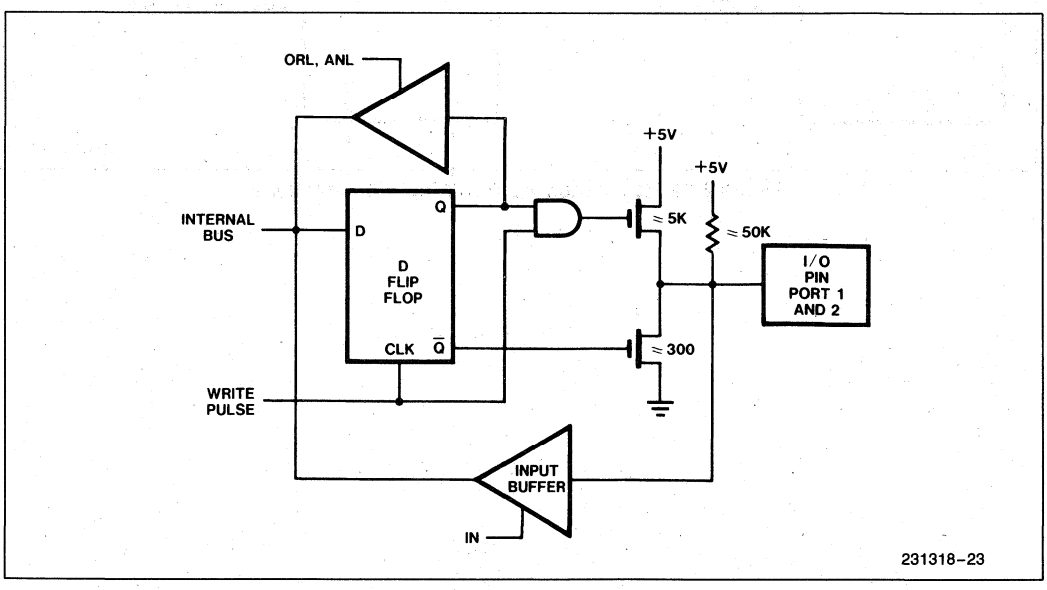


Figure 2-18. Quasi-Bidirectional Port Structure

instructions. This feature saves program space and design time, and improves the bit handling capability of the UPI-41A/41AH/42/42AH.

The lower half of PORT 2 provides an interface to the 8243 as illustrated in Figure 2-20. The PROG pin is used as a strobe to clock address and data information via the PORT 2 interface. The extra 16 I/O lines are referred to in UPI software as PORTS 4, 5, 6, and 7. Each PORT can be directly addressed and can be ANDed and ORed with an immediate data mask. Data can be moved directly to the accumulator from the expander PORTS (or vice-versa).

The 8243 I/O ports, PORTS 4, 5, 6, and 7, provide more drive capability than the UPI-41A/41AH/42/42AH bidirectional ports. The 8243 output is capable of driving about 5 standard TTL loads.

Multiple 8243's can be connected to the PORT 2 interface. In normal operation, only one of the 8243's would be active at the time an Input or Output command is executed. The upper half of PORT 2 is used to provide chip select signals to the 8043's. Figure 2-21 shows how four 8243's could be connected. Software is needed to select and set the proper PORT 2 pin before an INPUT or OUTPUT command to PORTS 4–7 is executed. In general, the software overhead required is very minor compared to the added flexibility of having a large number of I/O pins available.

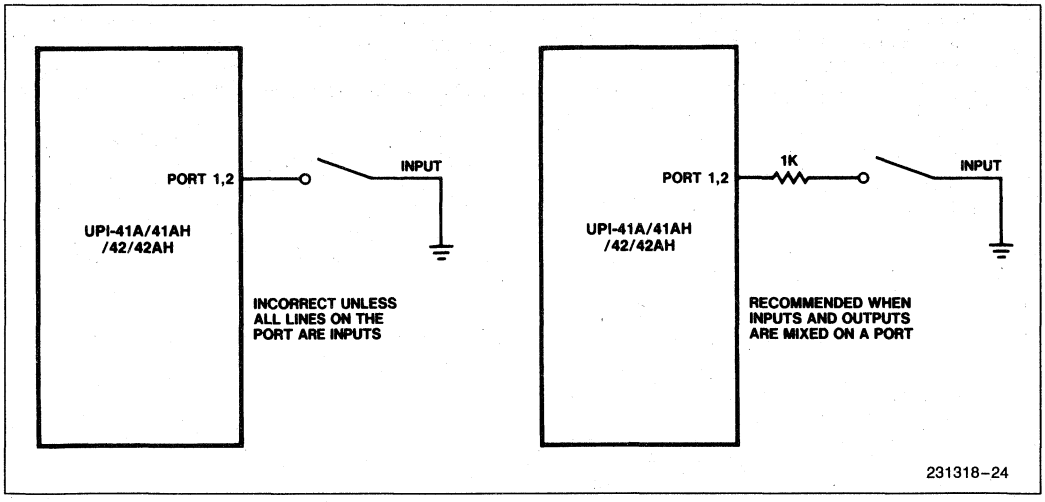


Figure 2-19. Recommended PORT Input Connections

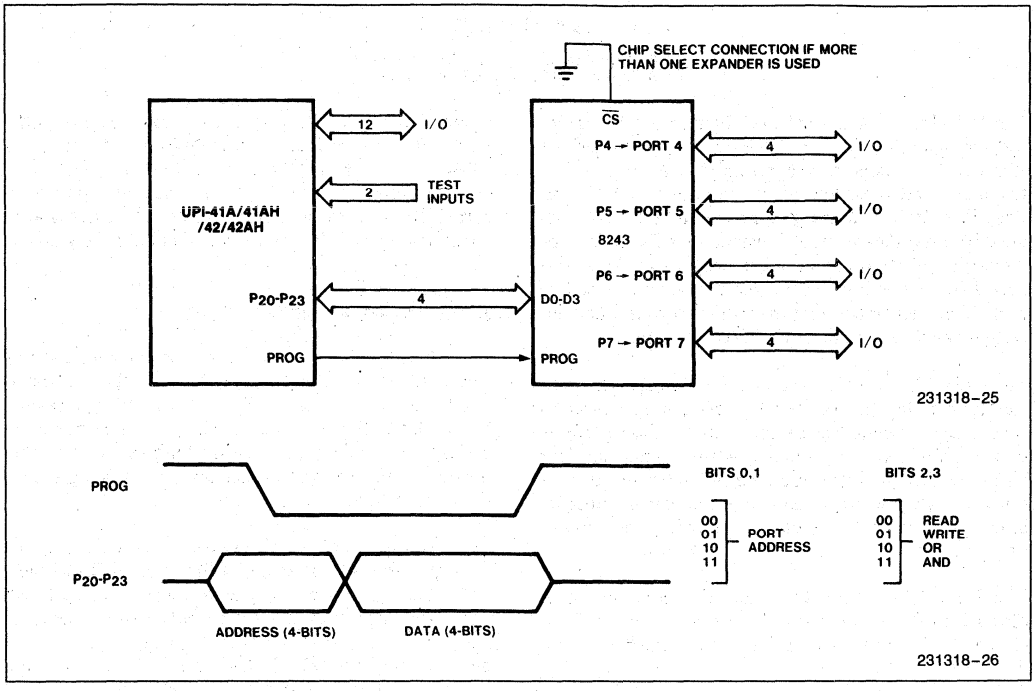


Figure 2-20. 8243 Expander Interface

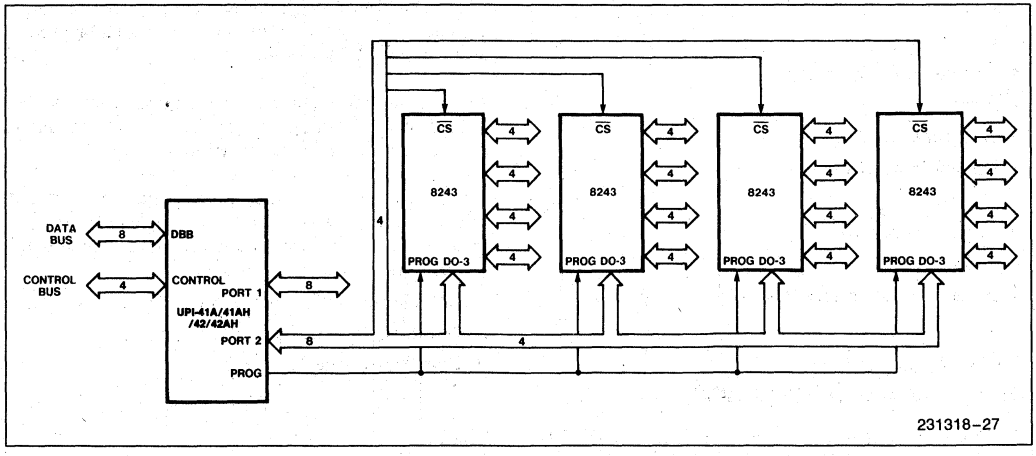


Figure 2-21. Multiple 8243 Expansion

CHAPTER 3
INSTRUCTION SET

The UPI-41A/41AH/42/42AH Instruction Set is op-code-compatible with the MCS-48 set except for the elimination of external program and data memory instructions and the addition of the data bus buffer instructions. It is very straightforward and efficient in its use of program memory. All instructions are either 1 or 2 bytes in length (over 70% are only 1 byte long) and over half of the instructions execute in one machine cycle. The remainder require only two cycles and include Branch, Immediate, and I/O operations.

The UPI-41A/41AH/42/42AH Instruction Set efficiently handles the single-bit operations required in control applications. Special instructions allow port bits to be set or cleared individually. Also, any accumulator bit can be directly tested via conditional branch instructions. Additional instructions are included to simplify loop counters, table look-up routines and N-way branch routines.

The UPI-41A/41AH/42/42AH Microcomputer handles arithmetic operations in both binary and BCD for efficient interface to peripherals such as keyboards and displays.

The instruction set can be divided into the following groups:

- Data Moves
- Accumulator Operations
- Flags
- Register Operations
- Branch Instructions
- Control
- Timer Operations
- Subroutines
- Input/Output Instructions

Data Moves (See Instruction Summary)

The 8-bit accumulator is the control point for all data transfers within the UPI-41A/41AH/42/42AH. Data can be transferred between the 8 registers of each working register bank and the accumulator directly (i.e., with a source or destination register specified by 3 bits in the instruction). The remaining locations in the RAM array are addressed either by R_0 or R_1 of the active register bank. Transfers to and from RAM require one cycle.

Constants stored in Program Memory can be loaded directly into the accumulator or the eight working registers. Data can also be transferred directly between the accumulator and the on-board timer/counter, the Status Register (STS), or the Program Status Word (PSW). Transfers to the STS register alter bits 4–7 only. Transfers to the PSW alter machine status accordingly and provide a means of restoring status after an interrupt or of altering the stack pointer if necessary.

Accumulator Operations

Immediate data, data memory, or the working registers can be added (with or without carry) to the accumulator. These sources can also be ANDed, ORed, or exclusive ORed to the accumulator. Data may be moved to or from the accumulator and working registers or data memory. The two values can also be exchanged in a single operation.

The lower 4 bits of the accumulator can be exchanged with the lower 4 bits of any of the internal RAM locations. This operation, along with an instruction which swaps the upper and lower 4-bit halves of the accumulator, provides easy handling of BCD numbers and other 4-bit quantities. To facilitate BCD arithmetic a Decimal Adjust instruction is also included. This instruction is used to correct the result of the binary addition of two 2-digit BCD numbers. Performing a decimal adjust on the result in the accumulator produces the desired BCD result.

The accumulator can be incremented, decremented, cleared, or complemented and can be rotated left or right 1 bit at a time with or without carry.

A subtract operation can be easily implemented in UPI software using three single-byte, single-cycle instructions. A value can be subtracted from the accumulator by using the following instructions:

- Complement the accumulator
- Add the value to the accumulator
- Complement the accumulator

Flags

There are four user accessible flags:

- Carry
- Auxiliary Carry
- F_0
- F_1

The Carry flag indicates overflow of the accumulator, while the Auxiliary Carry flag indicates overflow between BCD digits and is used during decimal adjust

operations. Both Carry and Auxiliary Carry are part of the Program Status Word (PSW) and are stored in the stack during subroutine calls. The F_0 and F_1 flags are general-purpose flags which can be cleared or complemented by UPI instructions. F_0 is accessible via the Program Status Word and is stored in the stack with the Carry flags. F_1 reflects the condition of the A_0 line, and caution must be used when setting or clearing it.

Register Operations

The working registers can be accessed via the accumulator as explained above, or they can be loaded with immediate data constants from program memory. In addition, they can be incremented or decremented directly, or they can be used as loop counters as explained in the section on branch instructions.

Additional Data Memory locations can be accessed with indirect instructions via R_0 and R_1.

Branch Instructions

The UPI-41A/41AH/42/42AH Instruction Set includes 17 jump instructions. The unconditional allows jumps anywhere in the 1K words of program memory. All other jump instructions are limited to the current page (256 words) of program memory.

Conditional jump instructions can test the following inputs and maching flags:

- TEST 0 input pin
- TEST 1 input pin
- Input Buffer Full flag
- Output Buffer Full flag
- Timer flag
- Accumulator zero
- Accumulator bit
- Carry flag
- F_0 flag
- F_1 flag

The conditions tested by these instructions are the instantaneous values at the time the conditional jump instruction is executed. For instance, the jump on accumulator zero instruction tests the accumulator itself, not an intermediate flag.

The decrement register and jump if not zero (DJNZ) instruction combines decrement and branch operations in a single instruction which is useful in implementing a loop counter. This instruction can designate any of the 8 working registers as a counter and can effect a branch to any address within the current page of execution.

A special indirect jump instruction (JMPP @A) allows the program to be vectored to any one of several different locations based on the contents of the accumulator. The contents of the accumulator point to a location in program memory which contains the jump address. As an example, this instruction could be used to vector to any one of several routines based on an ASCII character which has been loaded into the accumulator. In this way, ASCII inputs can be used to initiate various routines.

Control

The UPI-41A/41AH/42/42AH Instruction Set has six instructions for control of the DMA, interrupts, and selection of working registers banks.

The UPI-41A/41AH/42/42AH provides two instructions for control of the external microcomputer system. IBF and OBF flags can be routed to PORT 2 allowing interrupts of the external processor. DMA handshaking signals can also be enabled using lines from PORT 2.

The IBF interrupt can be enabled and disabled using two instructions. Also, the interrupt is automatically disabled following a RESET input or during an interrupt service routine.

The working register bank switch instructions allow the programmer to immediately substitute a second 8 register bank for the one in use. This effectively provides either 16 working registers or the means for quickly saving the contents of the first 8 registers in response to an interrupt. The user has the option of switching register banks when an interrupt occurs. However, if the banks are switched, the original bank will automatically be restored upon execution of a return and restore status (RETR) instruction at the end of the interrupt service routine.

Timer

The 8-bit on-board timer/counter can be loaded or read via the accumulator while the counter is stopped or while counting.

The counter can be started as a timer with an internal clock source or as an event counter or timer with an

external clock applied to the TEST 1 pin. The instruction executed determines which clock source is used. A single instruction stops the counter whether it is operating with an internal or an external clock source. In addition, two instructions allow the timer interrupt to be enabled or disabled.

Subroutines

Subroutines are entered by executing a call instruction. Calls can be made to any address in the 1K word program memory. Two separate return instructions determine whether or not status (i.e., the upper 4 bits of the PSW) is restored upon return from a subroutine.

Input/Output Instructions

Two 8-bit data bus buffer registers (DBBIN and DBBOUT) and an 8-bit status register (STS) enable the UPI-41A universal peripheral interface to communicate with the external microcomputer system. Data can be INputted from the DBBIN register to the accumulator. Data can be OUTputted from the accumulator to the DBBOUT register.

The STS register contains four user-definable bits (ST_4–ST_7) plus four reserved status bits (IBF, OBF, F_0 and F_1). The user-definable bits are set from the accumulator.

The UPI-41A/41AH/42/42AH peripheral interface has two 8-bit static I/O ports which can be loaded to and from the accumulator. Outputs are statically latched but inputs to the ports are sampled at the time an IN instruction is executed. In addition, immediate data from program memory can be ANDed and ORed directly to PORTS 1 and 2 with the result remaining on the port. This allows "masks" stored in program memory to be used to set or reset individual bits on the I/O ports. PORTS 1 and 2 are configured to allow input on a given pin by first writing a "1" to the pin.

Four additional 4-bit ports are available through the 8243 I/O expander device. The 8243 interfaces to the UPI-41A/41AH/42/42AH peripheral interface via four PORT 2 lines which form an expander bus. The 8243 ports have their own AND and OR instructions like the on-board ports, as well as move instructions to transfer data in or out. The expander AND or OR instructions, however, combine the contents of the accumulator with the selected port rather than with immediate data as is done with the on-board ports.

INSTRUCTION SET DESCRIPTION

The following section provides a detailed description of each UPI instruction and illustrates how the instructions are used.

For further information about programming the UPI, consult the *8048/8041AH Assembly Language Manual.*

Table 3-1. Symbols and Abbreviations Used

Symbol	Definition
A	Accumulator
C	Carry
DBBIN	Data Bus Buffer Input
DBBOUT	Data Bus Buffer Output
F_0, F_1	FLAG 0, FLAG 1 (C/D flag)
I	Interrupt
P	Mnemonic for "in-page" operation
PC	Program Counter
Pp	Port designator (p = 1, 2, or 4–7)
PSW	Program Status Word
Rr	Register designator (r = 0–7)
SP	Stack Pointer
STS	Status register
T	Timer
TF	Timer Flag
T_0, T_1	TEST 0, TEST 1
#	Immediate data prefix
@	Indirect address prefix
(( ))	Double parentheses show the effect of @, that is @RO is shown as ((RO)).
( )	Contents of

Table 3-2. Instruction Set Summary

Mnemonic	Description	Bytes	Cycle
ACCUMULATOR			
ADD A, Rr	Add register to A	1	1
ADD A, @Rr	Add data memory to A	1	1
ADD A, #data	Add immediate to A	2	2
ADDC A, Rr	Add register to A with carry	1	1
ADDC A, @Rr	Add data memory to A with carry	1	1
ADDC A, #data	Add immediate to A with carry	2	2
ANL A, Rr	And register to A	1	1
ANL A, @Rr	And data memory to A	1	1
ANL A, #data	And immediate to A	2	2
ORL A, Rr	Or register to A	1	1
ORL A, @Rr	Or data memory to A	1	1
ORL A, #data	Or immediate to A	2	2
XRL A, Rr	Exclusive Or register to A	1	1
XRL A, @Rr	Exclusive Or data memory to A	1	1
XRL A, #data	Exclusive Or immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal Adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
INPUT/OUTPUT			
IN A, Pp	Input port to A	1	2
OUTL Pp, A	Output A to port	1	2
ANL Pp, #data	And immediate to port	2	2
ORL Pp, #data	Or immediate to port	2	2
IN A,DBB	Input DDB to A, clear IBF	1	1
OUT DBB, A	Output A to DBB, Set OBF	1	1
MOV STS,A	A_4–A_7 to bits 4–7 of status	1	1
MOVD A,Pp	Input Expander port to A	1	2
MOVD Pp,A	Output A to Expander port	1	2
ANLD Pp,A	And A to Expander port	1	2
ORLD Pp,A	Or A to Expander port	1	2
DATA MOVES			
MOV A, Rr	Move register to A	1	1
MOV A, @Rr	Move data memory to A	1	1
MOV A, #data	Move immediate to A	2	2
MOV Rr, A	Move A to register	1	1
MOV @Rr, A	Move A to data memory	1	1
MOV Rr, #data	Move immediate to register	2	2
MOV @Rr, #data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1
XCH A, Rr	Exchange A and registers	1	1
XCH A, @Rr	Exchange A and data memory	1	1
XCHD A, @Rr	Exchange digit of A and register	1	1

Mnemonic	Description	Bytes	Cycle
DATA MOVES (Continued)			
MOVP A, @A	Move to A from current page	1	2
MOVP3 A, @A	Move to A from page 3	1	2
TIMER/COUNTER			
MOV A,T	Read Timer/Counter	1	1
MOV T,A	Load Timer/Counter	1	1
STRT T	Start Timer	1	1
STRT CNT	Start Counter	1	1
STOP TCNT	Stop Timer/Counter	1	1
EN TCNTI	Enable Timer/Counter Interrupt	1	1
DIS TCNTI	Disable Timer/Counter Interrupt	1	1
CONTROL			
EN DMA	Enable DMA Handshake Lines	1	1
EN I	Enable IBF interrupt	1	1
DIS I	Disable IBF interrupt	1	1
EN FLAGS	Enable Master Interrupts	1	1
SEL MB0	Select memory bank 0	1	1
SEL MB1	Select memory bank 1	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
NOP	No Operation	1	1
REGISTERS			
INC Rr	Increment register	1	1
INC @Rr	Increment data memory	1	1
DEC Rr	Decrement register	1	1
SUBROUTINE			
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2
FLAGS			
CLR C	Clear Carry	1	1
CPL C	Complement Carry	1	1
CLR F0	Clear Flag 0	1	1
CPL F0	Complement Flag 0	1	1
CLR F1	Clear F_1 Flag	1	1
CPL F1	Complement F_1 Flag	1	1
BRANCH			
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ Rr, addr	Decrement register and jump on non-zero	2	2
JC addr	Jump on Carry = 1	2	2
JNC addr	Jump on Carry = 0	2	2
JZ addr	Jump on A zero	2	2
JNZ addr	Jump on A not zero	2	2
JT0 addr	Jump on T_0 = 1	2	2
JNT0 addr	Jump on T_0 = 0	2	2
JT1 addr	Jump on T_1 = 1	2	2
JNT1 addr	Jump on T_1 = 0	2	2
JF0 addr	Jump on F_0 Flag = 1	2	2
JF1 addr	Jump on F_1 Flag = 1	2	2
JTF addr	Jump on Timer Flag = 1	2	2
JNIBF addr	Jump on IBF Flag = 0	2	2
JOBF addr	Jump on OBF Flag = 1	2	2
JBb addr	Jump on Accumulator Bit	2	2

ALPHABETIC LISTING

ADD A,Rr Add Register Contents to Accumulator

Opcode:

0 1 1 0	1 r_2 r_1 r_0

The contents of register 'r' are added to the accumulator. Carry is affected.
(A) ← (A) + (Rr) r = 0–7

Example:
```
ADDREG: ADD A,R6        ;ADD REG 6 CONTENTS
                        ;TO ACC
```

ADD A,@Rr Add Data Memory Contents to Accumulator

Opcode:

0 1 1 0	0 0 0 r

The contents of the standard data memory location address by register 'r' bits 0–5 are added to the accumulator. Carry is affected.

(A) ← (A) + ((Rr)) r = 0–1

Example:
```
ADDM: MOV RO,#47        ;MOVE 47 DECIMAL TO REG 0
      ADD A,@RO         ;ADD VALUE OF LOCATION
                        ;47 TO ACC
```

ADD A,#data Add !mmediate Data to Accumulator

Opcode:

0 0 0 0	0 0 1 1	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The specified data is added to the accumulator. Carry is affected.
(A) ← (A) + data

Example:
```
ADDID: ADD A,#ADDER     ;ADD VALUE OF SYMBOL
                        ;ADDER' TO ACC
```

ADDC A,Rr Add Carry and Register Contents to Accumulator

Opcode:

0 1 1 1	1 r_2 r_1 r_0

The content of the carry bit is added to accumulator location 0. The contents of register 'r' are then added to the accumulator. Carry is affected.
(A) ← (A) + (Rr) + (C) r = 0–7

Example:
```
ADDRGC: ADDC A,R4       ;ADD CARRY AND REG 4
                        ;CONTENTS TO ACC
```

ADDC A,@Rr Add Carry and Data Memory Contents to Accumulator

Opcode:

0 1 1 1	0 0 0 r

The content of the carry bit is added to accumulator location 0. Then the contents of the standard data memory location addressed by register 'r' bits 0–5 are added to the accumulator. Carry is affected.

(A) ← (A) + ((Rr)) + (C) r = 0–1

Example:

```
ADDMC: MOV R1,#40      ;MOV '40' DEC TO REG 1
       ADDC A,@R1      ;ADD CARRY AND LOCATION 40
                       ;CONTENTS TO ACC
```

ADDC A,#data Add Carry and Immediate Data to Accumulator

Opcode:

0 0 0 1	0 0 1 1	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The content of the carry bit is added to accumulator location 0. Then the specified data is added to the accumulator. Carry is affected.

(A) ← (A) + data + (C)

Example:

```
ADDC A,#255            ;ADD CARRY AND '225' DEC
                       ;TO ACC
```

ANL A,Rr Logical AND Accumulator With Register Mask

Opcode:

0 1 0 1	1 r_2 r_1 r_0

Data in the accumulator is logically ANDed with the mask contained in working register 'r'.

(A) ← (A) AND (Rr) r = 0–7

Example:

```
ANDREG: ANL A,R3       ;'AND' ACC CONTENTS WITH MASK
                       ;MASK IN REG 3
```

ANL A,@Rr Logical AND Accumulator With Memory Mask

Opcode:

0 1 0 1	0 0 0 r

Data in the accumulator is logically ANDed with the mask contained in the data memory location referenced by register 'r', bits 0–5.

(A) ← (A) AND ((Rr)) r = 0–1

Example:

```
ANDDM: MOV R0,#0FFH    MOV 'FF' HEX TO REG 0
       ANL A,#0AFH     ;'AND' ACC CONTENTS WITH
                       ;MASK IN LOCATION 63
```

ANL A,#data Logical AND Accumulator With Immediate Mask

Opcode:

0 1 0 1	0 0 1 1	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data in the accumulator is logically ANDed with an immediately-specified mask.
(A) ← (A) AND data

Example:

```
ANDID: ANL A,#0AFH      ;'AND' ACC CONTENTS
                        ;WITH MASK 10101111
       ANL A,#3+X/Y     ;'AND' ACC CONTENTS
                        ;WITH VALUE OF EXP
                        '3+X/Y'
```

ANL PP,#data Logical AND PORT 1–2 With Immediate Mask

Opcode:

1 0 0 1	1 0 p_1 p_0	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data on the port 'p' is logically ANDed with an immediately-specified mask.
(Pp) ← (Pp) AND data p = 1–2

Note: Bits 0–1 of the opcode are used to represent PORT 1 and PORT 2. If you are coding in binary rather than assembly language, the mapping is as follows:

Bits	p1	p0	Port
	0	0	X
	0	1	1
	1	0	2
	1	1	X

Example:

```
ANDP2: ANL P2,#OF0H     ;'AND' PORT 2 CONTENTS
                        ;WITH MASK'F0' HEX
                        ;(CLEAR P20–23)
```

ANLD Pp,A Logical AND Port 4–7 With Accumulator Mask

Opcode:

1 0 0 1	1 1 p_1 p_0

This is a 2-cycle instruction. Data on port 'p' on the 8243 expander is logically ANDed with the digit mask contained in accumulator bits 0–3.
(Pp) ← (Pp) AND (A0–3) p = 4–7

Note: The mapping of Port 'p' to opcode bits p_1, p_0 is as follows:

P1	P0	Port
0	0	4
0	1	5
1	0	6
1	1	7

Example:

```
ANDP4: ANLD P4,A        ;'AND' PORT 4 CONTENTS
                        ;WITH ACC BITS 0–3
```

CALL address Subroutine Call

Opcode:

$0\ a_9\ a_8\ 1$	$0\ 1\ 0\ 0$	•	$a_7\ a_6\ a_5\ a_4$	$a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. The program counter and PSW bits 4–7 are saved in the stack. The stack pointer (PSW bits 0–2) is updated. Program control is then passed to the location specified by 'address'.

Execution continues at the instruction following the CALL upon return from the subroutine.
((SP)) ← (PC), (PSW_{4-7})
(SP) ← (SP) + 1
(PC_{8-9}) ← $(addr_{8-9})$
(PC_{0-7}) ← $(addr_{0-7})$

Example: Add three groups of two numbers. Put subtotals in locations 50, 51 and total in location 52.

```
                MOV R0,#50          ;MOVE '50' DEC TO ADDRESS
                                    ;REG 0
BEGADD:         MOV A,R1            ;MOVE CONTENTS OF REG 1
                                    ;TO ACC
                ADD A,R2            ;ADD REG 2 TO ACC
                CALL SUBTOT         ;CALL SUBROUTINE 'SUBTOT'
                ADD A,R3            ;ADD REG 3 TO ACC
                ADD A,R4            ;ADD REG 4 TO ACC
                CALL SUBTOT         ;CALL SUBROUTINE 'SUBTOT'
                ADD A,R5            ;ADD REG 5 TO ACC
                ADD A,R6            ;ADD REG 6 TO ACC
                CALL SUBTOT         ;CALL SUBROUTINE 'SUBTOT'
                    .
                    .
                    .
SUBTOT:         MOV @R0,A           ;MOVE CONTENTS OF ACC TO
                                    ;LOCATION ADDRESSED BY
                                    ;REG 0
                INC R0              ;INCREMENT REG 0
                RET                 ;RETURN TO MAIN PROGRAM
```

CLR A Clear Accumulator

Opcode:

$0\ 0\ 1\ 0$	$0\ 1\ 1\ 1$

The contents of the accumulator are cleared to zero.
(A) ← 00H

CLR C Clear Carry Bit

Opcode:

$1\ 0\ 0\ 1$	$0\ 1\ 1\ 1$

During normal program execution, the carry bit can be set to one by the ADD, ADDC, RLC, CPLC, RRC, and DAA instructions. This instruction resets the carry bit to zero.
(C) ← 0

CLR F1 Clear Flag 1

Opcode:

$1\ 0\ 1\ 0$	$0\ 1\ 0\ 1$

The F_1 flag is cleared to zero.
(F_1) ← 0

CLR F0 Clear Flag 0

Opcode:

1 0 0 0	0 1 0 1

F_0 flag is cleared to zero.
$(F_0) \leftarrow 0$

CPL A Complement Accumulator

Opcode:

0 0 1 1	0 1 1 1

The contents of the accumulator are complemented. This is strictly a one's complement. Each one is changed to zero and vice-versa.
$(A) \leftarrow$ NOT (A)

Example: Assume accumulator contains 01101010.

```
CPLA: CPL A                ;ACC CONTENTS ARE COMPLE-
                           ;MENTED TO 10010101
```

CPL C Complement Carry Bit

Opcode:

1 0 1 0	0 1 1 1

The setting of the carry bit is complemented; one is changed to zero, and zero is changed to one.
$(C) \leftarrow$ NOT (C)

Example: Set C to one; current setting is unknown.

```
CT01: CLR C                ;C IS CLEARED TO ZERO
      CPL C                ;C IS SET TO ONE
```

CPL F0 COMPLEMENT FLAG 0

Opcode:

1 0 0 1	0 1 0 1

The setting of Flag 0 is complemented; one is changed to zero, and zero is changed to one.
$F_0 \leftarrow$ NOT (F0)

CPL F1 Complement Flag 1

Opcode:

1 0 1 1	0 1 0 1

The setting of the F_1 Flag is complemented; one is changed to zero, and zero is changed to one.
$(F_1) \leftarrow$ NOT (F_1)

DA A Decimal Adjust Accumulator

Opcode: | 0 1 0 1 | 0 1 1 1 |

The 8-bit accumulator value is adjusted to form two 4-bit Binary Coded Decimal (BCD) digits following the binary addition of BCD numbers. The carry bit C is affected. If the contents of bits 0–3 are greater than nine, or if AC is one, the accumulator is incremented by six.

The four high-order bits are then checked. If bits 4–7 exceed nine, or if C is one, these bits are increased by six. If an overflow occurs, C is set to one; otherwise, it is cleared to zero.

Example: Assume accumulator contains 9AH.

```
        DA A                ;ACC ADJUSTED TO 01H with C set
C   AC                      ACC
0   0                       9AH   INITIAL CONTENTS
                            06H   ADD SIX TO LOW DIGIT
0   0                       A1H
                            60H   ADD SIX TO HIGH DIGIT

1   0                       01H   RESULT
```

DEC A Decrement Accumulator

Opcode: | 0 0 0 0 | 0 1 1 1 |

The contents of the accumulator are decremented by one.
(A) ← (A) − 1

Example: Decrement contents of data memory location 63.

```
        MOV R0,#3FH     ;MOVE '3F' HEX TO REG 0
        MOV A,@R0       ;MOVE CONTENTS OF LOCATION 63
                        ;TO ACC
        DEC A           ;DECREMENT ACC
        MOV @R0,A       ;MOVE CONTENTS OF ACC TO
                        ;LOCATION 63
```

DEC Rr Decrement Register

Opcode: | 1 1 0 0 | 1 r_2 r_1 r_0 |

The contents of working register 'r' are decremented by one.
(Rr) ← (Rr) − 1 r = 0–7

Example:
```
DECR1: DEC R1           ;DECREMENT ADDRESS REG 1
```

DIS I Disable IBF Interrupt

Opcode: | 0 0 0 1 | 0 1 0 1 |

The input Buffer Full interrupt is disabled. The interrupt sequence is not initiated by $\overline{WR}$ and $\overline{CS}$, however, an IBF interrupt request is latched and remains pending until an EN I (enable IBF interrupt) instruction is executed.

Note: The IBF flag is set and cleared independent of the IBF interrupt request so that handshaking protocol can continue normally.

DIS TCNTI Disable Timer/Counter Interrupt

Opcode:

0 0 1 1	0 1 0 1

The timer/counter interrupt is disabled. Any pending timer interrupt request is cleared. The interrupt sequence is not initiated by an overflow, but the timer flag is set and time accumulation continues.

DJNZ Rr, address Decrement Register and Test

Opcode:

1 1 1 1	1 r_2 r_1 r_0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Register 'r' is decremented and tested for zero. If the register contains all zeros, program control falls through to the next instruction. If the register contents are not zero, control jumps to the specified address within the current page.
(Rr) ← (Rr) − 1
If R ≠ 0, then;
(PC_{0-7}) ← addr

Note: A 10-bit address specification does not cause an error if the DJNZ instruction and the jump target are on the same page. If the DJNZ instruction begins in location 255 of a page, it will jump to a target address on the following page. Otherwise, it is limited to a jump within the current page.

Example: Increment values in data memory locations 50–54.

```
    MOV R0,#50          ;MOVE '50' DEC TO ADDRESS
                        ;REG 0
    MOV R3,#05          ;MOVE '5' DEC TO COUNTER
                        ;REG 3
INCRT: INC @R0          ;INCREMENT CONTENTS OF
                        ;LOCATION ADDRESSED BY
                        ;REG 0
    INC R0              ;INCREMENT ADDRESS IN REG 0
    DJNZ R3,INCRT       ;DECREMENT REG 3——JUMP TO
                        ;'INCRT' IF REG 3 NONZERO
    NEXT——              ;'NEXT' ROUTINE EXECUTED
                        ;IF R3 IS ZERO
```

EN DMA Enable DMA Handshake Lines

Opcode:

1 1 1 0	0 1 0 1

DMA handshaking is enabled using P_{26} as DMA request (DRQ) and P_{27} as DMA acknowledge ($\overline{DACK}$). The $\overline{DACK}$ lines forces $\overline{CS}$ and A_0 low internally and clears DRQ.

EN FLAGS Enable Master Interrupts

Opcode:

1 1 1 1	0 1 0 1

The Output Buffer Full (OBF) and the Input Buffer Full (IBF) flags (IBF is inverted) are routed to P_{24} and P_{25}. For proper operation, a "1" should be written to P_{25} and P_{24} before the EN FLAGS instruction. A "0" written to P_{24} or P_{25} disables the pin.

EN I Enable IBF Interrupt

Opcode: | 0 0 0 0 | 0 1 0 1 |

The Input Buffer Full interrupt is enabled. A low signal on $\overline{WR}$ and $\overline{CS}$ initiates the interrupt sequence.

EN TCNTI Enable Timer/Counter Interrupt

Opcode: | 0 0 1 0 | 0 1 0 1 |

The timer/counter interrupt is enabled. An overflow of this register initiates the interrupt sequence.

IN A,DBB Input Data Bus Buffer Contents to Accumulator

Opcode: | 0 0 1 0 | 0 0 1 0 |

Data in the DBBIN register is transferred to the accumulator and the Input Buffer Full (IBF) flag is set to zero.
(A) ← (DBB)
(IBF) ← 0

Example:
```
INDBB: IN A,DBB          ;INPUT DBBIN CONTENTS TO
                         ;ACCUMULATOR
```

IN A,Pp Input Port 1-2 Data to Accumulator

Opcode: | 0 0 0 0 | 1 0 p_1 p_0 |

This is a 2-cycle instruction. Data present on port 'p' is transferred (read) to the accumulator.
(A) ← (Pp) p = 1-2 (see ANL instruction)

Example:
```
INP 12: IN A,P1          ;INPUT PORT 1 CONTENTS
                         ;TO ACC
        MOV R6,A         ;MOVE ACC CONTENTS TO
                         ;REG 6
        IN A,P2          ;INPUT PORT 2 CONTENTS
                         ;TO ACC
        MOV R7,A         ;MOVE ACC CONTENTS TO REG 7
```

INC A Increment Accumulator

Opcode: | 0 0 0 1 | 0 1 1 1 |

The contents of the accumulator are incremented by one.
(A) ← (A) + 1

Example: Increment contents of location 10 in data memory.
```
INCA: MOV R0,#10         ;MOV '10' DEC TO ADDRESS
                         ;REG 0
      MOV A,@R0          ;MOVE CONTENTS OF LOCATION
                         ;10 TO ACC
      INC A              ;INCREMENT ACC
      MOV @R0,A          ;MOVE ACC CONTENTS TO
                         ;LOCATION 10
```

INC Rr Increment Register

Opcode: | 0 0 0 1 | 1 r_2 r_1 r_0 |

The contents of working register 'r' are incremented by one.
(Rr) ← (Rr) + 1 r = 0–7

Example:
```
INCR0: INC R0          ;INCREMENT ADDRESS REG 0
```

INC @Rr Increment Data Memory Location

Opcode: | 0 0 0 1 | 0 0 0 r |

The contents of the resident data memory location addressed by register 'r' bits 0–5 are incremented by one.
((Rr)) ← ((Rr)) + 1 r = 0–1

Example:
```
INCDM: MOV R1,#OFFH    ;MOVE ONES TO REG 1
       INC @R1         ;INCREMENT LOCATION 63
```

JBb address Jump If Accumulator Bit is Set

Opcode: | b_2 b_1 b_0 1 | 0 0 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if accumulator bit 'b' is set to one.
(PC_{0-7}) addr if b = 1
(PC) ← (PC) + 2 if b = 0

Example:
```
JB4IS1: JB4 NEXT       ;JUMP TO 'NEXT' ROUTINE
                       ;IF ACC BIT 4 = 1
```

JC address Jump If Carry Is Set

Opcode: | 1 1 1 1 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if the carry bit is set to one.
(PC_{0-7}) ← addr if C = 1
(PC) ← (PC) + 2 if C = 0

Example:
```
JC1: JC OVERFLOW       ;JUMP TO 'OVFLOW' ROUTINE
                       ;IF C = 1
```

JF0 address Jump If Flag 0 is Set

Opcode: | 1 0 1 1 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if flag 0 is set to one.
(PC_{0-7}) ← addr if F_0 = 1

Example:
```
JF0IS1: JF0 TOTAL      ;JUMP TO 'TOTAL' ROUTINE
                       ;IF F0 = 1
```

JF1 address Jump If C/D Flag (F1) Is Set

Opcode:

0 1 1 1	0 1 1 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the C/D flag (F_1) is set to one.

$(PC_{0-7}) \leftarrow$ addr if $F_1 = 1$

Example:

```
JF 1IS1: JF1 FILBUF      ;JUMP TO 'FILBUF'
                         ;ROUTINE IF F1 = 1
```

JMP address Direct Jump Within 1K Block

Opcode:

a_{10} a_9 a_8 0	0 1 0 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Bits 0–9 of the program counter are replaced with the directly-specified address.

$(PC_{8-9}) \leftarrow$ addr 8–9

$(PC_{0-7}) \leftarrow$ addr 0–7

Example:

```
JMP SUBTOT               ;JUMP TO SUBROUTINE 'SUBTOT'
JMP $-6                  ;JUMP TO INSTRUCTION SIX LOCATIONS
                         ;BEFORE CURRENT LOCATION
JMP 2FH                  ;JUMP TO ADDRESS '2F' HEX
```

JMPP @A Indirect Jump Within Page

Opcode:

1 0 1 1	0 0 1 1

This is a 2-cycle instruction. The contents of the program memory location pointed to by the accumulator are substituted for the 'page' portion of the program counter (PC 0–7).

$(PC_{0-7}) \leftarrow ((A))$

Example:

```
Assume accumulator contains OFH
JMPPAG: JMPP @A          ;JMP TO ADDRESS STORED IN
                         ;LOCATION 15 IN CURRENT PAGE
```

JNC address Jump If Carry Is Not Set

Opcode:

1 1 1 0	0 1 1 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the carry bit is not set, that is, equals zero.

$(PC_{0-7}) \leftarrow$ addr if $C = 0$

Example:

```
JC0: JNC NOVFLO          ;JUMP TO 'NOVFLO' ROUTINE
                         ;IF C = 0
```

JNIBF address Jump If Input Buffer Full Flag Is Low

Opcode:

1 1 0 1	0 1 1 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the Input Buffer Full flag is low (IBF = 0).

$(PC_{0-7}) \leftarrow$ addr if IBF = 0

Example:

```
LOC 3:JNIBF LOC 3        ;JUMP TO SELF IF IBF = 0
                         ;OTHERWISE CONTINUE
```

JNTO address Jump if TEST 0 is Low

Opcode:

0 0 1 0	0 1 1 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address, if the TEST 0 signal is low. Pin is sampled during SYNC.

$(PC_{0-7}) \leftarrow$ addr if $T_0 = 0$

Example:
```
JT0LOW: JNT0 60          ;JUMP TO LOCATION 60 DEC
                         ;IF T0 = 0
```

JNT1 address Jump If TEST 1 is Low

Opcode:

0 1 0 0	0 1 1 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the TEST 1 signal is low. Pin is sampled during SYNC.

$(PC_{0-7}) \leftarrow$ addr if $T_1 = 0$

Example:
```
JT1LOW: JNT1 OBBH        ;JUMP TO LOCATION 'BB' HEX
                         ;IF T1 = 0
```

JNZ address Jump If Accumulator Is Not Zero

Opcode:

1 0 0 1	0 1 1 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the accumulator contents are nonzero at the time this instruction is executed.

$(PC_{0-7}) \leftarrow$ addr if $A \neq 0$

Example:
```
JACCNO: JNZ OABH         ;JUMP TO LOCATION 'AB' HEX
                         ;IF ACC VALUE IS NONZERO
```

JOBF Address Jump If Output Buffer Full Flag Is Set

Opcode:

1 0 0 0	0 1 1 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the Output Buffer Full (OBF) flag is set (= 1) at the time this instruction is executed.

$(PC_{0-7}) \leftarrow$ addr if OBF = 1

Example:
```
JOBFHI: JOBF OAAH        ;JUMP TO LOCATION 'AA' HEX
                         ;IF OBF = 1
```

JTF address Jump If Timer Flag is Set

Opcode:

0 0 0 1	0 1 1 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the timer flag is set to one, that is, the timer/counter register overflows to zero. The timer flag is cleared upon execution of this instruction. (This overflow initiates an interrupt service sequence if the timer-overflow interrupt is enabled.)

$(PC_{0-7}) \leftarrow$ addr if TF = 1

Example:
```
JTF1: JTF TIMER           ;JUMP TO 'TIMER' ROUTINE
                          ;IF TF = 1
```

JTO address Jump If TEST 0 Is High

Opcode: | 0 0 1 1 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if the TEST 0 signal is high (= 1). Pin is sampled during SYNC.
(PC_{0-7}) ← addr if $T_0 = 1$

Example:
```
JT0HI: JT0 53          ;JUMP TO LOCATION 53 DEC
                       ;IF T0 = 1
```

JT1 address Jump If TEST 1 Is High

Opcode: | 0 1 0 1 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if the TEST 1 signal is high (= 1). Pin is sampled during SYNC.
(PC_{0-7}) ← addr if $T_1 = 1$

Example:
```
JT1HI: JT1 COUNT       ;JUMP TO 'COUNT' ROUTINE
                       ;IF T1 = 1
```

JZ address Jump If Accumulator Is Zero

Opcode: | 1 1 0 0 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if the accumulator contains all zeros at the time this instruction is executed.
(PC_{0-7}) ← addr if A = 0

Example:
```
JACCO: JZ OA3H         ;JUMP TO LOCATION 'A3' HEX
                       ;IF ACC VALUE IS ZERO
```

MOV A,#data Move Immediate Data to Accumulator

Opcode: | 0 0 1 0 | 0 0 1 1 | • | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 |

This is a 2-cycle instruction. The 8-bit value spedified by 'data' is loaded in the accumulator.
(A) ← data

Example:
```
MOV A,#OA3H            ;MOV 'A3' HEX TO ACC
```

MOV A,PSW Move PSW Contents to Accumulator

Opcode: | 1 1 0 0 | 0 1 1 1 |

The contents of the program status word are moved to the accumulator.
(A) ← (PSW)

Example: Jump to 'RB1SET' routine if bank switch, PSW bit 4, is set.
```
BSCHK: MOV A,PSW       ;MOV PSW CONTENTS TO ACC
       JB4 RB1 SET     ;JUMP TO 'RB1SET' IF ACC
                       ;BIT 4 = 1
```

6

MOV A,Rr Move Register Contents to Accumulator

Opcode:

1 1 1 1	1 r_2 r_1 r_0

Eight bits of data are moved from working register 'r' into the accumulator.
(A) ← (Rr) r = 0–7

Example:
```
MAR: MOV A,R3          ;MOVE CONTENTS OF REG 3
                       ;TO ACC
```

MOV A,@Rr Move Data Memory Contents to Accumulator

Opcode:

1 1 1 1	0 0 0 r

The contents of the data memory location addressed by bits 0–5 of register 'r' are moved to the accumulator. Register 'r' contents are unaffected.
(A) ← ((Rr)) r = 0–1

Example: Assume R1 contains 00110110.
```
MADM: MOV A,@R1        ;MOVE CONTENTS OF DATA MEM
                       ;LOCATION 54 TO ACC
```

MOV A,T Move Timer/Counter Contents to Accumulator

Opcode:

0 1 0 0	0 0 1 0

The contents of the timer/event-counter register are moved to the accumulator. The timer/event-counter is not stopped.
(A) ← (T)

Example: Jump to "Exit" routine when timer reaches '64', that is, when bit 6 is set—assuming initialization to zero.
```
TIMCHK: MOV A,T        ;MOVE TIMER CONTENTS TO
                       ;ACC
        JB6 EXIT       ;JUMP TO 'EXIT' IF ACC BIT
                       ;6 = 1
```

MOV PSW,A Move Accumulator Contents to PSW

Opcode:

1 1 0 1	0 1 1 1

The contents of the accumulator are moved into the program status word. All condition bits and the stack pointer are affected by this move.
(PSW) ← (A)

Example: Move up stack pointer by two memory locations, that is, increment the pointer by one.
```
INCPTR: MOV A,PSW      ;MOVE PSW CONTENTS TO ACC
        INC A          ;INCREMENT ACC BY ONE
        MOV PSW,A      ;MOVE ACC CONTENTS TO PSW
```

MOV Rr,A Move Accumulator Contents to Register

Opcode: | 1 0 1 0 | 1 r_2 r_1 r_0 |

The contents of the accumulator are moved to register 'r'
(Rr) ← (A) r = 0–7

Example:
```
MRA MOV R0,A        ;MOVE CONTENTS OF ACC TO
                    ;REG 0
```

MOV Rr,#data Move Immediate Data to Register

Opcode: | 1 0 1 1 | 1 r_2 r_1 r_0 | • | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 |

This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to register 'r'.
(Rr) ← data r = 0–7

Example:
```
MIR4: MOV R4,#HEXTEN    ;THE VALUE OF THE SYMBOL
                        ;'HEXTEN' IS MOVED INTO
                        ;REG 4
MIR5: MOV R5,#PI*(R*R)  ;THE VAUE OF THE
                        ;EXPRESSION 'PI*(R*R)'
                        ;IS MOVED INTO REG 5
MIR6: MOV R6,#OADH      ;'AD' HEX IS MOVED INTO
                        REG 6
```

MOV @Rr,A Move Accumulator Contents to Data Memory

Opcode: | 1 0 1 0 | 0 0 0 r |

The contents of the accumulator are moved to the data memory location whose address is specified by bits 0–5 of register 'r'. Register 'r' contents are unaffected.
((Rr)) ← (A) r = 0–1

Example:
```
Assume R0 contains 11000111.
MDMA: MOV @R,A       ;MOVE CONTENTS OF ACC TO
                     ;LOCATION 7 (REG)
```

MOV @Rr,#data Move Immediate Data to Data Memory

Opcode: | 1 0 1 1 | 0 0 0 r | • | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 |

This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to the standard data memory location addressed by register 'r', bit 0–5.

Example: Move the hexadecimal value AC3F to locations 62–63.
```
MIDM: MOV R0,#62      ;MOVE '62' DEC TO ADDR REG0
      MOV @RO,#OACH   ;MOVE 'AC' HEX TO LOCATION 62
      INC R0          ;INCREMENT REG 0 TO '63'
      MOV @R0,#3FH    ;MOVE '3F' HEX TO LOCATION 63
```

MOV STS,A Move Accumulator Contents to STS Register

Opcode:

1 0 0 1	0 0 0 0

The contents of the accumulator are moved into the status register. Only bits 4–7 are affected.
$(STS_{4-7}) \leftarrow (A_{4-7})$

Example: Set ST_4–ST_7 to "1".

```
MSTS: MOV A,#0F0H        ;SET ACC
      MOV STS,A          ;MOVE TO STS
```

MOV T,A Move Accumulator Contents to Timer/Counter

Opcode:

0 1 1 0	0 0 1 0

The contents of the accumulator are moved to the timer/event-counter register.
$(T) \leftarrow (A)$

Example: Initialize and start event counter.

```
INITEC: CLR A            ;CLEAR ACC TO ZEROS
        MOV T,A          ;MOVE ZEROS TO EVENT COUNTER
        STRT CNT         ;START COUNTER
```

MOVD A,Pp Move Port 4–7 Data to Accumulator

Opcode:

0 0 0 0	1 1 p_1 p_0

This is a 2-cycle instruction. Data on 8243 port 'p' is moved (read) to accumulator bits 0–3. Accumulator bits 4–7 are zeroed.
$(A_{0-3}) \leftarrow Pp$ p = 4–7
$(A_{4-7}) \leftarrow 0$

Note: Bits 0–1 of the opcode are used to represent PORTS 4–7. If you are coding in binary rather than assembly language, the mapping is as follows:

Bits		Port
p_1	p_0	
0	0	4
0	1	5
1	0	6
1	1	7

Example:

```
INPPT5: MOVD A,P5        ;MOVE PORT 5 DATA TO ACC
                         ;BITS 0-3, ZERO ACC BITS 4-7
```

MOVD Pp,A Move Accumulator Data to Port 4, 5, 6 and 7

Opcode:

0 0 1 1	1 1 p_1 p_0

This is a 2-cycle instruction. Data in accumulator bits 0–3 is moved (written) to 8243 port 'p'. Accumulator bits 4–7 are unaffected. (See NOTE above regarding port mapping.)

Example: Move data in accumulator to ports 4 and 5.

```
OUTP45: MOVD P4,A        ;MOVE ACC BITS 0-3 TO PORT 4
        SWAP A           ;EXCHANGE ACC BITS 0-3 AND 4-7
        MOVD P5,A        ;MOVE ACC BITS 0-3 TO PORT 5
```

MOVP A,@A Move Current Page Data to Accumulator

Opcode:

1 0 1 0	0 0 1 1

This is a 2-cycle instruction. The contents of the program memory location addressed by the accumulator are moved to the accumulator. Only bits 0–7 of the program counter are affected, limiting the program memory reference to the current page. The program counter is restored following this operation.
(A) ← ((A))

Note: This a 1-byte, 2-cycle instruction. If it appears in location 255 of a program memory page, @A addresses a location in the following page.

Example:

```
MOV128: MOV A,#128      ;MOVE '128' DEC TO ACC
        MOVP A,@A       ;CONTENTS OF 129TH LOCATION
                        ;IN CURRENT PAGE ARE MOVED TO
                        ;ACC
```

MOVP3 A,@A Move Page 3 Data to Accumulator

Opcode:

1 1 1 0	0 0 1 1

This is a 2-cycle instruction. The contents of the program memory location within page 3, addressed by the accumulator, are moved to the accumulator. The program counter is restored following this operation.
(A) ← ((A)) within page 3

Example: Look up ASCII equivalent of hexadecimal code in table contained at the beginning of page 3. Note that ASCII characters are designated by a 7-bit code; the eighth bit is always reset.

```
TABSCH: MOV A,#0B8H     ;MOVE 'B8' HEX TO ACC (10111000)
        ANL A,#7FH      ;LOGICAL AND ACC TO MASK BIT
                        ;7 (00111000)
MOVP3, A,@A             ;MOVE CONTENTS OF LOCATION
                        ;'38' HEX IN PAGE 3 TO ACC
                        ;(ASCII '8')
```

Access contents of location in page 3 labelled TAB1. Assume current program location is not in page 3.

```
TABSCH: MOV A,#TAB1     ;ISOLATE BITS 0–7
                        ;OF LABEL
                        ;ADDRESS VALUE
        MOVP3 A,@A      ;MOVE CONTENT OF PAGE 3
                        ;LOCATION LABELED 'TAB1'
                        ;TO ACC
```

NOP The NOP Instruction

Opcode:

0 0 0 0	0 0 0 0

No operation is performed. Execution continues with the following instruction.

ORL A,Rr Logical OR Accumulator With Register Mask

Opcode:

0 1 0 0	1 r_2 r_1 r_0

Data in the accumulator is logically ORed with the mask contained in working register 'r'.
(A) ← (A) OR (Rr) r = 0–7

Example:

```
ORREG: ORL A,R4         ;'OR' ACC CONTENTS WITH
                        ;MASK IN REG 4
```

ORL A,@Rr Logical OR Accumulator With Memory Mask

Opcode:

0 1 0 0	0 0 0 r

Data in the accumulator is logically ORed with the mask contained in the data memory location referenced by register 'r', bits 0–5.
(A) ← (A) OR ((Rr)) r = 0–1

Example:
```
ORDM: MOVE R0,#3FH      ;MOVE '3F' HEX TO REG 0
      ORL A, @R0        ;'OR' ACC CONTENTS WITH MASK
                        ;IN LOCATION 63
```

ORL A,#Data Logical OR Accumulator With Immediate Mask

Opcode:

0 1 0 0	0 0 1 1	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data in the accumulator is logically ORed with an immediately-specified mask.
(A) ← (A) OR data

Example:
```
ORID: ORL A,#'X'        ;'OR' ACC CONTENTS WITH MASK
                        ;01011000 (ASCII VALUE OF 'X')
```

ORL Pp,#data Logical OR Port 1–2 With Immediate Mask

Opcode:

1 0 0 0	1 0 p_1 p_0	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data on port 'p' is logically ORed with an immediately-specified mask.
(Pp) ← (Pp) OR data p = 1–2 (see OUTL instruction)

Example:
```
ORP1: ORL P1,#0FH       ;'OR' PORT 1 CONTENTS WITH
                        ;MASK 'FF' HEX (SET PORT 1
                        'TO ALL ONES)
```

ORLD Pp,A Logical OR Port 4–7 With Accumulator Mask

Opcode:

1 0 0 0	1 1 p_1 p_0

This is a 2-cycle instruction. Data on 8243 port 'p' is logically ORed with the digit mask contained in accumulator bits 0–3,
(Pp) (Pp) OR (A_{0-3}) p = 4–7 (See MOVD instruction)

Example:
```
ORP7; ORLD P7,A         ;'OR' PORT 7 CONTENTS
                        ;WITH ACC BITS 0–3
```

OUT DBB,A Output Accumulator Contents to Data Bus Buffer

Opcode:

0 0 0 0	0 0 1 0

Contents of the accumulator are transferred to the Data Bus Buffer Output register and the Output Buffer Full (OBF) flag is set to one.
(DBB) ← (A)
OBF ← 1

Example:
```
OUTDBB: OUT DBB,A       ;OUTPUT THE CONTENTS OF
                        ;THE ACC TO DBBOUT
```

OUTL Pp,A Output Accumulator Data to Port 1 and 2

Opcode:

0 0 1 1	1 0 p_1 p_0

This is a 2-cycle instruction. Data residing in the accumulator is transferred (written) to port 'p' and latched.
(Pp) ← (A) P = 1–2

Note: Bits 0–1 of the opcode are used to represent PORT 1 and PORT 2. If you are coding in binary rather than assembly language, the mapping is as follows:

Bits		Port
P_1	P_0	
0	0	X
0	1	1
1	0	2
1	1	X

Example:
```
OUTLP; MOV A,R7      ;MOVE REG 7 CONTENTS TO ACC
       OUTL P2,A     ;OUTPUT ACC CONTENTS TO PORT2
       MOV A,R6      ;MOVE REG 6 CONTENTS TO ACC
       OUTL P1,A     ;OUTPUT ACC CONTENTS TO PORT 1
```

RET Return Without PSW Restore

Opcode:

1 0 0 0	0 0 1 1

This is a 2-cycle instruction. The stack pointer (PSW bits 0–2 is decremented. The program counter is then restored from the stack. PSW bits 4–7 are not restored.
(SP) ← (SP) − 1
(PC) ← ((SP))

RETR Return With PSW Restore

Opcode:

1 0 0 1	0 0 1 1

This is a 2-cycle instruction. The stack pointer is decremented. The program counter and bits 4–7 of the PSW are then restored from the stack. Note that RETR should be used to return from an interrupt, but should not be used within the interrupt service routine as it signals the end of an interrupt routine.
(SP) ← (SP) − 1
(PC) ← ((SP))
(PSW_{4-7}) ← ((SP))

RL A Rotate Left Without Carry

Opcode:

1 1 1 0	0 1 1 1

The contents of the accumulator are rotated left one bit. Bit 7 is rotated into the bit 0 position.
(A_{n+1}) ← (A_n) n = 0–6
(A_0) ← (A_7)

Example: Assume accumulator contains 10110001.
```
RLNC: RL A           ;NEW ACC CONTENTS ARE 01100011
```

RLC A Rotate Left Through Carry

Opcode:

1 1 1 1	0 1 1 1

The contents of the accumulator are rotated left one bit. Bit 7 replaces the carry bit; the carry bit is rotated into the bit 0 position.

$(A_{n+1}) \leftarrow (A_n)$ n = 0–6
$(A_0) \leftarrow (C)$
$(C) \leftarrow (A7)$

Example: Assume accumulator contains a 'signed' number; isolate sign without changing value.

```
RLTC: CLR C          ;CLEAR CARRY TO ZERO
      RLC A          ;ROTATE ACC LEFT, SIGN
                     ;BIT (7) IS PLACED IN CARRY
      RR A           ;ROTATE ACC RIGHT—VALUE
                     ;(BITS 0–6) IS RESTORED,
                     ;CARRY UNCHANGED, BIT 7
                     ;IS ZERO
```

RR A Rotate Right Without Carry

Opcode:

0 1 1 1	0 1 1 1

The contents of the accumulator are rotated right one bit. Bit 0 is rotated into the bit 7 position.

$(A) \leftarrow (A_{n+1})$ n = 0–6
$(A_7) \leftarrow (A_0)$

Example Assume accumulator contains 10110001.

```
RRNC: RRA            ;NEW ACC CONTENTS ARE 11011000
```

RRC A Rotate Right Through Carry

Opcode:

0 1 1 0	0 1 1 1

The contents of the accumulator are rotated one bit. Bit 0 replaces the carry bit; the carry bit is rotated into the bit 7 position.

$(A_n) \leftarrow (A_{n+1})$ n = 0–6
$(A_7) \leftarrow (C)$
$(C) \leftarrow (A_0)$

Example Assume carry is not set and accumulator contains 10110001.

```
RRTC: RRCA           ;CARRY IS SET AND ACC
                     ;CONTAINS 01011000
```

SEL MB0 Select Memory Bank 0 (Not in 8x41)

Opcode:

1 1 1 0	0 1 0 1

PC bit 11 is set to zero on next JMP or CALL instruction. All references to program memory addresses fall within the range 0–1023

(DBF) ← 0

Example: Assume program counter contains 834 Hex.

```
SEL MB0              ;SELECT MEMORY BANK 0
JMP $+20             ;JUMP TO LOCATION
                     ;48 HEX
```

SEL MB1 Select Memory Bank 1 (Not in 8x41)

Opcode:

1 1 1 1	0 1 0 1

PC bit 11 is set to one on next JMP or CALL instruction. All references to program memory addresses fall within the range 1024–2047.
(DBF) ← 1

SEL RB0 Select Register Bank 0

Opcode:

1 1 0 0	0 1 0 1

PSW BIT 4 is set to zero. References to working registers 0–7 address data memory locations 0–7. This is the recommended setting for normal program execution.
(BS) ← 0

SEL RB1 Select Register Bank 1

Opcode:

1 1 0 1	0 1 0 1

PSW bit 4 is set to one. References to working registers 0–7 address data memory locations 24–31. This is the recommended setting for interrupt service routines, since locations 0–7 are left intact. The setting of PSW bit 4 in effect at the time of an interrupt is restored by the RETR instruction when the interrupt service routine is completed.

Example: Assume an IBF interrupt has occurred, control has passed to program memory location 3, and PSW bit 4 was zero before the interrupt.

```
LOC3: JMP INIT              ;JUMP TO ROUTINE 'INIT'
        .
        .
        .
INIT: MOV R7,A              ;MOV ACC CONTENTS TO
                            ;LOCATION 7
      SEL RB1               ;SELECT REG BANK 1
      MOV R7,#OFAH          ;MOVE 'FA' HEX TO LOCATION 31
        .
        .
        .
      SEL RB0               ;SELECT REG BANK 0
      MOV A,R7              ;RESTORE ACC FROM LOCATION 7
      RETR                  ;RETURN——RESTORE PC AND PSW
```

STOP TCNT Stop Timer/Event Counter

Opcode:

0 1 1 0	0 1 0 1

This instruction is used to stop both time accumulation and event counting.

Example: Disable interrupt, but jump to interrupt routine after eight overflows and stop timer. Count overflows in register 7.

```
START: DIS TCNTI          ;DISABLE TIMER INTERRUPT
       CLR A              ;CLEAR ACC TO ZERO
       MOV T,A            ;MOV ZERO TO TIMER
       MOV R7,A           ;MOVE ZERO TO REG 7
       STRT T             ;START TIMER
MAIN:  JTF COUNT          ;JUMP TO ROUTINE 'COUNT'
                          ;IF TF = 1 AND CLEAR TIMER FLAG
       JMP MAIN           ;CLOSE LOOP
COUNT: INC R7             ;INCREMENT REG 7
       MOV A,R7           ;MOVE REG 7 CONTENTS TO ACC
       JB3 INT            ;JUMP TO ROUTINE 'INT' IF ACC
                          ;BIT 3 IS SET (REG 7 = 8)
       JMP MAIN           ;OTHERWISE RETURN TO ROUTINE
                          ;MAIN
        .
        .
        .
INT: STOP TCNT            ;STOP TIMER
       JMP 7H             ;JUMP TO LOCATION 7 (TIMER
                          ;INTERRUPT ROUTINE)
```

STRT CNT Start Event Counter

Opcode:

0 1 0 0	0 1 0 1

The TEST 1 (T_1) pin is enabled as the event-counter input and the counter is started. The event-counter register is incremented with each high to low transition on the T_1 pin.

Example: Initialize and start event counter. Assume overflow is desired with first T_1 input.

```
STARTC: EN TCNTI          ;ENABLE COUNTER INTERRUPT
        MOV A,#OFFH       ;MOVE 'FF' HEX (ONES) TO
                          ;ACC
        MOV T,A           ;MOVE ONES TO COUNTER
        STRT CNT          ;INPUT AND START
```

STRT T Start Timer

Opcode:

0 1 0 1	0 1 0 1

Timer accumulation is initiated in the timer register. The register is incremented every 32 instruction cycles. The prescaler which counts the 32 cycles is cleared but the timer register is not.

Example: Initialize and start timer.

```
STARTT: EN TCNTI          ;ENABLE TIMER INTERRUPT
        CLR A             :CLEAR ACC TO ZEROS
        MOV T,A           ;MOVE ZEROS TO TIMER
        STRT T            ;START TIMER
```

SWAP A Swap Nibbles Within Accumulator

Opcode:

0 1 0 0	0 1 1 1

Bits 0–3 of the accumulator are swapped with bits 4-7 of the accumulator.
$(A_{4-7}) \longleftrightarrow (A_{0-3})$

Example: Pack bits 0–3 of locations 50-51 into location 50.

```
PCKDIG: MOV R0,#50          ;MOVE '50' DEC TO REG 0
        MOV R1,#51          ;MOVE '51' DEC TO REG 1
        XCHD A,@R0          ;EXCHANGE BIT 0-3 OF ACC
                            ;AND LOCATION 50
        SWAP A              ;SWAP BITS 0-3 AND 4-7 OF ACC
        XCHD A,@ R1         ;EXCHANGE BITS 0-3 OF ACC AND
                            ;LOCATION 51
        MOV @R0,A           ;MOVE CONTENTS OF ACC TO
                            ;LOCATION 51
```

XCH ARr Exchange Accumulator-Register Contents

Opcode:

0 0 1 0	1 r_2 r_1 r_0

The contents of the accumulator and the contents of working register 'r' are exchanged.
(A) $\longleftrightarrow$ (Rr) r = 0–7

Example: Move PSW contents to Reg 7 without losing accumulator contents.

```
XCHAR7: XCH A,R7             ;EXCHANGE CONTENTS OF REG 7
                             ;AND ACC
        MOV A,PSW            ;MOVE PSW CONTENTS TO ACC
        XCH, A,R7            ;EXCHANGE CONTENTS OF REG 7
                             ;AND ACC AGAIN
```

XCH A,@Rr Exchange Accumulator and Data Memory Contents

Opcode:

0 0 1 0	0 0 0 r

The contents of the accumulator and the contents of the data memory location addressed by bits 0–5 of register 'r' are exchanged. Register 'r' contents are unaffected.
(A) $\longleftrightarrow$ ((Rr)) r = 0–1

Example: Decrement contents of location 52.

```
DEC 52: MOV R0,#52           ;MOVE '52' DEC TO ADDRESS
                             ;REG 0
        XCH A,@R0            ;EXCHANGE CONTENTS OF ACC
                             ;AND LOCATION 52
        DEC A                ;DECREMENT ACC CONTENTS
        XCH A,@R0            ;EXCHANGE CONTENTS OF ACC
                             ;AND LOCATION 52 AGAIN
```

XCHD A,@Rr Exchange Accumulator and Data Memory 4-bit Data

Opcode: | 0 0 1 1 | 0 0 0 r |

This instruction exchanges bits 0–3 of the accumulator with bits 0–3 of the data memory location addressed by bits 0–5 of register 'r'. Bits 4–7 of the accumulator, bits 4–7 of the data memory location, and the contents of register 'r' are unaffected.

$(A_{0-3}) \longleftrightarrow ((Rr_{0-3}))$ r = 0–1

Example: Assume program counter contents have been stacked in locations 22-23.

```
XCHNIB: MOV R0,#23      ;MOVE '23' DEC TO REG 0
        CLR A           ;CLEAR ACC TO ZEROS
        XCHD A,@R0      ;EXCHANGE BITS 0-3 OF ACC
                        ;AND LOCATION 23 (BITS 8-11
                        ;OF PC ARE ZEROED, ADDRESS
                        ;REFERS TO PAGE 0)
```

XRL A,Rr Logical XOR Accumulator With Register Mask

Opcode: | 1 1 0 1 | 1 r_2 r_1 r_0 |

Data in the accumulator is EXCLUSIVE ORed with the mask contained in working register 'r'.

(A) $\longleftrightarrow$ (A) XOR (Rr) r = 0–7

Example:

```
XORREG: XRL A,R5         ;'XOR' ACC CONTENTS WITH
                         ;MASK IN REG 5
```

XRL A,@Rr Logical XOR Accumulator With Memory Mask

Opcode: | 1 1 0 1 | 0 0 0 r |

Data in the accumulator is EXCLUSIVE ORed with the mask contained in the data memory location address by register 'r', bits 0–5.

(A) $\longleftarrow$ (A) XOR ((Rr)) r = 0–1

Example:

```
XORDM: MOV R1,#20H       ;MOVE '20' HEX TO REG 1
       XRL A,@R1         ;'XOR' ACC CONTENTS WITH MASK
                         ;IN LOCATION 32
```

XRL A,#data, Logical XOR Accumulator With Immediate Mask

Opcode: | 1 1 0 1 | 0 0 1 1 | • | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 |

This is a 2-cycle instruction. Data in the accumulator is EXCLUSIVE ORed with an immediately-specified mask.

(A) $\longleftarrow$ (A) XOR data

Example:

```
XORID: XOR A,#HEXTEN     ;XOR CONTENTS OF ACC WITH
                         ;MASK EQUAL VALUE OF SYMBOL
                         ;'HEXTEN'
```

CHAPTER 4
SINGLE-STEP AND PROGRAMMING POWER-DOWN MODES

SINGLE-STEP

The UPI family has a single-step mode which allows the user to manually step through his program one instruction at a time. While stopped, the address of the next instruction to be fetched is available on PORT 1 and the lower 2 bits of PORT 2. The single-step feature simplifies program debugging by allowing the user to easily follow program execution.

Figure 4-1 illustrates a recommended circuit for single-step operation, while Figure 4-2 shows the timing relationship between the SYNC output and the $\overline{SS}$ input. During single-step operation, PORT 1 and part of PORT 2 are used to output address information. In order to retain the normal I/O functions of PORTS 1 and 2, a separate latch can be used as shown in Figure 4-3.

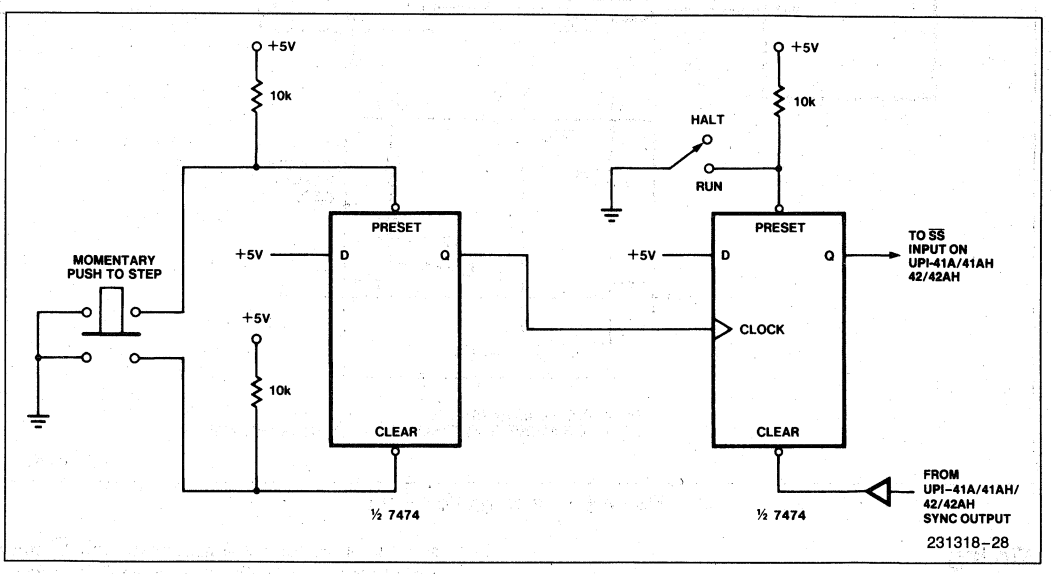


Figure 4-1. Single-Step Circuit

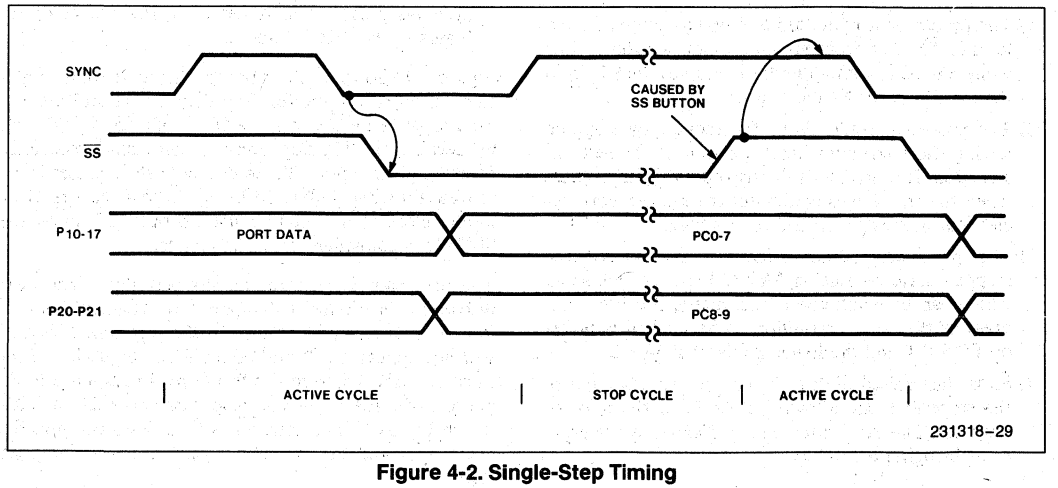


Figure 4-2. Single-Step Timing

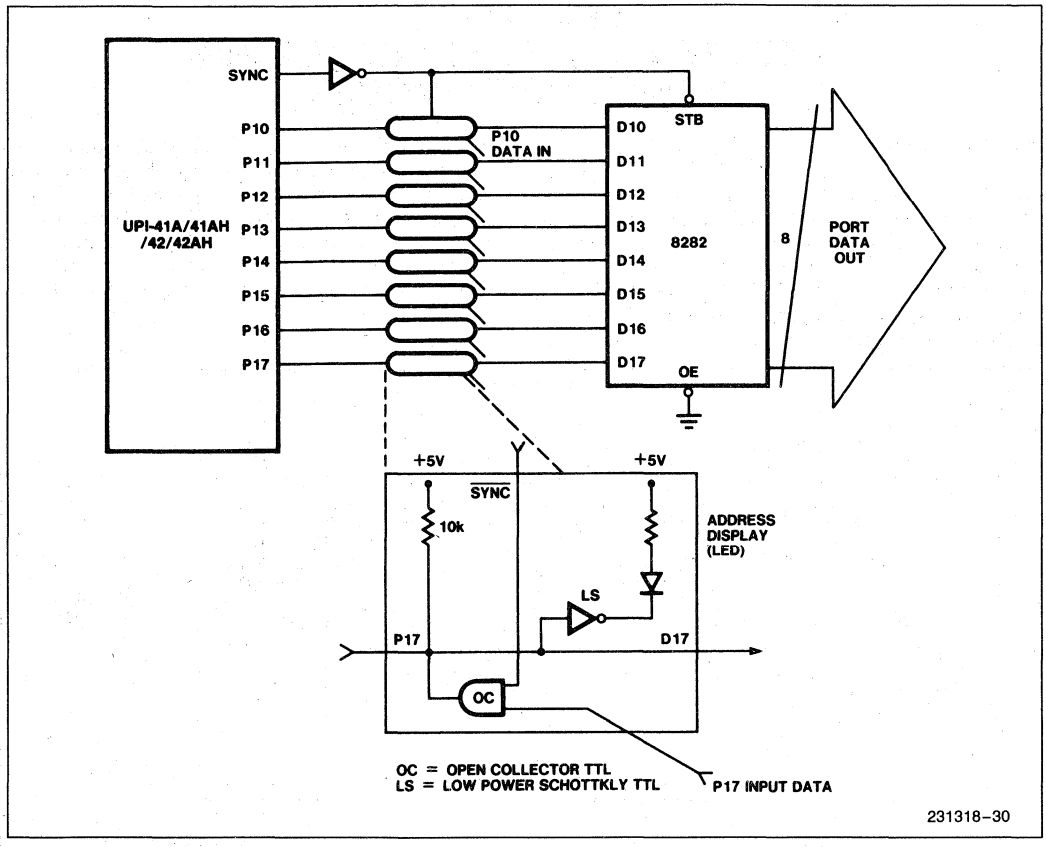


Figure 4-3. Latching Port Data

Timing

The sequence of single-step operation is as follows:

1) The processor is requested to stop by applying a low level on $\overline{SS}$. The $\overline{SS}$ input should not be brought low while SYNC is high. (The UPI samples the $\overline{SS}$ pin in the middle of the SYNC pulse).
2) The processor responds to the request by stopping during the instruction fetch portion of the next instruction. If a double cycle instruction is in progress when the single-step command is received, both cycles will be completed before stopping.
3) The processor acknowledges it has entered the stopped state by raising SYNC high. In this state, which can be maintained indefinitely, the 10-bit address of the next instruction to be fetched is preset on PORT 1 and the lower 2 bits of PORT 2.
4) $\overline{SS}$ is then raised high to bring the processor out of the stopped mode allowing it to fetch the next instruction.The exit from stop is indicated by the processor bringing SYNC low.
5) To stop the processor at the next instruction $\overline{SS}$ must be brought low again before the next SYNC pulse—the circuit in Figure 4-1 uses the trailing edge of the previous pulse. If $\overline{SS}$ is left high, the processor remains in the "RUN" mode.

Figure 4-1 shows a schematic for implementing single-step. A single D-type flip-flop with preset and clear is used to generate $\overline{SS}$. In the RUN mode $\overline{SS}$ is held high by keeping the flip-flop preset (preset has precedence over the clear input). To enter single-step, preset is removed allowing SYNC to bring $\overline{SS}$ low via the clear input. Note that SYNC must be buffered since the SN7474 is equivalent to 3 TTL loads.

The processor is now in the stopped state. The next instruction is initiated by stoppe state. The next instruction is initiated by clocking "1" the flip-flop. This "1" will not appear on $\overline{SS}$ unless SYNC is high (I.e., clear must be removed from the flip-flop). In response to $\overline{SS}$ going high, the processor begins an instruction fetch which brings SYNC low. $\overline{SS}$ is then reset through the clear input and the processor again enters the stopped state.

EXTERNAL ACCESS

The UPI family has an External Access mode (EA) which puts the processor into a test mode. This mode allows the user to disable the internal program memory and execute from external memory. External Access mode is useful in testing because it allows the user to test the processor's functions directly. It is only useful for testing since this mode uses D_0–D_7, PORTS 10–17 and PORTS 20–22.

This mode is invoked by connecting the EA pin to 5V. The 11-bit current program counter contents then come out on PORTS 10–17 and PORTS 20–22 after the SYNC output goes high. (PORT 10 is the least significant bit.) The desired instruction opcode is placed on D_0–D_7 before the start of state S_1. During state S_1, the opcode is sampled from D_0–D_7 and subsequently executed in place of the internal program memory contents.

The program counter contents are multiplexed with the I/O port data on PORTS 10–17 and PORTS 20–22. The I/O port data may be demultiplexed using an external latch on the rising edge of SYNC. The program counter contents may be demultiplexed similarly using the trailing edge of SYNC.

Reading and/or writing the Data Bus Buffer registers is still allowed although only when D_0–D_7 are not being sampled for opcode data. In practice, since this sampling time is not known externally, reads or writes on the system bus are done during SYNC high time. Approximately 600 ns are available for each read or write cycle.

POWER DOWN MODE (UPI-41AH/42AH ONLY)

Extra circuitry is included in the UPI-41AH/42AH version to allow low-power, standby operation. Power is removed from all system elements except the internal data RAM in the low-power mode. Thus the contents of RAM can be maintained and the device draws only 10 to 15% of its normal power.

The V_{CC} pin serves as the 5V power supply pin for all of the UPI-41AH/42AH version's circuitry except the data RAM array. The V_{DD} pin supplies only the RAM array. In normal operation, both V_{CC} and V_{DD} are connected to the same 5V power supply.

To enter the Power-Down mode, the $\overline{\text{RESET}}$ signal to the UPI is asserted. This ensures the memory will not be inadvertently altered by the UPI during power-down. The V_{CC} pin is then grounded while V_{DD} is maintained at 5V. Figure 4-4 illustrates a recommended Power-Down sequence. The sequence typically occurs as follows:

1) Imminent power supply failure is detected by user defined circuitry. The signal must occur early enough to guarantee the UPI-41AH/42AH can save all necessary data before V_{CC} falls outside normal operating tolerance.
2) A "Power Failure" signal is used to interrupt the processor (via a timer overflow interrupt, for instance) and call a Power Failure service routine.
3) The Power Failure routine saves all important data and machine status in the RAM array. The routine may also initiate transfer of a backup supply to the V_{DD} pin and indicate to external circuitry that the Power Failure routine is complete.
4) A $\overline{\text{RESET}}$ signal is applied by external hardware to guarantee data will not be altered as the power supply falls out of limits. $\overline{\text{RESET}}$ must be low until V_{CC} reaches ground potential.

Recovery from the Power-Down mode can occur as any other power-on sequence. An external 1 μfd capacitor on the $\overline{\text{RESET}}$ input will provide the necessary initialization pulse.

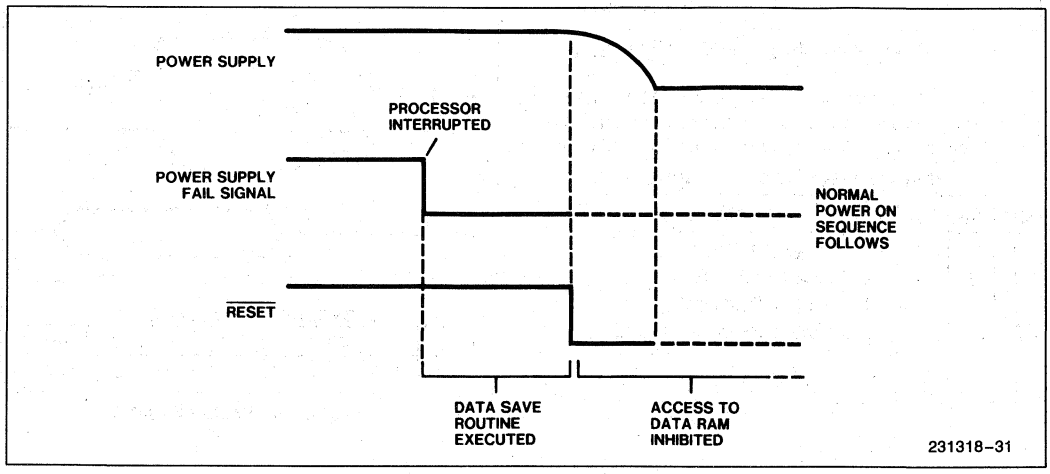


Figure 4-4. Power-Down Sequence

CHAPTER 5
SYSTEM OPERATION

BUS INTERFACE

The UPI-41A/41AH/42/42AH Microcomputer functions as a peripheral to a master processor by using the data bus buffer registers to handle data transfers. The DBB configuration is illustrated in Figure 5-1. The UPI Microcomputer's 8 three-state data lines (D_7–D_0) connect directly to the master processor's data bus. Data transfer to the master is controlled by 4 external inputs to the UPI:

- A_0 Address Input signifying command or data
- $\overline{CS}$ Chip Select
- $\overline{RD}$ Read strobe
- $\overline{WR}$ Write strobe

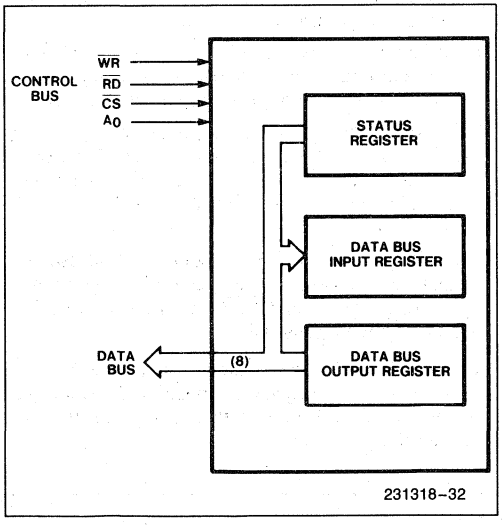


Figure 5-1. Data Bus Register Configuration

The master processor addresses the UPI-41A/41AH/42/42AH Microcomputer as a standard peripheral device. Table 5-1 shows the conditions for data transfer:

Table 5-1. Data Transfer Controls

$\overline{CS}$	A_0	$\overline{RD}$	$\overline{WR}$	Condition
0	0	0	1	Read DBBOUT
0	1	0	1	Read STATUS
0	0	1	0	Write DBBIN data, set $F_1 = 0$
0	1	1	0	Write DBBIN command set $F_1 = 1$
1	x	x	x	Disable DBB

Reading the DBBOUT Register

The sequence for reading the DBBOUT register is shown in Figure 5-2. This operation causes the 8-bit contents of the DBBOUT register to be placed on the system Data Bus. The OBF flag is cleared automatically.

Reading STATUS

The sequence for reading the UPI Microcomputer's 8 STATUS bits is shown in Figure 5-3. This operation causes the 8-bit STATUS register contents to be placed on the system Data Bus as shown.

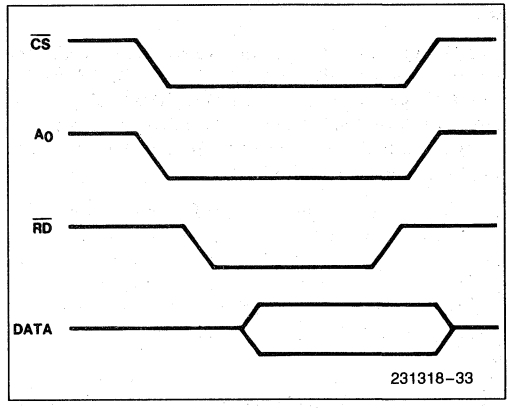


Figure 5-2. DBBOUT Read

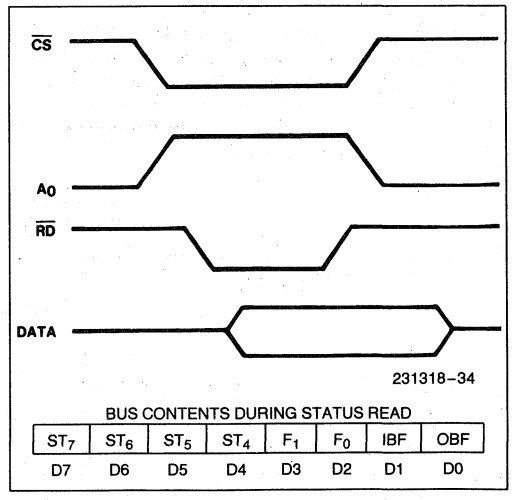


Figure 5-3. Status Read

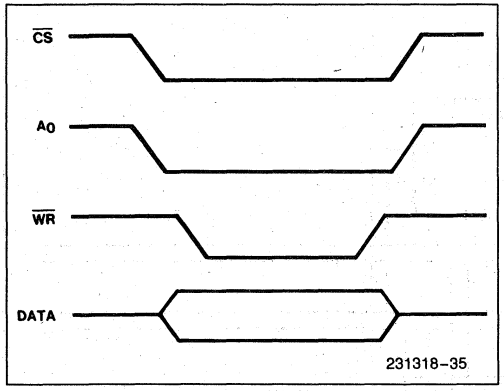


Figure 5-4. Writing Data to DBBIN

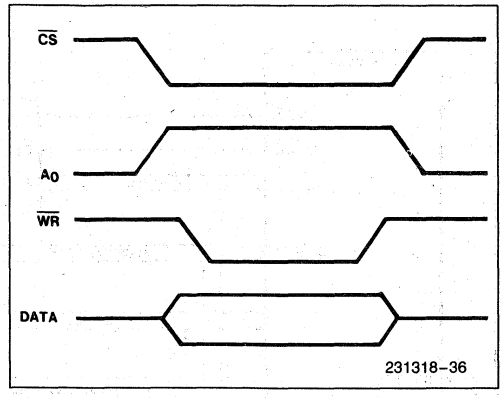


Figure 5-5. Writing Commands to DBBIN

Write Data to DBBIN

The sequence for writing data to the DBBIN register is shown in Figure 5-4. This operation causes the system Data Bus contents to be transferred to the DBBIN register and the IBF flag is set. Also, the F_1 flag is cleared ($F_1 = 0$) and an interrupt request is generated. When the IBF interrupt is enabled, a jump to location 3 will occur. The interrupt request is cleared upon entering the IBF service routine or by a system RESET input.

Writing Commands to DBBIN

The sequence for writing commands to the DBBIN register is shown in Figure 5-5. This sequence is identical to a data write except that the A_0 input is latched in the F_1 flag ($F_1 = 1$). The IBF flag is set and an interrupt request is generated when the master writes a command to DBB.

Operations of Data Bus Registers

The UPI-41A/41AH/42/42AH Microcomputer controls the transfer of DBB data to its accumulator by executing INput and OUTput instructions. An IN A,DBB instruction causes the contents to be transferred to the UPI accumulator and the IBF flag is cleared.

The OUT DBB,A instruction causes the contents of the accumulator to be transferred to the DBBOUT register. The OBF flag is set.

The UPI's data bus buffer interface is applicable to a variety of microprocessors including the 8086, 8088, 8085AH, 8080, and 8048.

A description of the interface to each of these processors follows.

DESIGN EXAMPLES

8085AH Interface

Figure 5-6 illustrates an 8085AH system using a UPI-41A/41AH/42/42AH. The 8085AH system uses a multiplexed address and data bus. During I/O the 8 upper address lines (A_8-A_{15}) contain the same I/O address as the lower 8 address/data lines (A_0-A_7); therefore I/O address decoding is done using only the upper 8 lines to eliminate latching of the address. An 8205 decoder provides address decoding for both the UPI and the 8237. Data is transferred using the two DMA handshaking lines of PORT 2. The 8237 performs the actual bus transfer operation. Using the UPI-41A/41AH/42/42AH's OBF master interrupt, the UPI notifies the 8085AH upon transfer completion using the RST 5.5 interrupt input. The $\overline{IBF}$ master interrupt is not used in this example.

8088 Interface

Figure 5-7 illustrates a UPI-41A/41AH/42/42AH interface to an 8088 minimum mode system. Two 8-bit latches are used to demultiplex the address and data bus. The address bus is 20-lines wide. For I/O only, the lower 16 address lines are used, providing an addressing range of 64K. UPI address selection is accomplished using an 8205 decoder. The A_0 address line of the bus is connected to the corresponding UPI input for register selection. Since the UPI is polled by the 8088, neither DMA nor master interrupt capabilities of the UPI are used in the figure.

8086 Interface

The UPI-41A/41AH/42/42AH can be used on an 8086 maximum mode system as shown in Figure 5-8. The address and data bus is demultiplexed using three

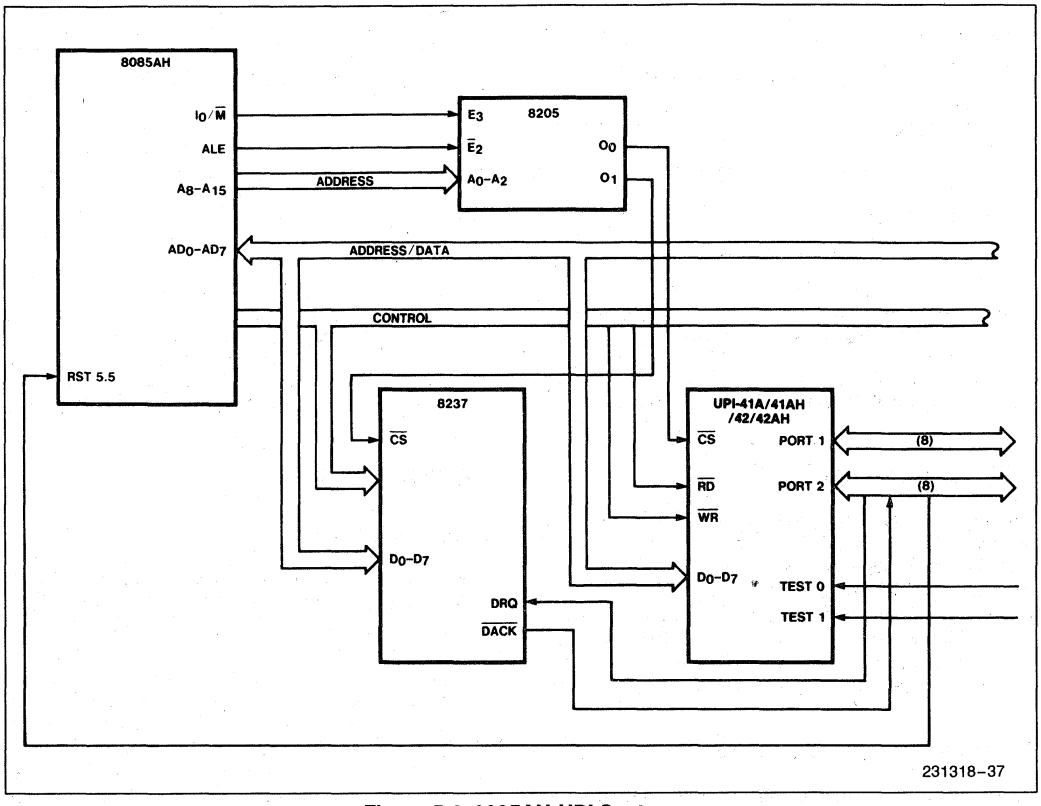


Figure 5-6. 8085AH-UPI System

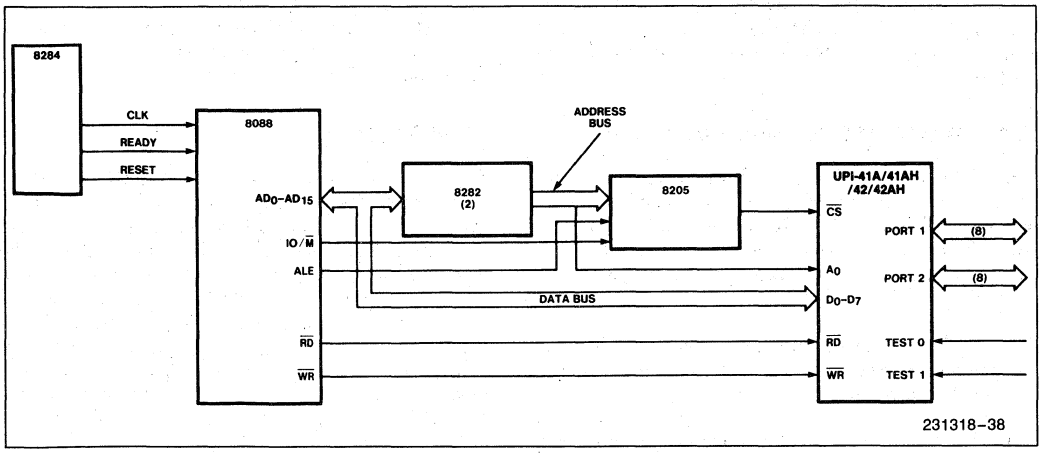


Figure 5-7. 8088-UPI Minimum Mode System

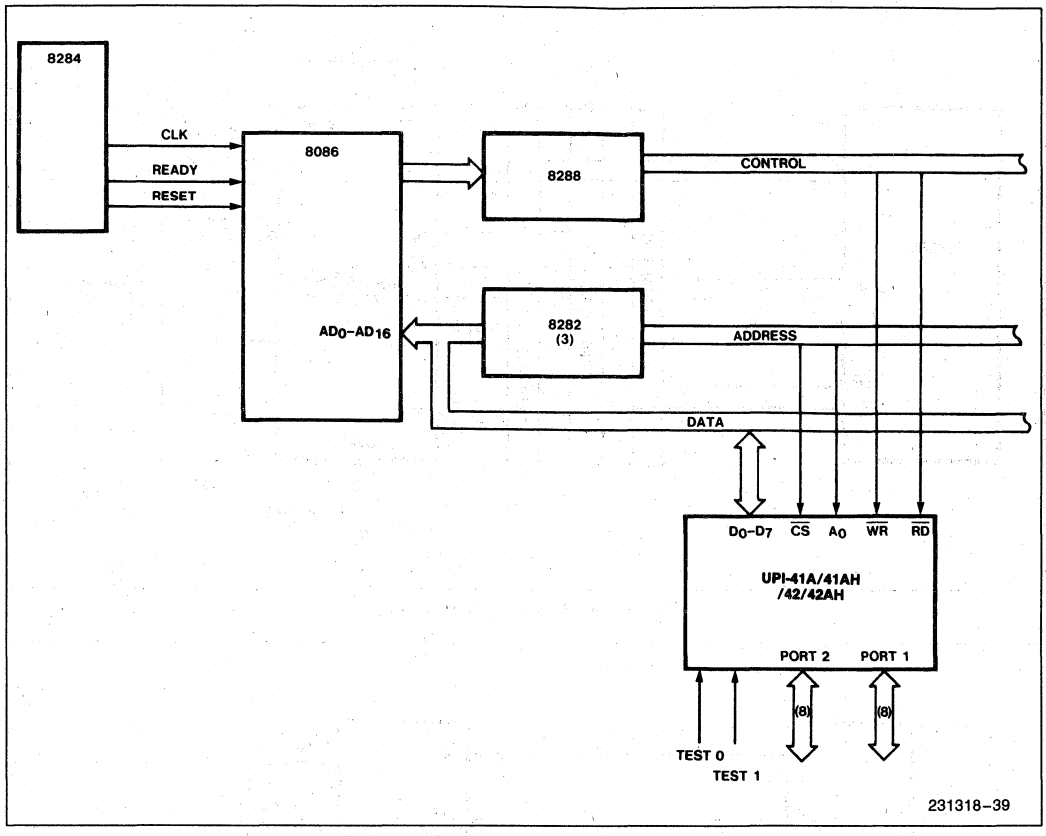


Figure 5-8. 8086-UPI Maximum Mode Systems

8282 latches providing separate address and data buses. The address bus is 20-lines wide and the data bus is 16-lines wide. Multiplexed control lines are decoded by the 8288. The UPI's $\overline{CS}$ input is provided by linear selection. Note that the UPI is both I/O mapped and memory mapped as a result of the linear addressing technique. An address decoder may be used to limit the UPI-41A/41AH/42/42AH to a specific I/O mapped address. Address line A_1 is connected to the UPI's A_0 input. This insures that the registers of the UPI will have even I/O addresses. Data will be transferred on D_0–D_7 lines only. This allows the I/O registers to be accessed using byte manipulation instructions.

8080 Interface

Figure 5-9 illustrates the interface to an 8080A system. In this example, a crystal and capacitor are used for UPI-41A/41AH/42/42AH timing reference and power-on RESET. If the 2-MHz 8080A 2-phase clock were used instead of the crystal, the UPI-41A/41AH/42/42AH would run at only 16% full speed.

The A_0 and $\overline{CS}$ inputs are direct connections to the 8080 address bus. In larger systems, however, either of these inputs may be decoded from the 16 address lines.

The $\overline{RD}$ and $\overline{WR}$ inputs to the UPI can be either the $\overline{IOR}$ and $\overline{IOW}$ or the $\overline{MEMR}$ and $\overline{MEMR}$ signals depending on the I/O mapping technique to be used.

The UPI can be addressed as an I/O device using INput and OUTput instructions in 8080 software.

8048 Interface

Figure 5-10 shows the UPI interface to an 8048 master processor.

The 8048 $\overline{RD}$ and $\overline{WR}$ outputs are directly compatible with the UPI. Figure 5-11 shows a distributed processing system with up to seven UPI's connected to a single 8048 master processor.

In this configuration the 8048 uses PORT 0 as a data bus. I/O PORT 2 is used to select one of the seven

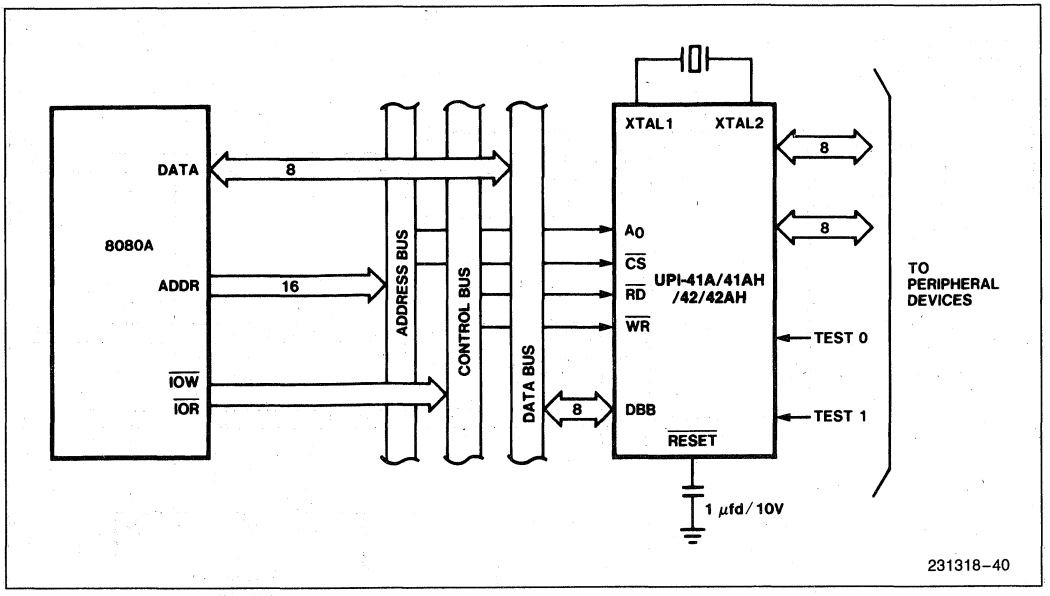


Figure 5-9. 8080A-UPI Interface

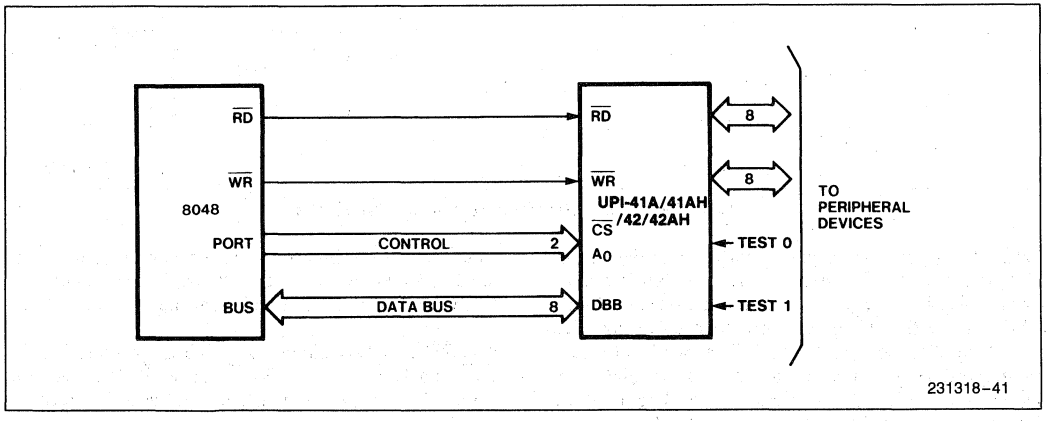


Figure 5-10. 8048-UPI Interface

UPI's when data transfer occurs. The UPI's are programmed to handle isolated tasks and, since they operate in parallel, system throughput is increased.

GENERAL HANDSHAKING PROTOCOL

1) Master reads STATUS register ($\overline{RD}$, $\overline{CS}$, A_0 = (0, 0, 1)) in polling or in response to either an $\overline{IBF}$ or an OBF interrupt.
2) If the UPI DBBIN register is empty (IBF flag = 0), Master writes a word to the DBBIN register ($\overline{WR}$, $\overline{CS}$, A_0 = (0, 0, 1) or (0, 0, 0)). If A_0 = 1, write command word, set F_1. If A_0 = 0, write data word, F_1 = 0.
3) If the UPI DBBOUT register is full (OBF flag = 1), Master reads a word from the DBBOUT register ($\overline{RD}$, $\overline{CS}$, A_0 = (0,0, 0)).
4) UPI recognizes IBF (via IBF interrupt or JNIBF). Input data or command word is processed, depending on F1; IBF is reset. Repeat step 1 above.
5) UPI recognizes OBF flag = 0 (via JOBF). Next word is output to DBBOUT register, OBF is set. Repeat step 1 above.

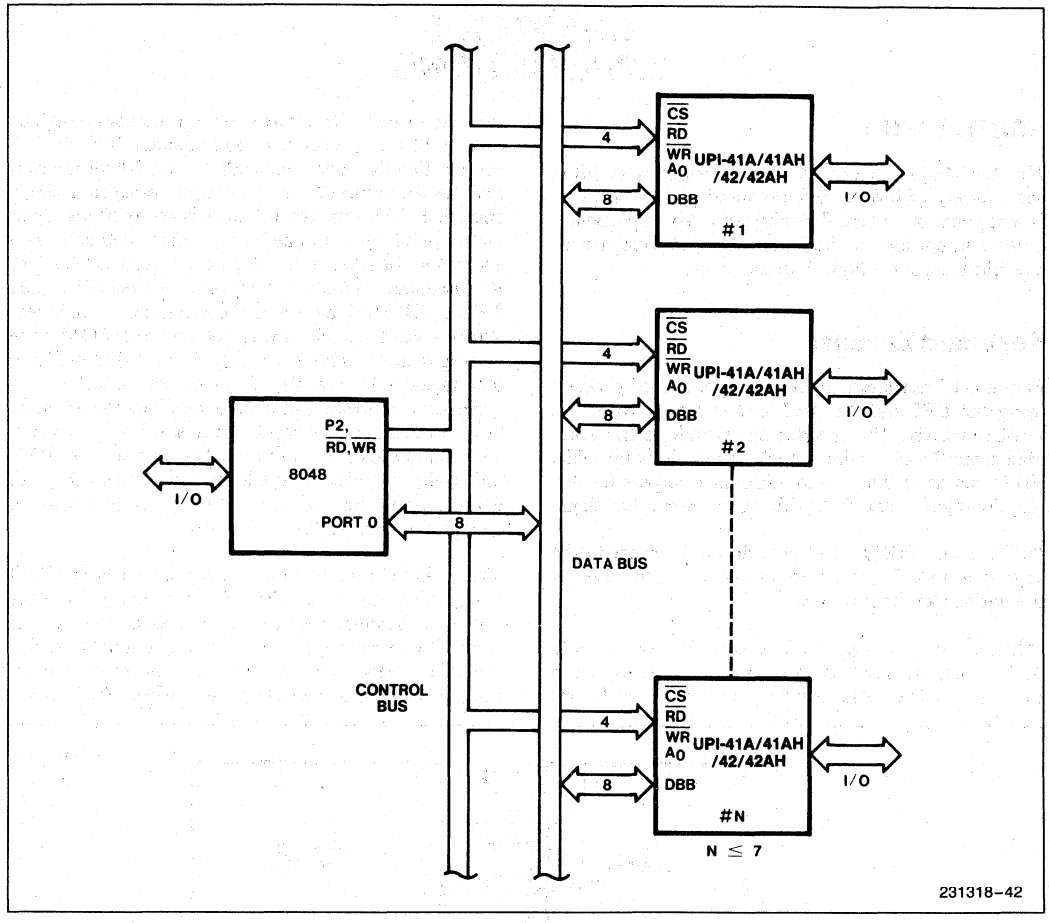


Figure 5-11. Distributed Processor System

CHAPTER 6
APPLICATIONS

ABSTRACTS

The UPI-41A/41AH/42/42AH is designed to fill a wide variety of low to medium speed peripheral interface applications where flexibility and easy implementation are important considerations. The following examples illustrate some typical applications.

Keyboard Encoder

Figure 6-1 illustrates a keyboard encoder configuration using the UPI and the 8243 I/O expander to scan a 128-key matrix. The encoder has switch matrix scanning logic, N-key rollover logic, ROM look-up table, FIFO character buffer, and additional outputs for display functions, control keys or other special functions.

PORT 1 and PORTs 4–7 provide the interface to the keyboard. PORT 1 lines are set one at a time to select the various key matrix rows.

When a row is energized all 16 columns (i.e., PORTs 4–7 inputs) are sampled to determine if any switch in the row is closed. The scanning software is code efficient because the UPI instruction set includes individual bit set/clear operations and expander PORTs 4–7 can be directly addressed with single, 2-byte instructions. Also, accumulator bits can be tested in a single operation. Scan time for 128 keys is about 10 ms. Each matrix point has a unique binary code which is used to address ROM when a key closure is detected. Page 3 of ROM contains a look-up table with useable codes (i.e., ASCII, EBCDIC, etc.) which correspond to each key. When a valid key closure is detected the ROM code corresponding to that key is stored in a FIFO buffer in data memory for transfer to the master processor. To avoid stray noise and switch bounce, a key closure must be detected on two consecutive scans before it is considered valid and loaded into the FIFO buffer. The FIFO buffer allows multiple keys to be processed as they are depressed without regard to when they are released, a condition known as N-key rollover.

The basic features of this encoder are fairly standard and require only about 500 bytes of memory. Since the UPI is programmable and has additional memory capacity it can handle a number of other functions. For example, special keys can be programmed to give an entry on closing as well as opening. Also, I/O lines are

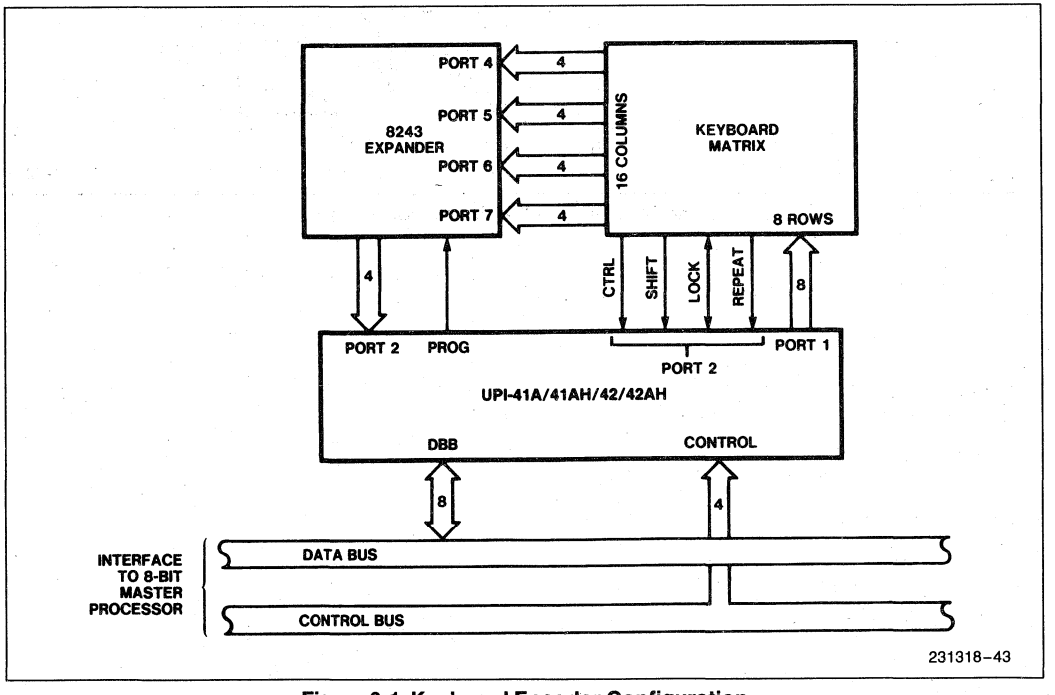


Figure 6-1. Keyboard Encoder Configuration

available to control a 16-digit, 7-segment display. The UPI can also be programmed to recognize special combinations of characters such as commands, then transfer only the decoded information to the master processor.

A complete keyboard application has been developed for the UPI-41A/41AH/42/42AH description is included in this section. The code for the application is available in the Intel Insite Library (program AB 147).

Matrix Printer Interface

The matrix printer interface illustrated in Figure 6-2 is a typical application for the UPI. The actual printer mechanism could be any of the numerous dot-matrix types and similar configurations can be shown for drum, spherical head, daisy wheel or chain type printers.

The bus structure shown represents a generalized, 8-bit system bus configuration. The UPI's three-state interface port and asynchronous data buffer registers allow it to connect directly to this type of system for efficient, two-way data transfer.

The UPI's two on-board I/O ports provide up to 16 input and output signals to control the printer mechanism. The timer/event counter is used for generating a timing sequence to control print head position, line feed, carriage return, and other sequences. The on-board program memory provides character generation for 5 x 7, 7 x 9, or other dot matrix formats. As an added feature a portion of the data memory can be used as a FIFO buffer so that the master processor can send a block of data at a high rate. The UPI can then output characters from the buffer at a rate the printer can accept while the master processor returns to other tasks.

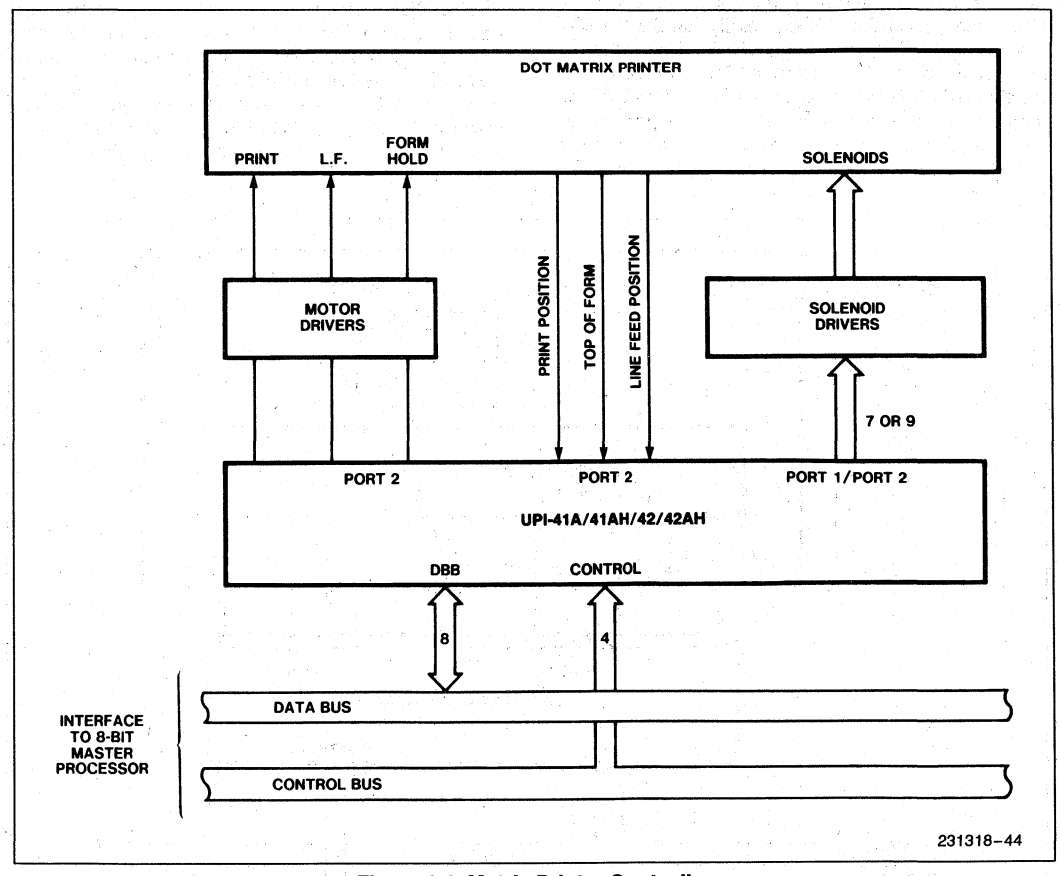


Figure 6-2. Matrix Printer Controller

The 8295 Printer Controller is an example of an UPI preprogrammed as a dot matrix printer interface.

Tape Cassette Controller

Figure 6-3 illustrates a digital cassette interface which can be implemented with the UPI. Two sections of the tape transport are controlled by the UPI: digital data/command logic, and motor servo control.

The motor servo requires a speed reference in the form of a monostable pulse whose width is proportional to the desired speed. The UPI monitors a prerecorded clock from the tape and uses its on-board interval timer to generate the required speed reference pulses at each clock transition.

Recorded data from the tape is supplied serially by the data/command logic and is converted to 8-bit words by the UPI, then transferred to the master processor. At 10 ips tape speed the UPI can easily handle the 8000 bps data rate. To record data, the UPI uses the two input lines to the data/command logic which control the flux direction in the recording head. The UPI also monitors 4 status lines from the tape transport including: end of tape, cassette inserted, busy, and write permit. All control signals can be handled by the UPI's two I/O ports.

Universal I/O Interface

Figure 6-4 shows an I/O interface design based on the UPI. This configuration includes 12 parallel I/O lines and a serial (RS232C) interface for full duplex data transfer up to 1200 baud. This type of design can be used to interface a master processor to a broad spectrum of peripheral devices as well as to a serial communication channel.

PORT 1 is used strictly for I/O in this example while PORT 2 lines provide five functions:

- P_{23}–P_{20} I/O lines (bidirectional)
- P_{24} Request to send (RTS)
- P_{25} Clear to send (CTS)
- P_{26} Interrupt to master
- P_{27} Serial data out

The parallel I/O lines make use of the bidirectional port structure of the UPI. Any line can function as an input or output. All port lines are automatically initialized to 1 by a system $\overline{\text{RESET}}$ pulse and remain latched. An external TTL signal connected to a port line will override the UPI's 50 KΩ internal pull-up so that an INPUT instruction will correctly sample the TTL signal.

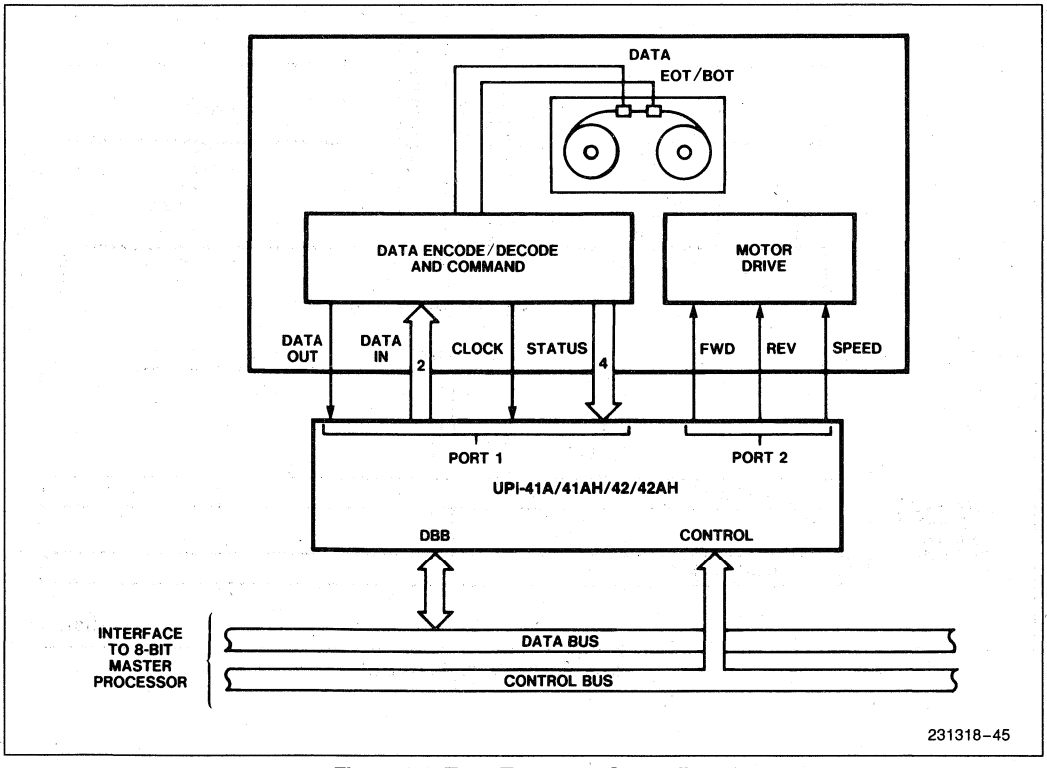


Figure 6-3. Tape Transport Controller

Four PORT 2 lines function as general I/O similar to PORT 1. Also, the RTS signal is generated on PORT 2 under software control when the UPI has serial data to send. The CTS signal is monitored via PORT 2 as an enable to the UPI to send serial data. A PORT 2 line is also used as a software generated interrupt to the master processor. The interrupt functions as a service request when the UPI has a byte of data to transfer or when it is ready to receive. Alternatively, the EN FLAGS instruction could be used to create the OBF and $\overline{IBF}$ interrupts on P_{24} and P_{25}.

The RS232C interface is implemented using the TEST 0 pin as a receive input and a PORT 2 pin as a transmit output. External packages (A_0, A_1) are used to provide RS232C drive requirements. The serial receive software is interrupt driven and uses the on-chip timer to perform time critical serial control. After a start bit is detected the interval timer can be preset to generate an interrupt at the proper time for sampling the serial bit stream. This eliminates the need for software timing loops and allows the processor to proceed to other tasks (i.e., parallel I/O operations) between serial bit samples. Software flags are used so the main program can determine when the interrupt driven receive program has a character assembled for it.

This type of configuration allows system designers flexibility in designing custom I/O interfaces for specific serial and parallel I/O applications. For instance, a second or third serial channel could be substituted in place of the parallel I/O if required. The UPI's data memory can buffer data and commands for up to 4 low-speed channels (110 baud teletypewriter, etc.)

Application Notes

The following application notes illustrate the various applications of the UPI family. Other related publications including the *Microcontroller Handbook* are available through the Intel Literature Department.

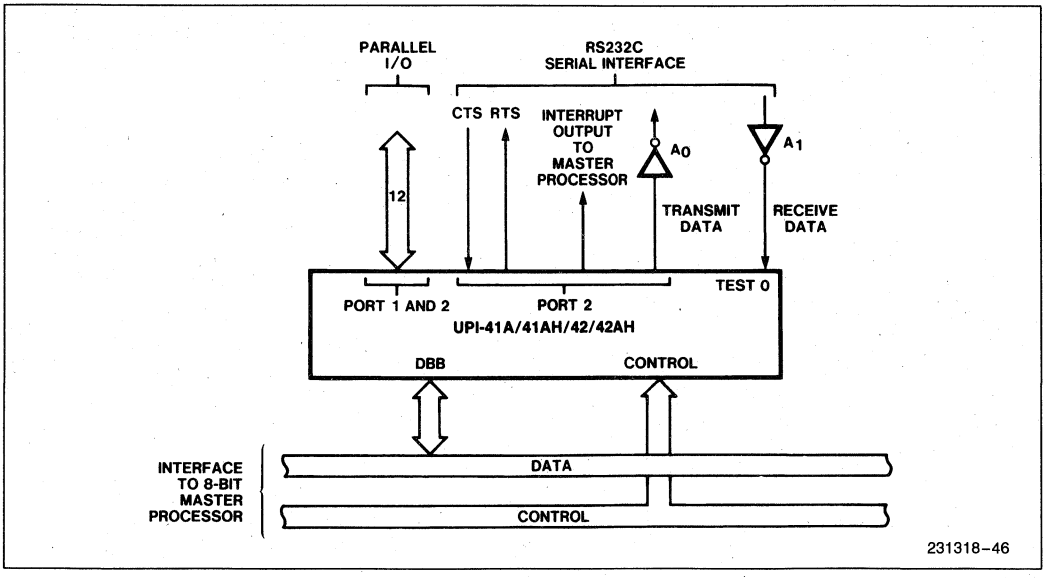


Figure 6-4. Universal I/O Interface

UPI-452
CHMOS PROGRAMMABLE I/O PROCESSOR

83C452 - 8K × 8 Mask Programmable Internal ROM

80C452 - External ROM/EPROM

- **83C452/80C452:3.5 to 14 MHz Clock Rate**
- **Software Compatible with the MCS-51 Family**
- **128-Byte Bi-Directional FIFO Slave Interface**
- **Two DMA Channels**
- **256 × 8-Bit Internal RAM**
- **34 Additional Special Function Registers**
- **40 Programmable I/O Lines**
- **Two 16-Bit Timer/Counters**
- **Boolean Processor**
- **Bit Addressable RAM**
- **8 Interrupt Sources**
- **Programmable Full Duplex Serial Channel**
- **64K Program Memory Space**
- **64K Data Memory Space**
- **68-Pin PGA and PLCC**
  (See Packaging Spec., Order: #231369)

The Intel UPI-452 (Universal Peripheral Interface) is a 68 pin CHMOS Slave I/O Processor with a sophisticated bi-directional FIFO buffer interface on the slave bus and a two channel DMA processor on-chip. The UPI-452 is the newest member of Intel's UPI family of products. It is a general-purpose slave I/O Processor that allows the designer to grow a customized interface solution.

The UPI-452 contains a complete 80C51 with twice the on-chip data and program memory. The sophisticated slave FIFO module acts as a buffer between the UPI-452 internal CPU and the external host CPU. To both the external host and the internal CPU, the FIFO module looks like a bi-directional bottomless buffer that can both read and write data. The FIFO manages the transfer of data independent of the UPI-452 core CPU and generates an interrupt or DMA request to either CPU, host or internal, as a FIFO service request.

The FIFO consists of two channels:the Input FIFO and the Output FIFO. The division of the FIFO module array, 128 bytes, between Input channel and Output channel is programmable by the user. Each FIFO byte has an additional logical ninth bit to distinguish between a data byte and a Data Stream Command byte. Additionally, Immediate Commands allow direct, interrupt driven, bi-directional communication between the UPI-452 internal CPU and external host CPU, bypassing the FIFO.

The on-chip DMA processor allows high speed data transfers from one writeable memory space to another. As many as 64K bytes can be transferred in a single DMA operation. Three distinct memory spaces may be used in DMA operations; Internal Data Memory, External Data Memory, and the Special Function Registers (including the FIFO IN, FIFO OUT, and Serial Channel Special Functions Registers).

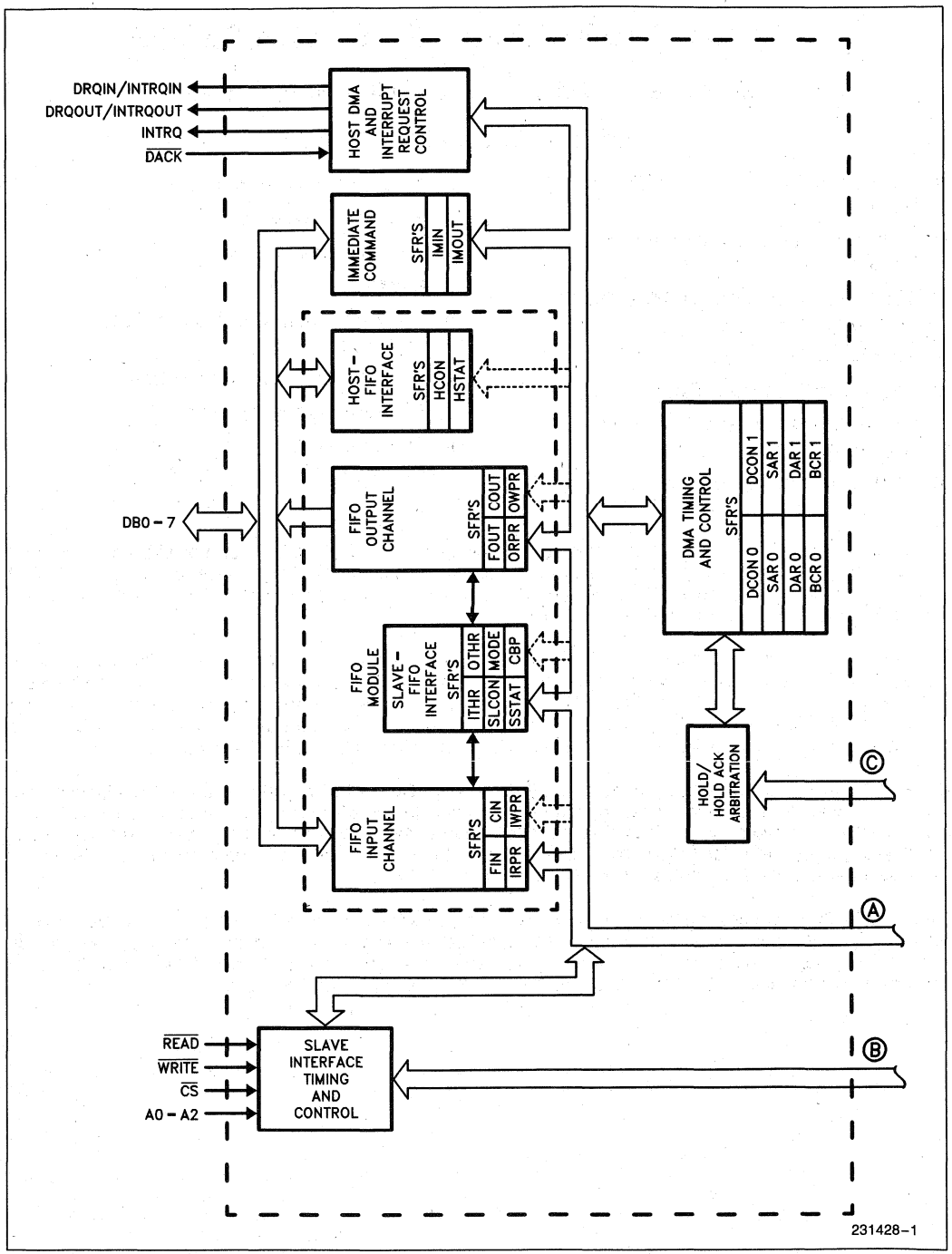


Figure 1. Architectural Block Diagram

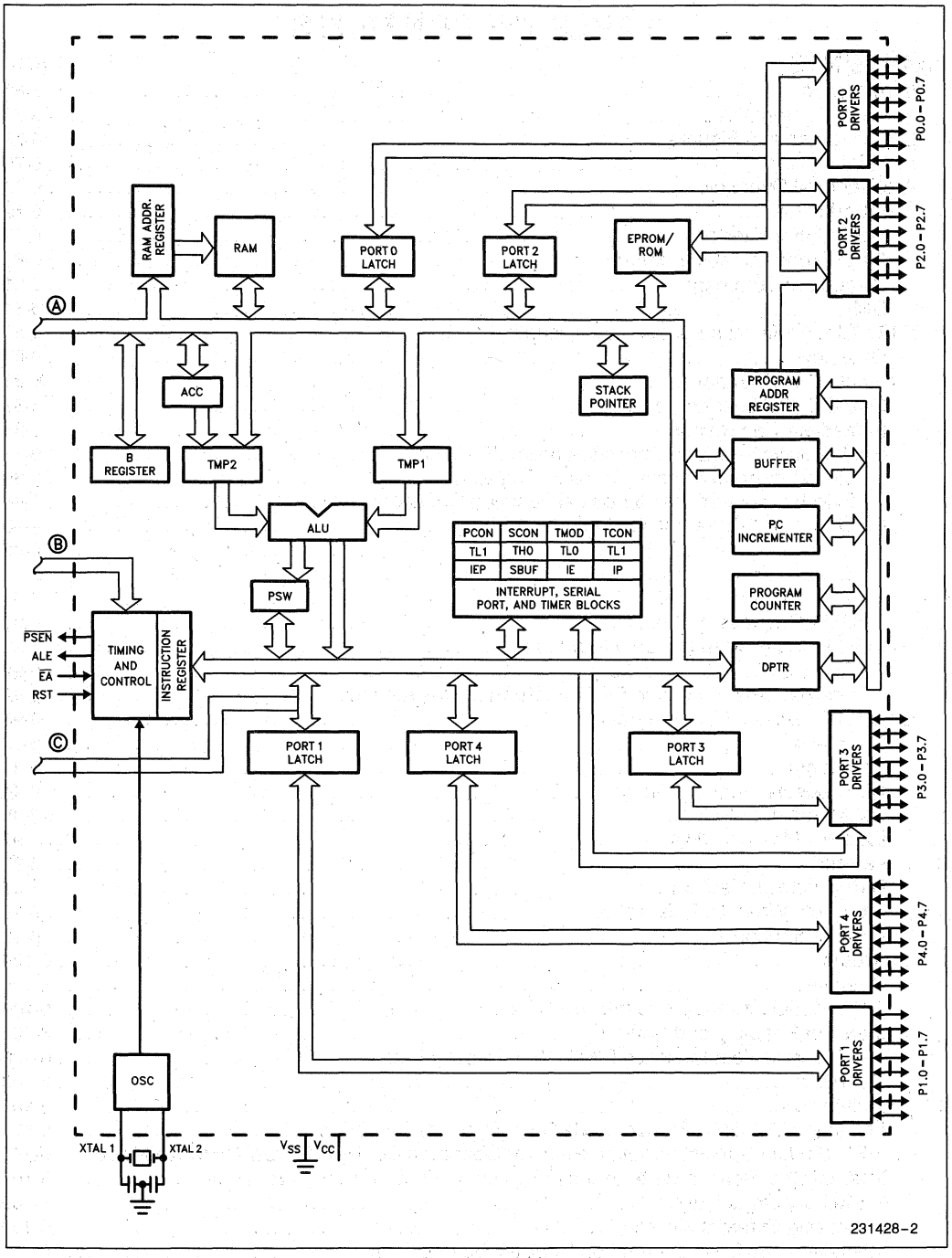


Figure 1. Architectural Block Diagram (Continued)

6

TABLE OF CONTENTS

CONTENTS

PAGE
Introduction ... 6-69
Table of Contents ... 6-72
List of Tables and Figures ... 6-73
Pin Description ... 6-76
Architectural Overview ... 6-79
Introduction ... 6-79
FIFO Buffer Interface ... 6-79
FIFO Programmable Features ... 6-80
Immediate Commands ... 6-81
DMA ... 6-81
FIFO/Slave Interface Functional Description ... 6-81
Overview ... 6-81
Input FIFO Channel ... 6-81
Output FIFO Channel ... 6-83
Immediate Commands ... 6-84
Host & Slave Interface Special Function Registers ... 6-86
Slave Interface Special Function Registers ... 6-86
External Host Interface Special Function Registers ... 6-88
FIFO Module—External Host Interface ... 6-90
Overview ... 6-90
Slave Interface Address Decoding ... 6-90
Interrupts to the Host ... 6-90
DMA Requests to the Host ... 6-92
FIFO Module—Internal CPU Interface ... 6-92
Overview ... 6-92
Internal CPU Access to FIFO via Software Instructions ... 6-92
General Purpose DMA Channels ... 6-93
Overview ... 6-93
Architecture ... 6-93
DMA Special Function Registers ... 6-94
DMA Transfer Modes ... 6-95
External Memory DMA ... 6-97
Latency ... 6-97
DMA Interrupt Vectors ... 6-97
Interrupts When DMA is Active ... 6-98
DMA Arbitration ... 6-98
Interrupts ... 6-100
Overview ... 6-100
FIFO Module Interrupts to Internal CPU ... 6-100
Interrupt Enabling and Priority ... 6-101
FIFO—External Host Interface FIFO DMA Freeze Mode ... 6-103
Overview ... 6-103
Initialization ... 6-103
Invoking FIFO DMA Freeze Mode During Normal Operation ... 6-104
FIFO Module Special Function Register Operation During FIFO DMA Freeze Mode ... 6-105
Internal CPU Read & Write of the FIFO During FIFO DMA Freeze Mode ... 6-109
Memory Organization ... 6-109
Accessing External Memory ... 6-109
Miscellaneous Special Function Register Descriptions ... 6-111

LIST OF TABLES AND FIGURES

Figures:

1. Architectural Block Diagram 6-70
2. UPI-452 68-Pin PGA Pinout Diagram 6-74
3. UPI-452 Conceptual Block Diagram 6-79
4. UPI-452 Functional Block Diagram 6-80
5. Input FIFO Channel Functional Block Diagram 6-82
6. Output FIFO Channel Functional Block Diagram 6-83

7a. Handshake Mechanisms for Handling Immediate Command IN Flowchart 6-85
7b. Handshake Mechanisms for Handling Immediate Command OUT Flowchart 6-85

8. DMA Transfer from: External to External Memory 6-99
9. DMA Transfer from: External to Internal Memory 6-99
10. DMA Transfer from: Internal to External Memory 6-99
11. DMA Transfer Waveform: Internal to Internal Memory 6-100
12. Disabling FIFO to Host Slave Interface Timing Diagram 6-104

Tables:

1. Input FIFO Channel Registers 6-81
2. Output FIFO Channel Registers 6-83
3. UPI-452 Address Decoding 6-91
4. DMA Accessible Special Function Registers 6-94
5. DMA Mode Control - PCON SFR 6-97
6. Interrupt Priority 6-100
7. Interrupt Vector Addresses 6-100
8. Slave Bus Interface Status During FIFO DMA Freeze Mode 6-103
9. FIFO SFR's Characteristics During FIFO DMA Freeze Mode 6-106
10. Threshold SFRs Range of Values and Number of Bytes to be Transferred 6-107

11a. Internal Memory Addressing 6-109
11b. 80C51 Special Function Registers 6-110
11c. UPI-452 Additional Special Function Registers 6-110

12. Program Status Word (PSW) 6-112
13. PCON Special Function Register 6-112

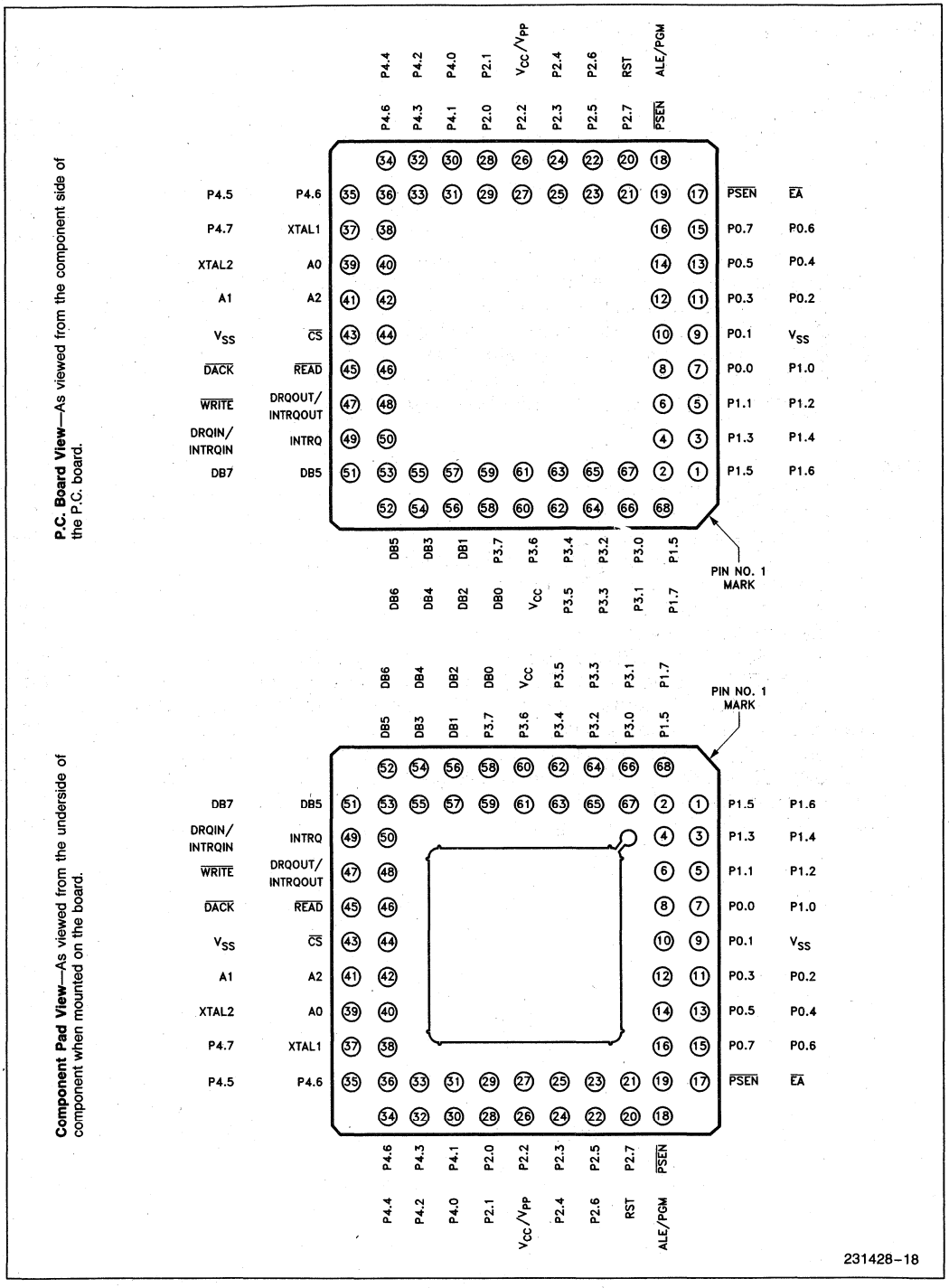
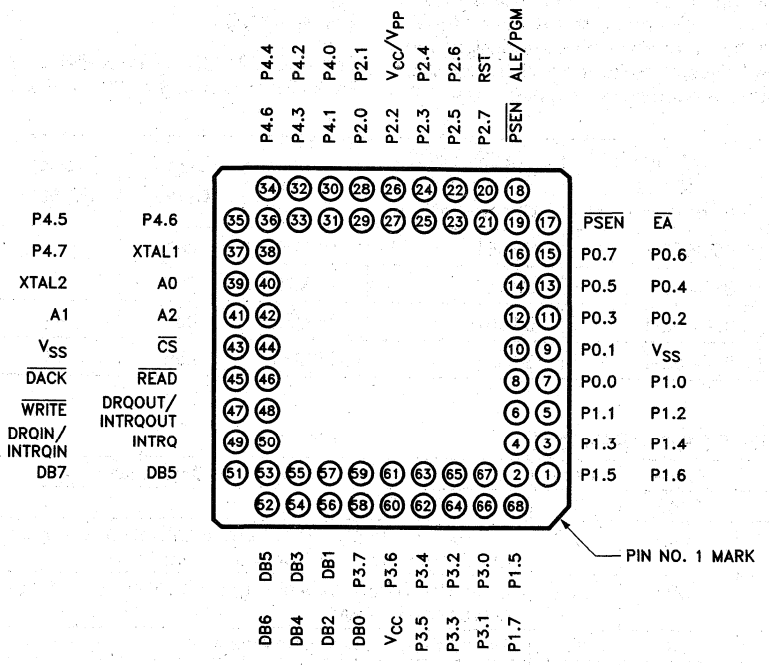
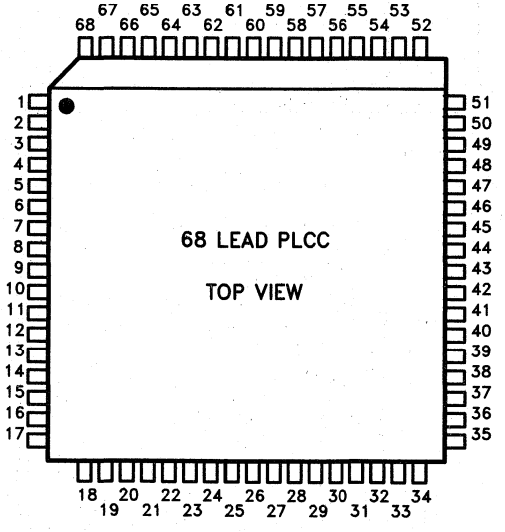


Figure 2. UPI-452 68-Pin PGA Pinout Diagram

P.C. Board View—As Viewed from the Component Side of the P.C. Board (Underside of Socket)





231428-32

Figure 2A. UPI 452 68-Pin PLCC Pinout Diagram

UPI MICROCONTROLLER FAMILY

The UPI-452 joins the current members of the UPI microcontroller family. UPI's are derivatives of the MCS™ family of microcontrollers. Because of their on-chip system bus interface, UPI's are designed to be system bus "slaves", while their microcontroller counterparts are intended as system bus "masters".

These UPI Microcontrollers are fully supported by Intel's development tools (ICE, ASM and PLM).

Packaging

The 80C452/83C452 is available in either a 68-pin PGA (Pin Grid Array) or 68-pin PLCC package.

UPI Family (Slave Configuration)	MCS Family (Master Configuration)	Speed	RAM (Bytes)	ROM (Bytes)
80C452	80C51	12 MHz	256	—
83C452	80C51	12 MHz	256	8K
80C452-1	80C51	14 MHz	256	—
83C452-1	80C51	14 MHz	256	8K

UPI-452 PIN DESCRIPTIONS

Symbol	Pin #	Type	Name and Function
V_{SS}	9/43	I	Circuit Ground.
V_{CC}	60	I	+5V power supply during normal and idle mode operation. It is also the standby power pin for power down mode.
XTAL1	38	I	Input to the oscillator's high gain amplifier. A crystal or external source can be used.
XTAL2	39	O	Output from the high gain amplifier.
Port 0 (AD0–AD7) P0.0 .1 .2 .3 .4 .5 .6 P0.7	 8 10 11 12 13 14 15 16	I/O	Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 can sink eight LS TTL inputs. It is also the multiplexed low-order address and data local expansion bus during accesses to external memory.

UPI-452 PIN DESCRIPTIONS (Continued)

Symbol	Pin #	Type	Name and Function
Port 1 (A0–A7) (HLD, HLDA) P1.0 .1 .2 .3 .4 .5 .6 P1.7	 7 6 5 4 3 2 1 68	I/O	Port 1 is an 8-bit quasi-bi-directional I/O port. Port 1 can sink four LS TTL inputs. The alternate functions can only be activated if the corresponding bit latch in the port SFR contains a 1. Otherwise, the port pin is stuck at 0. Pins P1.5 and P1.6 are multiplexed with $\overline{HLD}$ and $\overline{HLDA}$ respectively whose functions are defined as below: Port Pin — Alternate Function P1.5 — $\overline{HLD}$ —Local bus hold input/output signal P1.6 — $\overline{HLDA}$ —Local bus hold acknowledge input
Port 2 (A8–A15) P2.0 .1 .2 .3 .4 .5 .6 .7	 29 28 27 25 24 23 22 21	I/O	Port 2 is an 8-bit quasi-bi-directional I/O port. It also emits the high-order 8 bits of address when accessing local expansion bus external memory. Port 2 can sink four LS TTL inputs.
Port 3 P3.0 .1 .2 .3 .4 .5 .6 P3.7	 67 66 65 64 63 62 61 59	I/O	Port 3 is an 8-bit quasi-bi-directional I/O port. It is also multiplexed with the interrupt, timer, local serial channel, RD/ and WR/ functions that are used by various options. The alternate functions can only be activated if the corresponding bit latch in the port SFR contains a 1. Otherwise, the port pin is stuck at 0. Port 3 can sink four LS TTL inputs. The alternate functions assigned to the pins of Port 3 are as follows: Port Pin — Alternate Function P3.0 — RxD — Serial input port P3.1 — TxD — Serial output port P3.2 — INT0 — Interrupt 0 Input P3.3 — INT1 — Interrupt 1 Input P3.4 — T0 — Input to counter 0 P3.5 — T1 — Input to counter 1 P3.6 — WR/ — The write control signal latches the data from Port 0 outputs into the External Data Memory on the local bus. P3.7 — RD/ — The read control signal latches the data from Port 0 outputs on the local bus.

UPI-452 PIN DESCRIPTIONS (Continued)

Symbol	Pin #	Type	Name and Function
Port 4 P4.0 .1 .2 .3 .4 .5 .6 .7	 30 32 33 34 35 36 37	I/O	Port 4 is an 8-bit quasi-bi-directional I/O port. Port 4 can sink/source four TTL inputs.
RST	20	I	A high level on this pin for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits Power-on reset using only a capacitor connected to V_{CC}. This pin does not receive the power down voltage as is the case for HMOS MCS-51 family members. This function has been transferred to the V_{CC} pin.
ALE	18	O	Provides Address Latch Enable output used for latching the address into external memory during normal operation. ALE can sink/source eight LS TTL inputs.
$\overline{PSEN}$	19	O	The Program Store Enable output is a control signal that enables the external Program Memory to the bus during normal fetch operation. $\overline{PSEN}$ can sink/source eight LS TTL inputs.
$\overline{EA}$	17	I	When held at TTL high level, the UPI-452 executes instructions from the internal ROM when the PC is less than 8192 (8K, 2000H). When held at a TTL low level, the UPI-452 fetches all instructions from external Program Memory.
DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7	58 57 56 55 54 53 52 51	I/O	Host Bus Interface is an 8-bit bi-directional bus. It is used to transfer data and commands between the UPI-452 and the host processor. This bus can sink/source eight LS TTL inputs.
$\overline{CS}$	44	I	This pin is the Chip Select of the UPI-452.
A0 A1 A2	40 41 42	I	These three address lines are used to interface with the host system. They define the UPI-452 operations. The interface is compatible with the Intel microprocessors and the MULTIBUS.
$\overline{READ}$	46	I	This pin is the read strobe from the host CPU. Activating this pin causes the UPI-452 to place the contents of the Output FIFO (either a command or data) or the Host Status/Control Special Function Register on the Slave Data Bus.
$\overline{WRITE}$	47	I	This pin is the write strobe from the host. Activating this pin will cause the value on the Slave Data Bus to be written into the register specified by A0–A2.
DRQIN/ INTRQIN	49	O	This pin requests an input transfer from the host system whenever the Input Channel requires data.
DRQOUT/ INTRQOUT	48	O	This output pin requests an output transfer whenever the Output Channel requires service. If the external host to UPI-452 DMA is enabled, and a Data Stream Command is at the Output FIFO, DRQOUT is deactivated and INTRQ is activated (see 'GENERAL PURPOSE DMA CHANNELS' section).

UPI-452 PIN DESCRIPTIONS (Continued)

Symbol	Pin #	Type	Name and Function
INTRQ	50	O	This output pin is used to interrupt the host processor when an Immediate Command Out or an error condition is encountered. It is also used to interrupt the host processor when the FIFO requests service if the DMA is disabled and INTRQIN and INTRQOUT are not used.
$\overline{DACK}$	45	I	This pin is the DMA acknowledge for the host bus interface Input and Output Channels. When activated, a write command will cause the data on the Slave Data Bus to be written as data to the Input Channel (to the Input FIFO). A read command will cause the Output Channel to output data (from the Output FIFO) on to the Slave Data Bus. This pin should be driven high (+5V) in systems which do not have a DMA controller (see Address Decoding).
V_{CC}	26	I	+5V power supply during operation.

ARCHITECTURAL OVERVIEW

Introduction

The UPI-452 slave microcontroller incorporates an 80C51 with double the program and data memory, a slave interface which allows it to be connected directly to the host system bus as a peripheral, a FIFO buffer module, a two channel DMA processor, and a fifth I/O port (Figure 3). The UPI-452 retains all of the 80C51 architecture, and is fully compatible with the MCS-51 instruction set.

The Special Function Register (SFR) interface concept introduced in the MCS-51 family of microcontrollers has been expanded in the UPI-452. To the 20 Special Function Registers of the MCS-51, the UPI-452 adds 34 more. These additional Special Function Registers, like those of the MCS-51, provide access to the UPI-452 functional elements including the FIFO, DMA and added interrupt capabilities. Several of the 80C51 core Special Function Registers have also been expanded to support added features of the UPI-452.

This data sheet describes the unique features of the UPI-452. Refer to the 80C51 data sheet for a description of the UPI-452's core CPU functional blocks including;

- Timers/Counters
- I/O Ports
- Interrupt timing and control (other than FIFO and DMA interrupts)
- Serial Channel
- Local Expansion Bus
- Program/Data Memory structure
- Power-Saving Modes of Operation
- CHMOS Features
- Instruction Set

Figure 3 contains a conceptual block diagram of the UPI-452. Figure 4 provides a functional block diagram.

FIFO Buffer Interface

A unique feature of the UPI-452 is the incorporation of a 128 byte FIFO array at the host-slave interface. The FIFO allows asynchronous bi-directional transfers between the host CPU and the internal CPU.

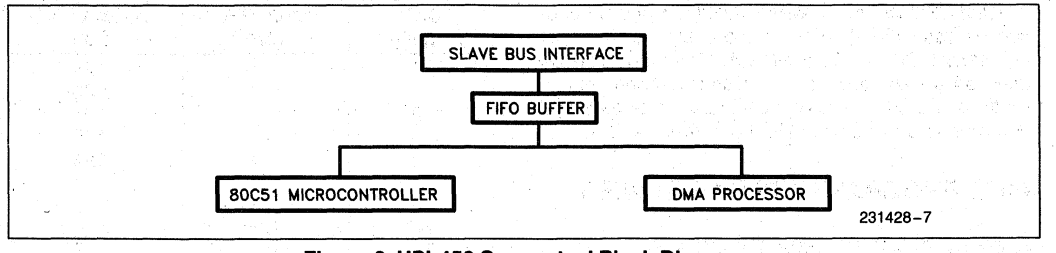


Figure 3. UPI-452 Conceptual Block Diagram

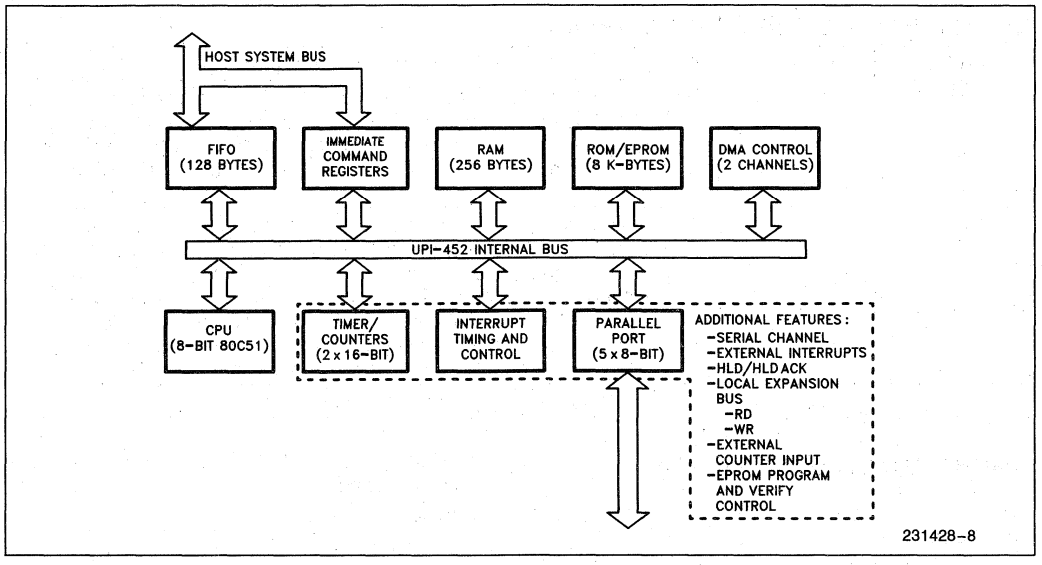


Figure 4. UPI-452 Functional Block Diagram

The division of the 128 bytes between Input and Output channels is user programmable allowing maximum flexibility. If the entire 128 byte FIFO is allocated to the Input channel, a high performance Host can transfer up to 128 bytes at one time, then dedicate its resources to other functions while the internal CPU processes the data in the FIFO. Various handshake signals allow the external Host to operate independently and without frequent monitoring of the UPI-452 internal CPU. The FIFO Buffer insures that the slave processor receives data in the same order that it was sent by the host without the need to keep track of addresses. Three slave bus interface handshake methods are supported by the UPI-452: DMA, Interrupt and Polled.

The FIFO is nine bits wide. The ninth bit acts as a command/data flag. Commands written to the FIFO by either the host or internal CPU are called Data Stream Commands or DSCs. DSCs are written to the input FIFO by the Host via a unique external address. DSCs are written to the output FIFO by the internal CPU via the COMMAND OUT Special Function Register (SFR). When encountered by the host or internal CPU a Data Stream Command can be used as an address vector to user defined service routines. DSCs provide synchronization of data and commands between the Host and internal CPU.

FIFO PROGRAMMABLE FEATURES

Size of Input/Output Channels

The 128 bytes of FIFO space can be allocated between the Input and Output channels via the Channel Boundary Pointer (CBP) SFR. This register contains the number of address locations assigned to the Input channel. The remaining address locations are automatically assigned to the Output FIFO. The CBP SFR can only be programmed by the internal CPU during FIFO DMA Freeze Mode (See FIFO-External Host Interface FIFO DMA Freeze Mode description). The CBP is initialized to 40H (64 bytes) upon reset.

The number in the Channel Boundary Pointer SFR is actually the first address location of the Output FIFO. Writing to the CBP SFR reassigns the Input and Output FIFO address space. Whenever the CBP is written, the Input FIFO pointers are reset to zero and the Output FIFO pointers are set to the value in the CBP SFR.

All of the FIFO space may be assigned to one channel. In such a situation the other channel's data path consists of a single SFR (FIFO IN/COMMAND IN or FIFO OUT/COMMAND OUT SFR) location.

CBP Register	Input FIFO Size	Output FIFO Size
0	1	128
1	1	128
2	2	126
3	3	125
4	4	124
•	•	•
7B	123	5
7C	124	4
7D	125	3
7E	128	1
7F	128	1

FIFO Read/Write Pointers

These normally operate in auto-increment (and auto-rollover) mode, but can be reassigned by the internal CPU during FIFO DMA Freeze Mode (See FIFO-External Host Interface FIFO DMA Freeze Mode description).

Threshold Register

The Input FIFO Threshold SFR contains the number of empty bytes that must be available in the Input FIFO to generate a Host interrupt. The Output FIFO Threshold SFR contains the number of bytes, data and/or DSC(s), that must be in the FIFO before an interrupt is generated. The Threshold feature prevents the Host from being interrupted each time the FIFO needs to load or unload one byte of data. The thresholds, therefore, allow the FIFO's operation to be adjusted to the speed of the Host, optimizing the overall interface performance.

Immediate Commands

The UPI-452 provides, in addition to data and DSCs, a third direct means of communication between the external Host and internal CPU called Immediate Commands. As the name implies, an Immediate Command is available to the receiving CPU immediately, via an interrupt, without being entered into the FIFO as are Data Stream Commands. Like Data Stream Commands, Immediate Commands are written either via a unique external address by the host CPU, or via dedicated SFR by the internal CPU.

The DSC and/or Immediate Command interface may be defined as either Interrupt or Polled under user program control via the Interrupt Enable (IE), Slave Control Register (SLCON), and Interrupt Enable Priority (IEP) Special Function Registers, for the internal CPU and via the Host Control SFR for the external Host CPU.

DMA

The UPI-452 contains a two channel internal DMA controller which allows transfer of data between any of the three writeable memory spaces: Internal Data Memory, External Load Expansion Bus Data Memory and the Special Function Register array. The Special Function Register array appears as a set of unique dedicated memory addresses which may be used as either the source or destination address of a DMA transfer. Each DMA channel is independently programmable via dedicated Special Function Registers for mode, source and destination addresses, and byte count to be transferred. Each DMA channel has four programmable modes:

— Alternate Cycle Mode

— Burst Mode

— FIFO or Serial Channel Demand Mode

— External Demand Mode

A complete description of each mode and DMA operation may be found in the section titled "General Purpose DMA Channels".

FIFO/SLAVE INTERFACE FUNCTIONAL DESCRIPTION

Overview

The FIFO is a 128 Byte RAM array with recirculating pointers to manage the read and write accesses. The FIFO consists of an Input and an Output channel. Access cycles to the FIFO by the internal CPU and external Host are interleaved and appear to be occurring concurrently to both the internal CPU and external Host. Interleaving access cycles ensures efficient use of this shared resource. The internal CPU accesses the FIFO in the same way it would access any of the Special Function Registers e.g., direct and register indirect addressing as well as arithmetic and logical instructions.

Input FIFO Channel

The Input FIFO Channel provides for data transfer from the external Host to the internal CPU (Figure 5). The registers associated with the Input Channel during normal operation are listed in Table 1*.

Table 1. Input FIFO Channel Registers*

	Register Name	Description
1)	Input Buffer Latch	Host CPU Write only
2)	FIFO IN SFR	Internal CPU Read only
3)	COMMAND IN SFR	Internal CPU Read only
4)	Input FIFO Read Pointer SFR	Internal CPU Read only
5)	Input FIFO Write Pointer SFR	Internal CPU Read only
6)	Input FIFO Threshold SFR	Internal CPU Read only

*See "FIFO-EXTERNAL HOST INTERFACE FIFO DMA FREEZE MODE" section for FIFO DMA Freeze Mode SFR characteristics description.

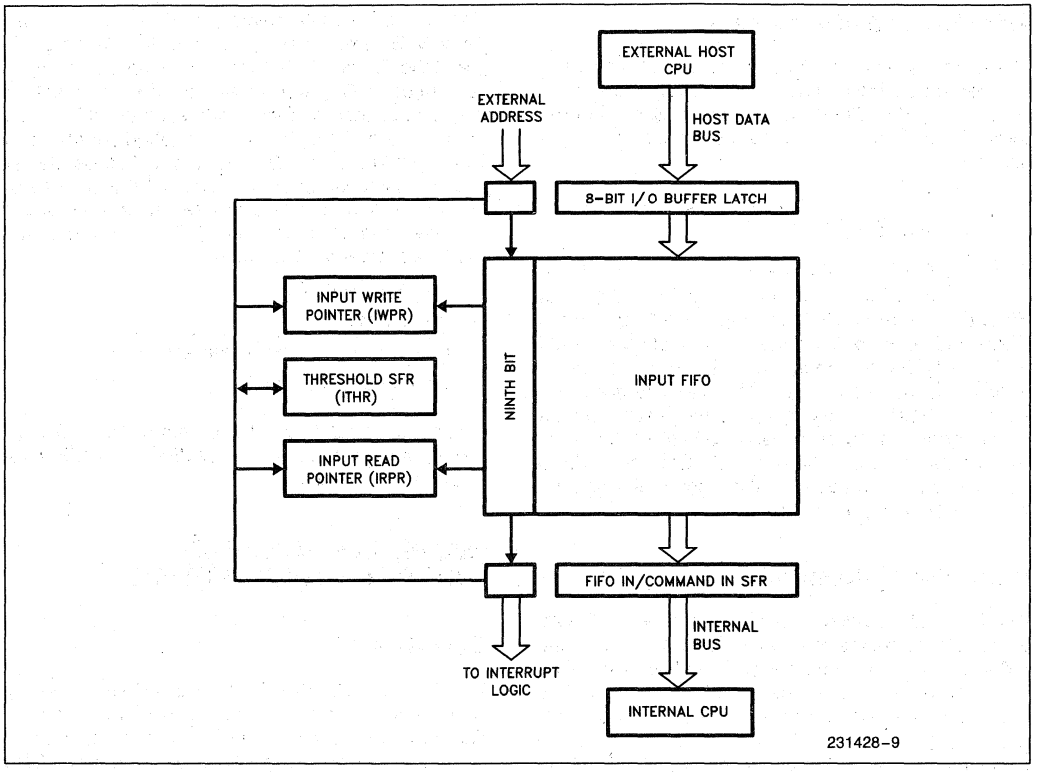


Figure 5. Input FIFO Channel Functional Block Diagram

The host CPU writes data and Data Stream Commands into the Input Buffer Latch on the rising edge of the external WR signal. External addressing determines whether the byte is a data byte or Data Stream Command and the FIFO logic sets the ninth bit of the FIFO accordingly as the byte is moved from the Input Buffer Latch into the FIFO. A "1" in the ninth bit indicates that the incoming byte is a Data Stream Command. The internal CPU reads data bytes via the FIFO IN SFR, and Data Stream Commands via the COMMAND IN SFR.

A Data Stream Command will generate an interrupt to the internal CPU prior to being read and after completion of the previous operation. The DSC can then be read via the COMMAND IN SFR. Data can only be read via the FIFO IN SFR and Data Stream Commands via the COMMAND IN SFR. Attempting to read Data Stream Commands as data by addressing the FIFO IN SFR will result in "0FFH" being read, and the Input FIFO Read Pointer will remain intact. (This prevents accidental misreading of Data Stream Commands.) Attempting to read data as Data Stream Commands will have the same consequence.

The Input FIFO Channel addressing is controlled by the Input FIFO Read and Write Pointer SFRs. These SFRs are read only registers during normal operation. However, during FIFO DMA Freeze Mode (See FIFO-External Host Interface FIFO DMA Freeze Mode description), the internal CPU has write access to them. Any write to these registers in normal mode will have no effect. The Input Write Pointer SFR contains the address location to which data/commands are written from the Input Buffer Latch. The write pointer is automatically incremented after each write and is reset to zero if equal to the CBP, as the Input FIFO operates as a circular buffer.

If a write is performed on an empty FIFO, the first byte is also written into the FIFO IN or COMMAND IN SFR. If the Host continues writing while the Input FIFO is full, an external interrupt, if enabled, is sent to the host to signal the overrun condition. The writes are ignored by the FIFO control logic. Similarly, an internal CPU read of an empty FIFO will cause an underrun error interrupt to be generated to the internal CPU and a value of "0FFH" will be read by the internal CPU.

The Read Pointer SFR holds the address of the next byte to be read from the Input FIFO. An Input FIFO read operation post-increments the Input Read Pointer SFR and loads a new data byte into the FIFO IN SFR or a Data Stream Command into the COMMAND IN SFR at the end of the read cycle.

An Input FIFO Request for Service (via DMA, Interrupt or a flag) is generated to the Host whenever more data can be written into the Input FIFO. For efficient utilization of the Host, a "threshold" value can be programmed into the Input FIFO Threshold SFR. The range of values of the Input FIFO Threshold SFR can be from 0 to (CBP-3). The Request for Service Interrupt is generated only after the Input FIFO has room to accommodate a threshold number of bytes or more. The threshold is equal to the total number of bytes assigned to the Input FIFO (CBP) minus the number of bytes programmed in the Input FIFO Threshold SFR. With this feature the Host is assured that it can write at least a threshold number of bytes to the Input FIFO channel without worrying about an overrun condition. Once the Request for Service is generated it remains active until the Input FIFO becomes full.

Output FIFO Channel

The Output FIFO Channel provides data transfer from the UPI-452 internal CPU to the external Host (Figure 6).

The registers associated with the Output Channel during normal operation are listed in Table 2*.

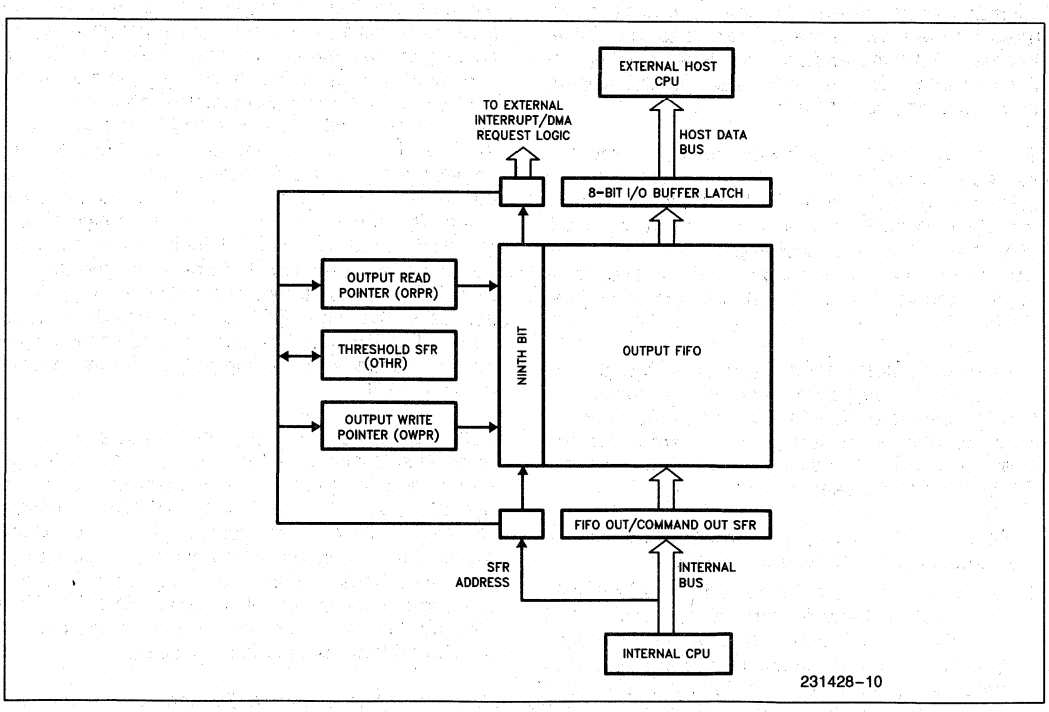


Figure 6. Output FIFO Channel Functional Block Diagram

Table 2. Output FIFO Channel Registers

	Register Name	Description
1)	Output Buffer Latch	Host CPU Read only
2)	FIFO OUT SFR	Internal CPU Read and Write
3)	COMMAND OUT SFR	Internal CPU Read and Write
4)	Output FIFO Read Pointer SFR	Internal CPU Read only
5)	Output FIFO Write Pointer SFR	Internal CPU Read only
6)	Output FIFO Threshold SFR	Internal CPU Read only

*See "FIFO-EXTERNAL HOST INTERFACE FIFO DMA FREEZE MODE" section for FIFO DMA Freeze Mode register characteristics description.

The UPI-452 internal CPU transfers data to the Output FIFO via the FIFO OUT SFR and commands via the COMMAND OUT SFR. If the byte is written to the COMMAND OUT SFR, the ninth bit is automatically set (=1) to indicate a Data Stream Command. If the byte is written to the FIFO OUT SFR the ninth bit is cleared (=0). Thus the FIFO OUT and COMMAND OUT SFRs are the same but the address determines whether the byte entered in the FIFO is a DSC or data byte.

The Output FIFO preloads a byte into the Output Buffer Latch. When the Host issues a RD/ signal, the data is immediately read from the Output Buffer Latch. The next data byte is then loaded into the Output Buffer Latch, a flag is set and an interrupt, if enabled, is generated if the byte is a DSC (ninth bit is set). The operation is carefully timed such that an interrupt can be generated in time for it to be recognized by the Host before its next read instruction. Internal CPU write and external Host read operations are interleaved at the FIFO so that they appear to be occurring concurrently.

The Output FIFO read and write pointer operation is the same as for the Input Channel. Writing to the FIFO OUT or COMMAND OUT SFRs will increment the Output Write Pointer SFR but reading from it will leave the write pointer unchanged. A rollover of the Output FIFO Write Pointer causes the pointer to be reset to the value in the Channel Boundary Pointer (CBP) SFR.

If the external host attempts to read a Data Stream Command as a data byte it will result in invalid data (0FFH) being read. The DSC is not lost because the invalid read does not increment the pointer. Similarly attempting to read a data byte as a Data Stream Command has the same result.

A Request for Service is generated to the external Host under the following two conditions:

1.) Whenever the internal CPU has written a threshold number of bytes or more into the Output FIFO (threshold = (OTHR) + 1). The threshold number should be chosen such that the bus latency time for the external Host does not result in a FIFO overrun error condition on the internal CPU side. The threshold limit should be large enough to make a bus request by the UPI-452 to the external host CPU worthwhile. Once a request for service is generated, the request remains active until the Output FIFO becomes empty. The range of values of the FIFO Output Threshold (OTHR) SFR is from 2 to {(80H-CBP)-1}. The threshold number can be programmed via the OTHR SFR.

2.) The second type of Request for Service is called "Flush Mode" and occurs when the internal CPU writes a Data Stream Command into the Output FIFO. Its purpose is to ensure that a data block entered into the Output FIFO, which is less than the programmed threshold, will generate a Request for Service interrupt, if enabled, and be read, or "Flushed" from the Output FIFO, by the external host CPU regardless of the status of the OTHR SFR.

Immediate Commands

Immediate Commands provide direct communication between the external Host and UPI-452. Unlike Data Stream Commands which are entered into the FIFO, the Immediate Command is available to the receiving CPU directly, bypassing the FIFO. The Immediate Command can serve as a program vector pointing into a jump table in the recipients software. Immediate Command Interrupts are generated, if enabled, and a bit in the appropriate Status Register is set when an Immediate Command is input or output. A similar bit is provided to acknowledge when an Immediate Command has been read and whether the register is available to receive another command. The bits are reset when the Immediate Commands are read. Two Special Function Registers are dedicated to the Immediate Command interface. External addressing determines whether the Host is accessing the Input FIFO or the Immediate Command IN (IMIN) SFR. The internal CPU writes Immediate Commands to the Immediate Command OUT (IMOUT) SFR.

Both processors have the ability to enable or disable Immediate Command Interrupts. By disabling the interrupt, the recipient of the Immediate Command can poll the status SFR and read the Immediate Command at its convenience. Immediate Commands should only be written when the appropriate Immediate Command SFR is empty (as indicated in the appropriate status SFR:HSTAT/SSTAT). Similarly, the Immediate Command SFR should only be read when there is data in the Register.

The flowcharts in Figure 7a and 7b illustrate the proper handshake mechanisms between the external Host and internal CPU when handling Immediate Commands.

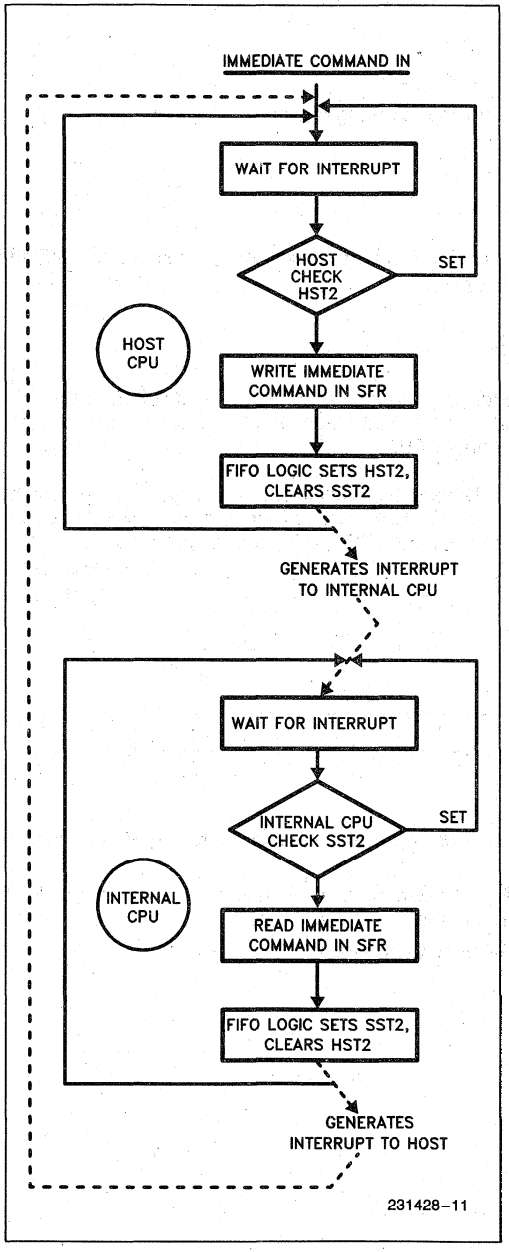


Figure 7a. Handshake Mechanisms for Handling Immediate Command IN Flowchart

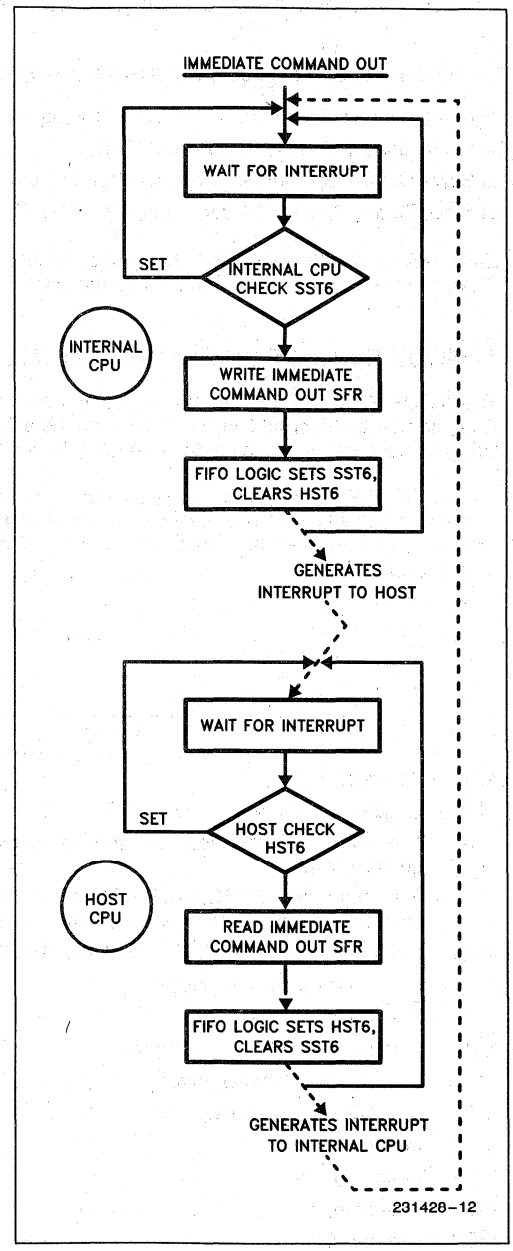


Figure 7b. Handshake Mechanisms for Handling Immediate Command OUT Flowchart

6

HOST & SLAVE INTERFACE SPECIAL FUNCTION REGISTERS

Slave Interface Special Function Registers

The Internal CPU interfaces with the FIFO slave module via the following registers:

1) Mode Special Function Register (MODE)

2) Slave Control Special Function Register (SLCON)

3) Slave Status Special Function Register (SSTAT)

Each register resides in the SFR Array and is accessible via all direct addressing modes except bit. Only the Slave Control Register (SLCON) is bit addressable.

1) MODE Special Function Register (MODE)

The MODE SFR provides the primary control of the external host-FIFO interface. It is included in the SFR Array so that the internal CPU can configure the external host-FIFO interface should the user decide that the UPI-452 slave initialize itself independent of the external host CPU.

The MODE SFR can be directly modified by the internal CPU through direct address instructions. It can also be indirectly modified by the external host CPU by setting up a MODE SFR service routine in the UPI-452 program memory and having the host issue a Command, either Immediate or DSC, to vector to that routine.

Symbolic Address									Physical Address
MODE	—	MD6	MD5	MD4	—	—	—	—	0F9H
	(MSB)							(LSB)	
	Status On Reset:								
	1*	0	0	0	1*	1*	1*	1*	

MD7 (reserved)**

MD6 Request for Service to external CPU via;

1 = DMA (DRQIN/DRQOUT) request to external host when the Input or Output FIFO channel requests service

0 = Interrupt (INTRQIN/INTRQOUT or INTRQ) to external host when the Input or Output FIFO channel requests service or a DSC is encountered in the I/O Buffer Latch

MD5 Configure DRQIN/INTRQIN and DRQOUT/INTRQOUT to be either;

1 = Enable (Actively driven)

0 = Disable (Tri-state)

MD4 Configure INTRQ to be either;

1 = Enable (Actively driven)

0 = Disable (Tri-state)

MD3 (reserved) **

MD2 (reserved) **

MD1 (reserved) **

MD0 (reserved) **

2) Slave Control SFR (SLCON)

The Slave Control SFR is used to configure the FIFO-internal CPU interface. All interrupts are to the internal CPU.

Symbolic Address									Physical Address
SLCON	IFI	OFI	ICII	ICOI	FRZ	—	IFRS	OFRS	0E8H
	(MSB)							(LSB)	
Status On Reset:									
	0	0	0	0	0	1*	0	0	

IFI Enable Input FIFO Interrupt (due to Underrun Error Condition, Data Stream Command or Request Service)

1 = Enable

0 = Disable

OFI Enable Output FIFO Interrupt (due to Overrun Error Condition or Request Service)

1 = Enable

0 = Disable

Note: If the DMA is configured to service a FIFO demand, then the Request for Service Interrupt is not generated.

ICII Generate Interrupt when a command is written to the Immediate Command in Register

1 = Enable

0 = Disable

ICOI Generate Interrupt when Immediate Command Out Register is Available

1 = Enable

0 = Disable

FRZ Enable FIFO DMA Freeze Mode

1 = Normal operation

0 = FIFO DMA Freeze Mode

SC2 (reserved) **

IFRS Input FIFO Channel Request for Service

1 = Request when Input FIFO not empty

0 = Request when Input FIFO full

OFRS Output FIFO Channel Request for Service

1 = Request when Output FIFO not full

0 = Channel Request when Output FIFO empty

NOTES:

*A '1' will be read from all SFR reserved locations except HCON SFR, HC0 and HC2.

**'reserved'—these locations are reserved for future use by Intel Corporation.

3) Slave Status SFR (SSTAT)

The bits in the Slave Status SFR reflect the status of the FIFO-internal CPU interface. It can be read during an internal interrupt service routine to determine the nature of the interrupt or read during a polling sequence to determine a course of action.

Symbolic Address									Physical Address
SSTAT	SST7	SST6	SST5	SST4	SST3	SST2	SST1	SST0	0E9H
	← Output FIFO Status →				← Input FIFO Status →				
Status On Reset:									
	1	0	0	0	1	1	1	1	
	(MSB)							(LSB)	

SST7 Output FIFO Overrun Error Condition
- 1 = No Error
- 0 = Error (latched until Slave Status SFR is read)

SST6 Immediate Command Out Register Status
- 1 = Full (i.e. Host CPU has not read previous Immediate Command Out sent by internal CPU)
- 0 = Available

SST5 FIFO DMA Freeze Mode Status
- 1 = Normal Operation
- 0 = FIFO DMA Freeze Mode in Progress

SST4 Output FIFO Request for Service Flag
- 1 = Output FIFO does not request service
- 0 = Output FIFO requests service

SST3 Input FIFO Underrun Error Condition Flag
- 1 = No Underrun Error
- 0 = Underrun Error (latched until Slave Status SFR is read)

SST2 Immediate Command In SFR Status
- 1 = Empty
- 0 = Immediate Command received from host CPU

SST1 Data Stream Command/Data at Input FIFO Flag
- 1 = Data (not DSC)
- 0 = DSC (at COMMAND IN SFR)

SST0 Input FIFO Request For Service Flag
- 1 = Input FIFO Does Not Request Service
- 0 = Input FIFO Request for Service

EXTERNAL HOST INTERFACE SPECIAL FUNCTION REGISTERS

The external host CPU has direct access to the following SFRs:

1) Host Control Special Function Register

2) Host Status Special Function Register

It can also access other SFRs by commanding the internal CPU to change them accordingly via Data Stream Commands or Immediate Commands. The protocol for implementing this is entirely determined by the user.

1) Host Control SFR (HCON)

By writing to the Host Control SFR, the host can enable or disable FIFO interrupts and DMA requests and can reset the UPI-452.

Symbolic Address									Physical Address
HCON	HC7	HC6	HC5	HC4	HC3	—	HC1	—	0E7H
	(MSB)							(LSB)	
	Status On Reset:								
	0	0	0	0	0	0*	0	0*	

HC7 Enable Output FIFO Interrupt due to Underrun Error Condition, Data Stream Command or Service Request
- 1 = Enable
- 0 = Disable

HC6 Enable Input FIFO Interrupt due to Overrun Error Condition, or Service Request
- 1 = Enable
- 0 = Disable

HC5 Enable the generation of the Interrupt due to Immediate Command Out being present
- 1 = Enable
- 0 = Disable

HC4 Enable the Interrupt due to the Immediate Command In Register being Available for a new Immediate Command byte
- 1 = Enable
- 0 = Disable

HC3 Reset UPI-452
- 1 = Software RESET
- 0 = Normal Operation

HC2 (reserved) **

HC1 Select between INTRQ and INTRQIN/INTRQOUT as Request for Service interrupt signal when DMA is disabled
- 1 = INTRQ
- 0 = INTRQIN or INTRQOUT

HC0 (reserved) **

NOTES:
*A '1' will be read from all SFR reserved locations except HCON SFR, HC0 and HC2.
**'reserved'—these locations are reserved for future use by Intel Corporation.

2) Host Status SFR (HSTAT)

The Host Status SFR provides information on the FIFO-Host Interface and can be used to determine the source of an external interrupt during polling. Like the Slave Status SFR, the Host Status SFR reflects the current status of the FIFO-external host interface.

Symbolic Address									Physical Address
HSTAT	HST7	HST6	HST5	HST4	HST3	HST2	HST1	HST0	0E6H
	← Output FIFO Status →				← Input FIFO Status →				
Status On Reset:									
	1	1	1	1	1	1/0*	1	1	
	(MSB)							(LSB)	

HST7 Output FIFO Underrun Error Condition
1 = No Underrun Error
0 = Underrun Error (latched until Host Status Register is read)

HST6 Immediate Command Out SFR Status
1 = Empty
0 = Immediate Command Present

HST5 Data Stream Command/Data at Output FIFO Status
1 = Data (not DSC)
0 = DSC (present at Output FIFO COMMAND OUT SFR)
(Note: Only if HST4 = 0, if HST4 = 1 then undetermined)

HST4 Output FIFO Request for Service Status
1 = No Request for Service
0 = Output FIFO Request for Service due to:

a. Output FIFO containing the threshold number of bytes or more

b. Internal CPU sending a block of data terminated by a DSC (DSC Flush Mode)

HST3 Input FIFO Overrun Error Condition
1 = No Overrun Error
0 = Overrun Error (latched until Host Status Register is read)

HST2 Immediate Command In SFR Status
1 = Full (i.e. Internal CPU has not read previous Immediate Command sent by Host)
0 = Empty

* Reset value;

'1' — if read by the external Host

'0' — if read by internal CPU (reads shadow latch - see FIFO DMA Freeze Mode description)

HST1 FIFO DMA Freeze Mode Status
1 = Freeze Mode in progress.
(In Freeze Mode, the bits of the Host Status SFR are forced to a '1' initially to prevent the external Host from attempting to access the FIFO. The definition of the Host Status SFR bits during FIFO DMA Freeze Mode can be found in FIFO DMA Freeze Mode description)
0 = Normal Operation

HST0 Input FIFO Request Service Status
1 = Input FIFO does not request service
0 = Input FIFO request service due to the Input FIFO containing enough space for the host to write the threshold number of bytes or more

FIFO MODULE - EXTERNAL HOST INTERFACE

Overview

The FIFO-external Host interface supports high speed asynchronous bi-directional 8-bit data transfers. The host interface is fully compatible with Intel microprocessor local busses and with MULTIBUS. The FIFO has two specialized DMA request pins for Input and Output FIFO channel DMA requests. These are multiplexed to provide a dedicated Request for Service interrupt (DRQIN/INTRQIN, DRQOUT/INTRQOUT).

The external Host can program, under user defined protocol, thresholds into the FIFO Input and Output Threshold SFRs which determine when the FIFO Request for Service interrupt is generated to the Host CPU. The FIFO module external Host interface is configured by the internal CPU via the MODE SFR. "The external Host can enable and disable Host interface interrupts via the Host Control SFR." Data Stream Commands in the Input FIFO channel allow the Host to influence the processing of data blocks and are sent with the data flow to maintain synchronization. Data Stream Commands in the Output FIFO Channel allow the internal CPU to perform the same function, and also to set the Output FIFO Request Service status logic to the host CPU regardless of the programmed value in the Threshold SFR.

Slave Interface Address Decoding

The UPI-452 determines the desired Host function through address decoding. The lower three bits of the address as well as the READ, WRITE, Chip Select ($\overline{CS}$) and DMA Acknowledge ($\overline{DACK}$) are used for decoding. Table 3 shows the pin states and the Read or Write operations associated with each configuration.

Interrupts to the Host

The UPI-452 interrupts the external Host via the INTRQ pin. In addition, the DRQIN and DRQOUT pins can be multiplexed as interrupt request lines, INTRQIN and INTRQOUT respectively, when DMA is disabled. This provides two special FIFO "Request for Service" interrupts.

There are eight FIFO-related interrupt sources; two from The Input FIFO; three from The Output FIFO; one from the Immediate Command Out SFR; one from the Immediate Command IN SFR; and one due to FIFO DMA Freeze Mode.

INPUT FIFO: The Input FIFO interrupt is generated whenever:

a. The Input FIFO contains space for a threshold number of bytes.

Table 3. UPI-452 Address Decoding

$\overline{DACK}$	$\overline{CS}$	A2	A1	A0	Read	Write
1	1	X	X	X	No Operation	No Operation
1	0	0	0	0	Data or DMA from Output FIFO Channel	Data or DMA to Input FIFO Channel
1	0	0	0	1	Data Stream Command from Output FIFO Channel	Data Stream Command to Input FIFO Channel
1	0	0	1	0	Host Status SFR Read	Reserved
1	0	0	1	1	Host Control SFR Read	Host Control SFR Write
1	0	1	0	0	Immediate Command SFR Read	Immediate Command to SFR Write
1	0	1	1	X	Reserved	Reserved
0	X	X	X	X	DMA Data from Output FIFO Channel	DMA Data to Input FIFO Channel
1	0	1	0	1	Reserved	Reserved

NOTES:
1. Attempting to read a DSC as a data byte will result in invalid data being read. The read pointers are not incremented so that the DSC is not lost. Attempting to read a data byte as a DSC has the same result.
2. If $\overline{DACK}$ is active the UPI-452 will attempt a DMA operation when $\overline{RD}$ or $\overline{WR}$ becomes active regardless of the DMA enable bit (MD6) in the MODE SFR. Care should be taken when using $\overline{DACK}$. For proper operation, $\overline{DACK}$ must be driven high (+5V) when not using DMA.

b. When an Input FIFO overrun error condition exists. The appropriate bits in the Host Status SFR are set and the interrupt is generated only if enabled.

OUTPUT FIFO: The Output FIFO Request for Service Interrupt operates in a similar manner as the Input FIFO interrupt:

a. When the FIFO contains the threshold number of bytes or more.

b. Output FIFO error condition interrupts are generated when the Output FIFO is underrun.

c. Data Stream Command present in the Output Buffer Latch.

A Data Stream Command interrupt is used to halt normal processing, using the command as a vector to a service routine. When DMA is disabled, the user may program (through HC1) INTRQ to include FIFO Request for Service Interrupts or use INTRQIN and INTRQOUT as Request for Service Interrupts.

IMMEDIATE COMMAND INTERRUPTS:

a. An Immediate Command Out Interrupt is generated, if enabled, to the Host and the corresponding Host Status SFR bit (HSTAT HST6) is cleared, when the internal CPU writes to the Immediate Command OUT (IMOUT) SFR. When the Host reads the Immediate Command OUT (IMOUT) SFR the corresponding bit in the Host Status (HSTAT) SFR is set. This causes the Slave Status Immediate Command OUT Status bit (SSTAT SST6) to be cleared indicating that the Immediate Command OUT (IMOUT) SFR is empty. If enabled, a FIFO-Slave Interface will also be generated to the internal CPU. (See Figure 7b, Immediate Command OUT Flowchart.)

b. An Immediate Command IN interrupt is generated, if enabled, to the Host when the internal CPU has read a byte from the Immediate Command IN (IMIN) SFR. The read operation clears the Host Status SFR Immediate Command IN Status bit (HSTAT HST2) indicating that the Immediate Command IN SFR is empty. The corresponding Slave Status (SSTAT) SFR bit is also set to indicate an empty status. Setting the Slave Status SFR bit generates a FIFO-Slave Interface interrupt, if enabled, to the internal CPU. (See Figure 7a, Immediate Command IN Flowchart.)

NOTE:
Immediate Command IN and OUT interrupts are actually specific Request For Service interrupts to the Host.

FIFO DMA FREEZE MODE: When the internal CPU invokes FIFO DMA Freeze Mode, for example at reset or to reconfigure the FIFO interface, INTRQ is activated. The INTRQ can only be deactivated by the external Host reading the Host Status SFR (HST1 remains active until FIFO DMA Freeze Mode is disabled by the internal CPU).

Once an interrupt is generated, INTRQ will remain high until no interrupt generating condition exists. For a FIFO underrun/overrun error interrupt, the interrupt condition is deactivated by the external Host reading the Host Status SFR. An interrupt is serviced by reading the Host Status SFR to determine the source of the interrupt and vectoring the appropriate service routine.

DMA Requests to the Host

The UPI-452 generates two DMA requests, DRQIN and DRQOUT, to facilitate data transfer between the Host and the Input and Output FIFO channels. A DMA acknowledge, $\overline{DACK}$, is used as a chip select and initiates a data transfer. The external $\overline{READ}$ and $\overline{WRITE}$ signals select the Input and Output FIFO respectively. The $\overline{CS}$ and address lines can also be used as a DMA acknowledge for processors with onboard DMA controllers which do not generate a $\overline{DACK}$ signal.

The internal CPU can configure the UPI-452 to request service from the external host via DMA or interrupts by programming Mode SFR MD6 bit. In addition the external Host enables DMA requests through bits 6 and 7 of the Host Control SFR. When a DMA request is invoked the number of bytes transferred to the Input FIFO is the total number of bytes in the Input FIFO (as determined by the CBP SFR) minus the value programmed in the Input FIFO Threshold SFR. The DMA request line is activated only when the Input FIFO has a threshold number of bytes that can be transferred.

The Output FIFO DMA request is activated when a DSC is written by the internal CPU at the end of a less than threshold size block of data (Flush Mode) or when the Output FIFO threshold is reached. The request remains active until the Input FIFO becomes full or the Output FIFO becomes empty. If a DSC is encountered during an Output FIFO DMA transfer, the DMA request is dropped until the DSC is read. The DMA request will be reactivated after the DSC is read and remains active until the Output FIFO becomes empty or another DSC is encountered.

FIFO MODULE - INTERNAL CPU INTERFACE

Overview

The Input and Output FIFOs are accessed by the internal CPU through direct addressing of the FIFO IN/COMMAND IN and FIFO OUT/COMMAND OUT Special Function Registers. All of the 80C51 instructions involving direct addressing may be used to access the FIFO's SFRs. The FIFO IN, COMMAND IN and Immediate Command In SFRs are actually read only registers, and their Output counterparts are write only. Internal DMA transfers data between Internal memory, External Memory and the Special Function Registers. The Special Function Registers appear as another group of dedicated memory addresses and are programmed as the source or destination via the DMA0/DMA1 Source Address or Destination Address Special Function Registers. The FIFO module manages the transfer of data between the external host and FIFO SFRs.

Internal CPU Access to FIFO Via Software Instructions

The internal CPU has access to the Input and Output FIFOs via the FIFO IN/COMMAND IN and FIFO OUT/COMMAND OUT SFRs which reside in the Special Function Register Array. At the end of every instruction that involves a read of the FIFO IN/COMMAND IN SFR, the SFR is written over by a new byte from the Input FIFO channel when available. At the end of every instruction that involves a write to the FIFO OUT/COMMAND OUT SFR, the new byte is written into the Output FIFO channel and the write pointer is incremented after the write operation (post incremented).

The internal CPU reads the Input FIFO by using the FIFO IN/COMMAND IN SFR as the source register in an instruction. Those instructions which read the Input FIFO are listed below:

```
ADD  A,FIFO IN/COMMAND IN
ADDC A,FIFO IN/COMMAND IN
PUSH FIFO IN/COMMAND IN
ANL  A,FIFO IN/COMMAND IN
ORL  A,FIFO IN/COMMAND IN
XRL  A,FIFO IN/COMMAND IN
CJNE A,FIFO IN/COMMAND IN, rel
SUBB A,FIFO IN/COMMAND IN
MOV  direct,FIFO IN/COMMAND IN
MOV  @Ri,FIFO IN/COMMAND IN
MOV  Rn,FIFO IN/COMMAND IN
MOV  A,FIFO IN/COMMAND IN
```

After each access to these registers, they are overwritten by a new byte from the FIFO.

NOTE:
Instructions which use the FIFO IN or COMMAND IN SFR as both a source and destination register will have the data destroyed as the next data byte is rewritten into the FIFO IN register at the end of the instruction. These instructions are not supported by the UPI-452 FIFO. Data can only be read through the FIFO IN SFR and DSCs through the COMMAND IN SFR. Data read through the COMMAND IN SFR will be read as 0FFH, and DSCs read through the FIFO IN SFR will be read as OFFH. The Immediate Command in SFR is read with the same instructions as the FIFO IN and COMMAND IN SFRs.

The FIFO IN, COMMAND IN and Immediate Command In SFRs are read only registers. Any write operation performed on these registers will be ignored and the FIFO pointers will remain intact.

The internal CPU uses the FIFO OUT SFR to write to the Output FIFO and any instruction which uses the FIFO OUT or COMMAND OUT SFR as a destination will invoke a FIFO write. DSCs are differentiated from data by writing to the COMMAND OUT SFR. In the FIFO, Data Stream Commands have the ninth bit associated with the command byte set to "1". The instructions used to write to the Output FIFO are listed below:

```
MOV FIFO OUT/COMMOUT, A
MOV FIFO OUT/COMMOUT, direct
MOV FIFO OUT/COMMOUT, Rn
POP FIFO OUT/COMMOUT
MOV FIFO OUT/COMMOUT, #data
MOV FIFO OUT/COMMOUNT, @Ri
```

NOTE:
Instructions which use the FIFO OUT/COMMAND OUT SFRs as both a source and destination register cause invalid data to be written into the Output FIFO. These instructions are not supported by the UPI-452 FIFO.

GENERAL PURPOSE DMA CHANNELS

Overview

There are two identical General Purpose DMA Channels on the UPI-452 which allow high speed data transfer from one writeable memory space to another. As many as 64K bytes can be transferred in a single DMA operation. The following memory spaces can be used with DMA channels:

- Internal Data Memory
- External Data Memory
- Special Function Registers

The Special Function Register array appears as a limited group of dedicated memory addresses. The Special Function Registers may be used in DMA transfer operations by specifying the SFR as the source or destination address. The Special Function Registers which may be used in DMA transfers are listed in Table 4. Table 4 also shows whether the SFR may be used as Source or Destination only, or both.

The FIFO can be accessed during DMA by using the FIFO IN SFR as the DMA Source Address Register (SAR) or the FIFO OUT SFR as the Destination Address Register (DAR). (Note: Since the FIFO IN SFR is a read only register, the DMA transfer will be ignored if it is used as a DMA DAR. This is also true if the FIFO OUT SFR is used as a DMA SAR.)

Each DMA channel is software programmable to operate in either Block Mode or Demand Mode. In the Block Mode, DMA transfers can be further programmed to take place in Burst Mode or Alternate Cycle mode. In Burst Mode, the processor halts its execution and dedicates its resources to the DMA transfer. In Alternate Cycle Mode, DMA cycles and instruction cycles occur alternately.

In Demand Mode, a DMA transfer occurs only when it is demanded. Demands can be accepted from an external device (through External Interrupt pins, EXT0/EXT1) or from either the Serial Channel or FIFO flags. In this way, a DMA transfer can be synchronized to an external device, the FIFO or the Serial Port. If the External Interrupt is configured in Edge Mode, a single byte transfer occurs per transition. The external interrupt itself will occur if enabled. If the External Interrupt is configured in Level Mode, DMA transfers continue until the External Interrupt request goes inactive or the byte count becomes zero. The following flags activate Demand Mode transfers of one byte to/from the FIFO or Serial Channel:

RI - Serial Channel Receiver Buffer Full

TI - Serial Channel Transmitter Buffer Empty

Architecture

There are three 16 bit and one 8 bit Special Function Registers associated with each DMA channel.

- The 16 bit Source Address SFR (SAR) points to the source byte.
- The 16 bit Destination Address SFR (DAR) points to the destination.
- The 16 bit Byte Count SFR (BCR) contains the number of bytes to be transferred and is decremented when a byte transfer is accomplished.
- The DMA Control SFR (DCON) is eight bits wide and specifies the source memory space, destination memory space and the mode of operation.

In Auto Increment mode, the Source Address and/or Destination Address is incremented when a byte is transferred. When a DMA transfer is complete (BCR = 0), the DONE bit is set and a maskable interrupt is generated. The GO bit must be set to start any DMA transfer (also, the Slave Control SFR FRZ bit must be set to disable FIFO DMA Freeze Mode). The two DMA channels are designated as DMA0 and DMA1, and their corresponding registers are suffixed by 0 or 1; e.g. SAR0, DAR1, etc.

Table 4. DMA Accessible Special Function Registers

SFR	Symbol	Address	Source Only	Destination Only	Either
Accumulator	A/ACC	0E0H			Y
B Register	B	0F0H			Y
FIFO IN	FIN	0EEH	Y		
COMMAND IN	CIN	0EFH	Y		
FIFO OUT	FOUT	0FEH		Y	
COMMAND OUT	COUT	0FFH		Y	
Serial Data Buffer	SBUF	099H			Y
Port 0	P0	080H			Y
Port 1	P1	090H			Y
Port 2	P2	0A0H			Y
Port 3	P3	0B0H			Y
Port 4	P4	0C0H			Y

DMA Special Function Registers

DMA Control SFR: DCON0, DCON1

Symbolic Address									Physical Address
DCON0	DAS	IDA	SAS	ISA	DM	TM	DONE	GO	092H
DCON1	DAS	IDA	SAS	ISA	DM	TM	DONE	GO	093H
	(MSB)							(LSB)	

Reset Status: DCON0 and DCON1 = 00H

Bit Definition:

DAS	IDA	Destination Address Space
0	0	External Data Memory without Auto-Increment
0	1	External Data Memory with Auto-Increment
1	0	Special Function Register
1	1	Internal Data Memory

SAS	ISA	Source Address Space
0	0	External Data Memory without Auto-Increment
0	1	External Data Memory with Auto-Increment
1	0	Special Function Register
1	1	Internal Data Memory

DM	TM	DMA Transfer Mode
0	0	Alternate-Cycle Transfer Mode
0	1	Burst Transfer Mode
1	0	FIFO or Serial Channel Demand Mode
1	1	External Demand Mode

DONE DMA transfer Flag:

0 DMA transfer is not completed.

1 DMA transfer is complete.

NOTE:
This flag is set when contents of the Byte Count SFR decrements to zero. It is reset automatically when the DMA vectors to its interrupt routine.

GO Enable DMA Transfer:

0 Disable DMA transfer (in all modes).

1 Enable DMA transfer. If the DMA is in the Block mode, start DMA transfer if possible. If it is in the Demand mode, enable the channel and wait for a demand.

NOTE:
The GO bit is reset when the BCR decrements to zero.

DMA Transfer Modes

The following four modes of DMA operation are possible in the UPI-452.

1. ALTERNATE-CYCLE MODE

General

Alternate cycle mode is useful when CPU processing must occur during the DMA transfers. In this mode, a DMA cycle and an instruction cycle occur alternately. The interrupt request is generated (if enabled) at the end of the process, i.e. when BCR decrements to zero. The transfer is initiated by setting the GO bit in the DCON SFR.

Alternate-Cycle FIFO Demand Mode

Alternate cycle demand mode is useful for FIFO transfers of a less urgent nature. As mentioned before, CPU instruction cycles are interleaved with DMA transfer cycles, allowing true parallel processing.

This mode differs from FIFO Demand Mode in that CPU instruction cycles must be interleaved with DMA transfers, even if the FIFO is demanding DMA. In FIFO Demand Mode, CPU cycles would never occur if the FIFO demand was present.

Input Channel

The DMA is configured as in FIFO Demand Mode and transfers are initiated whenever an Input FIFO service request is generated. DMA transfer cycles are alternated with instruction execution cycles. DMA transfers are terminated as in FIFO Demand Mode.

Output Channel

The DMA is configured as in FIFO Demand Mode and transfers are initiated whenever an Output FIFO requests service. DMA transfer cycles are alternated with instruction execution cycles. DMA transfers are terminated as in FIFO Demand Mode.

The FIFO logic resets the interrupt flag after transferring the byte, so the interrupt is never generated.

Once the DMA is programmed to service the FIFO, the request for service interrupt for the FIFO is inhibited until the DMA is done (BCR = 0).

2. BURST MODE

In BURST mode the DMA is initiated by setting the GO bit in the DCON SFR. The DMA operation continues until BCR decrements to zero (zero byte count), then an interrupt is generated (if enabled). No interrupts are recognized during this DMA operation once it has started.

Input Channel

The FIFO Input Channel can be used in burst mode by specifying the FIFO IN SFR as the DMA Source Address. DMA transfers begin when the GO bit in the DMA Control SFR is set. The number of bytes to be transferred must be specified in the Byte Count SFR (BCR) and auto-incrementing of the SAR must be disabled. Once the GO bit is set nothing can interrupt the transfer of data until the BCR is zero. In this mode, a Data Stream Command encountered in the FIFO will be held in the COMMAND IN SFR with the pointers frozen, and invalid data (FFH) will be read through the FIFO IN SFR. If the input FIFO becomes empty during the block transfer, an 0FFH will be read until BCR decrements to zero.

Output Channel

The Output FIFO Channel can be used in burst mode by specifying the FIFO OUT or COMMAND OUT SFR as the DMA Destination Address. DMA transfers begin when the GO bit is set. This mode can be used to send a block of data or a block of Data Stream Commands. If the FIFO becomes full during the block transfer, the remaining data will be lost.

NOTE:
All interrupts including FIFO interrupts are not recognized in Burst Mode. Burst Mode transfers should be used to service the FIFO only when the user is certain that no Data Stream Commands are in the block to be transferred (Input FIFO) and that the FIFO contains enough space to store the block to be transferred. In all other cases Alternate Cycle or Demand Mode should be used.

3. FIFO AND SERIAL CHANNEL DEMAND MODES

NOTES:
1. If the output FIFO is configured as a one byte buffer and the user program consists of two-cycle instructions only, then Alternate-Cycle Mode should be used.
2. In non-auto increment mode for internal to external, or external to internal transfers, the lower 8 bits of the external address should not correspond to the FIFO or Serial Port address.

FIFO Demand Mode

Although any DMA mode is possible using the FIFO buffer, only FIFO Demand and Alternate Cycle FIFO Demand Modes are recommended. FIFO Demand Mode DMA transfers using the input FIFO Channel are set-up by setting the GO bit and specifying the FIFO IN register as the DMA Source Address Register. The BCR should be set to the maximum number of expected transfers. The user must also program bit 1 of the Slave Control Register (SC1) to determine whether the Slave Status (SSTAT) SFR FIFO Request For Service Flag will be activated when the FIFO becomes not empty or full. Once the Request For Service Flag is activated by the FIFO, the DMA transfer begins, and continues until the request flag is deactivated. While the request is active, nothing can interrupt the DMA (i.e. it behaves like burst mode). The DMA Request is held active until one of the following occurs:

1) The FIFO becomes empty.
2) A Data Stream Command is encountered (this generates a FIFO interrupt and DMA operation resumes after the Data Stream Command is read).
3) BCR = 0 (this generates a DMA interrupt and sets the DONE bit).

DMA transfers to the Output FIFO Channel are similar. The FIFO OUT or COMMAND OUT SFR is the DMA Destination Address SFR and a transfer is started by setting the GO bit. The user programs bit 0 of the Slave Control SFR (SC0) to determine whether a demand occurs when the Output FIFO is not full or empty. DMA transfers begin when the Request For Service Flag is activated by the FIFO logic and continue as long as the flag is active. The Flag remains active until one of the following occurs:

1) The FIFO becomes full
2) BCR = 0 (this generates a DMA interrupt and sets the DONE bit).

As in Alternate Cycle FIFO Demand Mode, the FIFO logic resets the interrupt flag after transferring the byte, so the interrupt is never generated.

After the GO bit is set, the DMA is activated if one of the following conditions takes place:
SAR(0/1) = FIFO IN and HIFRS flag is set
DAR(0/1) = FIFO OUT and HOFRS flag is set

The HIFRS and HOFRS signals are internal flags which are not accessible by software. These flags are similar to the SST0 and SST4 flags in the Slave Status Register except that they are of the opposite polarity and once set they are not cleared until the Input FIFO becomes empty (HIFRS) or the Output FIFO becomes full (HOFRS).

Serial Channel Demand Mode

Serial Channel Demand Mode is the logical choice when using the Serial Port. The DMAs can be activated by one of the Serial Channel Flags. Receiver interrupt (RI) or Transmitter Interrupt (TI).

SAR(0/1) = SBUF and RI flag is set
DAR(0/1) = SBUF and TI flag is set

NOTE:
TI flag must be set by software to initiate the first transfer.

When the DMA transfer begins, only one byte is transferred at a time. The serial port hardware automatically resets the flag after completion of the transfer, so an interrupt will not be generated unless DMA servicing is held off due to the DMA being done (BCR = 0) or when the Hold/Hold Acknowledge logic is used and the DMA does not own the bus. In this case a Serial Port interrupt may be generated if enabled because of the status of the RI or TI flags.

In FIFO demand mode, Alternate cycle FIFO demand mode or Serial Port demand mode only one of the following registers (SBUF, FIN or FOUT) should be used as either the SAR or DAR registers to prevent undesired transfers. For example if SAR0 = FIN and DAR0 = SBUF in demand mode, the DMA transfer will start if either the HIFRS or TI flags are set.

4. EXTERNAL DEMAND MODE

The DMA can be initiated by an external device via External interrupt 0 and 1 (INT0/INT1) pins. The INT0 pin demands DMA0 (Channel 0) and INT1 demands DMA1 (Channel 1). If the interrupts are configured in edge mode, a single byte transfer is accomplished for every request. Interrupts also result (INT0 and INT1) after every byte transfer (if enabled). If the interrupts are configured in level mode, the DMA transfer continues until the request goes inactive or BCR = 0. In either case, a DMA interrupt is generated (if enabled) when BCR = 0. The GO bit must be set for the transfer to begin.

EXTERNAL MEMORY DMA

When transferring data to or from external memory via DMA, the HOLD (HLD) and HOLD-ACKNOWLEDGE (HLDA) signals are used for handshaking. The HOLD and HOLD-ACKNOWLEDGE are active low signals which arbitrate control of the local bus. The UPI-452 can be used in a system where multimasters are connected to a single parallel Address/Data bus. The HLD/HLDA signals are used to share resources (memory, peripherals, etc.) among all the processors on the local bus. The UPI-452 can be configured in any of three different External Memory Modes controlled by bits 5 and 6 (REQ & ARB) in the PCON SFR (Table 5). Each mode is described below:

REQUESTER MODE: In this mode, the UPI-452 is not the bus master, but must request the bus from another device. The UPI-452 configures port pin P1.5 as a HLD output and pin P1.6 as a HLDA input. The UPI-452 issues a HLD signal when it needs external access for a DMA channel. It uses the local bus after receiving the HLDA signal from the bus master, and will not release the bus until its DMA operation is complete.

ARBITER MODE: In this mode, the UPI-452 is the bus master. It configures port pin P1.5 as HLD input and pin P1.6 as HLDA output. When a device asserts the HLD signal to use the local bus, the UPI-452 asserts the HLDA signal after current instruction execution is complete. If the UPI-452 needs an external access via a DMA channel, it waits until the requester releases the bus, HLD goes inactive.

DISABLE MODE: When external program memory is accessed by an instruction or by program counter overflow beyond the internal ROM address or external data memory is accessed by MOVX instructions, it is a local memory access and the HLD/HLDA logic is not initiated. When a DMA channel attempts data transfer to/from the external data memory, the HLD/HLDA logic is initiated as described below. DMA transfers from the internal memory space to the internal memory space does not initiate the HLD/HLDA logic.

The balance of the PCON SFR bits are described in the "80C51 Register Description: Power Control SFR" section below.

Latency

When the GO bit is set, the UPI-452 finishes the current instruction before starting the DMA operation. Thus the maximum latency is 3.5 microseconds (at 14 MHz).

DMA Interrupt Vectors

Each DMA channel has a unique vectored interrupt associated with it. There are two vectored interrupts associated with the two DMA channels. The DMA interrupts are enabled and priorities set via the Interrupt Enable and Priority SFR (see "Interrupts" section). The interrupt priority scheme is similar to the scheme in 80C51.

Table 5. DMA MODE CONTROL - PCON SFR

Symbolic Address								Physical Address
PCON	—*	ARB	REQ	—*	—*	—*	—*	—*	87H
	(MSB)							(LSB)	

*Defined as per MLS-51 Data Sheet
Reset Status: 00H

Definition:

ARB	REQ	
0	0	HLD/HLDA logic is disabled.
0	1	The UPI-452 is in the Requester Mode.
1	0	The UPI-452 is in the Arbiter Mode.
1	1	Invalid

When a DMA operation is complete (BCR decrements to zero), the DONE flag in the respective DCON (DCON0 or DCON1) SFR is set. If the DMA interrupt is enabled, the DONE flag is reset automatically upon vectoring to the interrupt routine.

Interrupts When DMA is Active

If a Burst Mode DMA transfer is in progress, the interrupts are not serviced until the DMA transfer is complete. This is also true for level activated External Demand DMA transfers. During Alternate Cycle DMA transfers, however, the interrupts are serviced at the end of the DMA cycle. After that, DMA cycles and instruction execution cycles occur alternately. In the case of edge activated External Demand Mode DMA transfers, the interrupt is serviced at the end of DMA transfer of that single byte.

DMA Arbitration

Only one of the two DMA channels is active at a time, except when both are configured in the Alternate Cycle mode. In this case, the DMA cycles and Instruction Execution cycles occur in the following order:

1. DMA Cycle 0.
2. Instruction execution.
3. DMA Cycle 1.
4. Instruction execution.

DMA0 has priority over DMA1 during simultaneous activation of the two DMA channels. If one DMA channel is active, the other DMA channel, if activated, waits until the first one is complete.

If DMA0 is already in the Alternate Cycle mode and DMA1 is activated in Alternate Cycle Mode, it will take two instruction cycles before DMA1 is activated (due to the priority of DMA0). Once DMA1 becomes active, the execution will follow the normal sequence.

If DMA0 is already in the Alternate Cycle mode and DMA1 is activated in Burst Mode, the DMA1 Burst transfer will follow the DMA0 Alternate Cycle transfer (after the completion of the next instruction).

If the UPI-452 (as a Requester) asserts a HLD signal to request a DMA transfer (see "External Memory DMA")and its other DMA Channel requests a transfer before the HLDA signal is received, the channel having higher priority is activated first. A Burst Mode transfer on channel 0 can not be interrupted since DMA0 has the highest priority. A Demand Mode transfer on channel 0 is the only type of activity that can interrupt a block transfer on DMA1.

If, while executing a DMA transfer, the Arbiter receives a HLD signal, and then before it can acknowledge, its other DMA Channel requests a transfer, it then completes the second DMA transfer before sending the HLDA signal to release the bus to the HLD request.

DMA transfers may be held off under the following conditions:

1. A write to any of the DMA registers inhibits the DMA for one instruction cycle.

NOTE:

An instruction cycle may be executed in 1, 2 or 4 machine cycles dependent on the instruction being executed. DMA transfers are only executed after the completion of an instruction cycle never between machine cycles of a single instruction cycle. Similarly instruction cycles are only executed upon completion of a DMA transfer whether it be a one machine cycle transfer or two machine cycles (for ext. to ext. memory transfers).

2. A single machine cycle DMA register read operation (i.e. MOV A, DCON0) will inhibit the DMA for one instruction cycle. However a two cycle DMA register read operation will not inhibit the DMA (i.e. MOV P1, DCON0).

If the HOLD/HOLD Acknowledge logic is enabled in requestor mode the hold request will go active once the go bit has been set (for burst mode) and once the demand flag is set (for demand mode) regardless of whether the DMA is held off by one of the above conditions.

The DMA Transfer waveforms are in Figures 8-11.

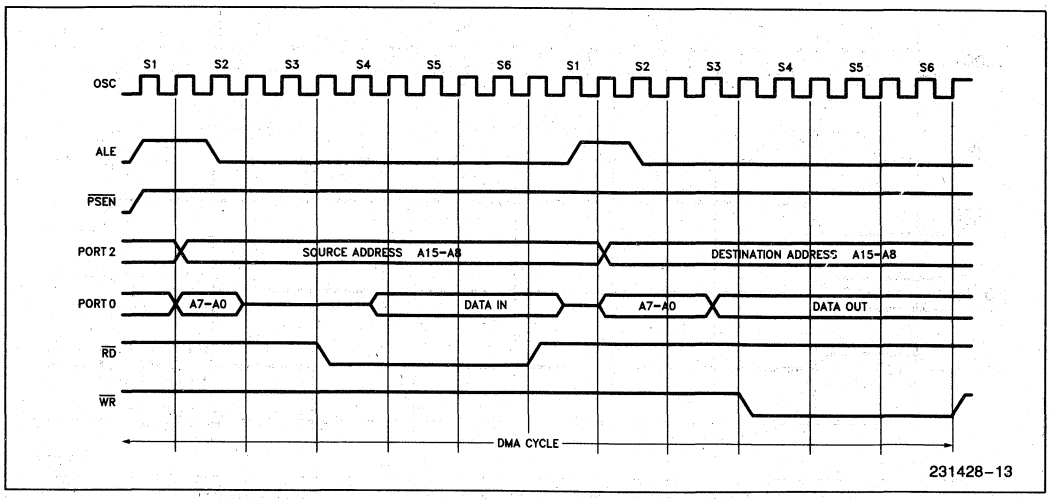


Figure 8. DMA Transfer from External Memory to External Memory

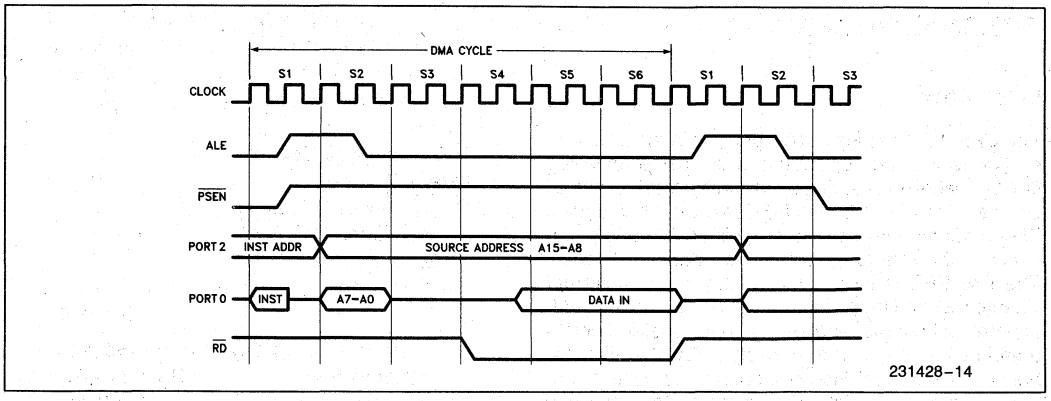


Figure 9. DMA Transfer from External Memory to Internal Memory

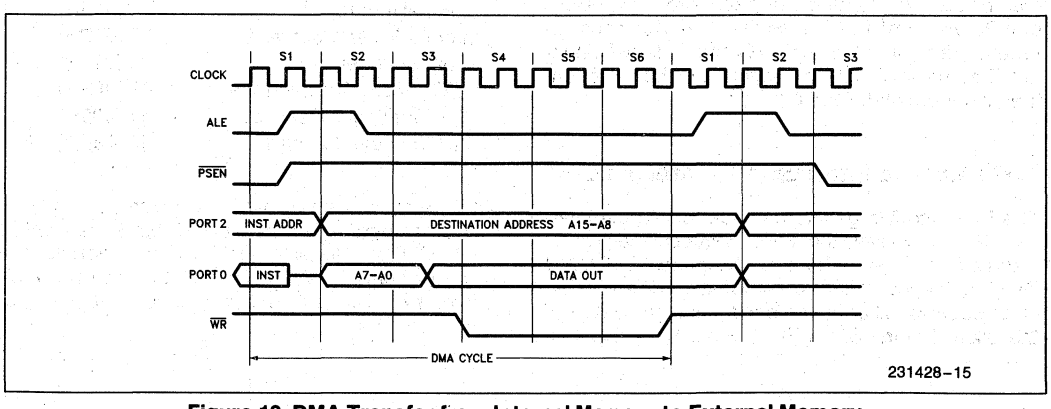


Figure 10. DMA Transfer from Internal Memory to External Memory

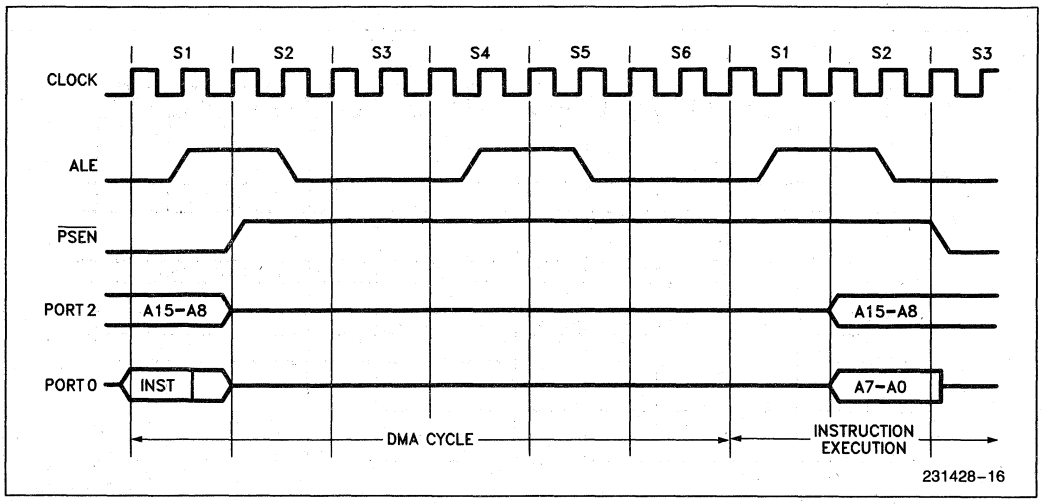


Figure 11. DMA Transfer from Internal Memory to Internal Memory

INTERNAL INTERRUPTS

Overview

The UPI-452 provides a total of eight interrupt sources (Table 6). Their operation is the same as in the 80C51, with the addition of three new interrupt sources for the UPI-452 FIFO and DMA features. These added interrupts have their enable and priority bits in the Interrupt Enable and Priority (IEP) SFR. The IEP SFR is in addition to the 80C51 Interrupt Enable (IE) and Interrupt Priority (IP) SFRs. The added interrupt sources are also globally enabled or disabled by the EA bit in the Interrupt Enable SFR. Table 6 lists the eight interrupt sources in order of priority. Table 7 lists the eight interrupt sources and their respective address vector location in program memory. (DMA interrupts are discussed in the "General Purpose DMA Channels" section. Additional interrupt information for Timer/Counter, Serial Channel, External Interrupt may be found in the Microcontroller Handbook for the 80C51.)

Table 6. Interrupt Priority

Interrupt Source	Priority Level
	(highest)
External Interrupt 0	0
Internal Timer/Counter 0	1
DMA Channel 0 Request	2
External Interrupt 1	3
DMA Channel 1 Request	4
Internal Timer/Counter 1	5
FIFO - Slave Bus Interface	6
Serial Channel	7
	(lowest)

Table 7. Interrupt Vector Addresses

Interrupt Source	Starting Address
External Interrupt 0	3 (003H)
Internal Timer/Counter 0	11 (00BH)
External Interrupt 1	19 (013H)
Internal Timer/Counter 1	27 (01BH)
Serial Channel	35 (023H)
FIFO - Slave Bus Interface	43 (02BH)
DMA Channel 0 Request	51 (033H)
DMA Channel 1 Request	59 (03BH)

FIFO Module Interrupts to Internal CPU

The FIFO module generates interrupts to the internal CPU whenever the FIFO requests service or when a Data Stream Command is in the COMMAND IN SFR. The Input FIFO will request service whenever it becomes full or not empty depending on bit 1 of the Slave Control SFR (IFRS). Similarly, the Output FIFO requests service when it becomes empty or not full as determined by bit 0 of the Slave Control SFR (OFRS). Request for Service interrupts are generated only if enabled by the internal CPU via the Interrupt Enable SFR, and the Slave Control Register.

A Data Stream Command Interrupt is generated whenever there is a Data Stream Command in the COMMAND IN SFR. The interrupt is generated to ensure that the internal interrupt is recognized before another instruction is executed.

Immediate Command Interrupts

a. An Immediate Command IN interrupt is generated, if enabled, to the internal CPU when the Host has written to the Immediate Command IN (IMIN) SFR. The write operation clears the Slave Status SFR bit (SSTAT SST2) and sets the Host Status SFR bit (HSTAT HST2) to indicate that a byte is present in the Immediate Command IN SFR. When the internal CPU reads the Immediate Command IN (IMIN) SFR the Slave Status SFR status bit is set, and the Host Status SFR status bit is cleared indicating the IMIN SFR is empty. Clearing the Host Status SFR bit will cause a Request For Service (INTRQ) interrupt, if enabled, to signal the Host that the IMIN SFR is empty. (See Figure 7a, Immediate Command IN Flowchart.)

b. An Immediate Command OUT interrupt is generated, if enabled, to the internal CPU when the Host has read the Immediate Command OUT SFR. The Host read causes the Slave Status Immediate Command OUT bit (SSTAT SST6) to be set and the corresponding Host Status bit (HSTAT HST6) to be cleared indicating the SFR is empty. When the internal CPU writes to the Immediate Command OUT SFR, the Host Status bit is set and Slave Status bit is cleared to indicate the SFR is full. (See Figure 7b, Immediate Command OUT Flowchart.)

NOTE:
Immediate Command IN and OUT interrupts are actually specific FIFO-Slave Interface interrupts to the internal CPU.

One instruction from the main program is executed between two consecutive interrupt service routines as in the 80C51. However, if the second interrupt service routine is due to a Data Stream Command Interrupt, the main program instruction is not executed (to prevent misreading of invalid data).

Interrupt Enabling and Priority

Each of the three interrupt special function registers (IE, IP and IEP) is listed below with its corresponding bit definitions.

Interrupt Enable SFR (IE)

Symbolic Address									Physical Address
IE	EA	—	—	ES	ET1	EX1	ET0	EX0	0A8H
	(MSB)							(LSB)	

Symbol	Position	Function
EA	IE.7	Enables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	(reserved)
—	IE.5	(reserved)
ES	IE.4	Serial Channel interrupt enable
ET1	IE.3	Internal Timer/Counter 1 Overflow Interrupt
EX1	IE.2	External Interrupt Request 1.
ET0	IE.1	Internal Timer/Counter 0 Overflow Interrupt
EX0	IE.0	External Interrupt Request 0.

Interrupt Priority SFR (IP)

A priority level of 0 or 1 may be assigned to each interrupt source, with 1 being higher priority level, through the IP and the IEP (Interrupt Enable and Priority) SFR. A priority level of 1 interrupt can interrupt a priority level 0 service routine to allow nesting of interrupts.

Symbolic Address									Physical Address
IP	—	—	—	PS	PT1	PX1	PT0	PX0	0B8H
	(MSB)							(LSB)	

Symbol	Position	Function	Priority Within A Level
			(lowest)
—	IP.7	(reserved)	—
—	IP.6	(reserved)	—
—	IP.5	(reserved)	—
PS	IP.4	Local Serial Channel	0.7
PT1	IP.3	Internal Timer/Counter 1	0.5
PX1	IP.2	External Interrupt Request 1	0.3
PT0	IP.1	Internal Timer/Counter 0	0.1
PX0	IP.0	External Interrupt Request 0	0.0
			(highest)

Interrupt Enable and Priority SFR (IEP)

The Interrupt Enable and Priority Register establishes the enabling and priority of those resources not covered in the Interrupt Enable and Interrupt Priority SFRs.

Symbolic Address									Physical Address
IEP	—	—	PFIFO	EDMA0	EDMA1	PDMA0	PDMA1	EFIFO	0F8H
	(MSB)							(LSB)	

Symbol	Position	Function	Priority Within a Level
—	IEP.7	(reserved)	
—	IEP.6	(reserved)	
PFIFO	IEP.5	FIFO Slave Bus Interface Interrupt Priority	0.6
EDMA0	IEP.4	DMA Channel 0 Interrupt Enable	
EDMA1	IEP.3	DMA Channel 1 Interrupt Enable	
PDMA0	IEP.2	DMA Channel 0 Priority	0.2
PDMA1	IEP.1	DMA Channel 1 Priority	0.4
EFIFO	IEP.0	FIFO Slave Bus Interface Interrupt Enable	

FIFO-EXTERNAL HOST INTERFACE FIFO DMA FREEZE MODE

Overview

During FIFO DMA Freeze Mode the internal CPU can reconfigure the FIFO interface. FIFO DMA Freeze Mode is provided to prevent the Host from accessing the FIFO during a reconfiguration sequence. The internal CPU invokes FIFO DMA Freeze Mode by clearing bit 3 of the Slave Control SFR (SC3). INTRQ becomes active whenever FIFO DMA Freeze Mode is invoked to indicate the freeze status. The interrupt can only be deactivated by the Host reading the Host Status SFR.

During FIFO DMA Freeze Mode only two operations are possible by the Host to the UPI-452 slave, the balance are disabled, as shown in Table 8. The internal DMA is disabled during FIFO DMA Freeze Mode, and the internal CPU has write access to all of the FIFO control SFRs (Table 9).

Initialization

At power on reset the FIFO Host interface is automatically frozen. The Slave Control Enable FIFO DMA Freeze Mode bit defaults to FIFO DMA Freeze Mode (SLCON FRZ=0). Below is a list of the FIFO Special Function Registers and their default power on reset values;

SFR Name	Label	Value
Channel Boundary Pointer	CBP	40H / 64D
Output Channel Read Pointers	ORPR	40H / 64D
Output Channel Write Pointers	OWPR	40H / 64D
Input Channel Read Pointers	IRPR	00H / 00D
Input Channel Write Pointers	IWPR	00H / 00D
Input Threshold	ITHR	80H / 128D
Output Threshold	OTHR	01H / 1D

The Input and Output FIFO channels can be reconfigured by programming any of these SFRs while the UPI-452 is in the Freeze Mode. The Host is notified when the Freeze Mode is active by a "1" in HST1 of the Host Status Register (HSTAT). The Host should interrogate HST1 to determine the status of the FIFO interface following reset before attempting to read from or write to the UPI-452 FIFO buffer.

NOTE:
During the initialization sequence of the UPI-452 FIFO SFRs, the OTHR should be changed from the default setting of 1 to a value between 2 and {(80H-CBP)-1}. Please refer to the section on Input and Output FIFO threshold SFRs for further information.

Table 8. Slave Bus Interface Status During FIFO DMA Freeze Mode

Interface Pins; $\overline{DACK}$	$\overline{CS}$	A2	A1	A0	$\overline{READ}$	$\overline{WRITE}$	Operation In Normal Mode	Status in FIFO DMA Freeze Mode
1	0	0	1	0	0	1	Read Host Status SFR	Operational
1	0	0	1	1	0	1	Read Host Control SFR	Operational
1	0	0	1	1	1	0	Write Host Control SFR	Disabled
1	0	0	0	0	0	1	Data or DMA Data from Output Channel	Disabled
1	0	0	0	0	1	0	Data or DMA Data to Input Channel	Disabled
1	0	0	0	1	0	1	Data Stream Command from Output Channel	Disabled
1	0	0	0	1	1	0	Data Stream Command to Input Channel	Disabled
1	0	1	0	0	0	1	Read Immediate Command Out from Output Channel	Disabled
1	0	1	0	0	1	0	Write Immediate Command In to Input Channel	Disabled
0	X	X	X	X	0	1	DMA Data from Output Channel	Disabled
0	X	X	X	X	1	0	DMA Data to Input Channel	Disabled

The UPI-452 can also be programmed to interrupt the Host following power on reset in order to indicate to the Host that FIFO DMA Freeze Mode is in progress. This is done by enabling the INTRQ interrupt output pin via the MODE SFR (MD4) before the Slave Control SFR Enable FIFO DMA Freeze Mode bit is set to Normal Mode. At power on reset the Mode SFR is forced to zero. This disables all interrupt and DMA output pins (INTRQ, DRQIN/INTRQIN and DRQOUT/INTRQOUT). Because the Host Status SFR FIFO DMA Freeze Mode In Progress bit is set, a Request For Service, INTRQ, interrupt is pending until the Host Status SFR is read. This is because the FIFO DMA Freeze Mode interrupt is always enabled. If the Slave Control FIFO DMA Freeze Mode bit (SLCON FRZ) is set to Normal Mode before the MODE SFR INTRQ bit is enabled, the INTRQ output will not go active when the MODE SFR INTRQ bit is enabled if the Host Status SFR has been read.

The default values for the FIFO and Slave Interface represents minimum UPI-452 internal initialization. No specific Special Function Register initialization is required to begin operation of the FIFO Slave Interface. The last initialization instruction must always set the UPI-452 to Normal Mode. This causes the UPI-452 to exit FIFO DMA Freeze Mode and enables Host read/write access of the FIFO.

Following reset, either hardware (via the RST pin) or software (via HCON SFR bit HC3) the UPI-452 requires 2 internal machine cycles (24 TCLCL) to update all internal registers.

Invoking FIFO DMA Freeze Mode During Normal Operation

When the UPI-452 is in normal operation, FIFO DMA Freeze Mode should not be arbitrarily invoked by clearing SC3 (SC3 = 0) because the external Host runs asynchronously to the internal CPU. Invoking FIFO DMA Freeze Mode without first stopping the external Host from accessing the UPI-452 will not guarantee a clean break with the external Host.

The proper way to invoke FIFO DMA Freeze Mode is by issuing an Immediate Command to the external host indicating that FIFO DMA Freeze Mode will be invoked. Upon receiving the Immediate Command, the external Host should complete servicing all pending interrupts and DMA requests, then send an Immediate Command back to the UPI-452 acknowledging the FIFO DMA Freeze Mode request. After issuing the first Immediate Command, the internal CPU should not perform any action on the FIFO until FIFO DMA Freeze Mode is invoked.

If FIFO DMA Freeze Mode is invoked without stopping the Host during Host transfers, only the last two bytes of data written into or read from the FIFO will be valid. The timing diagram for disabling the FIFO module to the external Host interface is illustrated in Figure 12. Due to this synchronization sequence, the UPI-452 might not go into FIFO DMA Freeze Mode immediately after SC3 is cleared. A special bit in the Slave Status Register (SST5) is provided to indicate the status of the FIFO DMA Freeze Mode. The FIFO DMA Freeze Mode operations described in this section are only valid after SST5 is cleared.

As FIFO DMA Freeze Mode is invoked, the DRQIN or DRQOUT will be deactivated (stopping the transferring of data), bit 1 of the Host Status SFR will be set (HST1 = 1), and SST5 will be cleared (SST5 = 0) to indicate to the external Host and internal CPU that the slave interface has been frozen. After the freeze becomes effective, any attempt by the external Host to access the FIFO will cause the overrun and underrun bits to be activated (bits HST7 (for reads) or HST3 (for writes)). These two bits, HST3 and HST7, will be set (deactivated) after the Host Status SFR has been read. If INTRQ is used to request service, the FIFO interface is frozen upon completion of any Host read or write operation in progress.

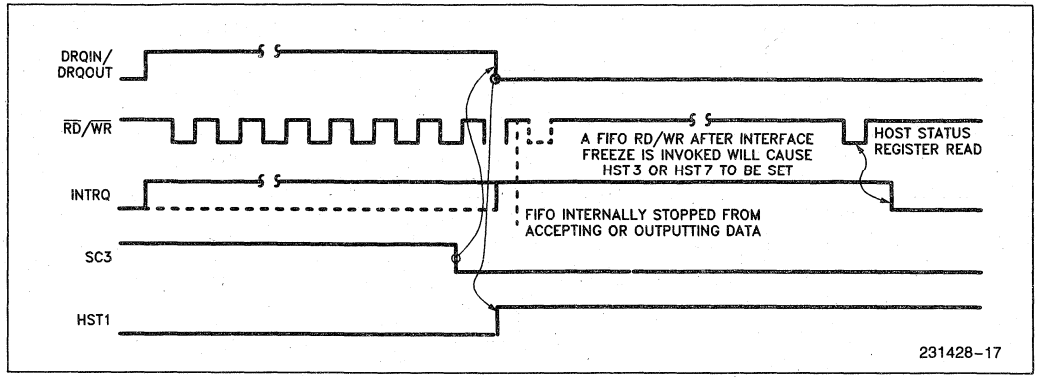


Figure 12. Disabling FIFO to Host Slave Interface Timing Diagram

External Host writing to the Immediate Command In SFR and the Host Control SFR is also inhibited when the slave bus interface is frozen. Writing to these two registers after FIFO DMA Freeze Mode is invoked will also cause HST3 (overrun) to be activated (HST3=0). Similarly, reading the Immediate Command Out Register by the external Host is disabled during FIFO DMA Freeze Mode, and any attempt to do so will cause the clearing (deactivating, "0") of HST7 bit (underrun).

After the slave bus interface is frozen, the internal CPU can perform the following operations on the FIFO Special Function Registers (these operations are allowed only during FIFO DMA Freeze Mode).

For FIFO Reconfiguration	1. Changing the Channel Boundary Pointer SFR. 2. Changing the Input and Output Threshold SFR.
To Enhance the Testability	3. Writing to the read and write pointers of the Input and Output FIFO's. 4. Writing to and reading the Host Control SFRs. 5. Controlling some bits of Host and Slave Status SFRS. 6. Reading the Immediate Command Out SFR and Writing to the Immediate Comand In SFR.

Description of each of these special functions are as follows:

FIFO Module SFRs During FIFO DMA Freeze Mode

Table 9 summarizes the characteristics of all the FIFO Special Function Registers during normal and FIFO DMA Freeze Modes. The registers that require special treatment in FIFO DMA Freeze Mode are: HCON, IWPR, IRPR, OWPR, ORPR, HSTAT, SSTAT, MIN & MOUT SFRs. They can be described in detail as follows:

Host Control SFR (HCON)

During normal operation, this register is written to or read by the external Host. However, in FIFO DMA Freeze Mode (i.e. SST5=0) the UPI-452 internal CPU has write access to the Host Control SFR and write operations to this SFR by the external Host will not be accepted. If the Host attempts to write to HCON, the Input Channel error condition flag (HST3) will be cleared.

Input FIFO Pointer Registers (IRPR & IWPR)

Once the FIFO module is in FIFO DMA Freeze Mode, error flags due to overrun and underrun of the Input FIFO pointers will be disabled. Any attempt to create an overrun or underrun condition by changing the Input FIFO pointers would result in an inconsistency in performance between the status flag and the threshold counter.

To enhance the speed of the UPI-452, read operations on the Input FIFO will look ahead by two bytes. Hence, every time the IRPR is changed during FIFO DMA Freeze Mode, two NOPs need to be executed so that the two byte pipeline can be updated with the new data bytes pointed to by the new IRPR. The Threshold Counter SFR also needs to change by the same number of bytes as the IRPR (increase Threshold Counter if IRPR goes forward or decrease if IRPR goes backward). This will ensure that future interrupts will still be generated only after a threshold number of bytes are available. (See "Input and Output FIFO Threshold SFR" section below.)

In FIFO DMA Freeze Mode, the internal CPU can also change the content of IWPR, and each change of IWPR also requires an update of the Threshold Counter SFR.

Normally, the internal CPU cannot write into the Input FIFO. It can, however, during FIFO DMA Freeze Mode by first reconfiguring the FIFO as an Output FIFO (Refer to "Input and Output FIFO Threshold SFR" section below). Changing the IRPR to be equal to IWPR generates an empty condition while changing IWPR to be equal to IRPR generates a full condition. The order in which the pointers are written determines whether a full or empty condition is generated.

Output FIFO Pointer SFR (ORPR and OWPR)

In FIFO DMA Freeze Mode the contents of OWPR can be changed by the internal CPU, but each change of OWPR or ORPR requires the Threshold Counter SFR to be updated as described in the next section. A NOP must be executed whenever a new value is written into ORPR, as just described for changes to IRPR. As before, changing ORPR to be equal to OWPR will generate an empty condition, Output FIFO overrun or underrun condition cannot be generated though. The FIFO pointers should not be set to a value outside of its range.

Table 9. FIFO SFR's Characteristics During FIFO DMA Freeze Mode

Label	Name	Normal Operation (SST5 = 1)	FIFO DMA Freeze Mode Operation (SST5 = 0)
HCON	Host Control	Not Accessible	Read & Write
HSTAT	Host Status	Read Only	Read & Write 4
SLCON	Slave Control	Read & Write	Read & Write
SSTAT	Slave Status	Read Only	Read & Write 4
IEP	Interrupt Enable & Priority	Read & Write	Read & Write
MODE	Mode Register	Read & Write	Read & Write
IWPR	Input FIFO Write Pointer	Read Only	Read & Write 5
IRPR	Input FIFO Read Pointer	Read Only	Read & Write 1, 5
OWPR	Output FIFO Write Pointer	Read Only	Read & Write 6
ORPR	Output FIFO Read Pointer	Read Only	Read & Write 2, 6
CBP	Channel Boundary Pointer	Read Only	Read & Write 3
IMIN	Immediate Command In	Read Only	Read & Write
IMOUT	Immediate Command Out	Read & Write	Read & Write
FIN	FIFO IN	Read Only	Read Only
CIN	COMMAND IN	Read Only	Read Only
FOUT	FIFO OUT	Read & Write	Read & Write
COUT	COMMAND OUT	Read & Write	Read & Write
ITHR	Input FIFO Threshold	Read Only	Read & Write
OTHR	Output FIFO Threshold	Read Only	Read & Write

NOTES:
1. Writing of IRPR will automatically cause the FIFO IN SFR to load the contents of the Input FIFO from that location.
2. Writing to ORPR will automatically cause the IOBL SFR to load the contents of the Output FIFO at that ORPR address.
3. Writing to the CBP SFR will cause automatic reset of the four pointers of the Input and Output FIFO channels.
4. The internal CPU cannot directly change the status of these registers. However, by changing the status of the FIFO channels, the internal CPU can indirectly change the contents of the status registers.
5. Changing the Input FIFO Read/Write Pointers also requires that a consistent update of the Input FIFO Threshold Counter SFR.
6. Changing the Output FIFO Read/Write Pointers also requires that a consistent update of the Output FIFO Threshold Counter SFR.

Input and Output FIFO Threshold SFR (ITHR & OTHR)

The Input and Output FIFO Threshold SFRs are also programmable by the internal CPU during FIFO DMA Freeze Mode. For proper operation of the Threshold feature, the Threshold SFR should be changed only when the Input and Output FIFO channels are empty, since they reflect the current number of bytes available to read/write before an interrupt is generated.

Table 10 illustrates the Threshold SFRs range of values and the number of bytes to be transferred when the Request For Service Flag is activated:

Table 10. Threshold SFRs Range of Values and Number of Bytes to be Transferred

ITHR (lower seven bits)	No. of Bytes Available to be Written	OTHR (lower seven bits)	No. of Bytes Available to be Read
0	CBP	2	3
1	CBP-1	3	4
2	CBP-2	•	•
•	•	•	•
•	•	•	•
•	•	(80H-CBP)-3	(80H-CBP)-2
CBP-3	3	(80H-CBP)-2	(80H-CBP)-1
		(80H-CBP)-1	(80H-CBP)

The eighth bit of the Input and Output FIFO Threshold SFR indicates the status of the service requests regardless of the freeze condition. If the eighth bit is a "1", the FIFO is requesting service from the external Host. In other words, when the Threshold SFR value goes below zero (2's complement), a service request is generated*. *The 8th bit of the ITHR SFR must be set during initialization if the Host interrupt request is desired immediately upon leaving Freeze Mode. Normally the ITHR SFR is decremented after each external Host write to the Input FIFO and incremented after each internal CPU read of the Input FIFO. The OTHR SFR is decremented by internal CPU writes and incremented by external Host reads. Thus if the pointers are moved when the FIFO's are not empty, these relationships can be used to calculate the offset for the Threshold SFRs. It is best to change the Threshold SFRs only when the FIFO's are empty to avoid this complication. The threshold registers should also be updated after the pointers have been manipulated.

NOTE:
The ITHR should only be programmed in the range from 0 to (CBP-3). An ITHR value of (CBP-2) could result in a failure to set the Input FIFO service request signal after the Input FIFO has been emptied.

Correspondingly, the OTHR should be programmed in the range from 2 to {(80H-CBP)-1}. An OTHR value of 1 could result in a failure to set the Output FIFO service request after subsequent writes by the UPI-452 have filled the Output FIFO.

NOTE:
When programming the ITHR SFR, the eighth bit should be set to 1 (OR'd with 80H). This causes HSTAT SFR HST0 = 0, Input FIFO Request For Service. If ITHR bit 7 = 0 then HSTAT HST0 = 1, Input FIFO Does Not Request Service, and no interrupt will be generated.

Host Status SFR (HSTAT)

When in FIFO DMA Freeze Mode, some bits in the Host Status SFR are forced high and will not reflect the new status until the system returns to normal operation. The definition of the register in FIFO DMA Freeze Mode is as follows:

NOTE:
The internal CPU reads this shadow latch value when reading the Host Status SFR. The shadow latch will keep the information for these bits so normal operation can be resumed with the right status. The following bits are set (= 1) when FIFO DMA Freeze Mode is invoked;

HST7 Output FIFO Error Condition Flag

1 = No error.

0 = An invalid read has been done on the output FIFO or the Immediate Command Out Register by the host CPU.

NOTE:
The normal underrun error condition status is disabled. If an Immediate Command Out (IMOUT) SFR read is attempted during FIFO DMA Freeze Mode, the contents of the IMOUT SFR is output on the Data Buffer and the error status is cleared (= 0).

HST6 Immediate Command Out SFR Status

During normal operation, this bit is cleared (=0) when the IMOUT SFR is written by the UPI-452 internal CPU and set (= 1) when the IMOUT SFR is read by the external Host. Once the host-slave interface is frozen (i.e. SST5 = 0), this bit will be read as a 1 by the host CPU. A shadow latch will keep the information for this bit so normal operation can be resumed with the correct status.

Shadow latch:

1 = Internal CPU reads the IMOUT SFR

0 = Internal CPU writes to the IMOUT SFR

6

HST5 Data Stream Command at Output FIFO

This bit is forced to a "1" during FIFO DMA Freeze Mode to prevent the external host CPU from trying to read the DSC. Once normal operation is resumed, HST5 will reflect the Data/Command status of the current byte in the Output FIFO.

Shadow Latch (read by the internal CPU):

1 = No Data Stream Command (DSC)

0 = Data Stream Command at Output FIFO

HST4 Output FIFO Service Request Status

When FIFO DMA Freeze Mode is invoked, this bit no longer reflects the Output FIFO Request Service Status. This bit wll be forced to a "1".

HST3 Input FIFO Error Condition Flag

1 = No error.

0 = One of the following operations has been attempted by the external host and is invalid:

1) Write into the Input FIFO

2) Write into the Host Control SFR

3) Write into the Immediate Command In SFR

NOTE:

The normal Input FIFO overrun condition is disabled.

HST2 Immediate Command In SFR Status

This bit is normally cleared when the internal CPU reads the IMIN SFR and set when the external host CPU writes into the IMIN SFR. When the host-slave interface is frozen, reading and writing of the IMIN by the internal CPU will change the shadow latch of this bit. This bit will be read as a "1" by the external Host.

Shadow latch.

1 = Internal CPU writes into IMIN SFR

0 = Internal CPU reads the IMIN SFR

HST1 FIFO DMA Freeze Mode Status

1 = FIFO DMA Freeze Mode.

0 = Normal Operation (non-FIFO DMA Freeze Mode).

NOTE:

This bit is used to indicate to the external Host that the host-slave interface has been frozen and hence the external Host functions are now reduced as shown in Table 8.

HST0 Input FIFO Request Service Satus

When slave interface is frozen this bit no longer reflects the Input FIFO Request Service Status. This bit will be forced to a "1".

Slave Status SFR (SSTAT)

The Slave Status SFR is a read-only SFR. However, once the slave interface is frozen, most of the bits of this SFR can be changed by the internal CPU by reconfiguring the FIFO and accessing the FIFO Special Function Registers.

SST7 Output FIFO Overrun Error Flag

Inoperative in FIFO DMA Freeze Mode.

SST6 Immediate Command Out SFR Status

In FIFO DMA Freeze Mode, this bit will be cleared when the internal CPU reads the Immediate Command Out SFR and set when the internal CPU writes to the Immediate Command Out Register.

SST5 FIFO-External Interface FIFO DMA Freeze Mode Status

This bit indicates to the internal CPU that FIFO DMA Freeze Mode is in progress and that it has write access to the FIFO Control, Host control and Immediate Command SFRs.

SST4 Output FIFO Request Service Status

During normal operation, this bit indicates to the internal CPU that the Output FIFO is ready for more data. The status of this bit reflects the position of the Output FIFO read and write pointers. Hence, in FIFO DMA Freeze Mode, this flag can be changed by the internal CPU indirectly as the read and write pointers change.

SST3 Input FIFO Underrun Flag

Inoperative during FIFO DMA Freeze Mode.

During normal operation, a read operation clears (=0) this bit when there are no data bytes in the Input FIFO and deactivated (=1) when the Slave Status SFR is read. In FIFO DMA Freeze Mode, this bit will not be cleared by an Input FIFO read underrun error condition, nor will it be reset by the reading of the Slave Status SFR.

SST2 Immediate Command In SFR Status

This bit is normally activated (=0) when the external host CPU writes into the Immediate Command In SFR and deactivated (=1) when it is read by the internal CPU. In FIFO DMA Freeze Mode, this bit will not be activated (=0) by the external Host's writing of the Immediate Command IN SFR since this function is disabled. However, this bit will be cleared (=0) if the internal CPU writes to the Immediate Command In SFR and it will be set =1) if it reads from the register.

SST1 Data Stream Command at Input FIFO Flag

In FIFO DMA Freeze Mode, this bit operates normally. It indicates whether the next byte of data from the Input FIFO is a DSC or data byte. If it is a DSC byte, reading from the FIFO IN SFR will result in reading invalid data (FFH) and vice versa. In FIFO DMA Freeze Mode, this bit still reflects the type of data byte available from the Input FIFO.

SST0 Input FIFO Service Request Flag

During normal operation, this bit is activated (=0) when the Input FIFO contains bytes that can be read by the internal CPU and deactivated (=1) when the Input FIFO does not need any service from the internal CPU. In FIFO DMA Freeze Mode, the status of this bit should not change unless the pointers of the Input FIFO are changed. In this mode, the internal CPU can indirectly change this bit by changing the read and write pointers of the Input FIFO but cannot change it directly.

Immediate Command In/Out SFR (IMIN/IMOUT)

If FIFO DMA Freeze Mode is in progress, writing to the Immediate Command In SFR by the external host will be disabled, and any such attempt will cause HST3 to be cleared (=0). Similarly, the Immediate Command Out SFR read operation (by the host) will be disabled internally and read attempts will cause HST7 to be cleared (=0).

Internal CPU Read and Write of the FIFO During FIFO DMA Freeze Mode

In normal operation, the Input FIFO can only be read by the internal CPU and similarly, the Output FIFO can only be written by the internal CPU. During FIFO DMA Freeze Mode, the internal CPU can read the entire contents of the Input FIFO by programming the CBP SFR to 7FH, setting the IRPR SFR to zero, and then the IWPR SFR to zero. Programming the pointer registers in this order generates a FIFO full signal to the FIFO logic and enables internal CPU read operations. If the IWPR and IRPR are already zero, the write pointer should be changed to a non-zero value to clear the empty status then the pointers can be set to zero. Writing to the IRDR SFR automatically updates the look ahead registers.

In a similar manner, the internal CPU can write to all 128 bytes of the FIFO by setting the CBP SFR to zero, setting OWPR SFR to zero, and then setting ORPR SFR to zero. This generates a FIFO empty signal and allows internal CPU write operations to all 128 bytes of the FIFO. The Threshold registers also need to be adjusted when the pointers are changed. (See "Input and Output FIFO Threshold SFR" section below.)

MEMORY ORGANIZATION

The UPI-452 has separate address spaces for Program Memory and Data Memory like the 80C51. The Program Memory can be up to 64K bytes. The lower 8K of Program Memory may reside on-chip. The Data Memory consists of 256 bytes of on-chip RAM, up to 64K bytes of off-chip RAM and a number of "SFRs" (Special Function Registers) which appear as yet another set of unique memory addresses.

Table 11a. Internal Memory Addressing

Memory Space	Addressing Method
Lower 128 Bytes of Internal RAM	Direct or Indirect
Upper 128 Bytes of Internal RAM	Indirect Only
UPI-452 SFR's	Direct Only

The 80C51 Special Function Registers are listed in Table 11a, and the additional UPI-452 SFRs are listed in Table 11b. A brief description of the 80C51 core SFRs is also provided below.

Accessing External Memory

As in the 80C51, accesses to external memory are of two types: Accesses to external Program Memory and accesses to external Data Memory.

External Program Memory is accessed under two conditions:

1) Whenever signal $\overline{EA}$ = 0; or
2) Whenever the program counter (PC) contains a number that is larger than 1FFFH.

This requires that the ROMless versions have $\overline{EA}$ wired low to enable the lower 8K program bytes to be fetched from external memory.

External Data Memory is accessed using either the MOVX @DPTR (16 bit address) or the MOVX @Ri (8 bit address) instructions, or during external data memory transfers.

6

Table 11b. 80C51 Special Function Registers

Symbol	Name	Address	Contents
*ACC	Accumulator	0E0H	00H
*B	B Register	0F0H	00H
*PSW	Program Status Word	0D0H	00H
SP	Stack Pointer	81H	07H
DPTR	Data Pointer (consisting of DPH and DPL)	82H	0000H
*P0	Port 0	80H	0FFH
*P1	Port 1	90H	0FFH
*P2	Port 2	0A0H	0FFH
*P3	Port 3	0B0H	0FFH
*IP	Interrupt Priority Control	0B8H	0E0H
*IE	Interrupt Enable Control	0A8H	60H
TMOD	Timer/Counter Mode Control	89H	00H
*TCON	Timer/Counter Control	88H	00H
TH0	Timer/Counter 0 (high byte)	8CH	00H
TL0	Timer/Counter 0 (low byte)	8AH	00H
TH1	Timer/Counter 1 (high byte)	8DH	00H
TL1	Timer/Counter 1 (low byte)	8BH	00H
*SCON	Serial Control	98H	00H
SBUF	Serial Data Buff	99H	I
PCON	Power Control	87H	I0H

I = Indeterminate
The SFRs marked with an asterisk (*) are both bit- and byte- addressable. The functions of the SFRs are as follows:

Table 11c. UPI-452 Additional Special Function Registers

Symbol	Name	Address	Contents
BCRL0	DMA Byte Count Low Byte/	0E2H	I
BCRH0	High Byte/ Channel 0	0E3H	I
BCRL1	Low Byte/	0F2H	I
BCRH1	Hi Byte/ Channel 1	0F3H	I
CBP	Channel Boundary Pointer	0ECH	40H
CIN	COMMAND IN	0EFH	I
COUT	COMMAND OUT DMA Destination Address	0FFH	I

Table 11c. UPI-452 Additional Special Function Registers (Continued)

Symbol	Name	Address	Contents
DARL0	Low Byte/	0C2H	I
DARH0	Hi Byte/ Channel 0	0C3H	I
DARL1	Low Byte/	0D2H	I
DARH1	Hi Byte/ Channel 1	0D3H	I
DCON0	DMA0 Control	92H	00H
DCON1	DMA1 Control	93H	00H
FIN	FIFO IN	0EEH	I
FOUT	FIFO OUT	0FEH	I
HCON	Host Control	0E7H	00H
HSTAT	Host Status	0E6H	0FBH
*IEP	Interrupt Enable and Priority	0F8H	0C0H
IMIN	Immediate Command In	0FCH	I
IMOUT	Immediate Command Out	0FDH	I
IRPR	Input Read Pointer	0EBH	00H
ITHR	Input FIFO Threshold	0F6H	80H
IWPR	Input Write Pointer	0EAH	00H
MODE	Mode Register	0F9H	8FH
ORPR	Output Read Pointer	0FAH	40H
OTHR	Output FIFO Threshold	0F7H	01H
OWPR	Output Write Threshold	0FBH	40H
*P4	Port 4 DMA Source Address	0C0H	0FFH
SARL0	Low Byte/	0A2H	I
SARH0	Hi Byte/ Channel 0	0A3H	I
SARL1	Low Byte/	0B2H	I
SARH1	Hi Byte/ Channel 1	0B3H	I
*SLCON	Slave Control	0E8H	04H
SSTAT	Slave Status	0E9H	08FH

I = Indeterminate
The SFRs marked with an asterisk (*) are both bit- and byte- addressable. The functions of the SFRs are as follows:

Miscellaneous Special Function Register Description

80C51 SFRs

ACCUMULATOR

ACC is the Accumuator SFR. The mnemonics for accumulator-specific instructions, however, refer to the accumulator simply as A.

B REGISTER

The B SFR is used during multiply and divide operations. For other instructions it can be treated as another scratch pad regster.

PROGRAM STATUS WORD

The PSW SFR contains program status information as detailed in Table 12.

STACK POINTER

The Stack Pointer register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H.

DATA POINTER

The Data Pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

PORTS 0 TO 4

P0, P1, P2, P3 and P4 are the SFR latches of Ports 0, 1, 2, 3 and 4, respectively.

SERIAL DATA BUFFER

The Serial Data Buffer is actually two separate registers, a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer where it is held for serial transmission. (Moving a byte to SBUF is what initiates the transmission.) When data is moved from SBUF, it comes from the receive buffer.

TIMER/COUNTER SFR

Register pairs (TH0, TL0), and (TH1, TL1) are the 16-bit counting registers for Timer/Counters 0 and 2.

POWER CONTROL SFR (PCON)

The PCON Register (Table 13) controls the power down and idle modes in the UPI-452, as well as providing the ability to double the Serial Channel baud rate. There are also two general purpose flag bits available to the user. Bits 5 and 6 are used to set the HOLD/HOLD Acknowledge mode (see "General Purpose DMA Channels" section), and bit 4 is not used.

Table 12. Program Status Word

Symbolic Address									Physical Address
PSW	CY	AC	FO	RS1	RS0	OV	—	P	0D0H
	(MSB)							(LSB)	

Symbol	Position	Name
CY	PSW.7	Carry Flag
AC	PSW.6	Auxiliary Carry (For BCD operations)
F0	PSW.5	Flag 0 (user assignable)
RS1	PSW.4	Register Bank Select bit 1*
RS0	PSW.3	Register Bank Select bit 0*
OV	PSW.2	Overflow Flag
—	PSW.1	(reserved)
P	PSW.0	Parity Flag

*(RS1, RS0) enable internal RAM register banks as follows:

RS1	RS0	Internal RAM Register Bank
0	0	Bank 0
0	1	Bank 1
1	0	Bank 2
1	1	Bank 3

Table 13. PCON Special Function Register

Symbolic Address									Physical Address
PCON	SMOD	ARB	REQ	—	GF1	GF0	PD	IDL	087H
	(MSB)							(LSB)	

Symbol	Position	Function
SMOD	PCON7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either Mode 1, 2 or 3.
ARB	PCON6	HLD/HLDA Arbiter control bit *
REQ	PCON5	HLD/HLDA Requestor control bit *
—	PCON4	(reserved)
GF1	PCON3	General-purpose flag bit
GF0	PCON2	General-purpose flag bit
PD	PCON1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON0	Idle Mode bit. Setting this bit activates idle mode operation.

*See "Ext. Memory DMA" description.

NOTE:
If 1's are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (000X0000).

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C†
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin to V_{SS} .....−0.5V to V_{CC} + 0.5V
Voltage on V_{CC} to V_{SS} .....−0.5V to +6.5V
Power Dissipation.....1.0W**

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage (except XTAL1, RST)	2.0	V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	3.9	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage (Ports 1, 2, 3, 4)		0.45	V	I_{OL} = 1.6 mA (Note 1)
V_{OL1}	Output Low Voltage (except Ports 1, 2, 3, 4)		0.45	V	I_{OL} = 3.2 mA (Note 1)
V_{OH}	Output High Voltage (Ports 1, 2, 3, 4)	2.4		V	I_{OH} = −60 μA, V_{CC} = 5V ±10%
		0.9 V_{CC}		V	I_{OH} = −10 μA
V_{OH1}	Output High Voltage (except Ports 1, 2, 3, 4 and Host Interface (Slave) Port)	2.4		V	I_{OH} = −400 μA, V_{CC} = 5V ±10%
		0.9 V_{CC}		V	I_{OH} = −40 μA (Note 2)
V_{OH2}	Output High Voltage (Host Interface (Slave) Port)	2.4		V	I_{OH} = −400 μA, V_{CC} = 5V ±10%
		V_{CC} − 0.4		V	I_{OH} = −10 μA
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3, 4)		−50	μA	V_{IN} = 0.45V
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3, 4)		−650	μA	V_{IN} = 2V

6

D.C. CHARACTERISTICS T_A = 0°C to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V (Continued)

Symbol	Parameter	Min	Max	Units	Test Conditions
I_{LI}	Input Leakage Current (except Ports 1, 2, 3, 4)		±10	μA	0.45V < V_{IN} < V_{CC}
I_{OZ}	Output Leakage Current (except Ports 1, 2, 3, 4)		±10	μA	0.45V < V_{OUT} < V_{CC}
I_{CC}	Operating Current		50	mA	V_{CC} = 5.5V, 14 MHz (Note 4)
I_{CCI}	Idle Mode Current		25	mA	V_{CC} = 5.5V, 14 MHz (Note 5)
I_{PD}	Power Down Current		100	μA	V_{CC} = 2V (Note 3)
RRST	Reset Pulldown Resistor	50	150	KΩ	
CIO	Pin Capacitance		20	pF	1 MHz, T_A = 25°C (sampled, not tested on all parts)

NOTES:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OLS} of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
2. Capacitive loading on Ports 0 and 2 may cause the V_{OH} on ALE and PSEN to momentarily fall before the 0.9 V_{CC} specification when the address bits are stabilizing.
3. Power DOWN I_{CC} is measured with all output pins disconnected; EA = Port 0 = V_{CC}; XTAL2 N.C.; RST = V_{SS}; DB = V_{CC}; $\overline{WR}$ = $\overline{RD}$ = $\overline{DACK}$ = $\overline{CS}$ = A0 = A1 = A2 = V_{CC}. Power Down Mode is not supported on the 87C452P.
4. I_{CC} is measured with all output pins disconnected; XTAL1 driven with TCLCH, TCHCL = 5 ns, V_{IL} = V_{SS} + 0.5V, V_{IH} = V_{CC} − 0.5V; XTAL2 N.C.; EA = RST = Port 0 = V_{CC}; $\overline{WR}$ = $\overline{RD}$ = $\overline{DACK}$ = $\overline{CS}$ = A0 = A1 = A2 = V_{CC}. I_{CC} would be slightly higher if a crystal oscillator is used.
5. Idle I_{CC} is measured with all output pins disconnected; XTAL1 driven with TCLCH, TCHCL = 5 ns, V_{IL} = V_{SS} + 0.5V, V_{IH} = V_{CC} − 0.5V; XTAL2 N.C.; Port 0 = V_{CC}; EA = RST = V_{SS}; $\overline{WR}$ = $\overline{RD}$ = $\overline{DACK}$ = $\overline{CS}$ = A0 = A1 = A2 = V_{CC}.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for:

A: Address.

C: Clock.

D: Input data.

H: Logic level HIGH.

I: Instruction (program memory contents).

L: Logic level LOW, or ALE.

P: PSEN.

Q: Output data.

R: READ signal.

T: Time.

V: Valid.

W: WRITE signal.

X: No longer a valid logic level.

Z: Float.

EXAMPLE

TAVLL = Time for Address Valid to ALE Low.
TLLPL = Time for ALE Low to $\overline{PSEN}$ Low.

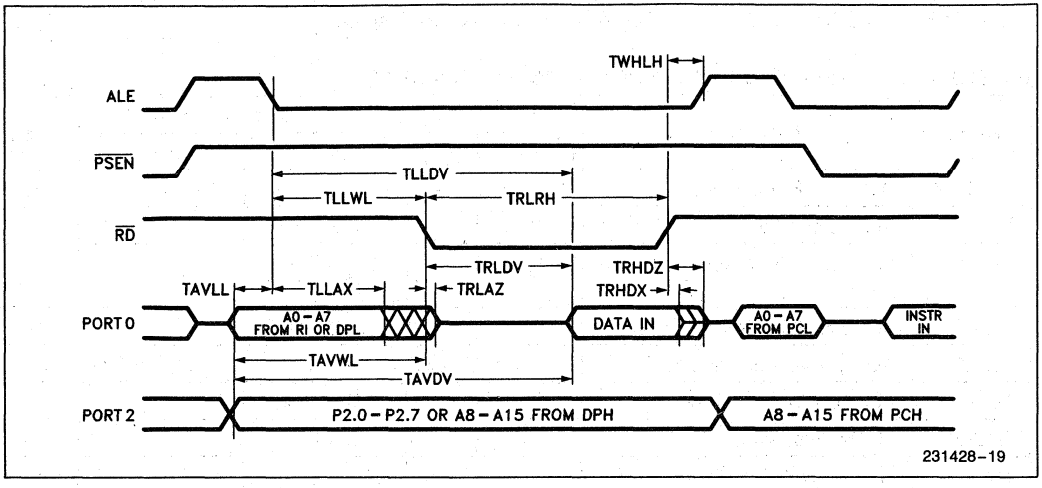
A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%, V_{SS} = 0V, Load Capacitance for Port 0, ALE, and PSEN = 100 pF, Load Capacitance for All Other Outputs = 80 pF

EXTERNAL PROGRAM AND DATA MEMORY CHARACTERISTICS

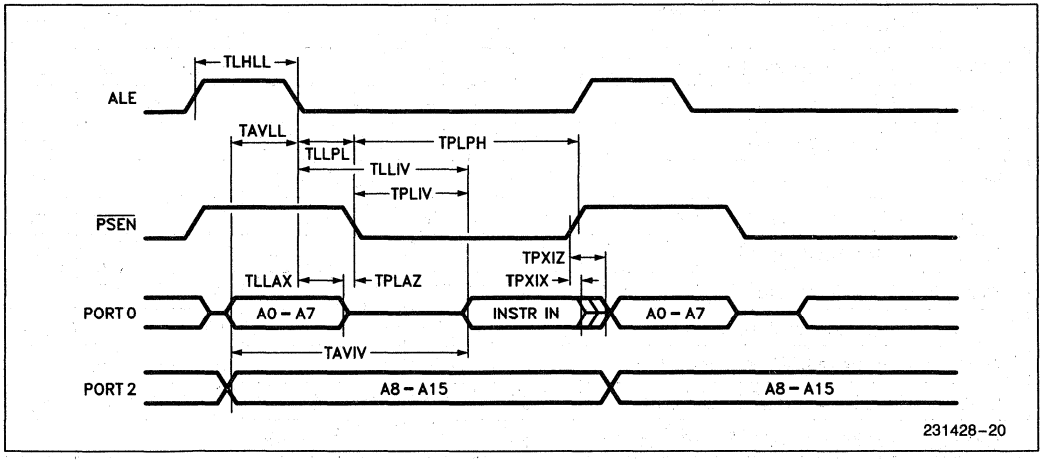
Symbol	Parameter	14 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency	3.5	14			MHz
TLHLL	ALE Pulse Width	103		2TCLCL−40		ns
TAVLL	Address Valid to ALE Low (Note 1)	25		TCLCL−55		ns
TLLAX	Address Hold after ALE Low	36		TCLCL−35		ns
TLLIV	ALE Low to Valid Instr In		185		4TCLCL−100	ns
TLLPL	ALE Low to $\overline{\text{PSEN}}$ Low	31		TCLCL−40		ns
TPLPH	$\overline{\text{PSEN}}$ Pulse Width	169		3TCLCL−45		ns
TPLIV	$\overline{\text{PSEN}}$ Low to Valid Instr In		110		3TCLCL−105	ns
TPXIX	Input Instr Hold after $\overline{\text{PSEN}}$	0		0		ns
TPXIZ	Input Instr Float after $\overline{\text{PSEN}}$ (Note 1)		57		TCLCL−25	ns
TAVIV	Address to Valid Instr In		252		5TCLCL−105	ns
TPLAZ	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
TRLRH	$\overline{\text{RD}}$ Pulse Width	329		6TCLCL−100		ns
TWLWH	$\overline{\text{WR}}$ Pulse Width	329		6TCLCL−100		ns
TRLDV	$\overline{\text{RD}}$ Low to Valid Data In		192		5TCLCL−165	ns
TRHDX	Data Hold after $\overline{\text{RD}}$	0		0		ns
TRHDZ	Data Float after $\overline{\text{RD}}$		73		2TCLCL−70	ns
TLLDV	ALE Low to Valid Data In		422		8TCLCL−150	ns
TAVDV	Address to Valid Data In		478		9TCLCL−165	ns
TLLWL	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	164	264	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address Valid to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	156		4TCLCL−130		ns
TQVWX	Data Valid to $\overline{\text{WR}}$ Transition	11		TCLCL−60		ns
TWHQX	Data Hold after $\overline{\text{WR}}$	21		TCLCL−50		ns
TRLAZ	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
TWHLH	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	31	111	TCLCL−40	TCLCL+40	ns
TQVWH	Data Valid to $\overline{\text{WR}}$ (Setup Time)	350		7TCLCL−150		ns

NOTE:
1. Use the value of 14 MHz specification or variable oscillator specification, whichever is greater.

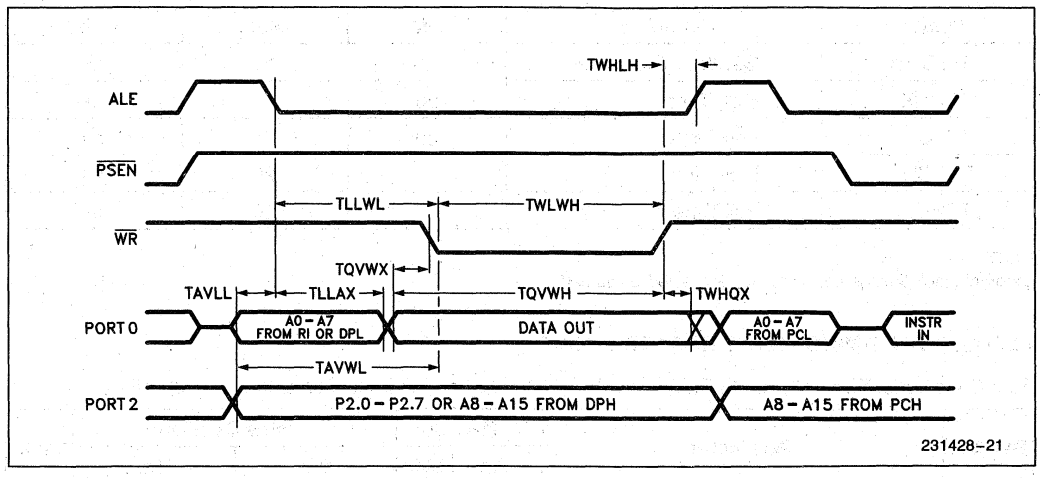
EXTERNAL DATA MEMORY READ CYCLE



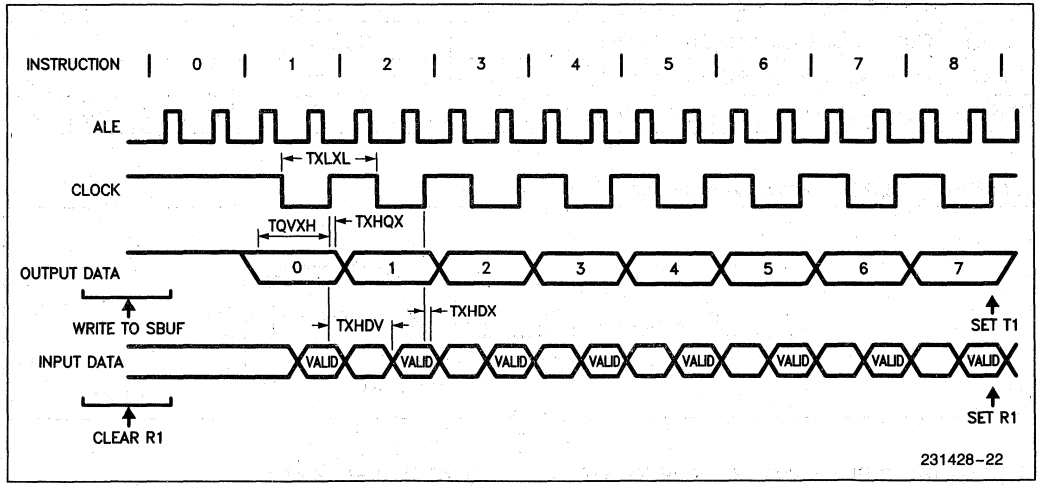
EXTERNAL PROGRAM MEMORY READ CYCLE



EXTERNAL DATA MEMORY WRITE CYCLE



SHIFT REGISTER MODE TIMING WAVEFORMS



6

EXTERNAL CLOCK DRIVE

Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency	3.5	14	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

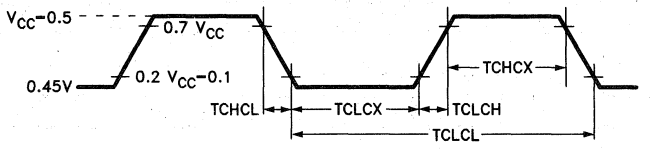
NOTE:
External clock timings are sampled, not tested on all parts.

SERIAL PORT TIMING—SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance = 80 pF

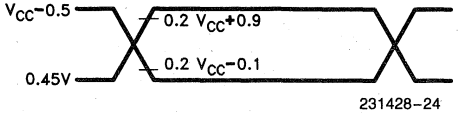
Symbol	Parameter	14 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	857		12TCLCL		ns
TQVXH	Output Data Setup to Clock Rising Edge	581		10TCLCL−133		ns
TXHQX	Output Data Hold after Clock Rising Edge	26		2TCLCL−117		ns
TXHDX	Input Data Hold after Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		581		10TCLCL−133	ns

EXTERNAL CLOCK DRIVE WAVEFORM



231428-23

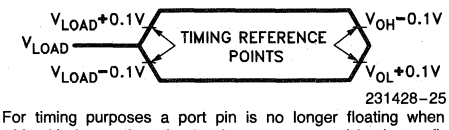
AC TESTING INPUT, OUTPUT WAVEFORMS



231428-24

AC inputs during testing are driven at V_{CC} −0.5V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at V_{IH} min. for a logic "1" and V_{IL} max. for a logic "0".

FLOAT WAVEFORMS

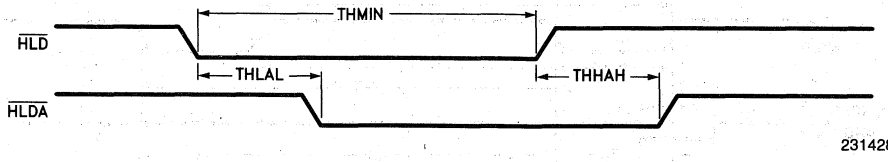


231428-25

For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

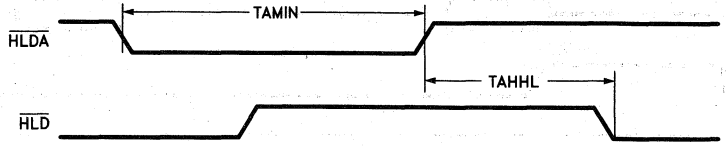
HLD/HLDA WAVEFORMS

Arbiter Mode



231428-26

Requestor Mode



231428-31

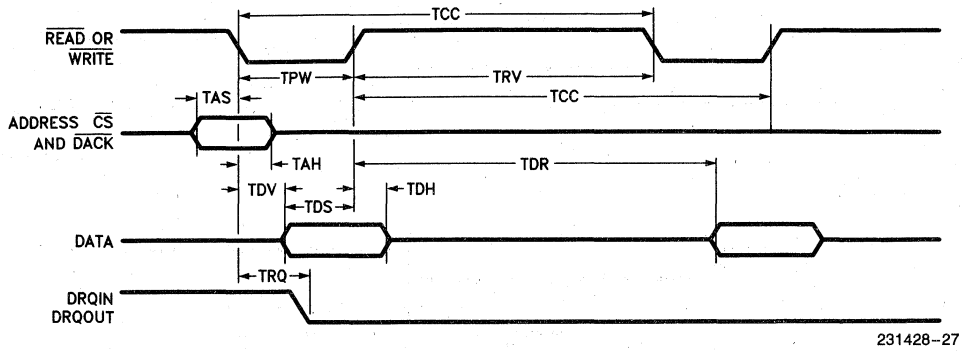
HLD/HLDA TIMINGS

Test Conditions: T_A = 0°C to +70°C; V_{CC} = 5V ±10%, V_{SS} = 0V; Load Capacitance = 80 pF

Symbol	Parameter	14 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
THMIN	$\overline{HLD}$ Pulse Width	386		4TCLCL+100		ns
THLAL	$\overline{HLD}$ to $\overline{HLDA}$ Delay if $\overline{HLDA}$ is Granted	186	672	4TCLCL−100	8TCLCL+100	ns
THHAH	HLD to HLDA Delay	186	672	4TCLCL−100	8TCLCL+100	ns
TAMIN	$\overline{HLDA}$ Pulse Width	386		4TCLCL+100		ns
TAHHL	$\overline{HLDA}$ Inactive to $\overline{HLD}$ Active	186		4TCLCL−100		ns

6

HOST PORT WAVEFORMS



HOST PORT TIMINGS

Test Conditions: T_A = 0°C to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance = 80 pF

Symbol	Parameter	14 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
TCC	Cycle Time	429		6TCLCL		ns
TPW	Command Pulse Width	100		100		ns
TRV	Recovery Time	60		60		ns
TAS	Address Setup Time	5		5		ns
TAH	Address Hold Time	30		30		ns
TDS	$\overline{WRITE}$ Data Setup Time	30		30		ns
TDHW	$\overline{WRITE}$ Data Hold Time	5		5		ns
TDHR	$\overline{READ}$ Data Hold Time	5	40	5	40	ns
TDV	$\overline{READ}$ Active to Read Data Valid Delay		92		92	ns
TDR	$\overline{WRITE}$ Inactive to Read Data Valid Delay (Applies only to Host Control SFR)		343		4.8TCLCL	ns
TRQ	$\overline{READ}$ or $\overline{WRITE}$ Active to DRQIN or DRQOUT Inactive Delay		150		150	ns

REVISION HISTORY

DOCUMENT: UPI-452 Data Sheet

OLD REVISION NUMBER: 231428-004

NEW REVISION NUMBER: 231428-005

1. Maximum Clock Rate was changed from 16 MHz to 14 MHz. This change is reflected in all Maximum Timing specifications.
2. The proper range of values for ITHR has been changed from [0 to (CBP-2)] to [0 to (CBP-3)] to ensure proper setting of the Input FIFO request for service bit. See the following sections: INPUT FIFO CHANNEL, and INPUT AND OUTPUT FIFO THRESHOLD SFR (ITHR & OTHR).
3. The proper range of values for OTHR has been changed from [1 to {(80H-CBP)-1}] to [2 to {(80-CBP)-1}] to ensure proper setting of the Output FIFO request for service bit. See the following sections: OUTPUT FIFO CHANNEL, FIFO-EXTERNAL HOST INTERFACE FIFO DMA FREEZE MODE, and INPUT AND OUTPUT FIFO THRESHOLD SFR (ITHR & OTHR).
4. The following D.C. Characteristics were deleted from the data sheet:

   V_{OH} = 0.75* V_{CC} @ I_{OH} = −25 μA,

   V_{OH1} = 0.75* V_{CC} @ I_{OH} = 150 μA,

   V_{OH2} = 3.0V @ I_{OH} = 1 mA, and

   I_{CC1} = 15 mA @ V_{CC} = 5.5V (87C452P).

   See D.C. CHARACTERISTICS TABLE.
5. The parameter descriptions for THHAH and THLAL has been reversed and their maximum specification for clock rates less than 14 MHz has been changed from [4TCLC + 100 ns] to [8TCLC + 100 ns]. See HLD/HLDA TIMINGS.
6. TAMIN specification has been removed from the Arbiter Mode waveform diagram and added to the Requestor Mode waveform diagram. See HLD/HLDA WAVEFORMS.

UPI™-41AH/42AH
UNIVERSAL PERIPHERAL INTERFACE
8-BIT SLAVE MICROCONTROLLER

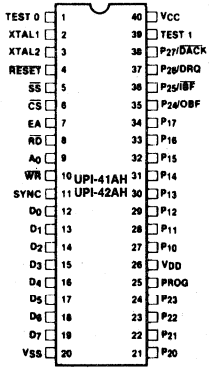
- **UPI-41: 6 MHz; UPI-42: 12.5 MHz**
- **Pin, Software and Architecturally Compatible with all UPI-41 and UPI-42 Products**
- **8-Bit CPU plus ROM/OTP™ EPROM, RAM, I/O, Timer/Counter and Clock in a Single Package**
- **2048 x 8 ROM/OTP, 256 x 8 RAM on UPI-42, 1024 x 8 ROM/OTP, 128 x 8 RAM on UPI-41, 8-Bit Timer/Counter, 18 Programmable I/O Pins**
- **One 8-Bit Status and Two Data Registers for Asynchronous Slave-to-Master Interface**
- **DMA, Interrupt, or Polled Operation Supported**
- **Fully Compatible with all Intel and Most Other Microprocessor Families**
- **Interchangeable ROM and OTP EPROM Versions**
- **Expandable I/O**
- **Sync Mode Available**
- **Over 90 Instructions: 70% Single Byte**
- **Available in EXPRESS — Standard Temperature Range**
- **int$_e$ligent Programming™ Algorithm — Fast OTP Programming**
- **Available in 40-Lead Plastic and 44-Lead Plastic Leaded Chip Carrier Packages**

(See Packaging Spec., Order #231369)

The Intel UPI-41AH and UPI-42AH are general-purpose Universal Peripheral Interfaces that allow the designer to develop customized solutions for peripheral device control.

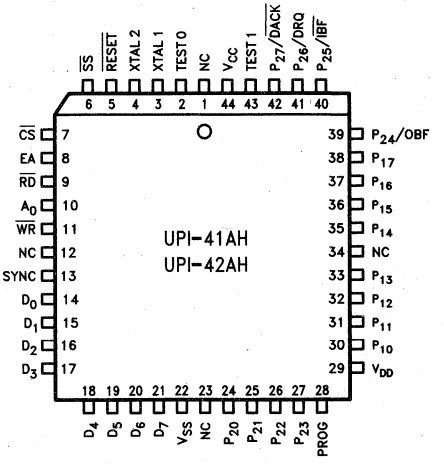
They are essentially "slave" microcontrollers, or microcontrollers with a slave interface included on the chip. Interface registers are included to enable the UPI device to function as a slave peripheral controller in the MCS™ Modules and iAPX family, as well as other 8-, 16-, and 32-bit systems.

To allow full user flexibility, the program memory is available in ROM and One-Time Programmable EPROM (OTP). All UPI-41AH and UPI-42AH devices are fully pin compatible for easy transition from prototype to production level designs.



210393-2

Figure 1. DIP Pin Configuration



210393-3

Figure 2. PLCC Pin Configuration

October 1990
Order Number: 210393-006

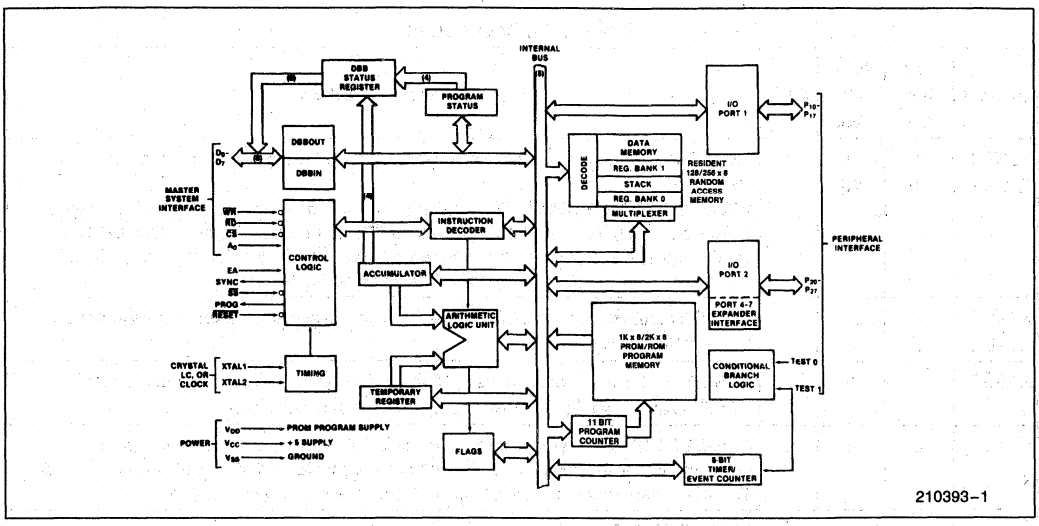


Figure 3. Block Diagram

UPI PRODUCT MATRIX

UPI Device	ROM	OTP™ EPROM	RAM	Programming Voltage
8042AH	2K	—	256	—
8242AH	2K	—	256	—
8742AH	—	2K	256	12.5V
8041AH	1K	—	128	—
8741AH	—	1K	128	12.5V

The 8242AH is an 8042AH programmed with Phoenix Technologies Ltd. keyboard controller firmware* for AT-compatible systems. The 8242AH supports all standard UPI-42AH features as well as the complete UPI-42AH command set.

The 8242AH keyboard controller firmware for AT-compatibles is fully compatible with all AT-compatible operating systems and applications.

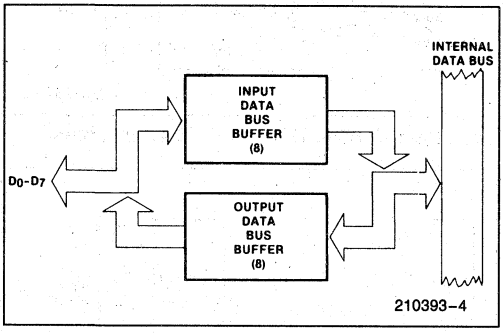
*Contact factory for current code revision available in the 8242AH.

Table 1. Pin Description

Symbol	DIP Pin No.	PLCC Pin No.	Type	Name and Function
TEST 0, TEST 1	1 39	2 43	I	**TEST INPUTS:** Input pins which can be directly tested using conditional branch instructions. **FREQUENCY REFERENCE:** TEST 1 (T_1) also functions as the event timer input (under software control). TEST 0 (T_0) is used during PROM programming and ROM/EPROM verification. It is also used during Sync Mode to reset the instruction state to S1 and synchronize the internal clock to PH1. See the Sync Mode Section.
XTAL 1, XTAL 2	2 3	3 4	I	**INPUTS:** Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
$\overline{\text{RESET}}$	4	5	I	**RESET:** Input used to reset status flip-flops and to set the program counter to zero. $\overline{\text{RESET}}$ is also used during EPROM programming and verification.
$\overline{\text{SS}}$	5	6	I	**SINGLE STEP:** Single step input used in conjunction with the SYNC output to step the program through each instruction (EPROM). This should be tied to +5V when not used. This pin is also used to put the device in Sync Mode by applying 12.5V to it.
$\overline{\text{CS}}$	6	7	I	**CHIP SELECT:** Chip select input used to select one UPI microcomputer out of several connected to a common data bus.
EA	7	8	I	**EXTERNAL ACCESS:** External access input which allows emulation, testing and ROM/EPROM verification. This pin should be tied low if unused.
$\overline{\text{RD}}$	8	9	I	**READ:** I/O read input which enables the master CPU to read data and status words from the OUTPUT DATA BUS BUFFER or status register.
A_0	9	10	I	**COMMAND/DATA SELECT:** Address Input used by the master processor to indicate whether byte transfer is data ($A_0 = 0$, F1 is reset) or command ($A_0 = 1$, F1 is set). $A_0 = 0$ during program and verify operations.
$\overline{\text{WR}}$	10	11	I	**WRITE:** I/O write input which enables the master CPU to write data and command words to the UPI INPUT DATA BUS BUFFER.
SYNC	11	13	O	**OUTPUT CLOCK:** Output signal which occurs once per UPI instruction cycle. SYNC can be used as a strobe for external circuitry; it is also used to synchronize single step operation.
D_0–D_7 (BUS)	12–19	14–21	I/O	**DATA BUS:** Three-state, bidirectional DATA BUS BUFFER lines used to interface the UPI microcomputer to an 8-bit master system data bus.
P_{10}–P_{17}	27–34	30–33 35–38	I/O	**PORT 1:** 8-bit, PORT 1 quasi-bidirectional I/O lines. P_{10}–P_{17} access the signature row and security bit.
P_{20}–P_{27}	21–24 35–38	24–27 39–42	I/O	**PORT 2:** 8-bit, PORT 2 quasi-bidirectional I/O lines. The lower 4 bits (P_{20}–P_{23}) interface directly to the 8243 I/O expander device and contain address and data information during PORT 4–7 access. The upper 4 bits (P_{24}–P_{27}) can be programmed to provide interrupt Request and DMA Handshake capability. Software control can configure P_{24} as Output Buffer Full (OBF) interrupt, P_{25} as Input Buffer Full ($\overline{\text{IBF}}$) interrupt, P_{26} as DMA Request (DRQ), and P_{27} as DMA ACKnowledge ($\overline{\text{DACK}}$).
PROG	25	28	I/O	**PROGRAM:** Multifunction pin used as the program pulse input during PROM programming. During I/O expander access the PROG pin acts as an address/data strobe to the 8243. This pin should be tied high if unused.
V_{CC}	40	44		**POWER:** +5V main power supply pin.
V_{DD}	26	29		**POWER:** +5V during normal operation. +12.5V during programming operation. Low power standby supply pin.
V_{SS}	20	22		**GROUND:** Circuit ground potential.

UPI-41AH and UPI-42AH FEATURES

1. Two Data Bus Buffers, one for input and one for output. This allows a much cleaner Master/Slave protocol.



2. 8 Bits of Status

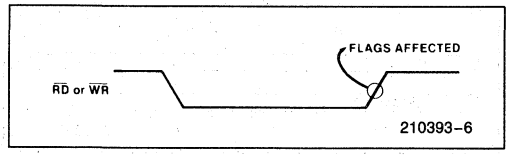
ST_7	ST_6	ST_5	ST_4	F_1	F_0	IBF	OBF
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

ST_4–ST_7 are user definable status bits. These bits are defined by the "MOV STS, A" single byte, single cycle instruction. Bits 4–7 of the acccumulator are moved to bits 4–7 of the status register. Bits 0–3 of the status register are not affected.

MOV STS, A Op Code: 90H

1	0	0	1	0	0	0	0
D_7							D_0

3. $\overline{RD}$ and $\overline{WR}$ are edge triggered. IBF, OBF, F_1 and $\overline{INT}$ change internally after the trailing edge of $\overline{RD}$ or $\overline{WR}$.

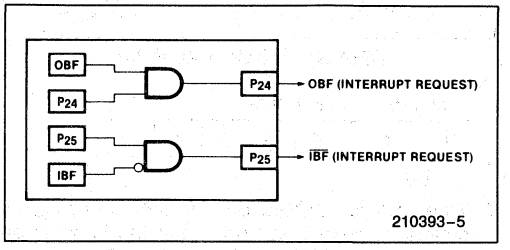


During the time that the host CPU is reading the status register, the UPI is prevented from updating this register or is 'locked out.'

4. P_{24} and P_{25} are port pins or Buffer Flag pins which can be used to interrupt a master processor. These pins default to port pins on Reset.

   If the "EN FLAGS" instruction has been executed, P_{24} becomes the OBF (Output Buffer Full) pin. A "1" written to P_{24} enables the OBF pin (the pin outputs the OBF Status Bit). A "0" written to P_{24} disables the OBF pin (the pin remains low). This pin can be used to indicate that valid data is available from the UPI (in Output Data Bus Buffer).

   If "EN FLAGS" has been executed, P_{25} becomes the $\overline{IBF}$ (Input Buffer Full) pin. A "1" written to P_{25} enables the $\overline{IBF}$ pin (the pin outputs the inverse of the IBF Status Bit. A "0" written to P_{25} disables the $\overline{IBF}$ pin (the pin remains low). This pin can be used to indicate that the UPI is ready for data.



Data Bus Buffer Interrupt Capability

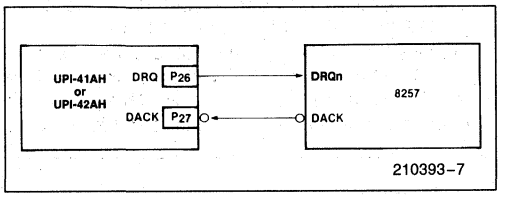
EN FLAGS Op Code: 0F5H

1	1	1	1	0	1	0	1
D_7							D_0

5. P_{26} and P_{27} are port pins or DMA handshake pins for use with a DMA controller. These pins default to port pins on Reset.

   If the "EN DMA" instruction has been executed, P_{26} becomes the DRQ (DMA Request) pin. A "1" written to P_{26} causes a DMA request (DRQ is activated). DRQ is deactivated by DACK•RD, DACK•WR, or execution of the "EN DMA" instruction.

   If "EN DMA" has been executed, P_{27} becomes the $\overline{DACK}$ (DMA ACKnowledge) pin. This pin acts as a chip select input for the Data Bus Buffer registers during DMA transfers.



DMA Handshake Capability

EN DMA Op Code: 0E5H

D_7							D_0
1	1	1	0	0	1	0	1

6. When EA is enabled on the UPI, the program counter is placed on Port 1 and the lower three bits of Port 2 (MSB = P_{22}, LSB = P_{10}). On the UPI this information is multiplexed with PORT DATA (see port timing diagrams at end of this data sheet).
7. The 8741AH and 8742AH support the intelligent Programming Algorithm. (See the Programming Section.)

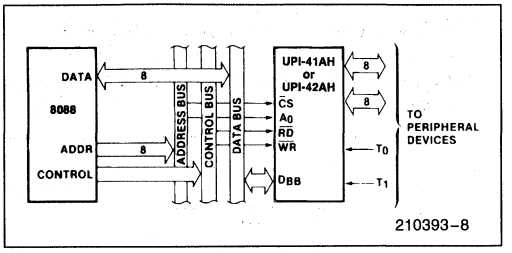


Figure 5. 8088-UPI-41AH/42AH Interface

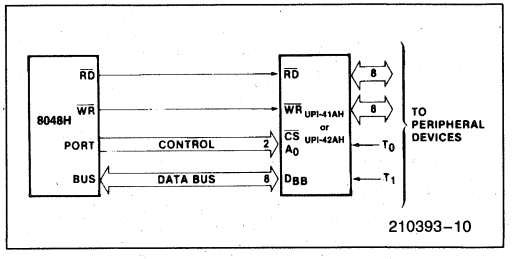


Figure 6. 8048H-UPI-41/42 Interface

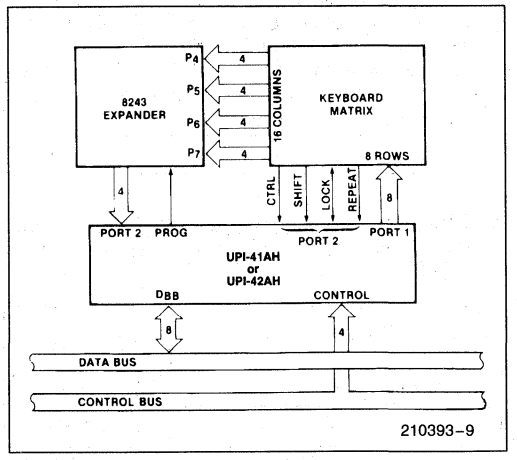


Figure 7. UPI-41/42-8243 Keyboard Scanner

APPLICATIONS

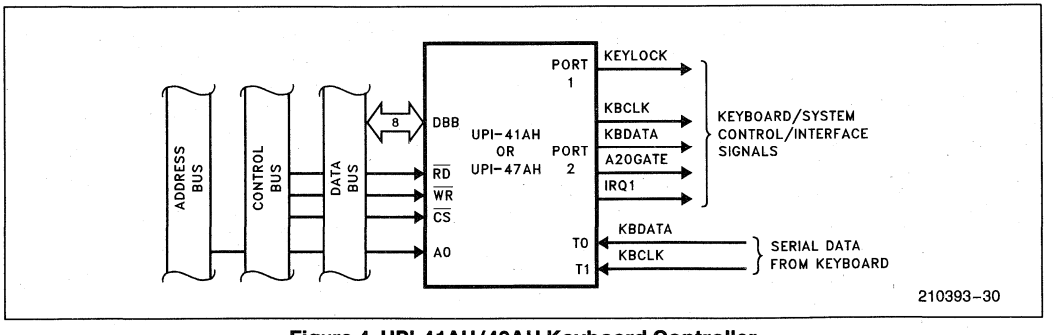


Figure 4. UPI-41AH/42AH Keyboard Controller

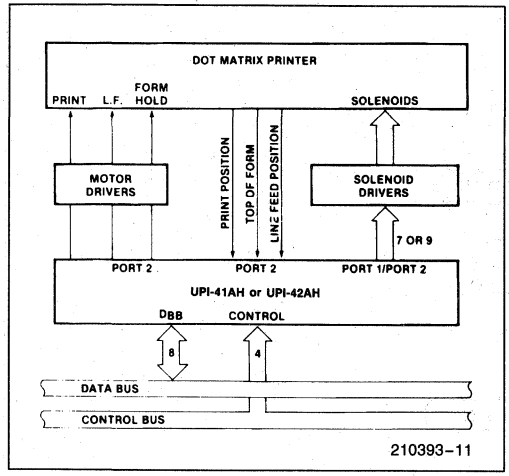


Figure 8. UPI-41AH/42AH 80-Column Matrix Printer Interface

PROGRAMMING AND VERIFYING THE 8741AH AND 8742AH OTP™ EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	2 Clock Inputs
$\overline{\text{Reset}}$	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Signature Row/Security Bit Modes
BUS	Address and Data Input Data Output During Verify
P_{20-22}	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING
An attempt to program a missocketed 8741AH or 8742AH will result in severe damage to the part. An indication of a properly socketed part is the appearance of the SYNC clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1. CS = 5V, V_{CC} = 5V, V_{DD} = 5V, RESET = 0V, A_0 = 0V, TEST 0 = 5V, clock applied or internal oscillator operating, BUS floating, PROG = 5V.
2. Insert 8741AH or 8742AH in programming socket
3. TEST 0 = 0V (select program mode)
4. EA = 12.5V (active program mode)
5. V_{CC} = 6V (programming supply)
6. V_{DD} = 12.5V (programming power)
7. Address applied to BUS and P_{20-22}
8. $\overline{\text{RESET}}$ = 5V (latch address)
9. Data applied to BUS
10. PROG = 5V followed by one 1 ms pulse to 0V
11. TEST 0 = 5V (verify mode)
12. Read and verify data on BUS
13. TEST 0 = 0V
14. Apply overprogram pulse
15. $\overline{\text{RESET}}$ = 0V and repeat from step 6
16. Programmer should be at conditions of step 1 when 8741AH or 8742AH is removed from socket

Please follow the int$_e$ligent Programming flow chart for proper programming procedure.

int$_e$ligent Programming™ Algorithm

The int$_e$ligent Programming Algorithm rapidly programs Intel 8741AH/8742AH EPROMs using an efficient and reliable method particularly suited to the production programming environment. Typical programming time for individual devices is on the order of 10 seconds. Programming reliability is also ensured as the incremental program margin of each byte is continually monitored to determine when it has been successfully programmed. A flowchart of the 8741AH/8742AH int$_e$ligent Programming Algorithm is shown in Figure 9.

The int$_e$ligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial PROG pulse(s) is one millisecond, which will then be followed by a longer overprogram pulse of length 3X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular 8741AH/8742AH location, before a correct verify occurs. Up to 25 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

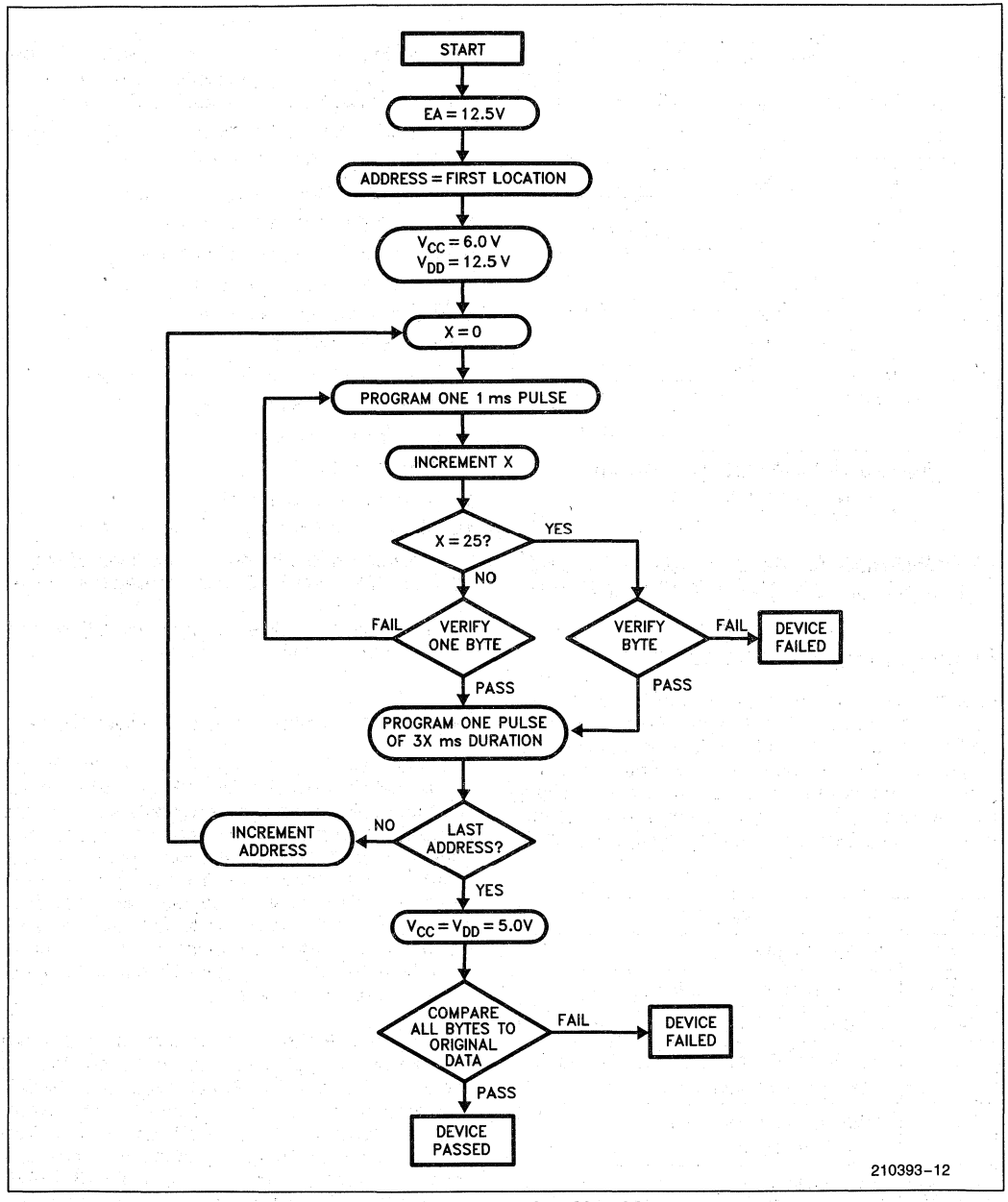


Figure 9. Programming Algorithm

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V and V_{DD} = 12.5V. When the int$_e$ligent Programming cycle has been completed, all bytes should be compared to the original data with V_{CC} = 5.0, V_{DD} = 5V.

Verify

A verify should be performed on the programmed bits to determine that they have been correctly programmed. The verify is performed with T0 = 5V, V_{DD} = 5V, EA = 12.5V, $\overline{SS}$ = 5V, PROG = 5V, A0 = 0V, and $\overline{CS}$ = 5V.

SECURITY BIT

The security bit is a single EPROM cell outside the EPROM array. The user can program this bit with the appropriate access code and the normal programming procedure, to inhibit any external access to the EPROM contents. Thus the user's resident program is protected. There is no direct external access to this bit. However, the security byte in the signature row has the same address and can be used to check indirectly whether the security bit has been programmed or not. The security bit has no effect on the signature mode, so the security byte can always be examined.

SECURITY BIT PROGRAMMING/ VERIFICATION

Programming

a. Read the security byte of the signature mode. Make sure it is 00H.

b. Apply access code to appropriate inputs to put the device into security mode.

c. Apply high voltage to EA and V_{DD} pins.

d. Follow the programming procedure as per the int$_e$ligent Programming Algorithm with known data on the databus. Not only the security bit, but also the security byte of the signature row is programmed.

e. Verify that the security byte of the signature mode contains the same data as appeared on the data bus. (If DB0–DB7 = high, the security byte will contain FFH.)

f. Read two consecutive known bytes from the EPROM array and verify that the wrong data are retrieved in at least one verification. If the EPROM can still be read, the security bit may have not been fully programmed though the security byte in the signature mode has.

Verification

Since the security bit address overlaps the address of the security byte of the signature mode, it can be used to check indirectly whether the security bit has been programmed or not. Therefore, the security bit verification is a mere read operation of the security byte of the signature row (0FFH = security bit programmed; 00H = security bit unprogrammed). Note that during the security bit programming, the reading of the security byte does not necessarily indicate that the security bit has been successfully programmed. Thus, it is recommended that two consecutive known bytes in the EPROM array be read and the wrong data should be read at least once, because it is highly improbable that random data coincides with the correct ones twice.

SIGNATURE MODE

The UPI-41AH/42AH has an additional 32 bytes of EPROM available for Intel and user signatures and miscellaneous purposes. The 32 bytes are partitioned as follows:

A. **Test code/checksum**—This can accommodate up to 25 bytes of code for testing the internal nodes that are not testable by executing from the external memory. The test code/checksum is present on ROMs, and OTPs.

B. **Intel signature**—This allows the programmer to read from the UPI-41AH/42AH the manufacturer of the device and the exact product name. It facilitates automatic device identification and will be present in the ROM and OTP versions. Location 10H contains the manufacturer code. For Intel, it is 89H. Location 11H contains the device code. The code is 43H and 42H for the 8042AH and OTP 8742AH, and 41H and 40H for the 8041AH and OTP 8741AH, respectively. The code is 44H for any device with the security bit set by Intel.

C. **User signature**—The user signature memory is implemented in the EPROM and consists of 2 bytes for the customer to program his own signature code (for identification purposes and quick sorting of previously programmed materials).

D. **Test signature**—This memory is used to store testing information such as: test data, bin number, etc. (for use in quality and manufacturing control).

E. **Security byte**—This byte is used to check whether the security bit has been programmed (see the security bit section).

The signature mode can be accessed by setting P10 = 0, P11–P17 = 1, and then following the programming and/or verification procedures. The location of the various address partitions are as follows:

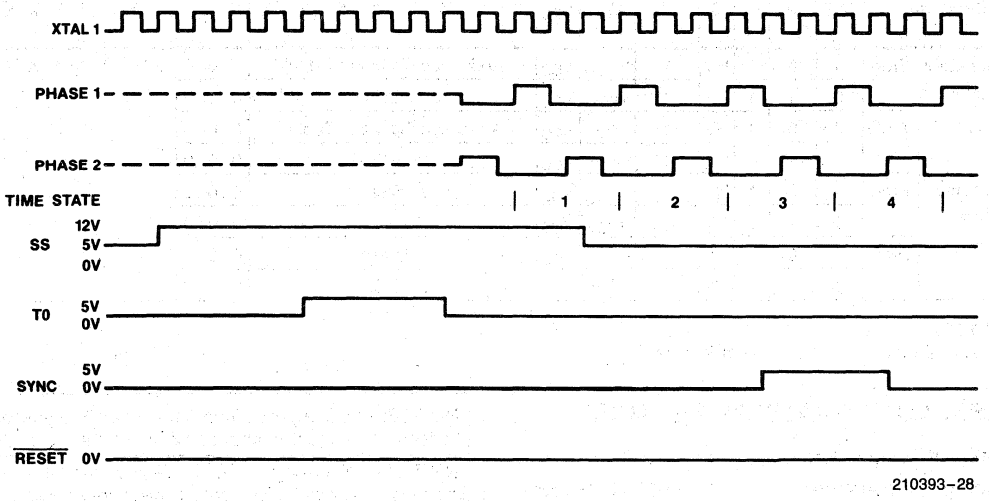
	Address		Device Type	No. of Bytes
Test Code/Checksum	0 16H	0FH 1EH	ROM/OTP	25
Intel Signature	10H	11H	ROM/OTP	2
User Signature	12H	13H	OTP	2
Test Signature	14H	15H	ROM/OTP	2
Security Byte	1FH		OTP	1

SYNC MODE

The Sync Mode is provided to ease the design of multiple controller circuits by allowing the designer to force the device into known phase and state time. The Sync Mode may also be utilized by automatic test equipment (ATE) for quick, easy, and efficient synchronizing between the tester and the DUT (device under test).

Sync Mode is enabled when $\overline{SS}$ pin is raised to high voltage level of +12 volts. To begin synchronization, T0 is raised to 5 volts at least four clock cycles after $\overline{SS}$. T0 must be high for at least four X1 clock cycles to fully reset the prescaler and time state generators. T0 may then be brought down during low state of X1. Two clock cycles later, with the rising edge of X1, the device enters into Time State 1, Phase 1. $\overline{SS}$ is then brought down to 5 volts 4 clocks later after T0. RESET is allowed to go high 5 tCY (75 clocks) later for normal execution of code.

SYNC MODE TIMING DIAGRAMS



210393-28

Minimum Specifications
SYNC Operation Time, t_{SYNC} = 3.5 XTAL 1 Clock cycles. Reset Time, t_{RS} = 4 t_{CY}.

NOTE:
The rising and falling edges of T0 should occur during low state of XTAL1 clock.

6

ACCESS CODE

The following table summarizes the access codes required to invoke the Sync Mode, Signature Mode, and the Security Bit, respectively. Also, the programming and verification modes are included for comparison.

Modes		Control Signals							Data Bus								Port 2	Access Code: Port 1							
		T0	RST	SS	EA	PROG	V_{DD}	V_{CC}	0	1	2	3	4	5	6	7	0 1 2	0	1	2	3	4	5	6	7
Programming Mode		0	0	1	HV	1	V_{DDH}	V_{CC}	Address								Addr	a_0	a_1	X	X	X	X	X	X
		0	1	1	HV	STB	V_{DDH}	V_{CC}	Data In								Addr								
Verification Mode		0	0	1	HV	1	V_{CC}	V_{CC}	Address								Addr	a_0	a_1	X	X	X	X	X	X
		1	1	1	HV	1	V_{CC}	V_{CC}	Data Out								Addr								
Sync Mode		STB High	0	HV	0	X	V_{CC}	V_{CC}	X	X	X	X	X	X	X	X	X X X	X	X	X	X	X	X	X	X
Signature Mode	Prog	0	0	1	HV	1	V_{DDH}	V_{CC}	Addr. (see Sig Mode Table)								0 0 0	0	1	1	1	1	X	X	1
		0	1	1	HV	STB	V_{DDH}	V_{CC}	Data In								0 0 0								
	Verify	0	0	1	HV	1	V_{CC}	V_{CC}	Addr. (see Sig Mode Table)								0 0 0								
		1	1	1	HV	1	V_{CC}	V_{CC}	Data Out								0 0 0								
Security Bit/Byte	Prog	0	0	1	HV	1	V_{DDH}	V_{CC}	Address								0 0 0								
		0	1	1	HV	STB	V_{DDH}	V_{CC}	Data In								0 0 0								
	Verify	0	0	1	HV	1	V_{CC}	V_{CC}	Address								0 0 0								
		1	1	1	HV	1	V_{CC}	V_{CC}	Data Out								0 0 0								

NOTES:
1. a_0 = 0 or 1; a_1 = 0 or 1. a_0 must = a_1.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . 0°C to +70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on Any Pin with Respect to Ground . . . . . −0.5V to +7V
Power Dissipation . . . . . 1.5 W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = V_{DD} = +5V ±10%

Symbol	Parameter	UPI-41AH/42AH		Units	Notes
		Min	Max		
V_{IL}	Input Low Voltage (Except XTAL1, XTAL2, RESET)	−0.5	0.8	V	
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, RESET)	−0.5	0.6	V	
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, RESET)	2.0	V_{CC}	V	
V_{IH1}	Input High Voltage (XTAL1, RESET)	3.5	V_{CC}	V	
V_{IH2}	Input High Voltage (XTAL2)	2.2	V_{CC}	V	
V_{OL}	Output Low Voltage (D_0–D_7)		0.45	V	I_{OL} = 2.0 mA

D.C. CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = V_{DD} = +5V \pm 10\%$ (Continued)

Symbol	Parameter	UPI-41AH/42AH		Units	Notes
		Min	Max		
V_{OL1}	Output Low Voltage ($P_{10}P_{17}$, $P_{20}P_{27}$, Sync)		0.45	V	$I_{OL} = 1.6$ mA
V_{OL2}	Output Low Voltage (PROG)		0.45	V	$I_{OL} = 1.0$ mA
V_{OH}	Output High Voltage (D_0–D_7)	2.4		V	$I_{OH} = -400\ \mu A$
V_{OH1}	Output High Voltage (All Other Outputs)	2.4			$I_{OH} = -50\ \mu A$
I_{IL}	Input Leakage Current (T_0, T_1, RD, WR, CS, A_0, EA)		±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{OFL}	Output Leakage Current (D_0–D_7, High Z State)		±10	μA	$V_{SS} + 0.45 \le V_{OUT} \le V_{CC}$
I_{LI}	Low Input Load Current ($P_{10}P_{17}$, $P_{20}P_{27}$)		0.3	mA	$V_{IL} = 0.8V$
I_{LI1}	Low Input Load Current (RESET, SS)		0.2	mA	$V_{IL} = 0.8V$
I_{DD}	V_{DD} Supply Current		20	mA	Typical = 8 mA
$I_{CC} + I_{DD}$	Total Supply Current		135	mA	Typical = 80 mA
I_{DD} Standby	Power Down Supply Current		20	mA	Typical = 8 mA
I_{IH}	Input Leakage Current (P_{10}–P_{17}, P_{20}–P_{27})		100	μA	$V_{IN} = V_{CC}$
C_{IN}	Input Capacitance		10	pF	$T_A = 25°C$ (1)
C_{IO}	I/O Capacitance		20	pF	$T_A = 25°C$ (1)

NOTE:
1. Sampled, not 100% tested.

D.C. CHARACTERISTICS—PROGRAMMING

$T_A = 25°C \pm 5°C$, $V_{CC} = 6V \pm 0.25V$, $V_{DD} = 12.5V \pm 0.5V$

Symbol	Parameter	Min	Max	Units
V_{DDH}	V_{DD} Program Voltage High Level	12	13	V(1)
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V
V_{PH}	PROG Program Voltage High Level	2.0	5.5	V
V_{PL}	PROG Voltage Low Level	−0.5	0.8	V
V_{EAH}	Input High Voltage for EA	12.0	13.0	V(2)
V_{EAL}	EA Voltage Low Level	−0.5	5.25	V
I_{DD}	V_{DD} High Voltage Supply Current		50.0	mA
I_{EA}	EA High Voltage Supply Current		1.0	mA

NOTES:
1. Voltages over 13V applied to pin V_{DD} will permanently damage the device.
2. V_{EAH} must be applied to EA before V_{DDH} and removed after V_{DDL}.
3. V_{CC} must be applied simultaneously or before V_{DD} and must be removed simultaneously or after V_{DD}.

A.C. CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{SS} = 0V$, $V_{CC} = V_{DD} = +5V \pm 10\%$

DBB READ

Symbol	Parameter	Min	Max	Units
t_{AR}	CS, A_0 Setup to RD ↓	0		ns
t_{RA}	CS, A_0 Hold After RD ↑	0		ns
t_{RR}	RD Pulse Width	160		ns
t_{AD}	CS, A_0 to Data Out Delay		130	ns
t_{RD}	RD ↓ to Data Out Delay	0	130	ns
t_{DF}	RD ↑ to Data Float Delay		85	ns

DBB WRITE

Symbol	Parameter	Min	Max	Units
t_{AW}	CS, A_0 Setup to WR ↓	0		ns
t_{WA}	CS, A_0 Hold After WR ↑	0		ns
t_{WW}	WR Pulse Width	160		ns
t_{DW}	Data Setup to WR ↑	130		ns
t_{WD}	Data Hold After WR ↑	0		ns

CLOCK

Symbol	Parameter	Min	Max	Units
t_{CY} (UPI-41AH/42AH)	Cycle Time	1.2	9.20	μs(1)
t_{CYC} (UPI-41AH/42AH)	Clock Period	80	613	ns
t_{PWH}	Clock High Time	30		ns
t_{PWL}	Clock Low Time	30		ns
t_R	Clock Rise Time		10	ns
t_F	Clock Fall Time		10	ns

NOTE:
1. $t_{CY} = 15/f(XTAL)$

A.C. CHARACTERISTICS DMA

Symbol	Parameter	Min	Max	Units
t_{ACC}	DACK to WR or RD	0		ns
t_{CAC}	RD or WR to DACK	0		ns
t_{ACD}	DACK to Data Valid	0	130	ns
t_{CRQ}	RD or WR to DRQ Cleared		110	ns(1)

NOTE:
1. $C_L = 150$ pF.

A.C. CHARACTERISTICS—PROGRAMMING

$T_A = 25°C \pm 5°C$, $V_{CC} = 6V \pm 0.25V$, $V_{DDL} = +5V \pm 0.25V$, $V_{DDH} = 12.5V \pm 0.5V$
(8741AH/8742AH ONLY)

Symbol	Parameter	Min	Max	Units
t_{AW}	Address Setup Time to RESET ↑	$4t_{CY}$		
t_{WA}	Address Hold Time After RESET ↑	$4t_{CY}$		
t_{DW}	Data in Setup Time to PROG ↓	$4t_{CY}$		
t_{WD}	Data in Hold Time After PROG ↑	$4t_{CY}$		
t_{PW}	Initial Program Pulse Width	0.95	1.05	ms(1)
t_{TW}	Test 0 Setup Time for Program Mode	$4t_{CY}$		
t_{WT}	Test 0 Hold Time After Program Mode	$4t_{CY}$		
t_{DO}	Test 0 to Data Out Delay		$4t_{CY}$	
t_{WW}	RESET Pulse Width to Latch Address	$4t_{CY}$		
t_r, t_f	PROG Rise and Fall Times	0.5	100	μs
t_{CY}	CPU Operation Cycle Time	2.5	3.75	μs
t_{RE}	RESET Setup Time Before EA ↑	$4t_{CY}$		
t_{OPW}	Overprogram Pulse Width	2.85	78.75	ms(2)
t_{DE}	EA High to V_{DD} High	$1t_{CY}$		

NOTES:
1. Typical Initial Program Pulse width tolerance = 1 ms ±5%.
2. This variation is a function of the iteration counter value, X.
3. If TEST 0 is high, t_{DO} can be triggered by RESET ↑.

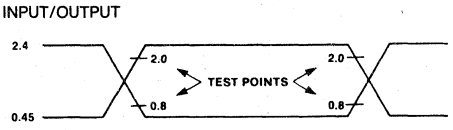
A.C. CHARACTERISTICS PORT 2 $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 10\%$

Symbol	Parameter	$f(t_{CY})$(3)	Min	Max	Units
t_{CP}	Port Control Setup Before Falling Edge of PROG	$1/15\ t_{CY} - 28$	55		ns(1)
t_{PC}	Port Control Hold After Falling Edge of PROG	$1/10\ t_{CY}$	125		ns(2)
t_{PR}	PROG to Time P2 Input Must Be Valid	$8/15\ t_{CY} - 16$		650	ns(1)
t_{PF}	Input Data Hold Time		0	150	ns(2)
t_{DP}	Output Data Setup Time	$2/10\ t_{CY}$	250		ns(1)
t_{PD}	Output Data Hold Time	$1/10\ t_{CY} - 80$	45		ns(2)
t_{PP}	PROG Pulse Width	$6/10\ t_{CY}$	750		ns

NOTES:
1. $C_L = 80$ pF.
2. $C_L = 20$ pF.
3. $t_{CY} = 1.25$ μs.

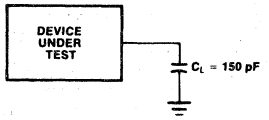
6

A.C. TESTING INPUT/OUTPUT WAVEFORM



210393-14

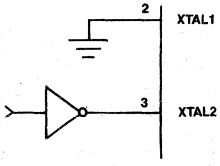
A.C. TESTING LOAD CIRCUIT



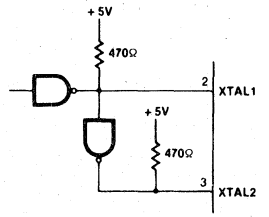
210393-15

DRIVING FROM EXTERNAL SOURCE-TWO OPTIONS

>6 MHz



210393-16

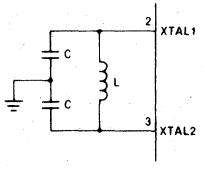


210393-17

Rise and Fall Times Should Not Exceed 10 ns. Resistors to V_{CC} are Needed to Ensure V_{IH} = 3.5V if TTL Circuitry is Used.

LC OSCILLATOR MODE

L	C	NOMINAL
45 H	20 pF	5.2 MHz
120 H	20 pF	3.2 MHz



$$f = \frac{1}{2\pi\sqrt{LC'}}$$

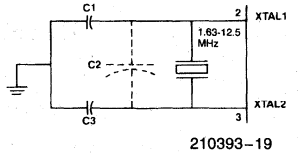
$$C' = \frac{C + 3Cpp}{2}$$

Cpp ≅ 5–10 pF
Pin-to-Pin Capacitance

210393-18

Each C Should be Approximately 20 pF, including Stray Capacitance.

CRYSTAL OSCILLATOR MODE



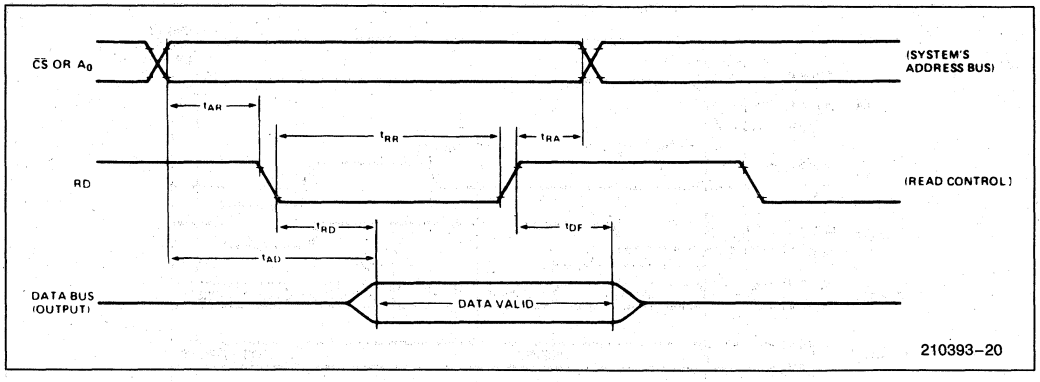
210393-19

C1 5 pF (STRAY 5 pF)
C2 (CRYSTAL + STRAY) 8 pF
C3 20–30 pF INCLUDING STRAY

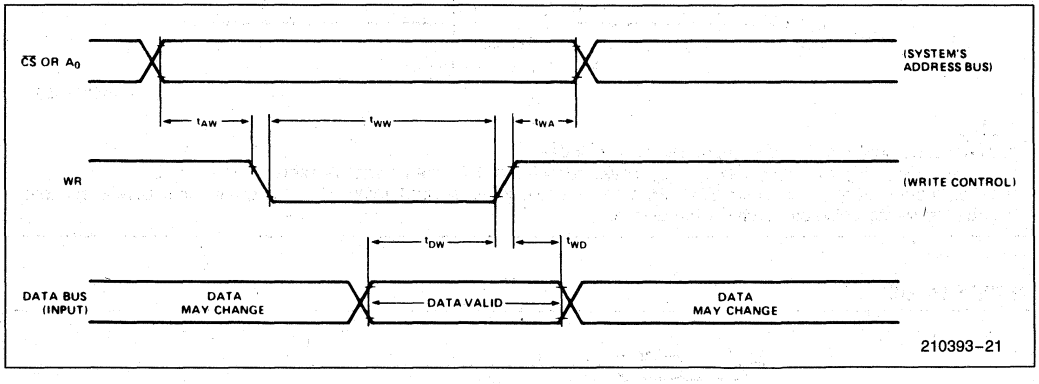
Crystal Series Resistance Should be Less Than 30Ω at 12.5 MHz.

WAVEFORMS

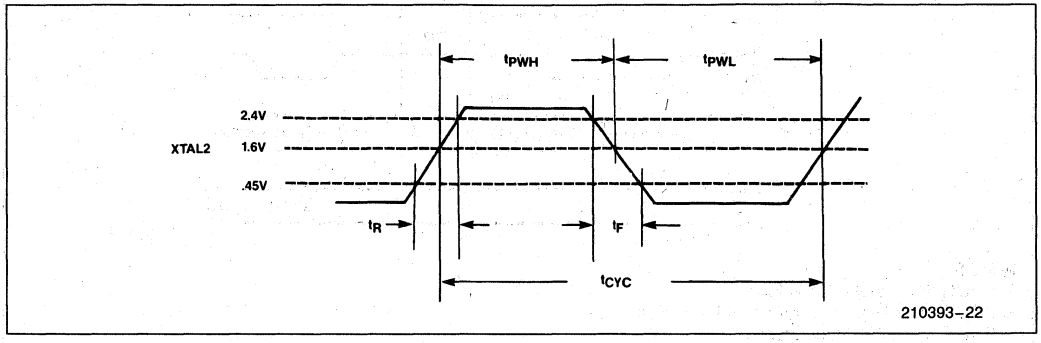
READ OPERATION—DATA BUS BUFFER REGISTER



WRITE OPERATION—DATA BUS BUFFER REGISTER

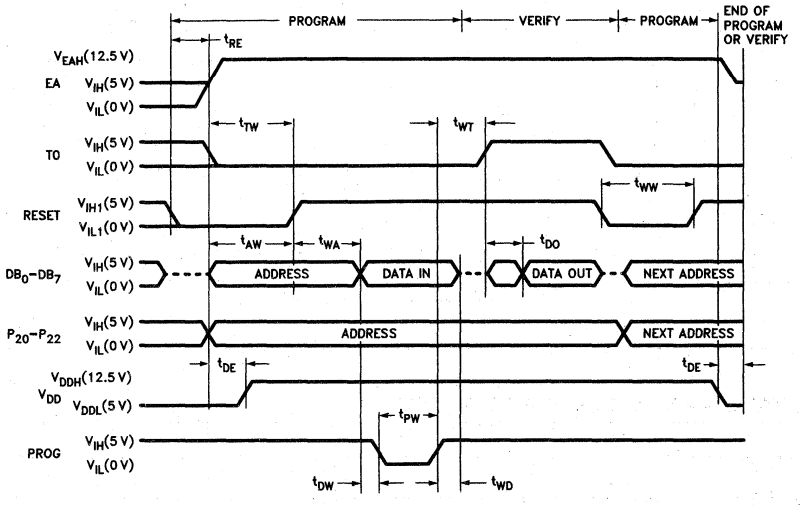


CLOCK TIMING



WAVEFORMS (Continued)

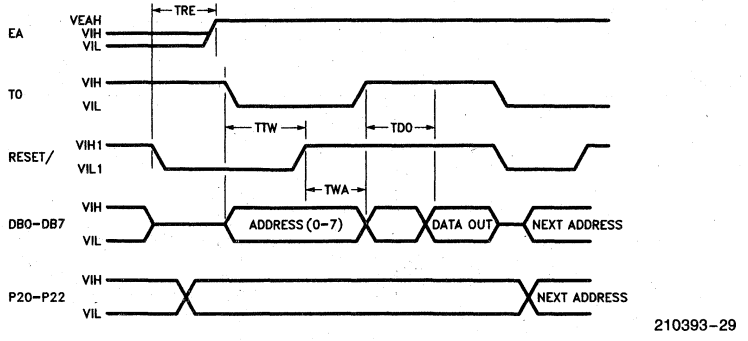
COMBINATION PROGRAM/VERIFY MODE



210393-23

NOTES:
1. A_9 must be held low (0V) during program/verify modes.
2. For V_{IH}, V_{IH1}, V_{IL}, V_{IL1}, V_{DDH}, and V_{DDL}, please consult the D.C. Characteristics Table.
3. When programming the 8741AH/8742AH, a 0.1 μF capacitor is required across V_{DD} and ground to suppress spurious voltage transients which can damage the device.

VERIFY MODE

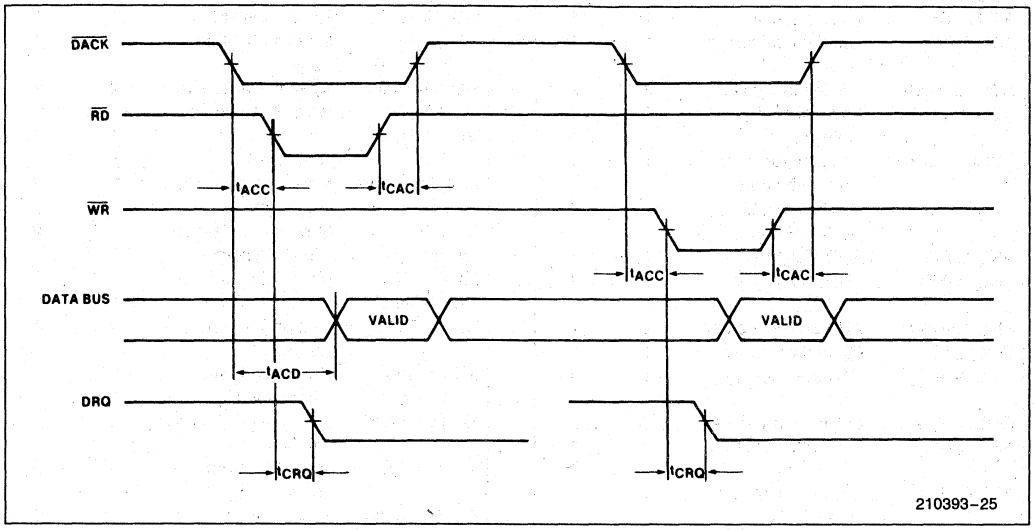


210393-29

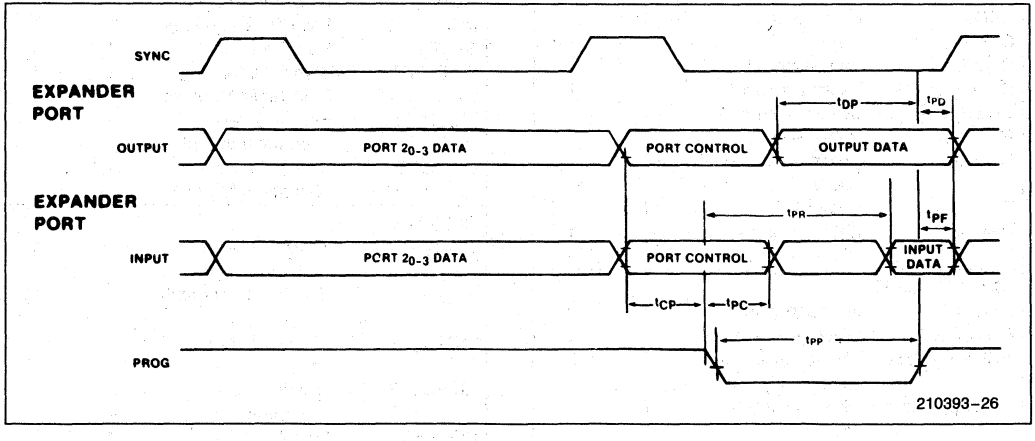
NOTES:
1. PROG must float if EA is low.
2. PROG must float or = 5V when EA is high.
3. P_{10}–P_{17} = 5V or must float.
4. P_{24}–P_{27} = 5V or must float.
5. A_9 must be held low during programming/verify modes.

WAVEFORMS (Continued)

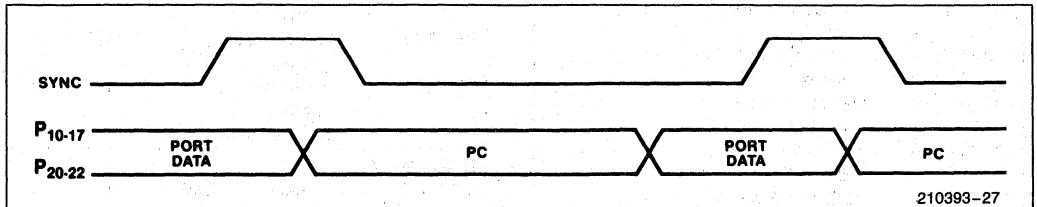
DMA



PORT 2



PORT TIMING DURING EXTERNAL ACCESS (EA)



On the Rising Edge of SYNC and EA is Enabled, Port Data is Valid and can be Strobed. On the Trailing Edge of Sync the Program Counter Contents are Available.

Table 2. UPI™ Instruction Set

Mnemonic	Description	Bytes	Cycles
ACCUMULATOR			
ADD A, Rr	Add register to A	1	1
ADD A, @Rr	Add data memory to A	1	1
ADD A, #data	Add immediate to A	2	2
ADDC A, Rr	Add register to A with carry	1	1
ADDC A, @Rr	Add data memory to A with carry	1	1
ADDC A, #data	Add immediate to A with carry	2	2
ANL A, Rr	AND register to A	1	1
ANL, A @Rr	AND data memory to A	1	1
ANL A, #data	AND immediate to A	2	2
ORL A, Rr	OR register to A	1	1
ORL, A, @Rr	OR data memory to A	1	1
ORL A, #data	OR immediate to A	2	2
XRL A, Rr	Exclusive OR register to A	1	1
XRL A, @Rr	Exclusive OR data memory to A	1	1
XRL A, #data	Exclusive OR immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal Adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
INPUT/OUTPUT			
IN A, Pp	Input port to A	1	2
OUTL Pp, A	Output A to port	1	2
ANL Pp, #data	AND immediate to port	2	2
ORL Pp, #data	OR immediate to port	2	2
IN A, DBB	Input DBB to A, clear IBF	1	1
OUT DBB, A	Output A to DBB, set OBF	1	1
MOV STS, A	A_4–A_7 to Bits 4–7 of Status	1	1
MOVD A, Pp	Input Expander port to A	1	2
MOVD Pp, A	Output A to Expander port	1	2
ANLD Pp, A	AND A to Expander port	1	2
ORLD Pp, A	OR A to Expander port	1	2

Mnemonic	Description	Bytes	Cycles
DATA MOVES			
MOV A, Rr	Move register to A	1	1
MOV A, @Rr	Move data memory to A	1	1
MOV A, #data	Move immediate to A	2	2
MOV Rr, A	Move A to register	1	1
MOV @Rr, A	Move A to data memory	1	1
MOV Rr, #data	Move immediate to register	2	2
MOV @Rr, #data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1
XCH A, Rr	Exchange A and register	1	1
XCH A, @Rr	Exchange A and data memory	1	1
XCHD A, @Rr	Exchange digit of A and register	1	1
MOVP A, @A	Move to A from current page	1	2
MOVP3, A, @A	Move to A from page 3	1	2
TIMER/COUNTER			
MOV A, T	Read Timer/Counter	1	1
MOV T, A	Load Timer/Counter	1	1
STRT T	Start Timer	1	1
STRT CNT	Start Counter	1	1
STOP TCNT	Stop Timer/Counter	1	1
EN TCNTI	Enable Timer/Counter Interrupt	1	1
DIS TCNTI	Disable Timer/Counter Interrupt	1	1
CONTROL			
EN DMA	Enable DMA Handshake Lines	1	1
EN I	Enable IBF Interrupt	1	1
DIS I	Diable IBF Interrupt	1	1
EN FLAGS	Enable Master Interrupts	1	1
SEL MB0	Select memory bank 0	1	1
SEL MB1	Select memory bank 1	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
NOP	No Operation	1	1
REGISTERS			
INC Rr	Increment register	1	1
INC @Rr	Increment data memory	1	1
DEC Rr	Decrement register	1	1

Table 2. UPI™ Instruction Set (Continued)

Mnemonic	Description	Bytes	Cycles
SUBROUTINE			
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2
FLAGS			
CLR C	Clear Carry	1	1
CPL C	Complement Carry	1	1
CLR F0	Clear Flag 0	1	1
CPL F0	Complement Flag 0	1	1
CLR F1	Clear F1 Flag	1	1
CPL F1	Complement F1 Flag	1	1
BRANCH			
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ Rr, addr	Decrement register and jump	2	2
JC addr	Jump on Carry = 1	2	2
JNC addr	Jump on Carry = 0	2	2
JZ addr	Jump on A Zero	2	2
JNZ addr	Jump on A not Zero	2	2
JT0 addr	Jump on T0 = 1	2	2
JNT0 addr	Jump on T0 = 0	2	2
JT1 addr	Jump on T1 = 1	2	2
JNT1 addr	Jump on T1 = 0	2	2
JF0 addr	Jump on F0 Flag = 1	2	2
JF1 addr	Jump on F1 Flag = 1	2	2
JTF addr	Jump on Timer Flag = 1, Clear Flag	2	2
JNIBF addr	Jump on IBF Flag = 0	2	2
JOBF addr	Jump on OBF Flag = 1	2	2
JBb addr	Jump on Accumula- for Bit	2	2

8741A
UNIVERSAL PERIPHERAL INTERFACE
8-BIT MICROCOMPUTER

- **8-Bit CPU plus ROM, RAM, I/O, Timer and Clock in a Single Package**
- **One 8-Bit Status and Two Data Registers for Asynchronous Slave-to-Master Interface**
- **DMA, Interrupt, or Polled Operation Supported**
- **1024 x 8 EPROM, 64 x 8 RAM, 8-Bit Timer/Counter, 18 Programmable I/O Pins**
- **Fully Compatible with All Microprocessor Families**
- **3.6 MHz 8741A-8 Available**
- **Expandable I/O**
- **RAM Power-Down Capability**
- **Over 90 Instructions: 70% Single Byte**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**

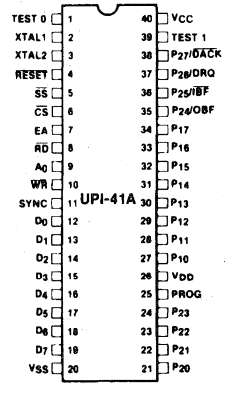
The Intel 8741A is a general purpose, programmable interface device designed for use with a variety of 8-bit microprocessor systems. It contains a low cost microcomputer with program memory, data memory, 8-bit CPU, I/O ports, timer/counter, and clock in a single 40-pin package. Interface registers are included to enable the UPI device to function as a peripheral controller in MCS®-48, MCS-80, MCS-85, MCS-86, and other 8-bit systems.

The UPI™-41A has 1K words of program memory and 64 words of data memory on-chip.

The device has two 8-bit, TTL compatible I/O ports and two test inputs. Individual port lines can function as either inputs or outputs under software control. I/O can be expanded with the 8243 device which is directly compatible and has 16 I/O lines. An 8-bit programmable timer/counter is included in the UPI device for generating timing sequences or counting external inputs. Additional UPI features include: single 5V supply, single-step mode for debug and dual working register banks.

Because it's a complete microcomputer, the UPI provides more flexibility for the designer than conventional LSI interface devices. It is designed to be an efficient controller as well as an arithmetic processor. Applications include keyboard scanning, printer control, display multiplexing and similar functions which involve interfacing peripheral devices to microprocessor systems.

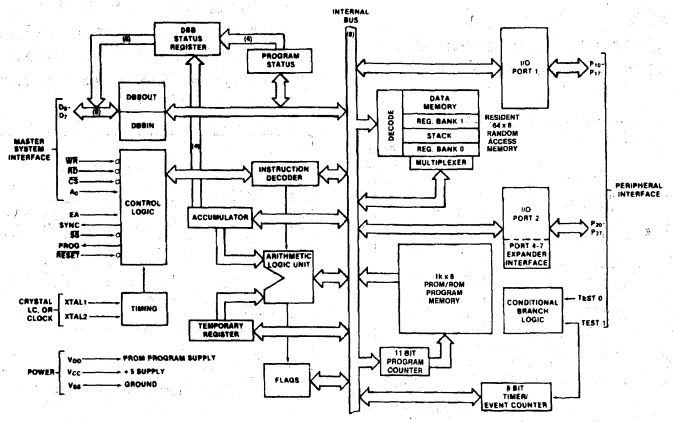
Pin Configuration



290241-2

October 1989
Order Number: 290241-001

Block Diagram



290241-1

Table 1. Pin Description

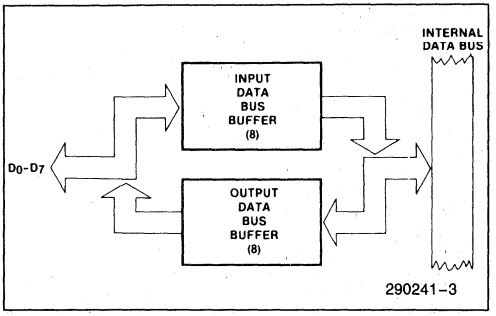
Signal	Description
D_0–D_7 (BUS)	Three-state, bidirectional DATA BUS BUFFER lines used to interface the UPI-41A to an 8-bit master system data bus.
P_{10}–P_{17}	8-bit, PORT 1 quasi-bidirectional I/O lines.
P_{20}–P_{27}	8-bit, PORT 2 quasi-bidirectional I/O lines. The lower 4 bits (P_{20}–P_{23}) interface directly to the 8243 I/O expander device and contain address and data information during PORT 4–7 access. The upper 4 bits (P_{24}–P_{27}) can be programmed to provide interrupt Request and DMA Handshake capability. Software control can configure P_{24} as OBF (Output Buffer Full), P_{25} as $\overline{IBF}$ (Input Buffer Full), P_{26} as DRQ (DMA Request), and P_{27} as $\overline{DACK}$ (DMA ACKnowledge).
$\overline{WR}$	I/O write input which enables the master CPU to write data and command words to the UPI-41A INPUT DATA BUS BUFFER.
$\overline{RD}$	I/O read input which enables the master CPU to read data and status words from the OUTPUT DATA BUS BUFFER or status register.
$\overline{CS}$	Chip select input used to select one UPI-41A out of several connected to a common data bus.
A_0	Address input used by the master processor to indicate whether byte transfer is data or command. During a write operation flag F_1 is set to the status of the A_0 input.
TEST 0, TEST 1	Input pins which can be directly tested using conditional branch instructions. (T_1) also functions as the event timer input (under software control). T_0 is used during PROM programming and verification in the 8741A.

Signal	Description
XTAL 1, XTAL 2	Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
SYNC	Output signal which occurs once per UPI-41A instruction cycle. SYNC can be used as a strobe for external circuitry; it is also used to synchronize single step operation.
EA	External access input which allows emulation, testing and PROM verification.
PROG	Multifunction pin used as the program pulse input during PROM programming. During I/O expander access the PROG pin acts as an address/data strobe to the 8243.
$\overline{RESET}$	Input used to reset status flip-flops and to set the program counter to zero. $\overline{RESET}$ is also used during PROM programming and verification. $\overline{RESET}$ should be held low for a minimum of 8 instruction cycles after power-up.
$\overline{SS}$	Single step input used in the 8741A in conjunction with the SYNC output to step the program through each instruction.
V_{CC}	+5V main power supply pin.
V_{DD}	+5V during normal operation. +25V during programming operation. Low power standby supply pin in ROM version.
V_{SS}	Circuit ground potential.

6

UPI-41A™ FEATURES AND ENHANCEMENTS

1. Two Data Bus Buffers, one for input and one for output. This allows a much cleaner Master/Slave protocol.



2. 8 Bits of Status

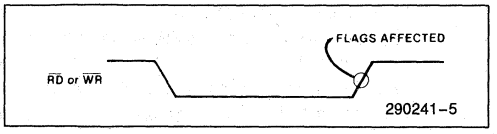
ST_7	ST_6	ST_5	ST_4	F_1	F_0	IBF	OBF
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

ST_4–ST_7 are user definable status bits. These bits are defined by the "MOV STS, A" single byte, single cycle instruction. Bits 4–7 of the accumulator are moved to bits 4–7 of the status register. Bits 0–3 of the status register are not affected.

MOV STS, A Op Code: 90H

1	0	0	1	0	0	0	0
D_7							**D_0**

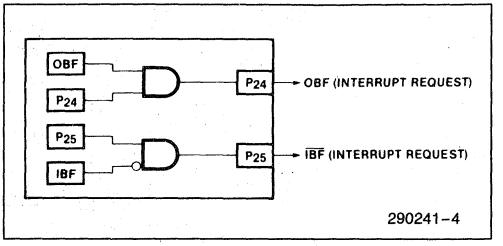
3. $\overline{RD}$ and $\overline{WR}$ are edge triggered. IBF, OBF, F_1 and INT change internally after the trailing edge of $\overline{RD}$ or $\overline{WR}$.



4. P_{24} and P_{25} are port pins or Buffer Flag pins which can be used to interrupt a master processor. These pins default to port pins on Reset.

   If the "EN FLAGS" instruction has been executed, P_{24} becomes the OBF (Output Buffer Full) pin. A "1" written to P_{24} enables the OBF pin (the pin outputs the OBF Status Bit). A "0" written to P_{24} disables the OBF pin (the pin remains low). This pin can be used to indicate that valid data is available from the UPI41A (in Output Data Bus Buffer).

   If "EN FLAGS" has been executed, P_{25} becomes the $\overline{IBF}$ (Input Buffer Full) pin. A "1" written to P_{25} enables the $\overline{IBF}$ pin (the pin outputs the inverse of the IBF Status Bit). A "0" written to P_{25} disables the $\overline{IBF}$ pin (the pin remains low). This pin can be used to indicate that the UPI is ready for data.



Data Bus Buffer Interrupt Capability

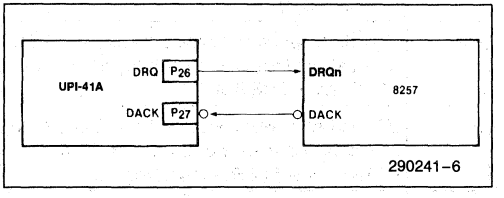
EN FLAGS Op Code: 0F5H

1	1	1	1	0	1	0	1
D_7							**D_0**

5. P_{26} and P_{27} are port pins or DMA handshake pins for use with a DMA controller. These pins default to port pins on Reset.

   If the "EN DMA" instruction has been executed, P_{26} becomes the DRQ (DMA Request) pin. A "1" written to P_{26} causes a DMA request (DRQ is activated). DRQ is deactivated by DACK•RD, DACK•WR, or execution of the "EN DMA" instruction.

   If "EN DMA" has been executed, P_{27} becomes the $\overline{DACK}$ (DMA Acknowledge) pin. This pin acts as a chip select input for the Data Bus Buffer registers during DMA transfers.



DMA Handshake Capability

EN DMA Op Code: 0E5H

1	1	1	0	0	1	0	1
D_7							**D_0**

APPLICATIONS

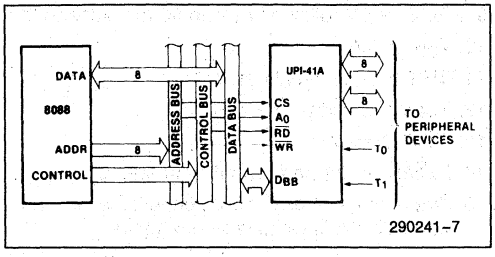


Figure 1. 8085A-8741A Interface

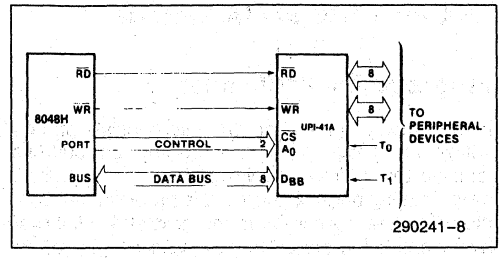


Figure 2. 8048-8741A Interface

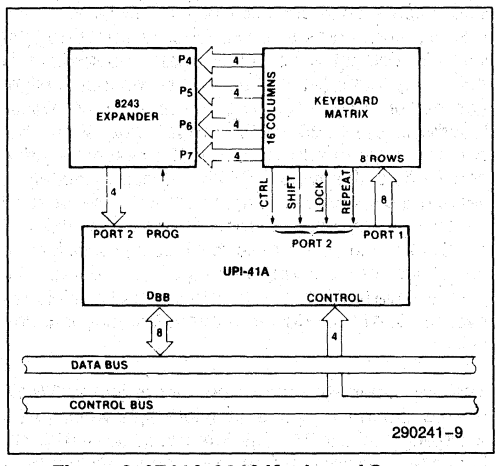


Figure 3. 8741A-8243 Keyboard Scanner

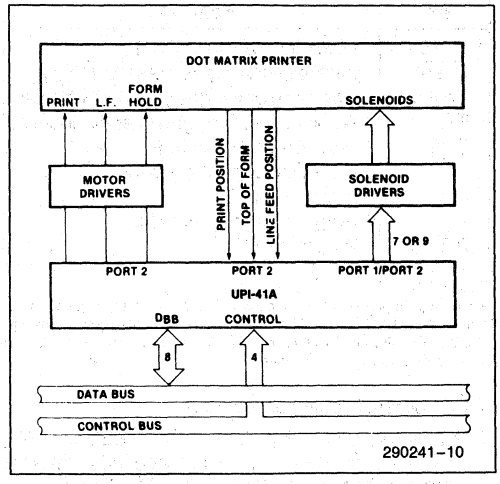


Figure 4. 8741A Matrix Printer Interface

6

PROGRAMMING, VERIFYING, AND ERASING THE 8741A EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input (1 to 6 MHz)
$\overline{\text{Reset}}$	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input
	Data Output during Verify
P20–1	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING:
An attempt to program a missocketed 8741A will result in severe damage to the part. An indication of a properly socketed part is the appearance of the SYNC clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1. A_0 = 0V, CS = 5V, EA = 5V, $\overline{\text{RESET}}$ = 0V, TEST0 = 5V, V_{DD} = 5V, clock applied or internal oscillator operating, BUS and PROG floating
2. Insert 8741A in programming socket
3. TEST 0 = 0V (select program mode)
4. EA = 23V (active program mode)
5. Address applied to BUS and P20–1
6. $\overline{\text{RESET}}$ = 5V (latch address)
7. Data applied to BUS
8. V_{DD} = 25V (programming power)
9. PROG = 0V followed by one 50 ms pulse to 23V
10. V_{DD} = 5V
11. TEST 0 = 5V (verify mode)
12. Read and verify data on BUS
13. TEST 0 = 0V
14. $\overline{\text{RESET}}$ = 0V and repeat from step 6
15. Programmer should be at conditions of step 1 when 8741A is removed from socket

8741A Erasure Characteristics

The erasure characteristics of the 8741A are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 8741A in approximately 3 years while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 8741A is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8741A window to prevent unintentional erasure.

The recommended erasure procedure for the 8741A is exposure to shortwave ultraviolet light which has a wavelength of 2537Å. The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 w-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The 8741A should be placed within one inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin with Respect to Ground.....0.5V to +7V
Power Dissipation.....1.5W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{SS} 0V, V_{CC} = V_{DD} = +5V ±10%

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage (except XTAL1, XTAL2, $\overline{RESET}$)	−0.5	0.8	V	
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, $\overline{RESET}$)	−0.5	0.6	V	
V_{IH}	Input High Voltage (except XTAL1, XTAL2, $\overline{RESET}$)	2.2	V_{CC}		
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$)	3.8	V_{CC}	V	
V_{OL}	Output Low Voltage (D_0–D_7)		0.45	V	I_{OL} = 2.0 mA
V_{OL1}	Output Low Voltage ($P_{10}P_{17}$, $P_{20}P_{27}$, Sync)		0.45	V	I_{OL} = 1.6 mA
V_{OL2}	Output Low Voltage (PROG)		0.45	V	I_{OL} = 1.0 mA
V_{OH}	Output High Voltage (D_0–D_7)	2.4		V	I_{OH} = −400 μA
V_{OH1}	Output High Voltage (All Other Outputs)	2.4		V	I_{OH} = −50 μA
I_{IL}	Input Leakage Current (T_0, T_1, $\overline{RD}$,$\overline{WR}$, $\overline{CS}$, A_0, EA)		±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{OZ}	Output Leakage Current (D_0–D_7, High Z State)		±10	μA	V_{SS} +0.45 $\le V_{IN} \le V_{CC}$
I_{LI}	Low Input Load Current ($P_{10}P_{17}$, $P_{20}P_{27}$)		0.5	mA	V_{IL} = 0.8V
I_{LI1}	Low Input Load Current ($\overline{RESET}$, $\overline{SS}$)		0.2	mA	V_{IL} = 0.8V
I_{DD}	V_{DD} Supply Current		15	mA	Typical = 5 mA
I_{CC} + I_{DD}	Total Supply Current		125	mA	Typical = 60 mA

A.C. CHARACTERISTICS T_A = 0°C to +70°C, V_{SS} = 0V, V_{CC} = V_{DD} = +5V ±10%

DBB READ

Symbol	Parameter	Min	Max	Unit	Test Conditions
t_{AR}	$\overline{CS}$, A_0 Setup to $\overline{RD}$ ↓	0		ns	
t_{RA}	$\overline{CS}$, A_0 Hold after $\overline{RD}$ ↑	0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	250		ns	
t_{AD}	$\overline{CS}$, A_0 to Data Out Delay		225	ns	C_L = 150 pF
t_{RD}	$\overline{RD}$ ↓ to Data Out Delay		225	ns	C_L = 150 pF
t_{DF}	$\overline{RD}$ ↑ to Data Float Delay		100	ns	
t_{CY}	Cycle Time (except 8741A-8)	2.5	15	μs	6.0 MHz XTAL
t_{CY}	Cycle Time (8741A-8)	4.17	15	μs	3.6 MHz XTAL

6

DBB WRITE

Symbol	Parameter	Min	Max	Units	Test Conditions
t_{AW}	$\overline{CS}$, A_0 Setup to $\overline{WR}$ ↓	0		ns	
t_{WA}	$\overline{CS}$, A_0 Hold after $\overline{WR}$ ↑	0		ns	
t_{WW}	$\overline{WR}$ Pulse Width	250		ns	
t_{DW}	Data Setup to $\overline{WR}$ ↑	150		ns	
t_{WD}	Data Hold after $\overline{WR}$ ↑	0		ns	

A.C. TIMING SPECIFICATION FOR PROGRAMMING

T_A = 0°C to +70°C, V_{CC} = +5V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
t_{AW}	Address Setup Time to $\overline{RESET}$ ↑	$4t_{CY}$			
t_{WA}	Address Hold Time after $\overline{RESET}$ ↑	$4t_{CY}$			
t_{DW}	Data in Setup Time to PROG ↑	$4t_{CY}$			
t_{WD}	Data in Hold Time after PROG ↓	$4t_{CY}$			
t_{PH}	$\overline{RESET}$ Hold Time to Verify	$4t_{CY}$			
t_{VDDW}	V_{DD} Setup Time to PROG ↑	$4t_{CY}$			
t_{VDDH}	V_{DD} Hold Time after PROG ↓	0			
t_{PW}	Program Pulse Width	50	60	ms	
t_{TW}	Test 0 Setup Time for Program Mode	$4t_{CY}$			
t_{WT}	Test 0 Hold Time after Program Mode	$4t_{CY}$			
t_{DO}	Test 0 to Data Out Delay		$4t_{CY}$		
t_{WW}	$\overline{RESET}$ Pulse Width to Latch Address	$4t_{CY}$			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	2.0	μs	
t_{CY}	CPU Operation Cycle Time	5.0		μs	
t_{RE}	$\overline{RESET}$ Setup Time before EA ↑	$4t_{CY}$			

NOTE:
1. If TEST 0 is high, t_{DO} can be triggered by $\overline{RESET}$ ↑.

D.C. SPECIFICATION FOR PROGRAMMING

T_A = 25°C ±5°C, V_{CC} = 5V ±5%, V_{DD} = 25V ±1V

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{DOH}	V_{DD} Program Voltage High Level	24.0	26.0	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	21.5	24.5	V	
V_{PL}	PROG Voltage Low Level		0.2	V	
V_{EAH}	EA Program or Verify Voltage High Level	21.5	24.5	V	
V_{EAL}	EA Voltage Low Level		5.25	V	
I_{DD}	V_{DD} High Voltage Supply Current		30.0	mA	
I_{PROG}	PROG High Voltage Supply Current		16.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

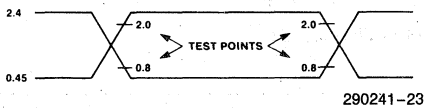
A.C. CHARACTERISTICS—DMA

Symbol	Parameter	Min	Max	Units	Test Conditions
t_{ACC}	$\overline{DACK}$ to $\overline{WR}$ or $\overline{RD}$	0		ns	
t_{CAC}	$\overline{RD}$ or $\overline{WR}$ to $\overline{DACK}$	0		ns	
t_{ACD}	$\overline{DACK}$ to Data Valid		225	ns	C_L = 150 pF
t_{CRQ}	$\overline{RD}$ or $\overline{WR}$ to DRQ Cleared		200	ns	

A.C. CHARACTERISTICS—PORT 2 T_A = 0°C to +70°C, V_{CC} = +5V ±10%

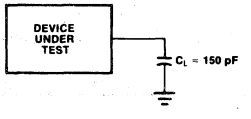
Symbol	Parameter	Min	Max	Units	Test Conditions
t_{CP}	Port Control Setup before Falling Edge of PROG	10		ns	
t_{PC}	Port Control Hold after Falling Edge of PROG	100		ns	
t_{PR}	PROG to Time P2 Input Must Be Valid		810	ns	
t_{PF}	Input Data Hold Time	0	150	ns	
t_{DP}	Output Data Setup Time	250		ns	
t_{PD}	Output Data Hold Time	65		ns	
t_{PP}	PROG Pulse Width	1200		ns	

A.C. TESTING INPUT/OUTPUT WAVEFORM



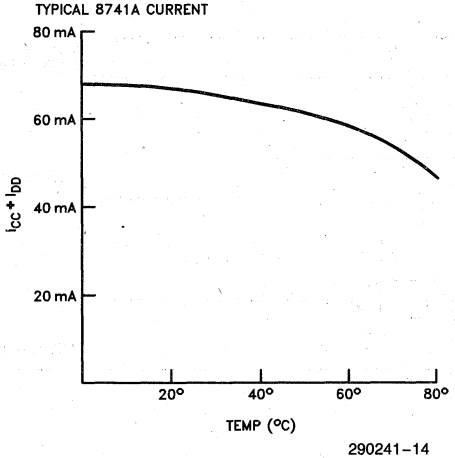
290241-23

A.C. TESTING LOAD CIRCUIT



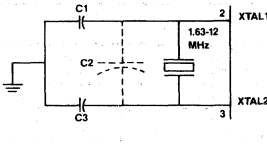
290241-15

TYPICAL 8741A CURRENT



290241-14

CRYSTAL OSCILLATOR MODE

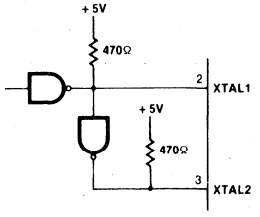


290241-11

Crystal series resistance should be
< 75Ω at 6 MHz; < 180Ω at 3.6 MHz.

6

DRIVING FROM EXTERNAL SOURCE

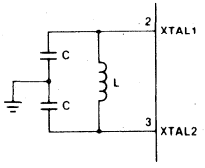


290241-12

Both XTAL1 and XTAL2 should be driven.
Resistors to V_{CC} are needed to ensure V_{IH} = 3.8V if TTL circuitry is used.

LC OSCILLATOR MODE

L	C	NOMINAL f
45 μH	20 pF	5.2 MHz
120 μH	20 pF	3.2 MHz



$$f = \frac{1}{2\pi\sqrt{LC'}}$$

$$C' = \frac{C + 3Cpp}{2}$$

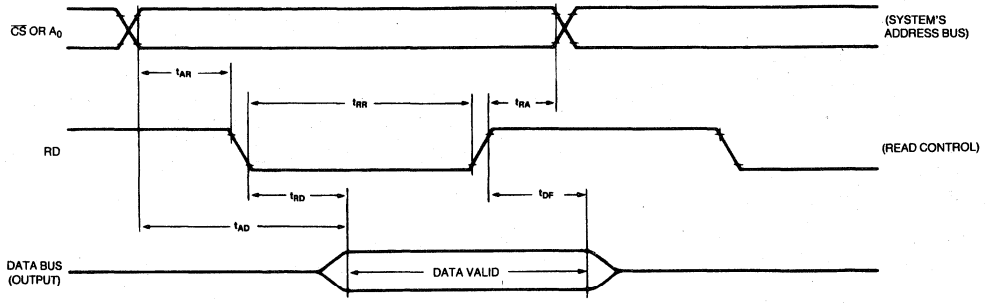
Cpp ≅ 5-10 pF
Pin-to-Pin Capacitance

290241-13

Each C should be approximately 20 pF, including stray capacitance.

WAVEFORMS

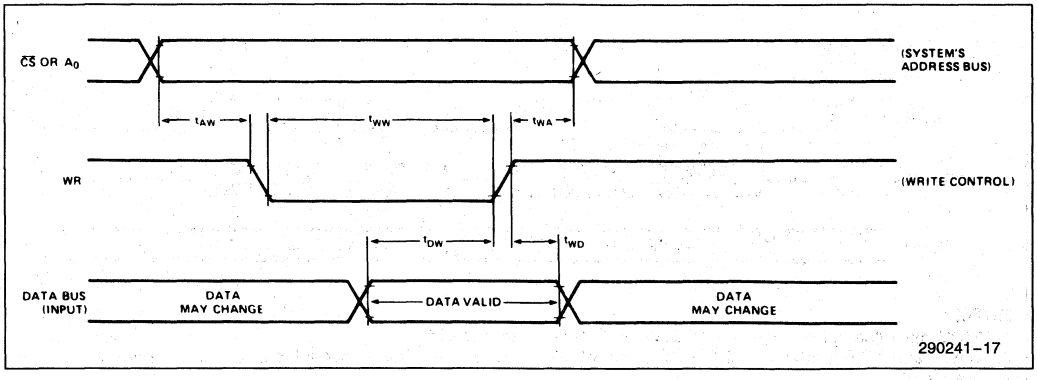
READ OPERATION—DATA BUS BUFFER REGISTER



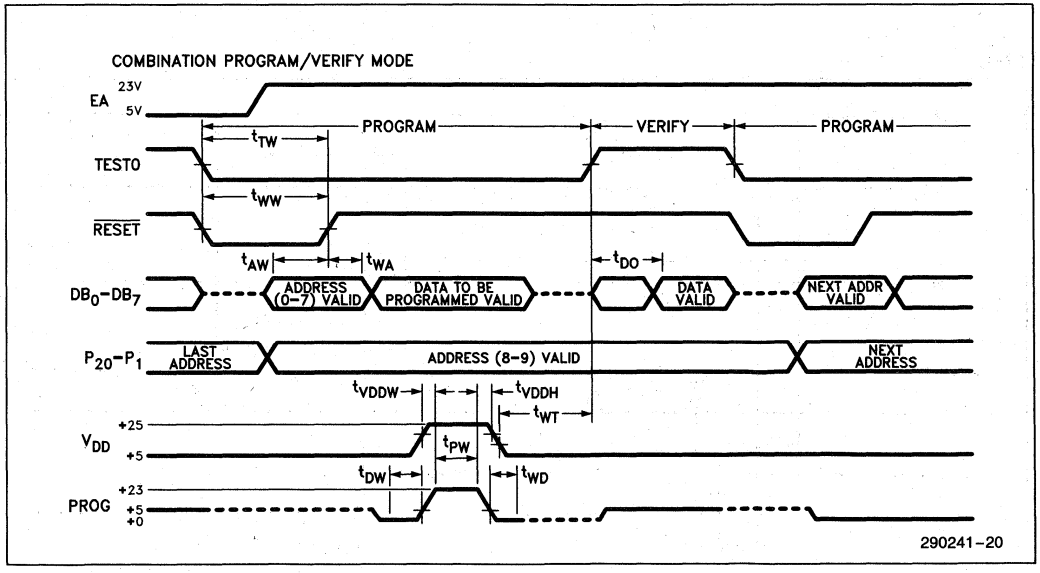
290241-16

WAVEFORMS

WRITE OPERATION—DATA BUS BUFFER REGISTER



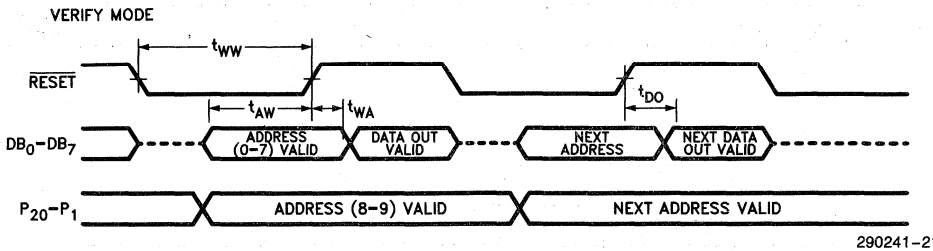
COMBINATION PROGRAM/VERIFY MODE



6

WAVEFORMS

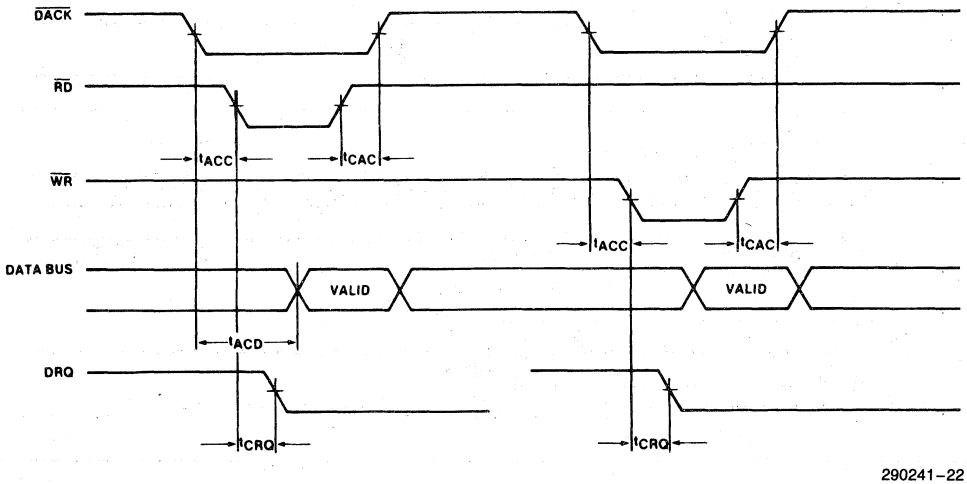
VERIFY MODE



290241-21

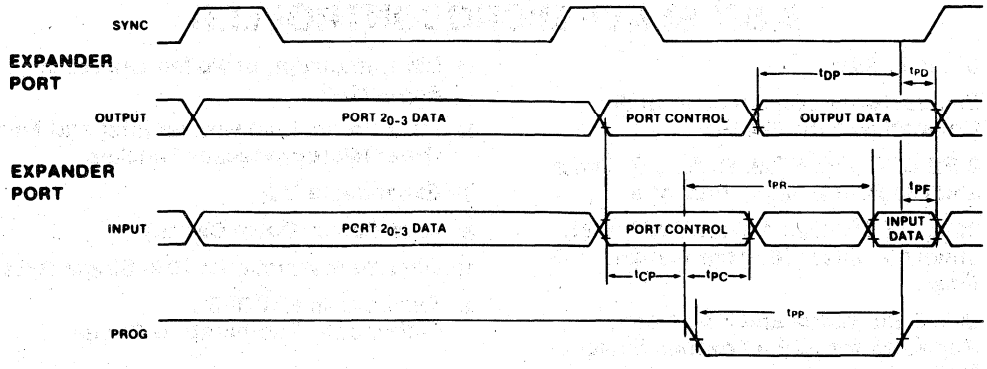
NOTES:
1. PROG must float if EA is low (i.e., ≠23V), or if T0 = 5V for the 8741A.
2. XTAL1 and XTAL2 driven by 3.6 MHz clock will give 7.17 μs t_{CY}. This is acceptable for 8741-8 parts as well as standard parts.
PROG must float or = 5V when EA is high.
3. A_0 must be held low (i.e., = 0V) during program/verify modes.

DMA



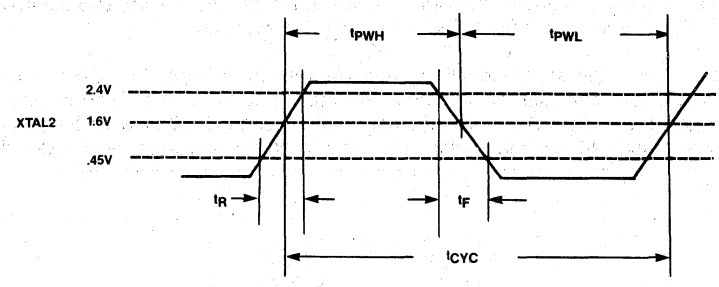
290241-22

PORT 2 TIMING



290241-19

PORT TIMING DURING EXTERNAL ACCESS (EA)



290241-18

On the rising edge of SYNC and EA is enabled, port data is valid and can be strobed. On the trailing edge of sync the program counter contents are available.

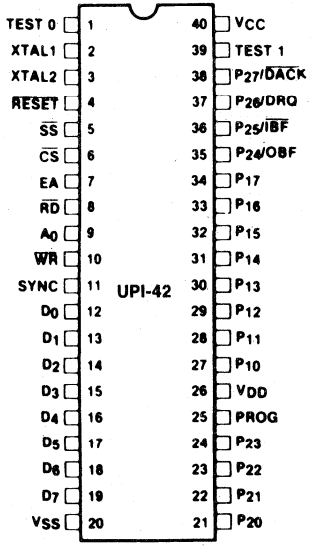
6

8742
UNIVERSAL PERIPHERAL INTERFACE 8-BIT SLAVE MICROCONTROLLER

- **8742: 12 MHz**
- **Pin, Software and Architecturally Compatible with 8741A**
- **8-Bit CPU plus ROM, RAM, I/O, Timer and Clock in a Single Package**
- **2048 x 8 EPROM, 128 x 8 RAM, 8-Bit Timer/Counter, 18 Programmable I/O Pins**
- **One 8-Bit Status and Two Data Registers for Asynchronous Slave-to-Master Interface**
- **DMA, Interrupt, or Polled Operation Supported**
- **Fully Compatible with all Intel and Most Other Microprocessor Families**
- **Expandable I/O**
- **RAM Power-Down Capability**
- **Over 90 Instructions: 70% Single Byte**
- **Available in EXPRESS — Standard Temperature Range**

The Intel 8742 is a general-purpose Universal Peripheral Interface that allows designers to grow their own customized solution for peripheral device control. It contains a low-cost microcomputer with 2K of program memory, 128 bytes of data memory, 8-bit timer/counter, and clock generator in a single 40-pin package. Interface registers are included to enable the UPI™ device to function as a peripheral controller in the MCS®-48, MCS-51, MCS-80, MCS-85, 8088, 8086 and other 8-, 16-bit systems.

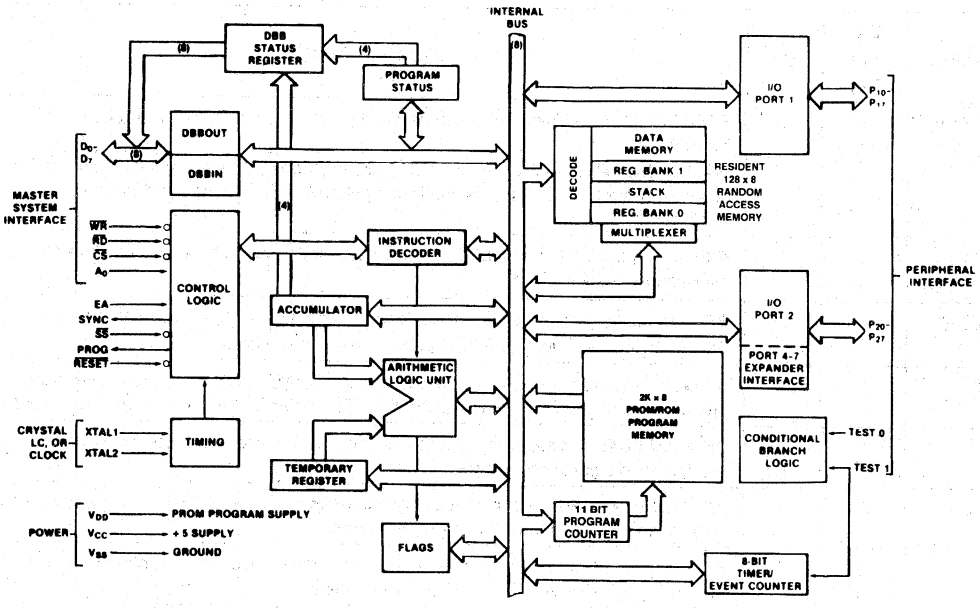
The 8742 is software, pin, and architecturally compatible with the 8741A. The 8742 doubles the on-chip memory space to allow for additional features and performance to be incorporated in upgraded 8741A designs. For new designs, the additional memory and performance of the 8742 extends the UPI concept to more complex motor control tasks, 80-column printers and process control applications as examples.



290256-2

Figure 1. Pin Configuration

October 1989
Order Number: 290256-001



290256-1

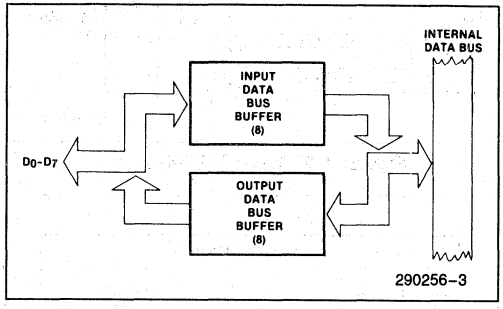
Figure 2. Block Diagram

Table 1. Pin Description

Symbol	DIP Pin No.	Type	Name and Function
TEST 0, TEST 1	1 39	I	**TEST INPUTS:** Input pins which can be directly tested using conditional branch instructions. **FREQUENCY REFERENCE:** TEST 1 (T_1) also functions as the event timer input (under software control). TEST 0 (T_0) is used during PROM programming and EPROM verification.
XTAL 1, XTAL 2	2 3	I	**INPUTS:** Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
$\overline{\text{RESET}}$	4	I	**RESET:** Input used to reset status flip-flops and to set the program counter to zero. $\overline{\text{RESET}}$ is also used during EPROM programming and verification.
$\overline{\text{SS}}$	5	I	**SINGLE STEP:** Single step input used in conjunction with the SYNC output to step the program through each instruction (EPROM). This should be tied to +5V when not used.
$\overline{\text{CS}}$	6	I	**CHIP SELECT:** Chip select input used to select one UPI microcomputer out of several connected to a common data bus.
EA	7	I	**EXTERNAL ACCESS:** External access input which allows emulation, testing and EPROM verification. This pin should be tied low if unused.
$\overline{\text{RD}}$	8	I	**READ:** I/O read input which enables the master CPU to read data and status words from the OUTPUT DATA BUS BUFFER or status register.
A_0	9	I	**COMMAND/DATA SELECT:** Address Input used by the master processor to indicate whether byte transfer is data (A_0 = 0, F1 is reset) or command (A_0 = 1, F1 is set). A_0 = 0 during program and verify operations.
$\overline{\text{WR}}$	10	I	**WRITE:** I/O write input which enables the master CPU to write data and command words to the UPI INPUT DATA BUS BUFFER.
SYNC	11	O	**OUTPUT CLOCK:** Output signal which occurs once per UPI instruction cycle. SYNC can be used as a strobe for external circuitry; it is also used to synchronize single step operation.
D_0–D_7 (BUS)	12–19	I/O	**DATA BUS:** Three-state, bidirectional DATA BUS BUFFER lines used to interface the UPI microcomputer to an 8-bit master system data bus.
P_{10}–P_{17}	27–34	I/O	**PORT 1:** 8-bit, PORT 1 quasi-bidirectional I/O lines.
P_{20}–P_{27}	21–24 35–38	I/O	**PORT 2:** 8-bit, PORT 2 quasi-bidirectional I/O lines. The lower 4 bits (P_{20}–P_{23}) interface directly to the 8243 I/O expander device and contain address and data information during PORT 4–7 access. The upper 4 bits (P_{24}–P_{27}) can be programmed to provide interrupt Request and DMA Handshake capability. Software control can configure P_{24} as Output Buffer Full (OBF) interrupt, P_{25} as Input Buffer Full ($\overline{\text{IBF}}$) interrupt, P_{26} as DMA Request (DRQ), and P_{27} as DMA ACKnowledge ($\overline{\text{DACK}}$).
PROG	25	I/O	**PROGRAM:** Multifunction pin used as the program pulse input during PROM programming. During I/O expander access the PROG pin acts as an address/data strobe to the 8243. This pin should be tied high if unused.
V_{CC}	40		**POWER:** +5V main power supply pin.
V_{DD}	26		**POWER:** +5V during normal operation. +21V during programming operation. Low power standby supply pin.
V_{SS}	20		**GROUND:** Circuit ground potential.

UPI-42 FEATURES

1. Two Data Bus Buffers, one for input and one for output. This allows a much cleaner Master/Slave protocol.



2. 8 Bits of Status

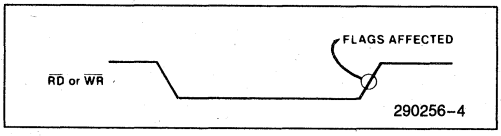
ST_7	ST_6	ST_5	ST_4	F_1	F_0	IBF	OBF
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

ST_4–ST_7 are user definable status bits. These bits are defined by the "MOV STS, A" single byte, single cycle instruction. Bits 4–7 of the acccumulator are moved to bits 4–7 of the status register. Bits 0–3 of the status register are not affected.

MOV STS, A Op Code: 90H

1	0	0	1	0	0	0	0
D_7							D_0

3. $\overline{RD}$ and $\overline{WR}$ are edge triggered. IBF, OBF, F_1 and $\overline{INT}$ change internally after the trailing edge of $\overline{RD}$ or $\overline{WR}$.

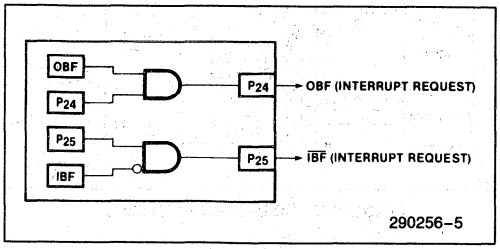


During the time that the host CPU is reading the status register, the 8742 is prevented from updating this register or is "locked out".

4. P_{24} and P_{25} are port pins or Buffer Flag pins which can be used to interrupt a master processor. These pins default to port pins on Reset.

   If the "EN FLAGS" instruction has been executed, P_{24} becomes the OBF (Output Buffer Full) pin. A "1" written to P_{24} enables the OBF pin (the pin outputs the OBF Status Bit). A "0" written to P_{24} disables the OBF pin (the pin remains low). This pin can be used to indicate that valid data is available from the UPI-41A (in Output Data Bus Buffer).

   If "EN FLAGS" has been executed, P_{25} becomes the $\overline{IBF}$ (Input Buffer Full) pin. A "1" written to P_{25} enables the $\overline{IBF}$ pin (the pin outputs the inverse of the IBF Status Bit. A "0" written to P_{25} disables the $\overline{IBF}$ pin (the pin remains low). This pin can be used to indicate that the UPI is ready for data.



Data Bus Buffer Interrupt Capability

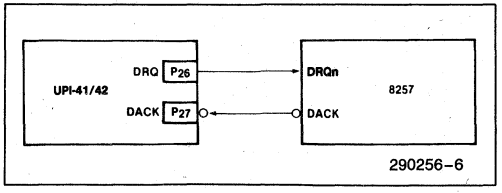
EN FLAGS Op Code: 0F5H

1	1	1	1	0	1	0	1
D_7							D_0

5. P_{26} and P_{27} are port pins or DMA handshake pins for use with a DMA controller. These pins default to port pins on Reset.

   If the "EN DMA" instruction has been executed, P_{26} becomes the DRQ (DMA Request) pin. A "1" written to P_{26} causes a DMA request (DRQ is activated). DRQ is deactivated by DACK•RD, DACK•WR, or execution of the "EN DMA" instruction.

   If "EN DMA" has been executed, P_{27} becomes the $\overline{DACK}$ (DMA Acknowledge) pin. This pin acts as a chip select input for the Data Bus Buffer registers during DMA transfers.



DMA Handshake Capability

EN DMA Op Code: 0E5H

1	1	1	0	0	1	0	1
D_7							D_0

6. The RESET input on the 8742, includes a 2-stage synchronizer to support reliable reset operation for 12 MHz operation.

7. When EA is enabled on the 8742, the program counter is placed on Port 1 and the lower three bits of Port 2 (MSB = P_{22}, LSB = P_{10}). On the 8742 this information is multiplexed with PORT DATA (see port timing diagrams at end of this data sheet).

APPLICATIONS

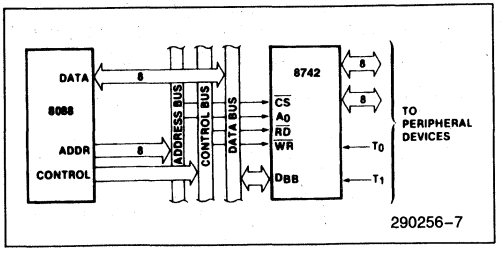


Figure 3. 8088-8742 Interface

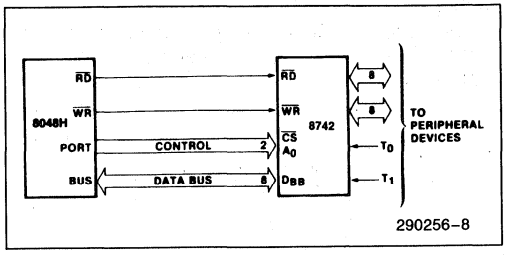


Figure 4. 8048H-8742 Interface

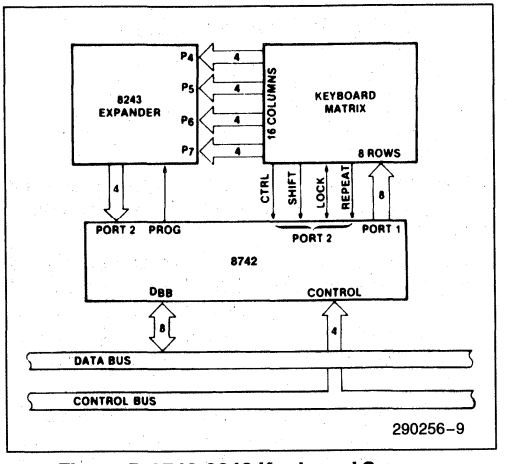


Figure 5. 8742-8243 Keyboard Scanner

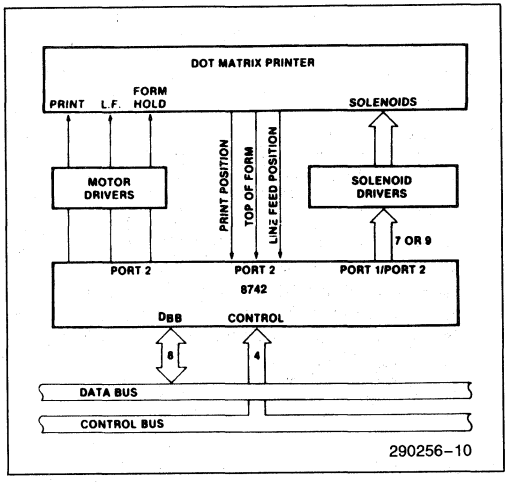


Figure 6. 8742 80-Column Matrix Printer Interface

PROGRAMMING, VERIFYING, AND ERASING THE 8742 EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock-Input
$\overline{\text{Reset}}$	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input Data Output During Verify
P_{20-12}	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING
An attempt to program a missocketed 8742 will result in severe damage to the part. An indication of a properly socketed part is the appearance of the SYNC clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1. A_0 = 0V, CS = 5V, EA = 5V, RESET = 0V, TESTO = 5V, V_{DD} = 5V, clock applied or internal oscillator operating, BUS floating, PROG = 5V.
2. Insert 8742 in programming socket
3. TEST 0 = 0V (select program mode)
4. EA = 18V (active program mode)
5. Address applied to BUS and P_{20-22}
6. $\overline{\text{RESET}}$ = 5V (latch address)
7. Data applied to BUS**
8. V_{DD} = 21V (programming power)
9. PROG = V_{CC} followed by one 50 ms pulse to 18V
10. V_{DD} = 5V
11. TEST 0 = 5V (verify mode)
12. Read and verify data on BUS
13. TEST 0 = 0V
14. $\overline{\text{RESET}}$ = 0V and repeat from step 5
15. Programmer should be at conditions of step 1 when 8742 is removed from socket

8742 Erasure Characteristics

The erasure characteristics of the 8742 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase the typical 8742 in approximately 3 years while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 8742 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8742 window to prevent unintentional erasure.

The recommended erasure procedure for the 8742 is exposure to shortwave ultraviolet light which has a wavelength of 2537Å. The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 w-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The 8742 should be placed within one inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

6

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin With Respect to Ground .....−0.5 to +7V
Power Dissipation.....1.5W

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS $T_A = 0°$ to $+70°C$, $V_{CC} = V_{DD} = +5V \pm 10\%$

Symbol	Parameter	8742		Units	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage (Except XTAL1, XTAL2, RESET)	−0.5	0.8	V	
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, RESET)	−0.5	0.6	V	
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, RESET)	2.0	V_{CC}	V	
V_{IH1}	Input High Voltage (XTLA1, XTAL2, RESET)	3.5	V_{CC}	V	
V_{OL}	Output Low Voltage (D_0–D_7)		0.45	V	$I_{OL} = 2.0$ mA
V_{OL1}	Output Low Voltage (P_{10}–P_{17}, P_{20}–P_{27}, Sync)		0.45	V	$I_{OL} = 1.6$ mA
V_{OL2}	Output Low Voltage (PROG)		0.45	V	$I_{OL} = 1.0$ mA
V_{OH}	Output High Voltage (D_0–D_7)	2.4		V	$I_{OH} = -400\ \mu A$
V_{OH1}	Output High Voltage (All Other Outupts)	2.4			$I_{OH} = -50\ \mu A$
I_{IL}	Input Leakage Current (T_0, T_1, RD, WR, CS, A_0, EA)		±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{OFL}	Output Leakage Current (D_0–D_7, High Z State)		±10	μA	$V_{SS} + 0.45 \le V_{OUT} \le V_{CC}$
I_{LI}	Low Input Load Current (P_{10}–P_{17}, P_{20}–P_{27})		0.3	mA	$V_{IL} = 0.8V$
I_{LI1}	Low Input Load Current (RESET, SS)		0.2	mA	$V_{IL} = 0.8V$
I_{DD}	V_{DD} Supply Current		10	mA	Typical = 5 mA
$I_{CC} + I_{DD}$	Total Supply Current		125	mA	Typical = 60 mA
I_{IH}	Input Leakage Current (P_{10}–P_{17}, P_{20}–P_{27})		100	μA	$V_{IN} = V_{CC}$
C_{IN}	Input Capacitance		10	pF	
$C_{1\,0}$	I/O Capacitance		20	pF	

D.C. CHARACTERISTICS—PROGRAMMING

$T_A = 25°C \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{DD} = 21V \pm 0.5V$

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{DOH}	V_{DD} Program Voltage High Level	20.5	21.5	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	17.5	18.5	V	
V_{PL}	PROG Voltage Low Level	$V_{CC}-0.5$	V_{CC}	V	
V_{EAH}	EA Program or Verify Voltage High Level	17.5	18.5	V	
V_{EAL}	EA Voltage Low Level		5.25	V	
I_{DD}	V_{DD} High Voltage Supply Current		30.0	mA	
I_{PROG}	PROG High Voltage Supply Current		1.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

A.C. CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{SS} = 0V$, $V_{CC} = V_{DD} = +5V \pm 10\%$

DBB READ

Symbol	Parameter	8742		Units
		Min	Max	
t_{AR}	CS, A_0 Setup to RD ↓	0		ns
t_{RA}	CS, A_0 Hold after RD ↑	0		ns
t_{RR}	RD Pulse Width	160		ns
t_{AD}	CS, A_0 to Data Out Delay		130	ns
t_{RD}	RD ↓ to Data Out Delay		130	ns
t_{DF}	RD ↑ to Data Float Delay		85	ns
t_{CY}	Cycle Time	1.25	15	μs(1)

DBB WRITE

Symbol	Parameter	Min	Max	Units
t_{AW}	CS, A_0 Setup to WR ↓	0		ns
t_{WA}	CS, A_0 Hold after WR ↑	0		ns
t_{WW}	WR Pulse Width	160		ns
t_{DW}	Data Setup to WR ↑	130		ns
t_{WD}	Data Hold after WR ↑	0		ns

NOTE:
1. $T_{CY} = 15/f(XTAL)$

A.C. CHARACTERISTICS $T_A = 25°C \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{DD} = +21V \pm 0.5$

PROGRAMMING

Symbol	Parameter	Min	Max	Units	Test Conditions
t_{AW}	Address Setup Time to RESET ↑	$4t_{CY}$			
t_{WA}	Address Hold Time after RESET ↑	$4t_{CY}$			
t_{DW}	Data in Setup Time to PROG ↑	$4t_{CY}$			
t_{WD}	Data in Hold Time after PROG ↓	$4t_{CY}$			
t_{PH}	RESET Hold Time to Verify	$4t_{CY}$			
t_{VDDW}	V_{DD} Setup Time to PROG ↑	0	1.0	mS	
t_{VDDH}	V_{DD} Hold Time after PROG ↑	0	1.0	mS	
t_{PW}	Program Pulse Width	50	60	mS	
t_{TW}	Test 0 Setup Time for Program Mode	$4t_{CY}$			
t_{WT}	Test 0 Hold Time after Program Mode	$4t_{CY}$			
t_{DO}	Test 0 to Data Out Delay		$4t_{CY}$		
t_{WW}	RESET Pulse Width to Latch Address	$4t_{CY}$			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	2.0	μs	
t_{CY}	CPU Operation Cycle Time	4.0		μs	
t_{RE}	RESET Setup Time before EA ↑	$4t_{CY}$			

NOTE:
If TEST 0 is high, t_{DO} can be triggered by RESET ↑.

A.C. CHARACTERISTICS DMA

Symbol	Parameter	8642/8742		Units
		Min	Max	
t_{ACC}	DACK to WR or RD	0		ns
t_{CAC}	RD or WR to DACK	0		ns
t_{ACD}	DACK to Data Valid		130	ns
t_{CRQ}	RD or WR to DRQ Cleared		100	ns(1)

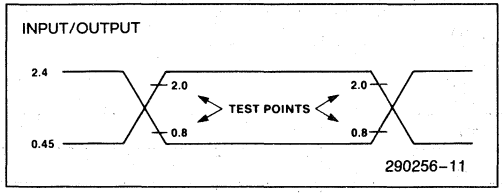
NOTE:
1. C_L = 150 pF.

A.C. CHARACTERISTICS PORT 2 T_A = 0°C to +70°C, V_{CC} = +5V ±10%

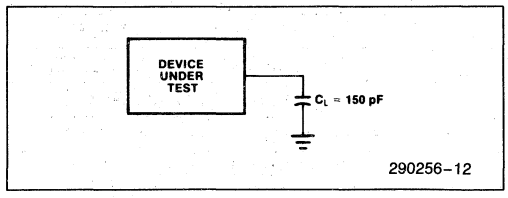
Symbol	Parameter	f(t_{CY})	8742/8642(3)		Units
			Min	Max	
t_{CP}	Port Control Setup before Falling Edge of PROG	1/15 t_{CY} − 28	55		ns(1)
t_{PC}	Port Control Hold after Falling Edge of PROG	1/10 t_{CY}	125		ns(2)
t_{PR}	PROG to Time P2 Input Must Be Valid	8/15 t_{CY} − 16		650	ns(1)
t_{PF}	Input Data Hold Time		0	150	ns(2)
t_{DP}	Output Data Setup Time	2/10 t_{CY}	250		ns(1)
t_{PD}	Output Data Hold Time	1/10 t_{CY} − 80	45		ns(2)
t_{PP}	PROG Pulse Width	6/10 t_{CY}	750		ns

NOTES:
1. C_L = 80 pF.
2. C_L = 20 pF.
3. t_{CY} = 1.25 μs.

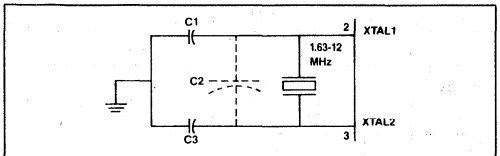
A.C. TESTING INPUT/OUTPUT WAVEFORM



A.C. TESTING LOAD CIRCUIT

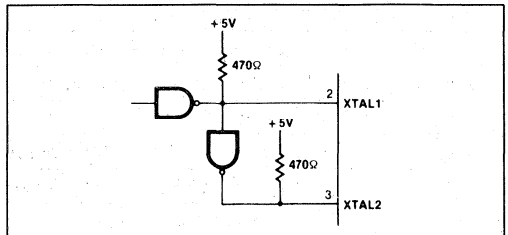


CRYSTAL OSCILLATOR MODE



Crystal Series Resistance Should be <750 at 12 MHz; <180Ω at 3.6 MHz.

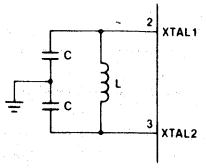
DRIVING FROM EXTERNAL SOURCE



Rise and Fall Times Should Not Exceed 20 ns. Resistors to V_{CC} are Needed to Ensure V_{IH} = 3.5V if TTL Circuitry is Used.

LC OSCILLATOR MODE

L	C	NOMINAL
45 H	20 pF	5.2 MHz
120 H	20 pF	3.2 MHz



$$f = \frac{1}{2\pi\sqrt{LC'}}$$

$$C' = \frac{C + 3Cpp}{2}$$

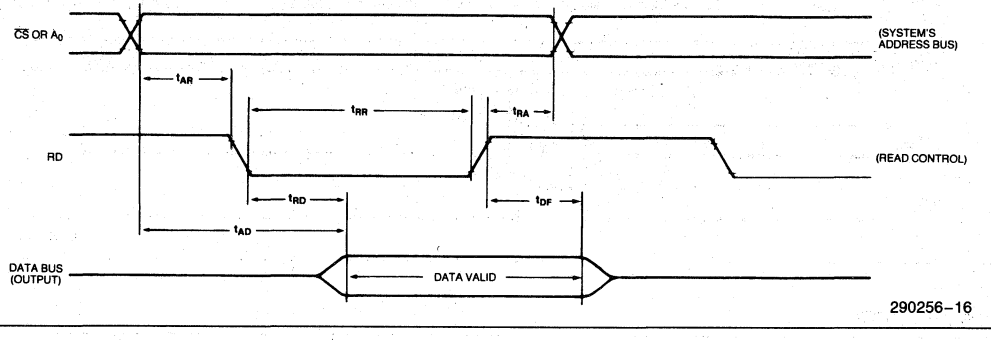
Cpp ≅ 5 pF–10 pF
Pin-to-Pin Capacitance

290256-15

Each C Should be Approximately 20 pF, including Stray Capacitance.

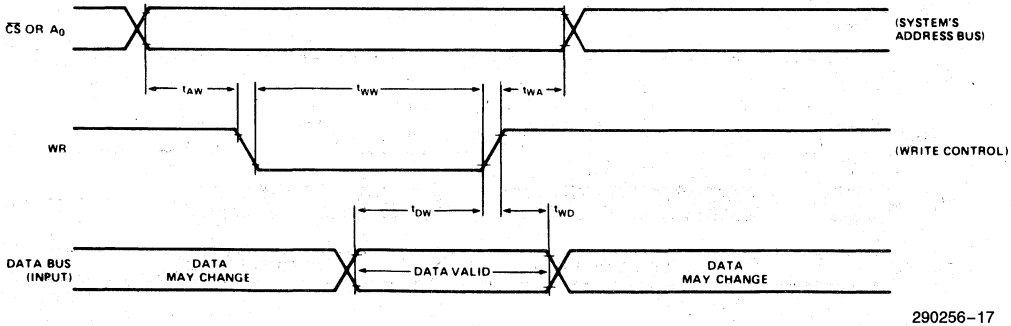
WAVEFORMS

READ OPERATION—DATA BUS BUFFER REGISTER



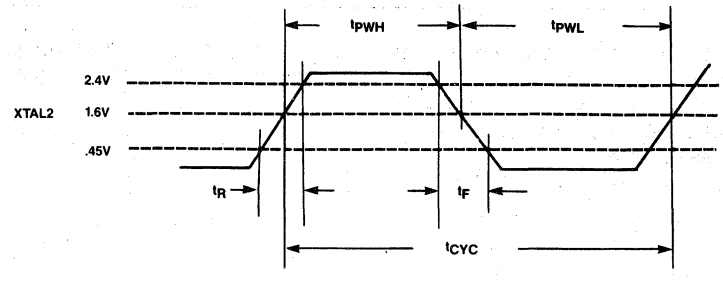
290256-16

WRITE OPERATION—DATA BUS BUFFER REGISTER



290256-17

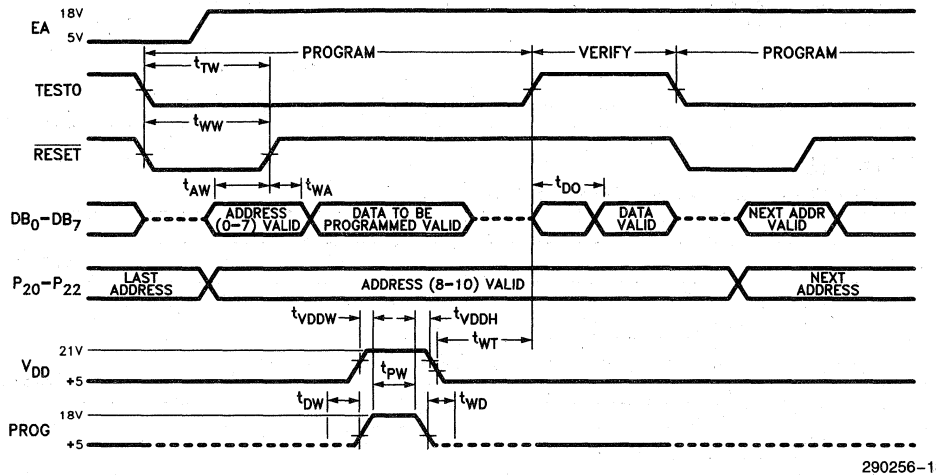
CLOCK TIMING



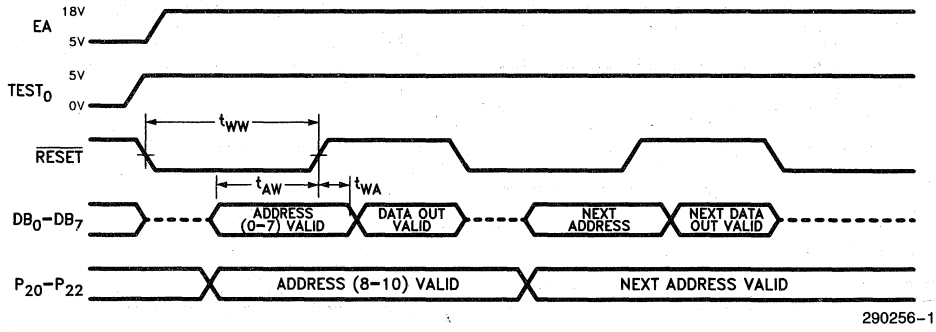
290256-23

WAVEFORMS

COMBINATION PROGRAM/VERIFY MODE



VERIFY MODE



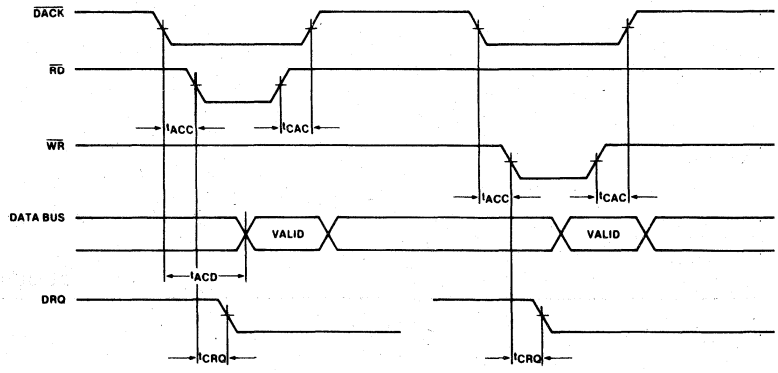
NOTES:
1. PROG must float if EA is low or EA is low or if $TEST_0$ = 5V.
2. A_0 must be held low (i.e., = 0V) during program/verify modes.
3. Test 0 must be held high.

The 8742 EPROM can be programmed by the following Intel products:

1. Universal PROM Programmer (UPP 103) peripheral of the Intellec® Development System with a UPP-549 Personality Card.
2. iUP-200/iUP-201 PROM Programmer with the iUP-F87/44 Personality Module.

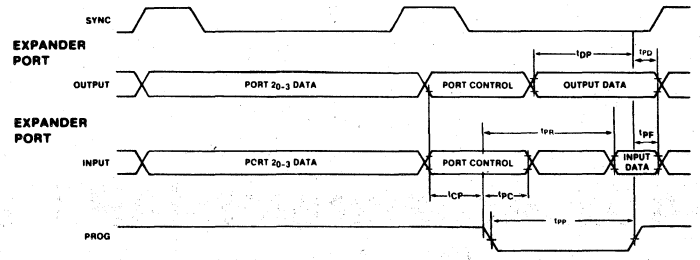
WAVEFORMS (Continued)

DMA



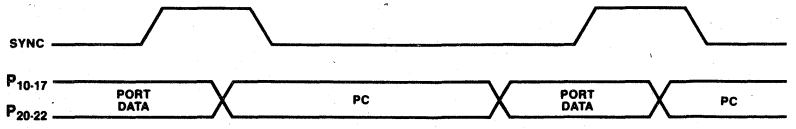
290256-20

PORT 2



290256-21

PORT TIMING DURING EXTERNAL ACCESS (EA)



290256-22

On the Rising Edge of SYNC and EA is Enabled, Port Data is Valid and can be Strobed on the Trailing Edge of Sync the Program Counter Contents are Available.

intel®

APPLICATION NOTE

AP-281

September 1989

UPI-452 Accelerates iAPX 286 Bus Performance

Order Number: 292018-001

UPI-452 ACCELERATES iAPX 286 BUS PERFORMANCE

CONTENTS

PAGE

INTRODUCTION

UPI-452 iAPX 286 System Configuration 6-169
Host UPI-452 FIFO Slave Interface 6-169
UPI-452 Initialization 6-171
FIFO Data Structures 6-172
FIFO Input/Output Channel Size 6-177
Interrupt Response Timing 6-178
DMA 6-180
Host FIFO DMA 6-180
UPI-452 Internal DMA Processor 6-182
Interface Latency 6-183
FIFO DMA Freeze Mode Interface 6-184

LIST OF FIGURES, TABLES AND EQUATIONS

FIGURES:

1. iAPX 286/UPI-452 System Block Diagram 6-170
2. Input FIFO Channel Functional Diagram 6-173
3. Output FIFO Channel Functional Diagram 6-174
4. Full Duplex FIFO Operation 6-177
5. Entire FIFO Assigned to Output Channel 6-178
6. UPI-452 Command Cycle Timing 6-181
7. Disabling FIFO to Host Slave Interface Timing Diagram 6-185
8. Sequence of Events to Invoke FIFO DMA Freeze Mode 6-186
9. Sequence of Events to Invoke FIFO DMA Freeze Mode Timings 6-186

TABLES:

1. FIFO Special Function Register Default Values 6-171
2. UPI-452 Initialization Event Sequence Example 6-171
3. UPI-452 External Address Decoding .. 6-175
4. UPI-452 to Host Interrupt Sources 6-179

CONTENTS

PAGE

5. Host UPI-452 Data Transfer Performance 6-180
6. UPI-452 Internal DMA Controller Cycle Timings 6-182
7. Host DMA FIFO Data Transfer Times 6-183
8. UPI-452 Internal DMA FIFO Data Transfer Times 6-184
9. Typical Instruction Cycle Timings 6-184
10. Slave Bus Interface Status During FIFO DMA Freeze Mode 6-187
11. FIFO SFR's Characteristics During FIFO DMA Freeze Mode 6-187

CONTENTS

PAGE

EQUATIONS:

1. Host Interrupt Response Time 6-179
2. Host FIFO DMA Transfer Rate—Input or Output Channel 6-180
3. Minimum Host/FIFO Transfer Rate Including Data Stream Command(s) .. 6-182
4. Effective Internal FIFO Transfer Time Using Internal DMA 6-184
5. Effective FIFO Transfer Time Using Individual Instructions 6-184

INTRODUCTION

The UPI-452 targets the leading problem in peripheral to host interfacing, the interface of a slow peripheral with a fast Host or "bus utilization". The solution is data buffering to reduce the delay and overhead of transferring data between the Host microprocessor and I/O subsystem. The Intel CMOS UPI-452 solves this problem by combining a sophisticated programmable FIFO buffer and a slave interface with an MSC-51 based microcontroller.

The UPI-452 is Intel's newest Universal Peripheral Interface family member. The UPI-452 FIFO buffer enables Host—peripheral communications to be through streams or bursts of data rather than by individual bytes. In addition the FIFO provides a means of embedding commands within a stream or block of data. This enables the system designer to manage data and commands to further off-load the Host.

The UPI-452 interfaces to the iAPX 286 microprocessor as a standard Intel slave peripheral device. READ, WRITE, CS and address lines from the Host are used to access all of the Host addressable UPI-452 Special Function Registers (SFR).

The UPI-452 combines an MSC-51 microcontroller, with 256 bytes of on-chip RAM and 8K bytes of ROM, twice that of the 80C51, a two channel DMA controller and a sophisticated 128 byte, two channel, bidirectional FIFO in a single device. The UPI-452 retains all of the 80C51 architecture, and is fully compatible with the MSC-51 instruction set.

This application note is a description of an iAPX 286 to UPI-452 slave interface. Included is a discussion of the respective timings and design considerations. This application note is meant as a supplement to the UPI-452 Advance Data Sheet. The user should consult the data sheet for additional details on the various UPI-452 functions and features.

UPI-452 iAPX 286 SYSTEM CONFIGURATION

The interface described in this application note is shown in Figure 1, iAPX 286 UPI-452 System Block Diagram. The iAPX 286 system is configured in a local bus architecture design. DMA between the Host and the UPI-452 is supported by the 82258 Advanced DMA Controller. The Host microprocessor accesses all UPI-452 externally addressable registers through address decoding (see Table 3, UPI-452 External Address Decoding). The timings and interface descriptions below are given in equation form with examples of specific calculations. The goal of this application note is a set of interface analysis equations. These equations are the tools a system designer can use to fully utilize the features of the UPI-452 to achieve maximum system performance.

HOST-UPI-452 FIFO SLAVE INTERFACE

The UPI-452 FIFO acts as a buffer between the external Host 80286 and the internal CPU. The FIFO allows the Host - peripheral interface to achieve maximum decoupling of the interface. Each of the two FIFO channels is fully user programmable. The FIFO buffer ensures that the respective CPU, Host or internal CPU, receives data in the same order as transmitted. Three slave bus interface handshake methods are supported by the UPI-452; DMA, Interrupt and Polled.

The interface between the Host 80286 and the UPI-452 is accomplished with a minimum of signals. The 8 bit data bus plus READ, WRITE, CS, and A0-2 provide access to all of the externally addressable UPI-452 registers including the two FIFO channels. Interrupt and DMA handshaking pins are tied directly to the interrupt controller and DMA controller respectively.

DMA transfers between the Host and UPI-452 are controlled by the Host processors DMA controller. In the example shown in Figure 1, the Host DMA controller is the 82258 Advanced DMA Controller. An internal DMA transfer to or from the FIFO, as well as between other internal elements, is controlled by the UPI-452 internal DMA processor. The internal DMA processor can also transfer data between Input and Output FIFO channels directly. The description that follows details the UPI-452 interface from both the Host processor's and the UPI-452's internal CPU perspective.

One of the unique features of the UPI-452 FIFO is its ability to distinguish between commands and data embedded in the same data block. Both interrupts and status flags are provided to support this operation in either direction of data transfer. These flags and interrupts are triggered by the FIFO logic independent of, and transparent to either the Host or internal CPUs. Commands embedded in the data block, or stream, are called Data Stream Commands.

Programmable FIFO channel Thresholds are another unique feature of the UPI-452. The Thresholds provide for interrupting the Host only when the Threshold number of bytes can be read or written to the FIFO buffer. This further decouples the Host UPI-452 interface by relieving the Host of polling the buffer to determine the number of bytes that can be read or written. It also reduces the chances of overrun and underrun errors which must be processed.

The UPI-452 also provides a means of bypassing the FIFO, in both directions, for an immediate interrupt of either the Host or internal CPU. These commands are called Immediate Commands. A complete description of the internal FIFO logic operation is given in the FIFO Data Structure section.

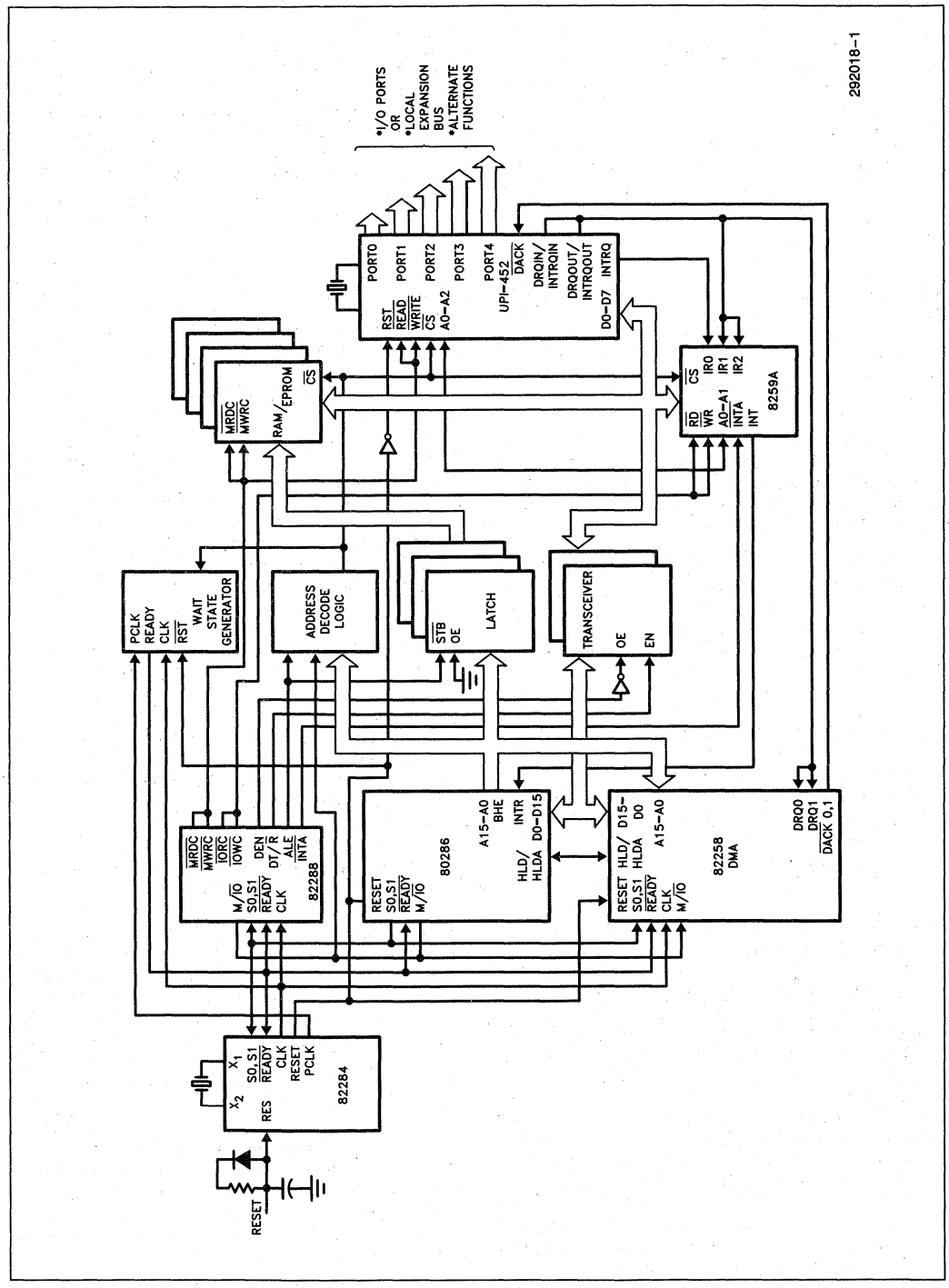


Figure 1. iAPX 286 UPI-452 System Block Diagram

UPI-452 INITIALIZATION

The UPI-452 at power-on reset automatically performs a minimum initialization of itself. The UPI-452 notifies the Host that it is in the process of initialization by setting a Host Status SFR bit. The user UPI-452 program must release the UPI-452 from initialization for the FIFO to be accessible by the Host. This is the minimum Host to UPI-452 initialization sequence. All further initialization and configuration of the UPI-452, including the FIFO, is done by the internal CPU under user program control. No interaction or programming is required by the Host 80286 for UPI-452 initialization.

At power-on reset the UPI-452 automatically enters FIFO DMA Freeze Mode by resetting the Slave Control (SLCON) SFR FIFO DMA Freeze/Normal Mode bit to FIFO DMA Freeze Mode (FRZ = "0"). This forces the Slave Status (SSTAT) and Host Status (HSTAT) SFR FIFO DMA Freeze/Normal Mode bits to FIFO DMA Freeze Mode In Progress. FIFO DMA Freeze Mode allows the FIFO interface to be configured, by the internal CPU, while inhibiting Host access to the FIFO.

The MODE SFR is forced to zero at reset. This disables, (tri-states) the DRQIN/INTRQIN, DRQOUT/INTRQOUT and INTRQ output pins. INTRQ is inhibited from going active to reflect the fact that a Host Status SFR bit, FIFO DMA Freeze Mode, is active. If the MODE SFR INTRQ configure bit is enabled (='1'), before the Slave Control and Host Status SFR FIFO DMA Freeze/Normal Mode bit is set to Normal Mode, INTRQ will go active immediately.

The first action by the Host following reset is to read the UPI-452 Host Status SFR Freeze/Normal Mode bit to determine the status of the interface. This may be done in response to a UPI-452 INTRQ interrupt, or by polling the Host Status SFR. Reading the Host Status SFR resets the INTRQ line low.

Any of the five FIFO interface SFRs, as well as a variety of additional features, may be programmed by the internal CPU following reset. At power-on reset, the five FIFO Special Function Registers are set to their default values as listed in Table 1. All reserved location bits are set to one, all other bits are set to zero in these three SFRs. The FIFO SFRs listed in Table 1 can be programmed only while the UPI-452 is in FIFO DMA Freeze Mode. The balance of the UPI-452 SFRs default values and descriptions are listed in the UPI-452 Advance Data Sheet in the Intel Microsystems Component Handbook Volume II and Microcontroller Handbook.

The above sequence is the minimum UPI-452 internal initialization required. The last initialization instruction must always set the UPI-452 to Normal Mode. This causes the UPI-452 to exit Freeze Mode and enables Host read/write access of the FIFO. The internal CPU sets the Slave Control (SLCON) SFR FIFO DMA Freeze/Normal Mode (FRZ) bit high (= 1) to activate Normal Mode. Ths causes the Slave Status (SSTAT) and Host Status (HSTAT) SFR FIFO DMA Freeze Mode bits to be set to Normal Mode. Table 2, UPI-452 Initialization Event Sequence Example, shows a summary of the initialization events described above.

Table 1. FIFO Special Function Register Default Values

SFR Name	Label	Reset Value
Channel Boundary Pointer	CBP	40H/64D
Output Channel Read Pointer	ORPR	40H/64D
Output Channel Write Pointer	OWPR	40H/64D
Input Channel Read Pointer	IRPR	00H/0D
Input Channel Write Pointer	IWPR	00H/0D
Input Threshold	ITH	00H/0D
Output Threshold	OTH	01H/1D

Table 2. UPI-452 Initialization Event Sequence Example

Event Description	SFR/bit
Power-on Reset	
UPI-452 forces FIFO DMA Freeze Mode (Host access to FIFO inhibited)	SLCON FRZ = 0
UPI-452 forces Slave Status and Host Status SFR to FIFO DMA Freeze Mode In Progress	SSTAT SST5 = 0 HSTAT HST1 = 1
UPI-452 forces all SFRs, including FIFO SFRs, to default values.	
* UPI-452 user program enables INTRQ, INTRQ goes active, high	MODE MD4 = 1
* Host $\overline{\text{READ}}$'s UPI-452 Host Status (HSTAT) SFR to determine interrupt source, INTRQ goes low	
* UPI-452 user program initializes any other SFRs; FIFO, Interrupts, Timers/Counters, etc.	
User program sets Slave Control SFR to Normal Mode (Host access to FIFO enabled)	SLCON FRZ = 1
UPI-452 forces Slave and Host Status SFRs bits to Normal Operation	SSTAT SST5 = 1 HSTAT HST1 = 0
* Host polls Host Status SFR to determine when it can access the FIFO - or - * Host waits for UPI-452 Request for Service interrupt to access FIFO	

* user option

FIFO DATA STRUCTURES

Overview

The UPI-452 provides three means of communication between the Host microprocessor and the UPI-452 in either direction;

Data
Data Stream Commands
Immediate Commands

Data and Data Stream Commands (DSC) are transferred between the Host and UPI-452 through the UPI-452 FIFO buffer. The third, Immediate Commands, provides a means of bypassing the FIFO entirely. These three data types are in addition to direct access by either Host or Internal CPU of dedicated Status and Control Special Function Registers (SFR).

The FIFO appears to both the Host 80286 and the internal CPU as 8 bits wide. Internally the FIFO is logically nine bits wide. The ninth bit indicates whether the byte is a data or a Data Stream Command (DSC) byte; 0 = data, 1 = DSC. The ninth bit is set by the FIFO logic in response to the address specified when writing to the FIFO by either Host or internal CPU. The FIFO uses the ninth bit to condition the UPI-452 interrupts and status flags as a byte is made available for a Host or internal CPU read from the FIFO. Figures 2 and 3 show the structure of each FIFO channel and the logical ninth bit.

It is important to note that both data and DSCs are actually entered into the FIFO buffer (see Figures 2 and 3). External addressing of the FIFO determines the state of the internal FIFO logic ninth bit. Table 3 shows the UPI-452 External Address Decoding used by the Host and the corresponding action. The internal CPU interface to the FIFO is essentially identical to the external Host interface. Dedicated internal Special Function Registers provide the interface between the FIFO, internal CPU and the internal two channel DMA processor. FIFO read and write operations by the Host and internal CPU are interleaved by the UPI-452 so they appear to be occurring simultaneously.

The ninth bit provides a means of supporting two data types within the FIFO buffer. This feature enables the Host and UPI-452 to transfer both commands and data while maintaining the decoupled interface a FIFO buffer provides. The logical ninth bit provides both a means of embedding commands within a block of data and a means for the internal CPU, or external Host, to discriminate between data and commands. Data or DSCs may be written in any order desired. Data Stream Commands can be used to structure or dispatch the data by defining the start and end of data blocks or packets, or how the data following a DSC is to be processed.

A Data Stream Command (DSC) acts as an internal service routine vector. The DSC generates an interrupt to a service routine which reads the DSC. The DSC byte acts as an address vector to a user defined service routine. The address can be any program or data memory location with no restriction on the number of DSCs or address boundaries.

A Data Stream Command (DSC) can also be used to clear data from the FIFO or "FLUSH" the FIFO. This is done by appending a DSC to the end of a block of data entered in the FIFO which is less than the programmed threshold number of bytes. The DSC will cause an interrupt, if enabled, to the respective receiving CPU. This ensures that a less than Threshold number of bytes in the FIFO will be read. Two conditions force a Request for Service interrupt, if enabled, to the Host. The first is due to a Threshold number of bytes having been written to the FIFO Output channel; the second is if a DSC is written to the Output FIFO channel. If less than the Threshold number of bytes are written to the Output FIFO channel, the Host Status SFR flag will not be set, and a Request for Service interrupt will not be generated, if enabled. By appending a DSC to end of the data block, the FIFO Request for Service flag and/or interrupt will be generated.

An example of a FIFO Flush application is a mass storage subsystem. The UPI-452 provides the system interface to a subsystem which supports tape and disk storage. The FIFO size is dynamically changed to provide the maximum buffer size for the direction of transfer. Large data blocks are the norm in this application. The FIFO Flush provides a means of purging the FIFO of the last bytes of a transfer. This guarantees that the block, no matter what its size, will be transmitted out of the FIFO.

Immediate Commands allow more direct communication between the Host processor and the UPI-452 by bypassing the FIFO in either direction. The Immediate Command IN and OUT SFRs are two more unique address locations externally and internally addressable. Both DSCs and Immediate Commands have internal interrupts and interrupt priorities associated with their operation. The interrupts are enabled or disabled by setting corresponding bits in the Slave Control (SLCON), Interrupt Enable (IEC), Interrupt Priority (IPC) and Interrupt Enable and Priority (IEP) SFRs. A detailed description of each of these may be found in the UPI-452 Advance Information Data Sheet.

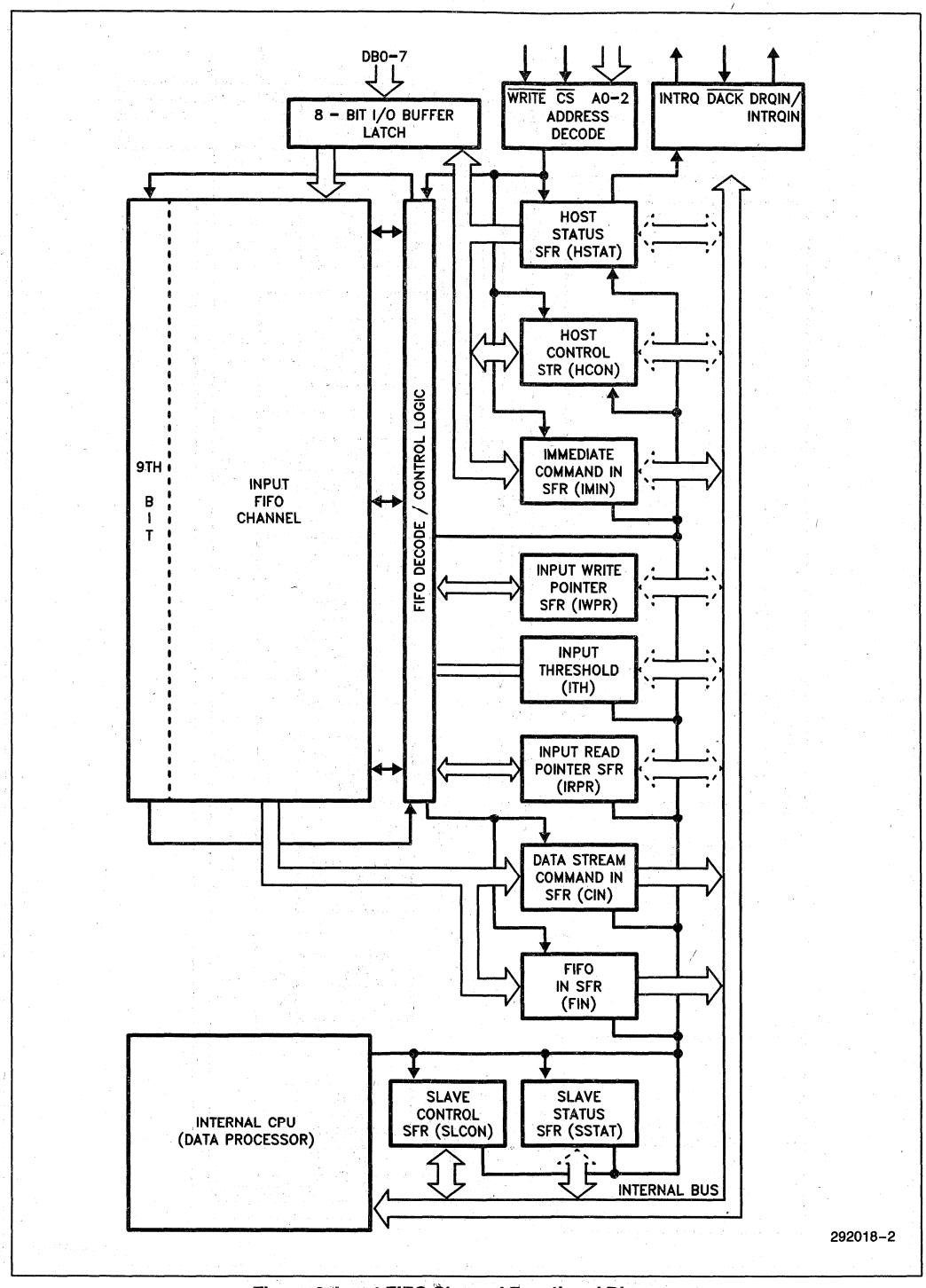


Figure 2. Input FIFO Channel Functional Diagram

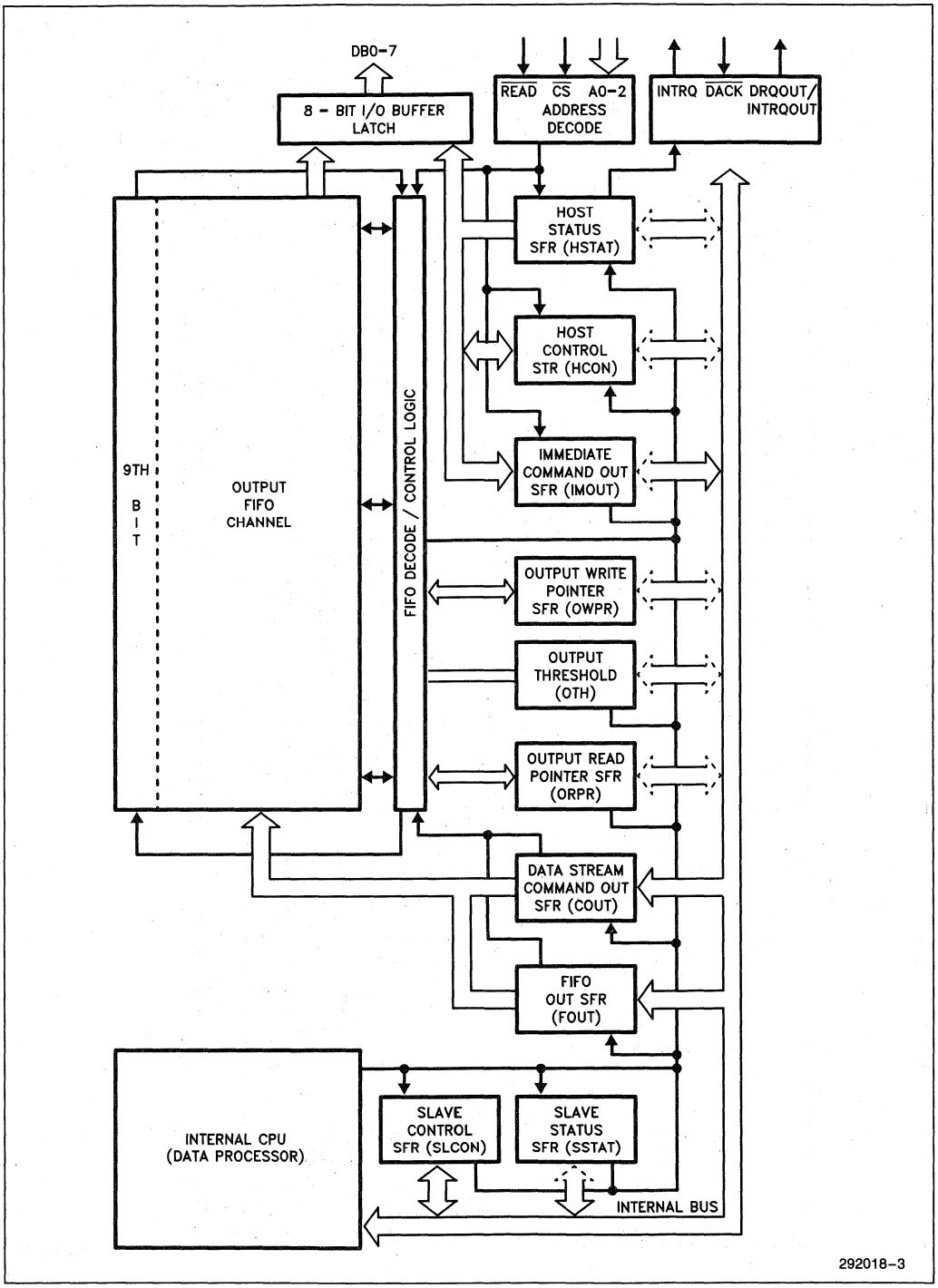


Figure 3. Output FIFO Channel Functional Diagram

Table 3. UPI-452 External Address Decoding

$\overline{DACK}$	$\overline{CS}$	A2	A1	A0	$\overline{READ}$	$\overline{WRITE}$
1	1	X	X	X	No Operation	No Operation
1	0	0	0	0	Data or DMA from Output FIFO Channel	Data or DMA to Input FIFO Channel
1	0	0	0	1	Data Stream Command from Output FIFO Channel	Data Stream Command to Input FIFO Channel
1	0	0	1	0	Host Status SFR Read	Reserved
1	0	0	1	1	Host Control SFR Read	Host Control SFR Write
1	0	1	0	0	Immediate Command SFR Read	Immediate Command SFR Write
1	0	1	1	X	Reserved	Reserved
0	X	X	X	X	DMA Data from Output FIFO Channel	DMA Data to Input FIFO Channel

Below is a detailed description of each FIFO channel's operation, including the FIFO logic response to the ninth bit, as a byte moves through the channel. The description covers each of the three data types for each channel. The details below provide a picture of the various FIFO features and operation. By understanding the FIFO structure and operation the user can optimize the interface to meet the requirements of an individual design.

OUTPUT CHANNEL

This section covers the data path from the internal CPU to the HOST. Data Stream Command or Immediate Command processing during Host DMA Operations is covered in the DMA section.

UPI-452 Internal Write to the FIFO

The internal CPU writes data and Data Stream Commands into the FIFO through the FIFO OUT (FOUT) and Command OUT (COUT) SFRs. When a Threshold number of bytes has been written, the Host Status SFR Output FIFO Request for Service bit is set and an interrupt, if enabled, is generated to the Host. Either the INTRQ or DRQOUT/INTRQOUT output pins can be used for this interrupt as determined by the MODE and Host Control (HCON) SFR setting. The Host responds to the Request for Service interrupt by reading the Host Status (HSTAT) SFR to determine the source of the interrupt. The Host then reads the Threshold number of bytes from the FIFO. The internal CPU may continue to write to the FIFO during the Host read of the FIFO Output channel.

Data Stream Commands may be written to the Output FIFO channel at any time during a write of data bytes. The write instruction need only specify the Command Out (COUT) SFR in the direct register instruction used. Immediate Commands may also be written at any time to the Immediate Command OUT (IMOUT) SFR. The Host reads Immediate Commands from the Immediate Command OUT (IMOUT).

The internal CPU can determine the number of bytes to write to the FIFO Output channel in one of three ways. The first, and most efficient, is by utilizing the internal DMA processor which will automatically manage the writing of data to avoid Underrun or Overrun Errors. The second is for the internal CPU to read the Output FIFO channels Read and Write Pointers and compare their values to determine the available space. The third method for determining the available FIFO space is to always write the programmed channel size number of bytes to the Output FIFO. This method would use the Overrun Error flag and interrupt to halt FIFO writing whenever the available space was less than the channel size. The interrupt service routine could read the channel pointers to determine or monitor the available channel space. The time required for the internal CPU to write data to the Output FIFO channel is a function of the individual instruction cycle time and the number of bytes to be written.

Host Read from the FIFO

The Host reads data or Data Stream Commands (DSC) from the FIFO in response to the Host Status (HSTAT) SFR flags and interrupts, if enabled. All Host read operations access the same UPI-452 internal I/O Buffer Latch. At the end of the previous Host FIFO read cycle a byte is loaded from the FIFO into the I/O Buffer Latch and Host Status (HSTAT) SFR bit 5 is set or cleared (1 = DSC, 0 = data) to reflect the state of the byte's FIFO ninth bit. If the FIFO ninth bit is set (= 1) indicating a DSC, an interrupt is generated to the external Host via INTRQ pin or INTRQIN/INTRQOUT pins as determined by Host Control (HCON) SFR bit 1. The Host then reads the Host Status (HSTAT) SFR to determine the source of the interrupt.

The most efficient Host read operation of the FIFO Output channel is through the use of Host DMA. The UPI-452 fully supports external DMA handshaking. The MODE and Host Control SFRs control the configuration of UPI-452 Host DMA handshake outputs. If Host DMA is used the Threshold Request for Service interrupt asserts the UPI-452 DMA Request (DRQOUT) output. The Host DMA processor acknowledges with $\overline{\text{DACK}}$ which acts as a chip select of the FIFO channels. The DMA transfer would stop when either the threshold byte count had been read, as programmed in the Host DMA processor, or when the DRQOUT output is brought inactive by the UPI-452.

INPUT CHANNEL

This section covers the data path from the HOST to the internal CPU or internal DMA processor. The details of Data Stream Command or Immediate Command processing during internal DMA operations are covered in the DMA section below.

Host Write to the FIFO

The Host writes data and Data Stream Commands into the FIFO through the FIFO IN (FIN) and Command IN (CIN) SFRs. When a Threshold number of bytes has been read out of the Input FIFO channel by the internal CPU, the Host Status SFR Input FIFO Request for Service bit is set and an interrupt, if enabled, is generated to the Host. The Input FIFO Threshold interrupt tells the Host that it may write the next block of data into the FIFO. Either the INTRQ or DRQIN/INTRQIN output pins can be used for this interrupt as determined by the MODE and Host Control (HCON) SFR settings. The Host may continue to write to the FIFO Input channel during the internal CPU read of the FIFO. Data Stream Commands may be written to the FIFO Input channel at any time during a write of data bytes. Immediate Commands may also be written at any time to the Immediate Command IN (IMIN) SFR.

The Host also has three methods for determining the available FIFO space. Two are essentially identical to that of the internal CPU. They involve reading the FIFO Input channel pointers and using the Host Status SFR Underrun and Overrun Error flags and Request for Service interrupts these would generate, if enabled in combination. The third involves using the UPI-452 Host DMA controller handshake signals and the programmed Input FIFO Threshold. The Host would receive a Request for Service interrupt when an Input FIFO channel has a Theshold number of bytes able to be written by the Host. The Host service routine would then write the Threshold number of bytes to the FIFO.

If a Host DMA is used to write to the FIFO Input channel, the Threshold Request for Service interrupt could assert the UPI-452 DRQIN output. The Host DMA processor would assert $\overline{DACK}$ and the FIFO write would be completed by Host the DMA processor. The DMA transfer would stop when either the Threshold byte count had been written or the DRQIN output was removed by the UPI-452. Additional details on Host and internal DMA operation is given below.

Internal Read of the FIFO

At the end of an internal CPU read cycle a byte is loaded from the FIFO buffer into the FIFO IN/Command IN SFR and Slave Status (SSTAT) SFR bit 1 is set or cleared (1 = data, 0 = DSC) to reflect the state of the FIFO ninth bit. If the byte is a DSC, the FIFO ninth bit is set (= 1) and an interrupt is generated, if enabled, to the Internal CPU. The internal CPU then reads the Slave Status (SSTAT) SFR to determine the source of the interrupt.

Immediate Commands are written by the Host and read by the internal CPU through the Immediate Command IN (IMIN) SFR. Once written, an Immediate Command sets the Slave Status (SSTAT) SFR flag bit and generates an interrupt, if enabled, to the internal CPU. In response to the interrupt the internal CPU reads the Slave Status (SSTAT) SFR to determine the source of the interrupt and service the Immediate Command.

FIFO INPUT/OUTPUT CHANNEL SIZE

Host

The Host does not have direct control of the FIFO Input or Output channel sizes or configuration. The Host can, however, issue Data Stream Commands or Immediate Commands to the UPI-452 instructing the UPI-452 to reconfigure the FIFO interface by invoking FIFO DMA Freeze Mode. The Data Stream Command or Immediate Command would be a vector to a service routine which performs the specific reconfiguration.

UPI-452 Internal

The default power-on reset FIFO channel sizes are listed in the "Initialization" section and can be set only by the internal CPU during FIFO DMA Freeze Mode. The FIFO channel size is selected to achieve the optimum application performance. The entire 128 byte FIFO can be allocated to either the Input or Output channel. In this case the other channel consists of a single SFR; FIFO IN/Command IN or FIFO OUT/Command OUT SFR. Figure 4 shows a FIFO division with a portion devoted to each channel. Figure 5 shows a FIFO configuration with all 128 bytes assigned to the Output channel.

The FIFO channel Threshold feature allows the user to match the FIFO channel size and the performance of the internal and Host data transfer rates. The programmed Threshold provides an elasticity to the data transfer operation. An example is if the Host FIFO

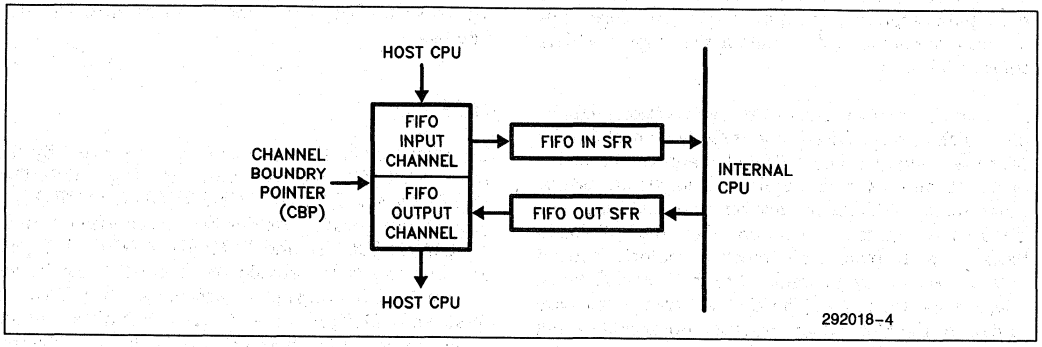


Figure 4. Full Duplex FIFO Operation

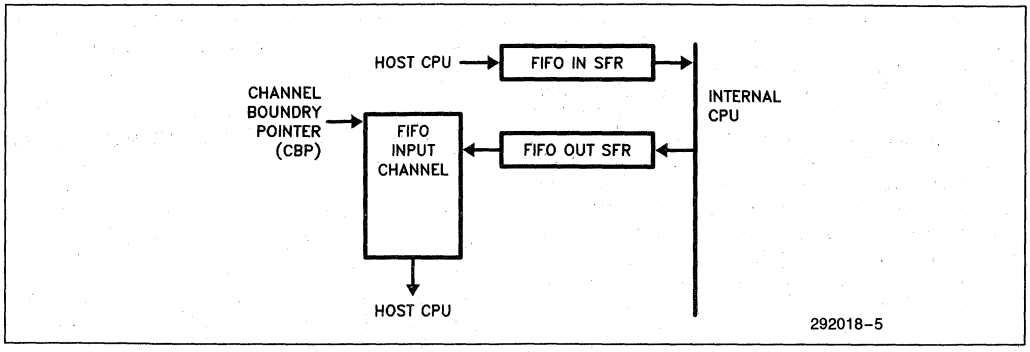


Figure 5. Entire FIFO Assigned to Output Channel

data transfer rate is twice as fast as the internal FIFO DMA data transfer rate. In this example the FIFO Input channel size is programmed to be 64 bytes and the Input channel Threshold is programmed to be 20 bytes. The Host writes the first 64 bytes to the Input FIFO. When the internal DMA processor has read 20 bytes the Threshold interrupt, or DMA request (DRQIN), is generated to signal the Host to begin writing more data to the Input FIFO channel. The internal DMA processor continues to read data from the Input channel as the Host, or Host DMA processor, writes to the FIFO. The Host can write 40 bytes to the FIFO Input channels in the time it takes for the internal DMA processor to read 20 more bytes from it. This will keep both the Host and internal DMA operating at their maximum rates without forcing one to wait for the other.

Two methods of managing the FIFO size are possible; fixed and variable channel size. A fixed channel size is one where the channel is configured at initialization and remains unchanged throughout program execution. In a variable FIFO channel size, the configuration is changed dynamically to meet the data transmission requirements as needed. An example of a variable channel size application is the mass storage subsystem described earlier. To meet the demands of a large data block transfer the FIFO size could be fully allocated to the Input or Output channel as needed. The Thresholds are also reprogrammed to match the respective data transfer rates.

An example of a fixed channel size application might be one which uses the UPI-452 to directly control a series of stepper motors. The UPI-452 manages the motor operation and status as required. This would include pulse train, acceleration, deceleration and feedback. The Host transmits motor commands to the UPI-452 in blocks of 6–10 bytes. Each block of motor command data is preceded by a command to the UPI-452 which selects a specific motor. The UPI-452 transmits blocks of data to the Host which provides motor and overall system status. The data and embedded commands structure, indicating the specific motor, is the same. In this example the default 64 bytes per channel might be adequate for both channels.

INTERRUPT RESPONSE TIMING

Interrupts enable the Host UPI-452 FIFO buffer interface and the internal CPU FIFO buffer interface to operate with a minimum of overhead on the respective CPU. Each CPU is "interrupted" to service the FIFO on an as needed basis only. In configuring the FIFO buffer Thresholds and choosing the appropriate internal DMA Mode the user must take into account the interrupt response time for both CPUs. These response times will affect the DMA transfer rates for each channel. By choosing FIFO channel Thresholds which reflect both the respective DMA transfer rate and the interrupt response time the user will achieve the maximum data throughput and system bus decoupling. This in turn will mean the overall available system bus bandwidth will increase.

The following section describes the FIFO interrupt interface to the Host and internal CPU. It also describes an analysis of sample interrupt response times for the Host and UPI-452 internal CPU. These equations and the example times shown are then used in the DMA section to further analyze an optimum Host UPI-452 interface.

HOST

Interrupts to the Host processor are supported by the three UPI-452 output pins; INTRQ, DRQIN/INTRQIN and DRQOUT/INTRQOUT. INTRQ is a general purpose Request For Service interrupt output. DRQIN/INTRQIN and DRQOUT/INRQOUT pins are multiplexed to provide two special "Request for Service" FIFO interrupt request lines when DMA is disabled. A FIFO Input or Output channel Request for Service interrupt is generated based upon the value programmed in the respective channel's Threshold SFRs; Input Threshold (ITHR), and Output Threshold

(OTHR) SFRs. Additional interrupts are provided for FIFO Underrun and Overrun Errors, Data Stream Commands, and Immediate Commands. Table 4 lists the eight UPI-452 interrupt sources as they appear in the HSTAT SFR to the Host processor.

Table 4. UPI-452 to Host Interrupt Sources

HSTAT SFR Bit	Interrupt Source
HST7	Output FIFO Underrun Error
HST6	Immediate Command Out SFR Status
HST5	Data Stream Command/Data at Output FIFO Status
HST4	Output FIFO Request for Service Status
HST3	Input FIFO Overrun Error Condition
HST2	Immediate Comamnd In SFR Status
HST1	FIFO DMA Freeze/Normal Mode Status
HST0	Input FIFO Request for Service

The interrupt response time required by the Host processor is application and system specific. In general, a typical sequence of Host interrupt response events and the approximate times associated with each are listed in Equation 1.

The example assumes the hardware configuration shown in Figure 1, iAPX 286/UPI-452 Block Diagram, with an 8259A Programmable Interrupt Controller. The timing analysis in Equation 1 also assumes the following; no other interrupt is either in process or pending, nor is the 286 in a LOCK condition. The current instruction completion time is 8 clock cycles (800 ns @ 10 MHz), or 4 bus cycles. The interrupt service routine first executes a PUSHA instruction, PUSH All General Registers, to save all iAPX 286 internal registers. This requires 19 clocks (or 2.0 μs @ 10 MHz), or 10 bus cycles (rounded to complete bus cycle). The next service routine instruction reads the UPI-452 Host Status SFR to determine the interrupt source.

It is important to note that any UPI-452 INTRQ interrupt service routine should ALWAYS mask for the Freeze Mode bit first. This will insure that Freeze Mode always has the highest priority. This will also save the time required to mask for bits which are forced inactive during Freeze Mode, before checking the Freeze Mode bit. Access to the FIFO channels by the Host is inhibited during Freeze Mode. Freeze Mode is covered in more detail below.

To initiate the interrupt the UPI-452 activates the INTRQ output. The interrupt acknowledge sequence requires two bus cycles, 400 ns (10 MHz iAPX 286), for the two INTA pulse sequence.

Equation 1. Host Interrupt Response Time

Action	Time	Bus Cycles*
Current instruction execution completion	800 ns	4
INTA sequence	400 ns	2
Interrupt service routine (time to host first $\overline{\text{READ}}$ of UPI-452)	2000 ns	10
Total Interrupt Response Time	2.3 μs	16

NOTE:
10 MHz iAPX 286 bus cycle, 200 ns each

UPI-452 Internal

The internal CPU FIFO interrupt interface is essentially identical to that of the Host to the FIFO. Three internal interrupt sources support the FIFO operation; FIFO-Slave bus Interface Buffer, DMA Channel 0 and DMA Channel 1 Requests. These interrupts provide a maximum decoupling of the FIFO buffer and the internal CPU. The four different internal DMA Modes available add flexibility to the interface.

The FIFO-Slave Bus Interface interrupt response is also similar to the Host response to an INTRQ Request for Service interrupt. The internal CPU responds to the interrupt by reading the Slave Status (SSTAT) SFR to determine the source of the interrupt. This allows the user to prioritize the Slave Status flag response to meet the users application needs.

The internal interrupt response time is dependent on the current instruction execution, whether the interrupt is enabled, and the interrupt priority. In general, to finish execution of the current instruction, respond to the interrupt request, push the Program Counter (PC) and vector to the first instruction of the interrupt service routine requires from 38 to 86 oscillator periods (2.38 to 5.38 μs @ 16 MHz). If the interrupt is due to an Immediate Command or DSC, additional time is required to read the Immediate Command or DSC SFR and vector to the appropriate service routine. This means two service routines back to back. One service routine to read the Slave Status (SSTAT) SFR to determine the source of the Request for Service interrupt, and second the service routine pointed to by the Immediate Command or DSC byte read from the respective SFR.

DMA

DMA is the fastest and most efficient way for the Host or internal CPU to communicate with the FIFO buffer. The UPI-452 provides support for both of these DMA paths. The two DMA paths and operations are fully independent of each other and can function simultaneously. While the Host DMA processor is performing a DMA transfer to or from the FIFO, the UPI-452 internal DMA processor can be doing the same.

Below are descriptions of both the Host and internal DMA operations. Both DMA paths can operate asynchronously and at different transfer rates. In order to make the most of this simultaneous asynchronous operation it is necessary to calculate the two transfer rates and accurately match their operations. Matching the different transfer rates is done by a combination of accurately programmed FIFO channel size and channel Thresholds. This provides the maximum Host and internal CPU to FIFO buffer interface decoupling. Below is a description of each of the two DMA operations and sample calculations for determining transfer rates. The next section of this application note, "Interface Latency", details the considerations involved in analyzing effective transfer rates when the overhead associated with each transfer is considered.

HOST FIFO DMA

DMA transfers between the Host and UPI-452 FIFO buffer are controlled by the Host CPU's DMA controller, and is independent of the UPI-452's internal two channel DMA processor. The UPI-452's internal DMA processor supports data transfers between the UPI-452 internal RAM, external RAM (via the Local Expansion Bus) and the various Special Function Registers including the FIFO Input and Output channel SFRs.

The maximum DMA transfer rate is achieved by the minimum DMA transfer cycle time to accomplish a source to destination move. The minimum Host UPI-452 FIFO DMA cycle time possible is determined by the $\overline{READ}$ and $\overline{WRITE}$ pulse widths, UPI-452 command recovery times in relation to the DMA transfer timing and DMA controller transfer mode used. Table 5 shows the relationship between the iAPX-286, iAPX-186 and UPI-452 for various DMA as well as non-DMA byte by byte transfer modes versus processor frequencies.

Host processor speed vs wait states required with UPI-452 running at 16 MHz:

Table 5. Host UPI-452 Data Transfer Performance

Processor & Speed	Wait States: Back to Back $\overline{READ}$/$\overline{WRITE}$'s	DMA: Single Cycle	Two Cycle
iAPX-186* 8 MHz	0	N/A*	0
10 MHz	0	N/A*	0
12.5 MHz	1	N/A*	0
iAPX-286** 6 MHz	0	0	0
8 MHz	1	1	0
10 MHz	2	2	0

NOTES:
* iAPX 186 On-chip DMA processor is two cycle operation only.
** iAPX 286 assumes 82258 ADMA (or other DMA) running 286 bus cycles at 286 clock rate.

In this application note system example, shown in Figure 1, DMA operation is assumed to be two bus cycle I/O to memory or memory to I/O. Two cycle DMA consists of a fetch bus cycle from the source and a store bus cycle to the destination. The data is stored in the DMA controller's registers before being sent to the destination. Single cycle DMA transfers involve a simultaneous fetch from the source and store to the destination. As the most common method of I/O-memory DMA operation, two cycle DMA transfers are the focus of this application note analysis. Equation 2 illustrates a calculation of the overall transfer rate between the FIFO buffer and external Host for a maximum FIFO size transfer. The equation does not account for the latency of initiating the DMA transfer.

Equation 2. Host FIFO DMA Transfer Rate—Input or Output Channel

2 Cycle DMA Transfer-I/O (UPI-452) to System Memory
= FIFO channel size* (DMA $\overline{READ}$/$\overline{WRITE}$ FIFO time + DMA $\overline{WRITE}$/$\overline{READ}$ Memory Time)
= 128 bytes* (200 ns + 200 ns)
= 51.2 μs
= 256 bus cycles*

NOTES:
*10 MHz iAPX 286, 200 ns bus cycles.

The UPI-452 design is optimized for high speed DMA transfers between the Host and the FIFO buffer. The

UPI-452 internal FIFO buffer control logic provides the necessary synchronization of the external Host event and the internal CPU machine cycle during UPI-452 $\overline{RD}/\overline{WR}$ accesses. This internal synchronization is addressed by the TCC AC specification of the UPI-452 shown in Figure 6. TCC is the time from the leading or trailing edge of a UPI-452 $\overline{RD}/\overline{WR}$ to the same edge of the next UPI-452 $\overline{RD}/\overline{WR}$. The TCC time is effectively another way of measuring the system bus cycle time with reference to UPI-452 accesses.

In the iAPX-286 10 MHz system depicted in this application note the bus cycle time is 200 ns. Alternate cycle accesses of the UPI-452 during two cycle DMA operation yields a TCC time of 400 ns which is more than the TCC minimum time of 375 ns. Back to back Host UPI-452 $\overline{READ}/\overline{WRITE}$ accesses may require wait states as shown in Table 5. The difference between 10 MHz iAPX-186 and 10 MHz iAPX 286 required wait states is due to the number of clock cycles in the respective bus cycle timings. The four clocks in a 10 MHz iAPX 186 bus cycle means a minimum TCC time of 400 ns versus 200 ns for a 10 MHz iAPX 286 with two clock cycle zero wait state bus cycle.

DMA handshaking between the Host DMA controller and the UPI-452 is supported by three pins on the UPI-452; DRQIN/INTRQIN, DRQOUT/INTRQOUT and $\overline{DACK}$. The DRQIN/INTRQIN and DRQOUT/INTRQOUT outputs are two multiplexed DMA or interrupt request pins. The function of these pins is controlled by MODE SFR bit 6 (MD6). DRQIN and DRQOUT provide a direct interface to the Host system DMA controller (see Figure 1). In response to a DRQIN or DRQOUT request, the Host DMA controller initiates control of the system bus using HLD/HLDA. The FIFO Input or Output channel transfer is accomplished with a minimum of Host overhead and system bus bandwidth.

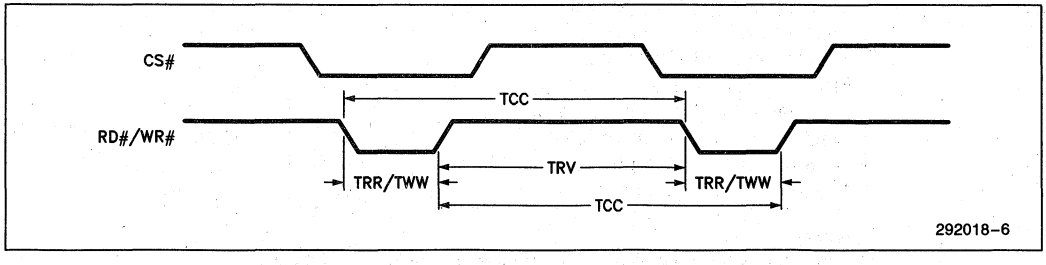
The third handshake signal pin is $\overline{DACK}$ which is used as a chip select during DMA data transfers. The UPI-452 Host $\overline{READ}$ and $\overline{WRITE}$ input signals select the respective Input and Output FIFO channel during DMA transfers. The $\overline{CS}$ and address lines provide DMA acknowledge for processors with onboard DMA controllers which do not generate a $\overline{DACK}$ signal.

The iAPX 286 Block I/O Instructions provide an alternative to two cycle DMA data transfers with approximately the same data rate. The String Input and Output instructions (INS & OUTS) when combined with the Repeat (REP) prefix, modifies INS and OUTS to provide a means of transferring blocks of data between I/O and Memory. The data transfer rate using REP INS/OUTS instructions is calculated in the same way as two cycle DMA transfer times. Each $\overline{READ}$ or $\overline{WRITE}$ would be 200 ns in a 10 MHz iAPX 286 system. The maximum transfer rate possible is 2.5 MBytes/second. The Block I/O FIFO data transfer calculation is the same as that shown in Equation 2 for two cycle DMA data transfers including TCC timing effects.

FIFO Data Structure and Host DMA

During a Host DMA write to the FIFO, if a DSC is to be written, the DMA transfer is stopped, the DSC is written and the DMA restarted. During a Host DMA read from the FIFO, if a DSC is loaded into the I/O Buffer Latch the DMA request, DRQOUT, will be deactivated (see Figure 2 above). The Host Status (HSTAT) SFR Data Stream Command bit is set and the INTRQ interrupt output goes active, if enabled. The Host responds to the interrupt as described above.

6



Symbol	Description	Var. Osc.	@16 MHz
TCC	Command Cycle Time	6 * Tclcl	375 ns min
TRV	Command Recovery Time	75	75 ns min

Figure 6. UPI-452 Command Cycle Timing

Once INTRQ is deactivated and the DSC has been read by the Host, the DMA request, $\overline{DRQOUT}$, is reasserted by the UPI-452. The DMA request then remains active until the transfer is complete or another DSC is loaded into the I/O Buffer Latch.

An Immediate Command written by the internal CPU during a Host DMA FIFO transfer also causes the Host Status flag and INTRQ to go active if enabled. In this case the Immediate Command would not terminate the DMA transfer unless terminated by the Host. The INTRQ line remains active until the Host reads the Host Status (HSTAT) SFR to determine the source of the interrupt.

The net effect of a Data Stream Command (DSC) on DMA data transfer rates is to add an additional factor to the data transfer rate equation. This added factor is shown in Equation 3. An Immediate Command has the same effect on the data transfer rate if the Immediate Command interrupt is recognized by the Host during a DMA transfer. If the DMA transfer is completed before the Immediate Command interrupt is recognized, the effect on the DMA transfer rate depends on whether the block being transmitted is larger than the FIFO channel size. If the block is larger than the programmed FIFO channel size the transfer rate depends on whether the Immediate Command flag or interrupt is recognized between partial block transfers.

The FIFO configuration shown in Equation 3 is arbitrary since there is no way of predicting the size relative to when a DSC would be loaded into the I/O Buffer Latch. The Host DMA rate shown is for a UPI-452 (Memory Mapped or I/O) to 286 System Memory transfer as described earlier. The equations do not account for the latency of intiating the DMA transfer.

Equation 3. Minimum host FIFO DMA Transfer Rate Including Data Stream Command(s)

Minimum Host/FIFO DMA Transfer Rate w/ DSC
= FIFO size* Host DMA 2 cycle time transfer rate + iAPX 286 interrupt response time (Eq. #1)
= (32 bytes* (200 ns + 200 ns)) + 2.3 μs
= 15.1 μs
= 75.5 bus cycles (@10 MHz iAPX286, 200 ns bus cycle)

UPI-452 INTERNAL DMA PROCESSOR

The two identical internal DMA channels allow high speed data transfers from one UPI-452 writable memory space to another. The following UPI-452 memory spaces can be used with internal DMA channels:

Internal Data Memory (RAM)
External Data Memory (RAM)
Special Function Registers (SFR)

The FIFO can be accessed during internal DMA operations by specifying the FIFO IN (FIN) SFR as the DMA Source Address (SAR) or the FIFO OUT (FOUT) SFR as the Destination Address (DAR). Table 6 lists the four types of internal DMA transfers and their respective timings.

Table 6. UPI-452 Internal DMA Controller Cycle Timings

Source	Destination	Machine Cycles**	@12 MHz	@16 MHz
Internal Data Mem. or SFR	Internal Data Mem. or SFR	1	1 μs	750 ns
Internal Data Mem. or SFR	External Data Mem.	1	1 μs	750 ns
External Data Mem.	Internal Data Mem. or SFR	1	1 μs	750 ns
*External Data Memory	External Data Memory	2	2 μs	1.5 μs

NOTES:
*External Data Memory DMA transfer applies to UPI-452 Local Bus only.
**MSC-51 Machine cycle = 12 clock cycles (TCLCL).

FIFO Data Structure and Internal DMA

The effect of Data Stream Commands and Immediate Commands on the internal DMA transfers is essentially the same as the effect on Host FIFO DMA transfers. Recognition also depends upon the programmed DMA Mode, the interrupts enabled, and their priorities. The net internal effect is the same for each possible internal case. The time required to respond to the Immediate or Data Stream Command is a function of the instruction time required. This must be calculated by the user based on the instruction cycle time given in the MSC-51 Instruction Set description in the Intel Microcontroller Handbook.

It is important to note that the internal DMA processor modes and the internal FIFO logic work together to automatically manage internal DMA transfers as data moves into and out of the FIFO. The two most appropriate internal DMA processor modes for the FIFO are FIFO Demand Mode and FIFO Alternate Cycle Mode. In FIFO Demand Mode, once the correct Slave Control and DMA Mode bits are set, the internal Input FIFO channel DMA transfer occurs whenever the Slave Control Input FIFO Request for Service flag is set. The DMA transfer continues until the flag is cleared or when the Input FIFO Read Pointer SFR (IRPR) equals zero. If data continues to be entered by the Host, the internal DMA continues until an internal interrupt of higher priority, if enabled, interrupts the DMA transfer, the internal DMA byte count reaches zero or until the Input FIFO Read Pointer equals zero. A complete description of interrupts and DMA Modes can be found in the UPI-452 Data Sheet.

DMA Modes

The internal DMA processor has four modes of operation. Each DMA channel is software programmable to operate in either Block Mode or Demand Mode. Demand Mode may be further programmed to operate in Burst or Alternate Cycle Mode. Burst Mode causes the internal processor to halt its execution and dedicate its resources exclusively to the DMA transfer. Alternate Cycle Mode causes DMA cycles and instruction cycles to occur alternately. A detailed description of each DMA Mode can be found in the UPI-452 Data Sheet.

INTERFACE LATENCY

The interface latency is the time required to accommodate all of the overhead associated with an individual data transfer. Data transfer rates between the Host system and UPI-452 FIFO, with a block size less than or equal to the programmed FIFO channel size, are calculated using the Host system DMA rate. (see Host DMA description above). In this case, the entire block could be transferred in one operation. The total latency is the time required to accomplish the block DMA transfer, the interrupt response or poll of the Host Status SFR response time, and the time required to initate the Host DMA processor.

A DMA transfer between the Host and UPI-452 FIFO with a block size greater than the programmed FIFO channel size introduces additional overhead. This additional overhead is from three sources; first, is the time to actually perform the DMA transfer. Second, the overhead of initializing the DMA processor, third, the handshaking between each FIFO block required to transfer the entire data block. This could be time to wait for the FIFO to be emptied and/or the interrupt response time to restart the DMA transfer of the next portion of the block. A fourth component may also be the time required to respond to Underrun and Overrun FIFO Errors.

Table 7 shows six typical FIFO Input/Output channel sizes and the Host DMA transfers times for each. The timings shown reflect a 10 MHz system bus two cycle I/O to Memory DMA transfer rate of 2.5 MBytes/second as shown in Equation 1. The times given would be the same for iAPX 286 I/O block move instructions REP INS and REP OUTS as described earlier.

Table 7. Host DMA FIFO Data Transfer Times

FIFO Size:	32	43	64	85	96	128	bytes
Full or Empty	1/4	1/3	1/2	2/3	3/4	Full or Empty	
Time	12.8	17.2	25.6	34.0	38.4	51.2	μs

Table 8 shows six typical FIFO Input/Output channel sizes and the internal DMA processor data transfers times for each. The timings shown are for a UPI-452 single cycle Burst Mode transfer at 16 MHz or 750 ns per machine cycle in or out of the FIFO channels. The

machine cycle time is that of the MSC-51 CPU; 6 states, 2 XTAL2 clock cycles each or 12 clock cycles per machine cycle. Details on the MSC-51 machine cycle timings and operation may be found in the Intel Microcontroller Handbook.

Table 8. UPI-452 Internal DMA FIFO Data Transfer Times

FIFO Size:	32	43	64	85	96	128	bytes
Full or Empty	1/4	1/3	1/2	2/3	3/4	Full or Empty	
Time	24.0	32.3	48.0	64.6	72.0	96.0	μs

A larger than programmed FIFO channel size data block internal DMA transfer requires internal arbitration. The UPI-452 provides a variety of features which support arbitration between the two internal DMA channels and the FIFO. An example is the internal DMA processor FIFO Demand Mode described above. FIFO Demand Mode DMA transfers occur continuously until the Slave Status Request for Service Flag is deactivated. Demand Mode is especially useful for continuous data transfers requiring immediate attention. FIFO Alternate Cycle Mode provides for interleaving DMA transfers and instruction cycles to achieve a maximum of software flexibility. Both internal DMA channels can be used simultaneously to provide continuous transfer for both Input and Output FIFO channels.

Byte by byte transfers between the FIFO and internal CPU timing is a function of the specific instruction cycle time. Of the 111 MCS-51 instructions, 64 require 12 clock cycles, 45 require 24 clock cycles and 2 require 48 clock cycles. Most instructions involving SFRs are 24 clock cycles except accumulator (for example, MOV direct, A) or logical operations (ANL direct, A). Typical instruction and their timings are shown in Table 9.

Oscillator Period: @ 12 MHz = 83.3 ns
@ 16 MHz = 62.5 ns

Table 9. Typical Instruction Cycle Timings

Instruction	Oscillator Periods	@12 MHz	@16 MHz
MOV direct†, A	12	1 μs	750 ns
MOV direct, direct	24	2 μs	1.5 μs

NOTE:
† Direct = 8-bit internal data locations address. This could be an Internal Data RAM location (0–255) or a SFR [i.e., I/O port, control register, etc.]

Byte by byte FIFO data transfers introduce an additional overhead factor not found in internal DMA operations. This factor is the FIFO block size to be transferred; the number of empty locations in the Output channel, or the number of bytes in the Input FIFO channel. As described above in the FIFO Data Structure section, the block size would have to be determined by reading the channel read and write pointer and calculating the space available. Another alternative uses the FIFO Overrun and Underrun Error flags to manage the transfers by accepting error flags. In either case the instructions needed have a significant impact on the internal FIFO data transfer rate latency equation.

A typical effective internal FIFO channel transfer rate using internal DMA is shown in Equation 4. Equation 5 shows the latency using byte by byte transfers with an arbitrary factor added for internal CPU block size calculation. These two equations contrast the effective transfer rates when using internal DMA versus individual instructions to transfer 128 bytes. The effective transfer rate is approximately four times as fast using DMA versus using individual instructions (96 μs with DMA versus 492 μs non-DMA).

Equation 4. Effective Internal FIFO Transfer Time Using Internal DMA

Effective Internal FIFO Transfer Rate with DMA
= FIFO channel size * Internal DMA Burst Mode Single Cycle DMA Time
= 128 Bytes * 750 ns
= 96 μs

Equation 5. Effective FIFO Transfer Time Using Individual Instructions

Effective Internal FIFO Transfer Rate without DMA
= FIFO channel size * Instruction Cycle Time + Block size calculation Time
= 128 Bytes * (24 oscillator periods @ 16 MHz) + 20 instructions (24 oscillator period each @ 16 MHz)
= 128 * 1.5 μs + 300 μs
= 492 μs

FIFO DMA FREEZE MODE INTERFACE

FIFO DMA Freeze Mode provides a means of locking the Host out of the FIFO Input and Output channels. FIFO DMA Freeze Mode can be invoked for a variety of reasons, for example, to reconfigure the UPI-452 Local Expansion Bus, or change the baud rate on the serial channel. The primary reason the FIFO DMA Freeze Mode is provided is to ensure that the Host does not read from or write to the FIFO while the FIFO interface is being altered. ONLY the internal CPU has the capability of altering the FIFO Special Function Registers, and these SFRs can ONLY be altered during FIFO DMA Freeze Mode. FIFO DMA Freeze Mode inhibits Host access of the FIFO while the internal CPU reconfigures the FIFO.

FIFO DMA Freeze Mode should not be arbitrarily invoked while the UPI-452 is in normal operation. Because the external CPU runs asynchronously to the internal CPU, invoking freeze mode without first properly resolving the FIFO Host interface may have serious consequences. Freeze Mode may be invoked only by the internal CPU.

The internal CPU invokes Freeze Mode by setting bit 3 of the Slave Control SFR (SC3). This automatically forces the Slave and Host Status SFR FIFO DMA Freeze Mode to In Progress (SSTAT SST5 = 0, HSTAT SFR HST1 = 1). INTRQ goes active, if enabled by MODE SFR bit 4, whenever FIFO DMA Freeze Mode is invoked to notify the Host. The Host reads the Host Status SFR to determine the source of the interrupt. INTRQ and the Slave and Host Status FIFO DMA Freeze Mode bits are reset by the Host READ of the Host Status SFR.

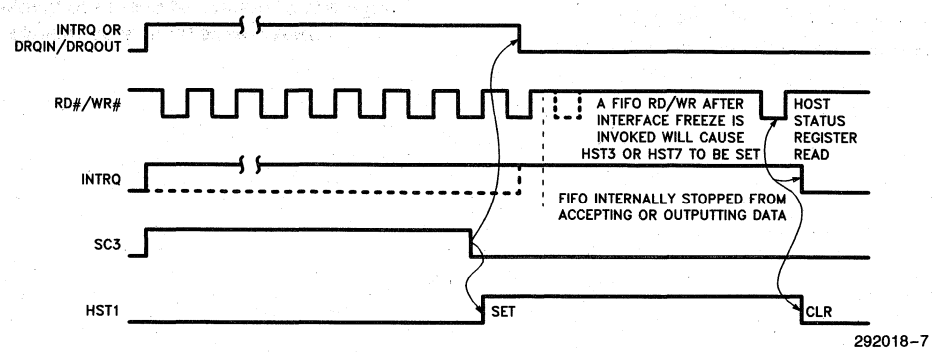
During FIFO DMA Freeze Mode the Host has access to the Host Status and Control SFRs. All other Host FIFO interface access is inhibited. Table 10 lists the FIFO DMA Freeze Mode status of all slave bus interface Special Function Registers. The internal DMA processor is disabled during FIFO DMA Freeze Mode and the internal CPU has write access to all of the FIFO control SFRs (Table 11).

If FIFO DMA Freeze Mode is invoked without stopping the host, only the last two bytes of data written into or read from the FIFO will be valid. The timing diagram for disabling the FIFO module to the external Host interface is illustrated in Figure 7. Due to this synchronization sequence, the UPI-452 might not go into FIFO DMA Freeze Mode immediately after the Slave Control SFR FIFO 7 DMA Freeze Mode bit (SC3) is set = 0. A special bit in the Slave Status SFR (SST5) is provided to indicate the status of the FIFO DMA Freeze Mode. The FIFO DMA Freeze Mode operations described in this section are only valid after SST5 is cleared.

Either the Host or internal CPU can request FIFO DMA Freeze Mode. The first step is to issue an Immediate Command indicating that FIFO DMA Freeze Mode will be invoked. Upon receiving the Immediate Command, the external CPU should complete servicing all pending interrupts and DMA requests, then send an Immediate Command back to the internal CPU acknowledging the FIFO DMA Freeze Mode request. After issuing the first Immediate Command, the internal CPU should not perform any action on the FIFO until FIFO DMA Freeze Mode is invoked. The handshaking is the same in reverse if the HOST CPU initiates FIFO DMA Freeze Mode.

After the slave bus interface is frozen, the internal CPU can perform the operations listed below on the FIFO Special Function Registers. These operations are allowed only during FIFO DMA Freeze Mode. Table 11 summarizes the characteristics of all the FIFO Special Function Registers during Normal and FIFO DMA Freeze Modes.

For FIFO Reconfiguration	1. Changing the Channel Boundary Pointer SFR.
	2. Changing the Input and Output Threshold SFR.
To Enhance the testability	3. Writing to the read and write pointers of the Input and Output FIFO's.
	4. Writing to and reading the Host Control SFRs.
	5. Controlling some bits of Host and Slave Status SFRs.
	6. Reading the Immediate Command Out SFR and Writing to the Immediate Command in SFR.



NOTE:
Timing Diagram of disabling of FIFO Module—External Host Interface.

Figure 7. Disabling FIFO to Host Slave Interface Timing Diagram

The sequence of events for invoking FIFO DMA Freeze Mode are listed in Figure 8.

1. Immediate Command to request FIFO DMA Freeze Mode (interrupt)
2. Host/internal CPU interrupt response/service
3. Host/internal CPU clear/service all pending interrupts and FIFO data
4. Internal CPU sets Slave Control (SLCON) FIFO DMA

   Freeze Mode bit = 0, FIFO DMA Freeze Mode, Host Status SFR FIFO DMA Freeze Mode Status bit = 1, INTRQ active (high)
5. Host $\overline{\text{READ}}$ Host Status SFR
6. Internal CPU reconfigures FIFO SFRs
7. Internal CPU resets Slave Control (SLCON) FIFO DMA

   Freeze Mode bit = 1, Normal Mode, Host Status FIFO DMA Freeze Mode Status bit = 0.
8. Internal CPU issues Immediate Command to Host indicating that FIFO DMA Freeze Mode is complete

   or

   Host polls Host Status SFR FIFO DMA Freeze Mode bit to determine end of reconfiguration

Figure 8. Sequence of Events to Invoke FIFO DMA Freeze Mode

EXAMPLE CONFIGURATION

An example of the time required to reconfigure the FIFO 180 degrees, for example from 128 bytes Input to 128 bytes Output, is shown in Figure 9. The example approximates the time based on several assumptions;

1. The FIFO Input channel is full-128 bytes of data
2. Output FIFO channel is empty-1 byte
3. No Data Stream Commands in the FIFO.
4. The Immediate Command interrupt is responded to immediately—highest priority—by Host and internal CPU.
5. Respective interrupt response times
   a. Host (Equation 3 above) = approximately 1.6 μs
   b. Internal CPU is 86 oscillator periods or approximately 5.38 μs worst case.

Event	Time
Immediate Command from Host to UPI-452 to request FIFO DMA Freeze Mode (iAPX286 $\overline{\text{WRITE}}$)	0.30 μs
Internal CPU interrupt response/service	5.38 μs
Internal CPU clears FIFO-128 bytes DMA	96.00 μs
Internal CPU sets Slave Control Freeze Mode bit	0.75 μs
Immediate Command to Host-Freeze Mode in progress Host Immediate Command interrupt response	2.3 μs
Internal CPU reconfigures FIFO SFRs	
Channel Boundary Pointer SFR	0.75 μs
Input Threshold SFR	0.75 μs
Output Threshold SFR	0.75 μs
Internal CPU resets Slave Control (SLCON) Freeze Mode bit = 1, Normal Mode, and automatically resets Host Status FIFO DMA Freeze Mode bit	2.3 μs
Internal CPU writes Immediate Command Out	0.75 μs
Host Immediate Command interrupt service	2.3 μs
Total Minimum Time to Reconfigure FIFO	112.33 μs

Figure 9. Sequence of Events to Invoke FIFO DMA Freeze Mode and Timings

Table 10. Slave Bus Interface Status During FIFO DMA Freezer Mode

Interface Pins;							Operation In Normal Mode	Status In Freeze Mode
$\overline{DACK}$	$\overline{CS}$	A2	A1	A0	$\overline{READ}$	$\overline{WRITE}$		
1	0	0	1	0	0	1	Read Host Status SFR	Operational
1	0	0	1	1	0	1	Read Host Control SFR	Operational
1	0	0	1	1	1	0	Write Host Control SFR	Disabled
1	0	0	0	0	0	1	Data or DMA data from Output Channel	Disabled
1	0	0	0	0	1	0	Data or DMA data to Input Channel	Disabled
1	0	0	0	1	0	1	Data Stream Command from Output Channel	Disabled
1	0	0	0	1	1	0	Data Stream Command to Input Channel	Disabled
1	0	1	0	0	0	1	Read Immediate Command Out from Output Channel	Disabled
1	0	1	0	0	1	0	Write Immediate Command In to Input Channel	Disabled
0	X	X	X	X	0	1	DMA Data from Output Channel	Disabled
0	X	X	X	X	1	0	DMA Data to Input Channel	Disabled

NOTE:
X = don't care

Table 11. FIFO SFR's Characteristics During FIFO DMA Freeze Mode

Label	Name	Normal Operation (SST5 = 1)	Freeze Mode Operation (SST5 = 0)
HCON	Host Control	Not Accessible	Read & Write
HSTAT	Host Status	Read Only	Read & Write
SLCON	Slave Control	Read & Write	Read & Write
SSTAT	Slave Status	Read Only	Read & Write
IEP	Interrupt Enable & Priority	Read & Write	Read & Write
MODE	Mode Register	Read & Write	Read & Write
IWPR	Input FIFO Write Pointer	Read Only	Read & Write
IRPR	Input FIFO Read Pointer	Read Only	Read & Write
OWPR	Output FIFO Write Pointer	Read Only	Read & Write
ORPR	Output FIFO Read Pointer	Read Only	Read & Write
CBP	Channel Boundary Pointer	Read Only	Read & Write
IMIN	Immediate Command In	Read Only	Read & Write
IMONT	Immediate Command Out	Read & Write	Read & Write
FIN	FIFO IN	Read Only	Read Only
CIN	COMMAND IN	Read Only	Read Only
FOUT	FIFO OUT	Read & Write	Read & Write
COUT	COMMAND OUT	Read & Write	Read & Write
ITHR	Input FIFO Threshold	Read Only	Read & Write
OTHR	Other FIFO Threshold	Read Only	Read & Write

iUP-200A/iUP-201A UNIVERSAL PROM PROGRAMMERS

MAJOR iUP-200A/iUP-201A FEATURES:

- **Provides Programming Support for Intel and Intel-Compatible EPROMs, PLDs, Microcontrollers, and Peripherals**
- **PROM Programming Software (iPPS) Makes Programming Easy with IBM PC XT*, PC AT*, and PC Compatibles**
- **Supports Personality Modules and GUPI Base W/Adaptors**
- **iUP-200A Provides On-Line Operation with a Built-In Serial RS232 Interface and Software for a PC Environment**
- **iUP-201A Provides Same On-Line Performance and Adds Keyboard and Display for Stand-Alone Use**
- **iUP-201A Stand-Alone Capability Includes Device Previewing, Editing, Duplication, and Download from any Source Over RS232C Port**

The iUP-200A and iUP-201A universal programmers program and verify data in Intel and Intel compatible, programmable devices. The iUP-200A and iUP-201A universal programmers provide on-line programming and verification in a growing variety of development environments using the Intel PROM programming software (iPPS). In addition, the iUP-201A universal programmer supports off-line, stand-alone program editing, duplication, and memory locking. The iUP-200A universal programmer is expandable to an iUP-201A model.

210319-1

These products manufactured by Intel Puerto Rico, Inc.

*IBM PC XT and PC AT are registered trademarks of International Business Machines Corporation.

September 1990
Order Number: 210319-007

FUNCTIONAL DESCRIPTION

The iUP-200A universal programmer operates in on-line mode. The iUP-201A universal programmer operates in both on-line and off-line mode.

On-Line System Hardware

The iUP-200A and iUP-201A universal programmers are free-standing units that, when connected to a host computer with at least 64K bytes of memory, provide on-line programming and verification of Intel programmable devices. In addition, the universal programmer can read the contents of the ROM versions of supported devices.

The universal programmer communicates with the host through a standard RS232C serial data link. Different versions of the iUP-200A and iUP-201A are equipped with different cables, including the cable most commonly used for interfacing to that host. Care should be taken that the version with the correct cable for your particular system is selected, as cable requirements can vary with your host configuration.

Each universal programmer contains the CPU, selectable power supply, static RAM, programmable timer, interface for personality modules, RS232C interface for the host system, and control firmware in EPROM. The iUP-201A also has a keyboard and display.

A personality module or GUPI Adaptor adapts the universal programmer to a family of devices; it contains all the hardware and software necessary to program either a family of Intel devices or a single Intel device. The user inserts the personality module into the universal programmer front panel.

On-Line System Software

The iUP-200A and iUP201A includes your choice of one copy of Intel's PROM Programming software iPPS, selected from a list of versions for different operating systems and hosts. Each version includes the software implementation designed for that host and O.S. and the RS232C cable most commonly used. Additional versions may be purchased separately if you decide to change hosts at a later date. The iPPS software provides user control through an easy-to-use interactive interface. The iPPS software performs the following functions to make EPROM programming quick and easy:

- Reads devices
- Programs devices directly or from a file
- Verifies device data with buffer data
- Locks device memory from unauthorized access (on devices which support this feature)
- Prints device contents on the network or development system printer
- Performs interactive formatting operations such as interleaving, nibble swapping, bit reversal, and block moves
- Programs multiple devices from the source file, prompting the user to insert new devices
- Uses a buffer to change device contents

All iPPS commands, as well as program address and data information, are entered through the host system ASCII keyboard and displayed on the system CRT.

The iPPS software supports data manipulation in the following Intel formats: 8080 hexadecimal ASCII, 8080 absolute object, 8086 hexadecimal ASCII, 8086 absolute object, and 80286 absolute object. Addresses and data can be displayed in binary, octal, decimal, or hexadecimal. The user can easily change default data formats as well as number bases. iPPS can also access disk files.

For programming Intel EPLDs, the iUP-200A/201A can be controlled by Intel's Logic Programming Software (LPS) and Advanced Programming Tool (APT). LPS and APT program EPLDs from JEDEC files produced by Intel's logic compiler. (iPPS can also program EPLDs, but only from pre-programmed device masters.)

System Expansion

The iUP-200A universal programmer can be easily upgraded (by the user) to an iUP-201A universal programmer for off-line operation. The upgrade kit (iUP-PAK-A) is available from Intel or your local Intel distributor.

Off-Line System

The iUP-201A universal programmer has all the on-line features of the iUP-200A universal programmer plus off-line editing, device duplication, program verification, and locking of device memory independent of the host system. The iUP-201A universal programmer also accepts Intel hexadecimal programs developed on non-Intel development systems. Just a few keystrokes download the program into the iUP RAM for editing and loading into a device.

Off-line commands are entered via a 16-character keypad. A 24-character display shows programmer status.

PERSONALITY MODULES

For some devices, a personality module is the interface between the iUP-200A/iUP-201A universal programmer and a selected device. Personality modules contain all the hardware and firmware for reading and programming a family of Intel devices. Table 1 lists the devices supported by the different modules.

For most devices, the GUPI module and interchangeable GUPI Adaptors provide the interface between the programmer and the device being programmed (see Figure 1). the GUPI (Generic Universal Programmer Interface) module is a base module that intefaces to the iUP-200A/201A and GUPI Adaptors. GUPI Adaptors tailor the GUPI module base signals to a family of devices or an individual device. The GUPI module and GUPI Adaptors provide a lower-cost method of device support than if unique Personality Modules were offered for each device/family. Tables 2 through 4 show which Adaptors support which devices. Note that these tables are current at the time of printing. Contact your Intel sales representative for information on current support.

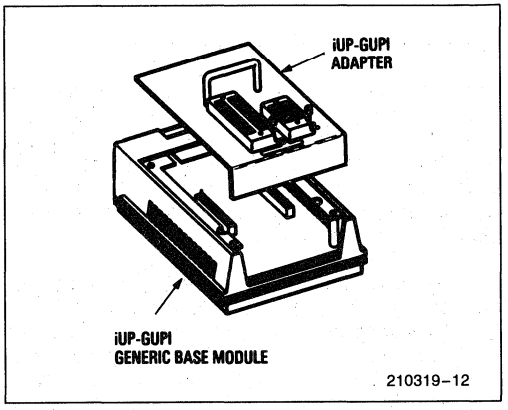


Figure 1. GUPI Adaptor

Table 1. iUP Personality Programming Modules

Device Type	Fast 27/K Module	F27/128 Module	F87/44A Module	F87/51A Module
EPROM	See GUPI EPROM28	2716 2732 2732A 2764 27128		
Microcontroller			8041A 8042 8044AH 8741H 8742 8744H 8755A	8748 8748H 8749H 8751 8751H 8048 8048H 8049 8049H 8050H 8051

*Quick-Pulse Programming™ algorithm

Table 2. Programming Adaptors for Memories

Device Type	GUPI 27010	GUPI 27011	GUPI 27210	GUPI 27960	GUPI EPROM28
EPROM	27010 27C010 27C100 27C020 27C040	27011	27210 27C202 27C210 27C220 27C240 87C75PF	27960CX 27960KX	2764A 27C64 87C64 27128A 27C128 27256 27C256 68C257 87C257 27512 27C512 27513 27C513 27011 27C011 27C021
Package Types	DIP	DIP	DIP		

Table 3. Programming Adaptors for EPLDs

Device Type	GUPI Logic-IID	GUPI 20D20J	GUPI 24D28J	GUPI 40D44J	GUPI Logic-18	GUPI Logic-18PGA	GUPI 85EPLD28
EPLD	5C060 85C060 5C090 85C090 5AC312	85C220	85C224	5AC324	5C180	5C180G	85C508 85C960
Package Types	DIP*	DIP PLCC	DIP PLCC	DIP PLCC	PLCC	PGA	DIP PLCC

*ADAPT units available to adapt DIP socket for PLCC package.

Table 4. Programming Adaptors for Microcontrollers

Device Type	GUPI 8742	GUPI MCS-51	GUPI 8796	GUPI 87C51GB	GUPI 87C51GBP	GUPI MCS-96LCC
Peripheral	8741AH 8742AH					
Microcontroller		8751H 8751BH 87C51 8752BH 87C51FA 87C51FB	8794BH 8795BH 8796BH 8797BH	87C51GB	87C51GB	8796JC 8797BH 87C196KB
Package Types	DIP	PLCC DIP	PGA DIP	LCC	PLCC	LCC

iUP-200A/iUP201A SPECIFICATIONS

Control Processor

Intel 8085A microprocessor
6.144 MHz clock rate

Memory

RAM—4.3 bytes static
ROM—12K bytes EPROM

Interfaces

Keyboard: 16-character hexadecimal and 12-function keypad (iUP-201A model only)

Display: 24-character alphanumeric (iUP-201A model only)

Software

Monitor— system controller in pre-programmed EPROM

iPPS — Intel PROM programming software on supplied diskette

Physical Characteristics

Depth: 15 inches (38.1 cm)
Width: 15 inches (38.1 cm)
Height: 6 inches (15.2 cm)
Weight: 15 pounds (6.9 kg)

Electrical Characteristics

Selectable 100, 120, 200, or 240 Vac ± 10%; 50-60 Hz

Maximum power consumption—80 watts

Environmental Characteristics

Reading Temperature:	10°C to 40°C
Programming Temperature:	25°C ±5°
Operating Humidity:	10% to 85% relative humidity

Reference Material

166041-001— *iUP-200A/201A Universal Programmer User's Guide.*

166042-001— *Getting Started with the iUP-200A/201A (For ISIS/iNDX Users).*

166043-001— *Getting Started with the iUP-200A/201A (For DOS Users).*

164853 — *iUP-200A/201A Universal Programmer Pocket Reference.*

6

ORDERING INFORMATION

Product Order Code	Description
iUP-200A 216D	On-Line PROM programmer with iPPS rel 2.0 for PC/DOS, and cable for PC or XT
iUP-200A 217D	On-Line PROM programmer with iPPS rel 2.0 for PC/DOS, and cable for AT
iUP-201A 216D	Off-Line and on-line PROM programmer with iPPS rel 2.0 for PC/DOS, and cable for PC or XT
iUP-201A 217D	Off-Line and on-line PROM programmer with iPPS rel 2.0 for PC/DOS, and cable for AT
iUP-200/201 U1* Upgrade Kit	Upgrades an iUP-200/201 universal programmer to an iUP-200A/201A universal programmer
iUP-DL	Download Support Kit for iUP-200A/201A upgrades programmer to support adaptors that use software programming (.DSS) files.
iUP-PAK-A Upgrade Kit	Upgrades an iUP-200/A universal programmer to an iUP-201A universal programmer

*Most personality modules can be used only with an iUP-200A/201A universal programmer or an iUP-200/iUP201 universal programmer upgraded to an A with the iUP-200/iUP-201 U1 upgrade kit.

Product Order Code	Description
piUP-GUPI	Generic Universal Programmer Interface (Base)

Software Sold Separately

Product Order Code	Description
216D	PROM programming software rel 2.0 for PC/DOS with cable for PC or PC XT
217D	PROM programming software rel 2.0 for PC/DOS with cable for PC AT

DOMESTIC SALES OFFICES

ALABAMA

Intel Corp.
5015 Bradford Dr., #2
Huntsville 35805
Tel: (205) 830-4010
FAX: (205) 837-2640

ARIZONA

†Intel Corp.
410 North 44th Street
Suite 500
Phoenix 85008
Tel: (602) 231-0386
FAX: (602) 244-0446

Intel Corp.
7225 N. Mona Lisa Rd.
Suite 215
Tucson 85741
Tel: (602) 544-0227
FAX: (602) 544-0232

CALIFORNIA

†Intel Corp.
21515 Vanowen Street
Suite 116
Canoga Park 91303
Tel: (818) 704-8500
FAX: (818) 340-1144

†Intel Corp.
300 N. Continental Blvd.
Suite 100
El Segundo 90245
Tel: (213) 640-6040
FAX: (213) 640-7133

Intel Corp.
1 Sierra Gate Plaza
Suite 280C
Roseville 95678
Tel: (916) 782-8086
FAX: (916) 782-8153

†Intel Corp.
9665 Chesapeake Dr.
Suite 325
San Diego 92123
Tel: (619) 292-8086
FAX: (619) 292-0628

†Intel Corp.*
400 N. Tustin Avenue
Suite 450
Santa Ana 92705
Tel: (714) 835-9642
TWX: 910-595-1114
FAX: (714) 541-9157

†Intel Corp.*
San Tomas 4
2700 San Tomas Expressway
2nd Floor
Santa Clara 95051
Tel: (408) 986-8086
TWX: 910-338-0255
FAX: (408) 727-2620

COLORADO

Intel Corp.
4445 Northpark Drive
Suite 100
Colorado Springs 80907
Tel: (719) 594-6622
FAX: (303) 594-0720

†Intel Corp.*
600 S. Cherry St.
Suite 700
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289
FAX: (303) 322-8670

CONNECTICUT

†Intel Corp.
301 Lee Farm Corporate Park
83 Wooster Heights Rd.
Danbury 06810
Tel: (203) 748-3130
FAX: (203) 794-0339

FLORIDA

†Intel Corp.
800 Fairway Drive
Suite 160
Deerfield Beach 33441
Tel: (305) 421-0506
FAX: (305) 421-2444

†Intel Corp.
5850 T.G. Lee Blvd.
Suite 340
Orlando 32822
Tel: (407) 240-8000
FAX: (407) 240-8097

Intel Corp.
11300 4th Street North
Suite 170
St. Petersburg 33716
Tel: (813) 577-2413
FAX: (813) 578-1607

GEORGIA

†Intel Corp.
20 Technology Parkway
Suite 150
Norcross 30092
Tel: (404) 449-0541
FAX: (404) 605-9762

ILLINOIS

†Intel Corp.*
Woodfield Corp. Center III
300 N. Martingale Road
Suite 400
Schaumburg 60173
Tel: (708) 605-8031
FAX: (708) 706-9762

INDIANA

†Intel Corp.
8910 Purdue Road
Suite 350
Indianapolis 46268
Tel: (317) 875-0623
FAX: (317) 875-8938

IOWA

Intel Corp.
1930 St. Andrews Drive N.E.
2nd Floor
Cedar Rapids 52402
Tel: (319) 393-5510

KANSAS

†Intel Corp.
10985 Cody St.
Suite 140
Overland Park 66210
Tel: (913) 345-2727
FAX: (913) 345-2076

MARYLAND

†Intel Corp.*
10010 Junction Dr.
Suite 200
Annapolis Junction 20701
Tel: (301) 206-2860
FAX: (301) 206-3677
(301) 206-3678

MASSACHUSETTS

†Intel Corp.*
Westford Corp. Center
3 Carlisle Road
2nd Floor
Westford 01886
Tel: (508) 692-0960
TWX: 710-343-6333
FAX: (508) 692-7867

MICHIGAN

†Intel Corp.
7071 Orchard Lake Road
Suite 100
West Bloomfield 48322
Tel: (313) 851-8096
FAX: (313) 851-8770

MINNESOTA

†Intel Corp.
3500 W. 80th St.
Suite 360
Bloomington 55431
Tel: (612) 835-6722
TWX: 910-576-2867
FAX: (612) 831-6497

MISSOURI

†Intel Corp.
4203 Earth City Expressway
Suite 131
Earth City 63045
Tel: (314) 291-1990
FAX: (314) 291-4341

NEW JERSEY

†Intel Corp.*
Parkway 109 Office Center
328 Newman Springs Road
Red Bank 07701
Tel: (201) 747-2233
FAX: (201) 747-0983

Intel Corp.
280 Corporate Center
75 Livingston Avenue
First Floor
Roseland 07068
Tel: (201) 740-0111
FAX: (201) 740-0626

NEW YORK

Intel Corp.*
850 Crosskeys Office Park
Fairport 14450
Tel: (716) 425-2750
TWX: 510-253-7391
FAX: (716) 223-2561

†Intel Corp.*
2950 Express Dr., South
Suite 130
Islandia 11722
Tel: (516) 231-3300
TWX: 510-227-6236
FAX: (516) 348-7939

†Intel Corp.
Westage Business Center
Bldg. 300, Route 9
Fishkill 12524
Tel: (914) 897-3860
FAX: (914) 897-3125

NORTH CAROLINA

†Intel Corp.
5800 Executive Center Dr.
Suite 105
Charlotte 28212
Tel: (704) 568-8966
FAX: (704) 535-2236

†Intel Corp.
5540 Centerview Dr.
Suite 215
Raleigh 27606
Tel: (919) 851-9537
FAX: (919) 851-8974

OHIO

†Intel Corp.*
3401 Park Center Drive
Suite 220
Dayton 45414
Tel: (513) 890-5350
TWX: 810-450-2528
FAX: (513) 890-8658

†Intel Corp.*
25700 Science Park Dr.
Suite 100
Beachwood 44122
Tel: (216) 464-2736
TWX: 810-427-9298
FAX: (804) 282-0673

OKLAHOMA

Intel Corp.
6801 N. Broadway
Suite 115
Oklahoma City 73162
Tel: (405) 848-8086
FAX: (405) 840-9819

OREGON

†Intel Corp.
15254 N.W. Greenbrier Parkway
Building B
Beaverton 97005
Tel: (503) 645-8051
TWX: 910-467-8741
FAX: (503) 645-8181

PENNSYLVANIA

†Intel Corp.*
925 Harvest Drive
Suite 200
Blue Bell 19422
Tel: (215) 641-1000
FAX: (215) 641-0785

†Intel Corp.*
400 Penn Center Blvd.
Suite 610
Pittsburgh 15235
Tel: (412) 823-4970
FAX: (412) 829-7578

PUERTO RICO

†Intel Corp.
South Industrial Park
P.O. Box 910
Las Piedras 00671
Tel: (809) 733-8616

TEXAS

Intel Corp.
8911 Capital of Texas Hwy.
Austin 78759
Tel: (512) 794-8086
FAX: (512) 338-9335

†Intel Corp.*
12000 Ford Road
Suite 400
Dallas 75248
Tel: (214) 241-8087
FAX: (214) 484-1180

†Intel Corp.*
7322 S.W. Freeway
Suite 1490
Houston 77074
Tel: (713) 988-8086
TWX: 910-881-2490
FAX: (713) 988-3660

UTAH

†Intel Corp.
428 East 6400 South
Suite 104
Murray 84107
Tel: (801) 263-8051
FAX: (801) 268-1457

VIRGINIA

†Intel Corp.
1504 Santa Rosa Road
Suite 108
Richmond 23288
Tel: (804) 282-5668
FAX: (216) 464-2270

WASHINGTON

†Intel Corp.
155 108th Avenue N.E.
Suite 386
Bellevue 98004
Tel: (206) 453-8086
TWX: 910-443-3002
FAX: (206) 451-9556

Intel Corp.
408 N. Mullan Road
Suite 102
Spokane 99206
Tel: (509) 928-8086
FAX: (509) 928-9467

WISCONSIN

Intel Corp.
330 S. Executive Dr.
Suite 102
Brookfield 53005
Tel: (414) 784-8087
FAX: (414) 796-2115

CANADA

BRITISH COLUMBIA

Intel Semiconductor of
Canada, Ltd.
4585 Canada Way
Suite 202
Burnaby V5G 4L6
Tel: (604) 298-0387
FAX: (604) 298-8234

ONTARIO

†Intel Semiconductor of
Canada, Ltd.
2650 Queensview Drive
Suite 250
Ottawa K2B 8H6
Tel: (613) 829-9714
FAX: (613) 820-5936

†Intel Semiconductor of
Canada, Ltd.
190 Attwell Drive
Suite 500
Rexdale M9W 6H8
Tel: (416) 675-2105
FAX: (416) 675-2438

QUEBEC

†Intel Semiconductor of
Canada, Ltd.
1 Rue Holiday
Suite 115
Tour East
Pt. Claire H9R 5N3
Tel: (514) 694-9130
FAX: 514-694-0064

†Sales and Service Office
*Field Application Location

DOMESTIC DISTRIBUTORS

ALABAMA

Arrow Electronics, Inc.
1015 Henderson Road
Huntsville 35805
Tel: (205) 837-6955
FAX: 205-751-1581

Hamilton/Avnet Computer
4930 I Corporate Drive
Huntsville 35805

Hamilton/Avnet Electronics
4940 Research Drive
Huntsville 35805
Tel: (205) 837-7210
FAX: 205-721-0356

MTI Systems Sales
4950 Corporate Drive
Suite 120
Huntsville 35806
Tel: (205) 830-9526
FAX: (205) 830-9557

Pioneer/Technologies Group, Inc.
4825 University Square
Huntsville 35805
Tel: (205) 837-9300
FAX: 205-837-9358

ALASKA

Hamilton/Avnet Computer
1400 W. Benson Blvd., Suite 400
Anchorage 99503

ARIZONA

†Arrow Electronics, Inc.
4134 E. Wood Street
Phoenix 85040
Tel: (602) 437-0750
TWX: 910-951-1550

Hamilton/Avnet Computer
30 South McKemy Avenue
Chandler 85226

Hamilton/Avnet Computer
90 South McKemy Road
Chandler 85226

†Hamilton/Avnet Electronics
505 S. Madison Drive
Tempe 85281
Tel: (602) 231-5140
TWX: 910-950-0077

Hamilton/Avnet Electronics
30 South McKemy
Chandler 85226
Tel: (602) 961-6669
FAX: 602-961-4073

Wyle Distribution Group
4141 E. Raymond
Phoenix 85040
Tel: (602) 249-2232
TWX: 910-371-2871

CALIFORNIA

Arrow Commercial System Group
1502 Crocker Avenue
Hayward 94544
Tel: (415) 489-5371
FAX: (415) 489-9393

Arrow Commercial System Group
14242 Chambers Road
Tustin 92680
Tel: (714) 544-0200
FAX: (714) 731-8438

†Arrow Electronics, Inc.
19748 Dearborn Street
Chatsworth 91311
Tel: (213) 701-7500
TWX: 910-493-2086

†Arrow Electronics, Inc.
9511 Ridgehaven Court
San Diego 92123
Tel: (619) 565-4800
FAX: 619-279-8062

†Arrow Electronics, Inc.
521 Weddell Drive
Sunnyvale 94086
Tel: (408) 745-6600
TWX: 910-339-9371

†Arrow Electronics, Inc.
2961 Dow Avenue
Tustin 92680
Tel: (714) 838-5422
TWX: 910-595-2860

Hamilton/Avnet Computer
3170 Pullman Street
Costa Mesa 92626

Hamilton/Avnet Computer
1361B West 190th Street
Gardena 90248

Hamilton/Avnet Computer
4103 Northgate Blvd.
Sacramento 95834

Hamilton/Avnet Computer
4545 Viewridge Avenue
San Diego 92123

Hamilton/Avnet Computer
1175 Bordeaux Drive
Sunnyvale 94089

Hamilton/Avnet Electronics
21150 Califa Street
Woodland Hills 91367

†Hamilton/Avnet Electronics
3170 Pullman Street
Costa Mesa 92626
Tel: (714) 641-4150
TWX: 910-595-2638

†Hamilton/Avnet Electronics
1175 Bordeaux Drive
Sunnyvale 94086
Tel: (408) 743-3300
TWX: 910-339-9332

†Hamilton/Avnet Electronics
4545 Ridgeview Avenue
San Diego 92123
Tel: (619) 571-7500
TWX: 910-595-2638

†Hamilton/Avnet Electronics
21150 Califa St.
Woodland Hills 91376
Tel: (818) 594-0404
FAX: 818-594-8233

†Hamilton/Avnet Electronics
10950 W. Washington Blvd.
Culver City 20230
Tel: (213) 558-2458
TWX: 910-340-6364

†Hamilton/Avnet Electronics
1361B West 190th Street
Gardena 90248
Tel: (213) 217-6700
TWX: 910-340-6364

†Hamilton/Avnet Electronics
4103 Northgate Blvd.
Sacramento 95834
Tel: (916) 920-3150

Pioneer/Technologies Group, Inc.
134 Rio Robles
San Jose 95134
Tel: (408) 954-9100
FAX: 408-954-9113

Wyle Distribution Group
124 Maryland Street
El Segundo 90254
Tel: (213) 322-8100

Wyle Distribution Group
7431 Chapman Ave.
Garden Grove 92641
Tel: (714) 891-1717
FAX: 714-891-1621

†Wyle Distribution Group
2951 Sunrise Blvd., Suite 175
Rancho Cordova 95742
Tel: (916) 638-5282

†Wyle Distribution Group
9525 Chesapeake Drive
San Diego 92123
Tel: (619) 565-9171
TWX: 910-335-1590

†Wyle Distribution Group
3000 Bowers Avenue
Santa Clara 95051
Tel: (408) 727-2500
TWX: 408-988-2747

†Wyle Distribution Group
17872 Cowan Avenue
Irvine 92714
Tel: (714) 863-9953
TWX: 910-371-7127

†Wyle Distribution Group
26677 W. Agoura Rd.
Calabasas 91302
Tel: (818) 880-9000
TWX: 372-9232

COLORADO

Arrow Electronics, Inc.
7060 South Tucson Way
Englewood 80112
Tel: (303) 790-4444

Hamilton/Avnet Computer
9605 Maroon Circle, Ste. 200
Engelwood 80112

†Hamilton/Avnet Electronics
9605 Maroon Circle
Suite 200
Englewood 80112
Tel: (303) 799-0663
TWX: 910-935-0787

†Wyle Distribution Group
451 E. 124th Avenue
Thornton 80241
Tel: (303) 457-9953
TWX: 910-936-0770

CONNECTICUT

†Arrow Electronics, Inc.
12 Beaumont Road
Wallingford 06492
Tel: (203) 265-7741
TWX: 710-476-0162

Hamilton/Avnet Computer
Commerce Industrial Park
Commerce Drive
Danbury 06810

†Hamilton/Avnet Electronics
Commerce Industrial Park
Commerce Drive
Danbury 06810
Tel: (203) 797-2800
TWX: 710-456-9974

†Pioneer/Standard Electronics
112 Main Street
Norwalk 06851
Tel: (203) 853-1515
FAX: 203-838-9901

FLORIDA

†Arrow Electronics, Inc.
400 Fairway Drive
Suite 102
Deerfield Beach 33441
Tel: (305) 429-8200
FAX: 305-428-3991

†Arrow Electronics, Inc.
37 Skyline Drive
Suite 3101
Lake Marv 32746
Tel: (407) 323-0252
FAX: 407-323-3189

Hamilton/Avnet Computer
6801 N.W. 15th Way
Ft. Lauderdale 33309

Hamilton/Avnet Computer
3247 Spring Forest Road
St. Petersburg 33702

†Hamilton/Avnet Electronics
6801 N.W. 15th Way
Ft. Lauderdale 33309
Tel: (305) 971-2900
FAX: 305-971-5420

†Hamilton/Avnet Electronics
3197 Tech Drive North
St. Petersburg 33702
Tel: (813) 573-3930
FAX: 813-572-4329

†Hamilton/Avnet Electronics
6947 University Boulevard
Winter Park 32792
Tel: (407) 628-3888
FAX: 407-678-1878

†Pioneer/Technologies Group, Inc.
337 Northlake Blvd., Suite 1000
Alta Monte Springs 32701
Tel: (407) 834-9090
FAX: (404) 623-0865

Pioneer/Technologies Group, Inc.
674 S. Military Trail
Deerfield Beach 33442
Tel: (305) 428-8877
FAX: 305-481-2950

GEORGIA

Arrow Commercial System Group
3400 C. Corporate Way
Deluth 30139
Tel: (404) 623-8825
FAX: (404) 623-8802

†Arrow Electronics, Inc.
4250 E. Rivergreen Parkway
Deluth 30136
Tel: (404) 497-1300
TWX: 810-766-0439

Hamilton/Avnet Computer
5825 D. Peachtree Corners E.
Norcross 30092

†Hamilton/Avnet Electronics
5825 D Peachtree Corners
Norcross 30092
Tel: (404) 447-7500
TWX: 810-766-0432

Pioneer/Technologies Group, Inc.
3100 F Northwoods Place
Norcross 30071
Tel: (404) 448-1711
FAX: 404-446-8270

ILLINOIS

†Arrow Electronics, Inc.
1140 W. Thorndale
Itasca 60143
Tel: (708) 250-0500
TWX: 708-250-0916

Hamilton/Avnet Computer
1130 Thorndale Avenue
Bensenville 60106

†Hamilton/Avnet Electronics
1130 Thorndale Avenue
Bensenville 60106
Tel: (708) 860-7780
TWX: 708-860-8530

MTI Systems Sales
1100 W. Thorndale
Itasca 60143
Tel: (708) 773-2300

†Pioneer/Standard Electronics
2171 Executive Dr., Suite 200
Addison 60101
Tel: (708) 495-9680
FAX: 708-495-9831

INDIANA

†Arrow Electronics, Inc.
7108 Lakeview Parkway West Drive
Indianapolis 46268
Tel: (317) 299-2071
FAX: 317-299-0255

Hamilton/Avnet Computer
485 Gradle Drive
Carmel 46032

Hamilton/Avnet Electronics
485 Gradle Drive
Carmel 46032
Tel: (317) 844-9333
FAX: 317-844-5921

†Pioneer/Standard Electronics
9350 Priority Way
West Drive
Indianapolis 46250
Tel: (317) 573-0880
FAX: 317-573-0979

†Certified Technical Distributor

DOMESTIC DISTRIBUTORS (Contd.)

IOWA

Hamilton/Avnet Computer
915 33rd Avenue SW
Cedar Rapids 52404

Hamilton/Avnet Electronics
915 33rd Avenue, S.W.
Cedar Rapids 52404
Tel: (319) 362-4757

KANSAS

Arrow Electronics, Inc.
8208 Melrose Dr., Suite 210
Lenexa 66214
Tel: (913) 541-9542
FAX: 913-541-0328

Hamilton/Avnet Computer
15313 W. 95th Street
Lenexa 61219

†Hamilton/Avnet Electronics
15313 W. 95th
Overland Park 66215
Tel: (913) 888-8900
FAX: 913-541-7951

KENTUCKY

Hamilton/Avnet Electronics
805 A. Newtown Circle
Lexington 40511
Tel: (606) 259-1475

MARYLAND

†Arrow Electronics, Inc.
8300 Guilford Drive
Suite H, River Center
Columbia 21046
Tel: (301) 995-6002
FAX: 301-381-3854

Hamilton/Avnet Computer
6822 Oak Hall Lane
Columbia 21045

†Hamilton/Avnet Electronics
6822 Oak Hall Lane
Columbia 21045
Tel: (301) 995-3500
FAX: 301-995-3593

†Mesa Technology Corp.
9720 Patuxent Woods Dr.
Columbia 21046
Tel: (301) 290-8150
FAX: 301-290-6474

†Pioneer/Technologies Group, Inc.
9100 Gaither Road
Gaithersburg 20877
Tel: (301) 921-0660
FAX: 301-921-4255

MASSACHUSETTS

Arrow Electronics, Inc.
25 Upton Dr.
Wilmington 01887
Tel: (508) 658-0900
TWX: 710-393-6770

Hamilton/Avnet Computer
10 D Centennial Drive
Peabody 01960

†Hamilton/Avnet Electronics
10D Centennial Drive
Peabody 01960
Tel: (508) 532-9838
FAX: 508-596-7802

†Pioneer/Standard Electronics
44 Hartwell Avenue
Lexington 02173
Tel: (617) 861-9200
FAX: 617-863-1547

Wyle Distribution Group
15 Third Avenue
Burlington 01803
Tel: (617) 272-7300
FAX: 617-272-6809

MICHIGAN

†Arrow Electronics, Inc.
19880 Haggerty Road
Livonia 48152
Tel: (313) 665-4100
TWX: 810-223-6020

Hamilton/Avnet Computer
2215 S.E. A-5
Grand Rapids 49508

Hamilton/Avnet Computer
41650 Garden Rd., Ste. 100
Novi 48050

Hamilton/Avnet Electronics
2215 29th Street S.E.
Space A5
Grand Rapids 49508
Tel: (616) 243-8805
FAX: 616-698-1831

Hamilton/Avnet Electronics
41650 Garden Brook
Novi 48050
Tel: (313) 347-4271
FAX: 313-347-4021

†Pioneer/Standard Electronics
4505 Broadmoor S.E.
Grand Rapids 49508
Tel: (616) 698-1800
FAX: 616-698-1831

†Pioneer/Standard Electronics
13485 Stamford
Livonia 48150
Tel: (313) 525-1800
FAX: 313-427-3720

MINNESOTA

†Arrow Electronics, Inc.
5230 W. 73rd Street
Edina 55435
Tel: (612) 830-1800
TWX: 910-576-3125

Hamilton/Avnet Computer
12400 Whitewater Drive
Minnetonka 55343

†Hamilton/Avnet Electronics
12400 Whitewater Drive
Minnetonka 55434
Tel: (612) 932-0600
TWX: 910-576-2720

†Pioneer/Standard Electronics
7625 Golden Triange Dr.
Suite G
Eden Prairie 55343
Tel: (612) 944-3355
FAX: 612-944-3794

MISSOURI

†Arrow Electronics, Inc.
2380 Schuetz
St. Louis 63141
Tel: (314) 567-6888
FAX: 314-567-1164

Hamilton/Avnet Computer
739 Goddard Avenue
Chesterfield 63005

†Hamilton/Avnet Electronics
741 Goddard
Chesterfield 63005
Tel: (314) 537-1600
FAX: 314-537-4248

NEW HAMPSHIRE

Hamilton/Avnet Computer
2 Executive Park Drive
Bedford 03102

Hamilton/Avnet Computer
444 East Industrial Park Dr.
Manchester 03103

NEW JERSEY

†Arrow Electronics, Inc.
4 East Stow Road
Unit 11
Marlton 08053
Tel: (609) 596-8000
FAX: 609-596-9632

†Arrow Electronics
6 Century Drive
Parsipanny 07054
Tel: (201) 538-0900
FAX: 201-538-0900

Hamilton/Avnet Computer
1 Keystone Ave., Bldg. 36
Cherry Hill 08003

Hamilton/Avnet Computer
10 Industrial Road
Fairfield 07006

†Hamilton/Avnet Electronics
1 Keystone Ave., Bldg. 36
Cherry Hill 08003
Tel: (609) 424-0110
FAX: 609-751-2552

†Hamilton/Avnet Electronics
10 Industrial
Fairfield 07006
Tel: (201) 575-3390
FAX: 201-575-5839

†MTI Systems Sales
9 Law Drive
Fairfield 07006
Tel: (201) 227-5552
FAX: 201-575-6336

†Pioneer/Standard Electronics
14-A Madison Rd.
Fairfield 07006
Tel: (201) 575-3510
FAX: 201-575-3454

NEW MEXICO

Alliance Electronics Inc.
10510 Research Avenue
Albuquerque 87123
Tel: (505) 292-3360
FAX: 505-292-6537

Hamilton/Avnet Computer
5659 Jefferson, N.E. Suites A & B
Albuquerque 87109

†Hamilton/Avnet Electronics
5659A Jefferson N.E.
Albuquerque 87109
Tel: (505) 765-1500
FAX: 505-243-1395

NEW YORK

†Arrow Electronics, Inc.
3375 Brighton Henrietta Townline Rd.
Rochester 14623
Tel: (716) 427-0300
TWX: 510-253-4766

Arrow Electronics, Inc.
20 Oser Avenue
Hauppauge 11788
Tel: (516) 231-1000
TWX: 510-227-6623

Hamilton/Avnet Computer
933 Motor Parkway
Haupauge 11788

Hamilton/Avnet Computer
2060 Townline
Rochester 14623

†Hamilton/Avnet Electronics
933 Motor Parkway
Hauppauge 11788
Tel: (516) 231-9800
TWX: 510-224-6166

†Hamilton/Avnet Electronics
2060 Townline Rd.
Rochester 14623
Tel: (716) 272-2744
TWX: 510-253-5470

Hamilton/Avnet Electronics
103 Twin Oaks Drive
Syracuse 13206
Tel: (315) 437-0288
TWX: 710-541-1560

†MTI Systems Sales
38 Harbor Park Drive
Port Washington 11050
Tel: (516) 621-6200
FAX: 510-223-0846

Pioneer/Standard Electronics
68 Corporate Drive
Binghamton 13904
Tel: (607) 722-9300
FAX: 607-722-9562

Pioneer/Standard Electronics
40 Oser Avenue
Hauppauge 11787
Tel: (516) 231-9200
FAX: 510-227-9869

†Pioneer/Standard Electronics
60 Crossway Park West
Woodbury, Long Island 11797
Tel: (516) 921-8700
FAX: 516-921-2143

†Pioneer/Standard Electronics
840 Fairport Park
Fairport 14450
Tel: (716) 381-7070
FAX: 716-381-5955

NORTH CAROLINA

†Arrow Electronics, Inc.
5240 Greensdairy Road
Raleigh 27604
Tel: (919) 876-3132
TWX: 510-928-1856

Hamilton/Avnet Computer
3510 Spring Forest Road
Raleigh 27604

†Hamilton/Avnet Electronics
3510 Spring Forest Drive
Raleigh 27604
Tel: (919) 878-0819
TWX: 510-928-1836

Pioneer/Technologies Group, Inc.
9401 L-Southern Pine Blvd.
Charlotte 28210
Tel: (919) 527-8188
FAX: 704-522-8564

Pioneer Technologies Group, Inc.
2810 Meridian Parkway
Suite 148
Durham 27713
Tel: (919) 544-5400
FAX: 919-544-5885

OHIO

Arrow Commercial System Group
284 Cramer Creek Court
Dublin 43017
Tel: (614) 889-9347
FAX: (614) 889-9680

†Arrow Electronics, Inc.
6238 Cochran Road
Solon 44139
Tel: (216) 248-3990
TWX: 810-427-9409

Hamilton/Avnet Computer
7764 Washington Village Dr.
Dayton 45459

Hamilton/Avnet Computer
30325 Bainbridge Rd., Bldg. A
Solon 44139

†Hamilton/Avnet Electronics
7760 Washington Village Dr.
Dayton 45459
Tel: (513) 439-6733
FAX: 513-439-6711

†Hamilton/Avnet Electronics
30325 Bainbridge
Solon 44139
Tel: (216) 349-5100
TWX: 810-427-9452

Hamilton/Avnet Computer
777 Brooksedge Blvd.
Westerville 43081
Tel: (614) 882-7004
FAX: 614-882-8650

Hamilton/Avnet Electronics
777 Brooksedge Blvd.
Westerville 43081
Tel: (614) 882-7004

MTI Systems Sales
23400 Commerce Park Road
Beachwood 44122
Tel: (216) 464-6688

†Pioneer/Standard Electronics
4433 Interpoint Boulevard
Dayton 45424
Tel: (513) 236-9900
FAX: 513-236-8133

†Pioneer/Standard Electronics
4800 E. 131st Street
Cleveland 44105
Tel: (216) 587-3600
FAX: 216-663-1004

†Certified Technical Distributor

DOMESTIC DISTRIBUTORS (Contd.)

OKLAHOMA

Arrow Electronics, Inc.
4719 South Memorial Dr.
Tulsa 74145

†Hamilton/Avnet Electronics
12121 E. 51st St., Suite 102A
Tulsa 74146
Tel: (918) 252-7297

OREGON

†Almac Electronics Corp.
1885 N.W. 169th Place
Beaverton 97005
Tel: (503) 629-8090
FAX: 503-645-0611

Hamilton/Avnet Computer
9409 Southwest Nimbus Ave.
Beaverton 97005

†Hamilton/Avnet Electronics
9409 S.W. Nimbus Ave.
Beaverton 97005
Tel: (503) 627-0201
FAX: 503-641-4012

Wyle
9640 Sunshine Court
Bldg. G, Suite 200
Beaverton 97005
Tel: (503) 643-7900
FAX: 503-646-5466

PENNSYLVANIA

Arrow Electronics, Inc.
650 Seco Road
Monroeville 15146
Tel: (412) 856-7000

Hamilton/Avnet Computer
2800 Liberty Ave., Bldg. E
Pittsburgh 15222

Hamilton/Avnet Electronics
2800 Liberty Ave.
Pittsburgh 15238
Tel: (412) 281-4150

Pioneer/Standard Electronics
259 Kappa Drive
Pittsburgh 15238
Tel: (412) 782-2300
FAX: 412-963-8255

†Pioneer/Technologies Group, Inc.
Delaware Valley
261 Gibralter Road
Horsham 19044
Tel: (215) 674-4000
FAX: 215-674-3107

TENNESSEE

Arrow Commercial System Group
3635 Knight Road
Suite 7
Memphis 38118
Tel: (901) 367-0540
FAX: (901) 367-2081

TEXAS

Arrow Electronics, Inc.
3220 Commander Drive
Carrollton 75006
Tel: (214) 380-6464
FAX: (214) 248-7208

Hamilton/Avnet Computer
1807A West Braker Lane
Austin 78758

Hamilton/Avnet Computer
Forum 2
4004 Beltline, Suite 200
Dallas 75244

Hamilton/Avnet Computer
4850 Wright Rd., Suite 190
Stafford 77477

†Hamilton/Avnet Electronics
1807 W. Braker Lane
Austin 78758
Tel: (512) 837-8911
TWX: 910-874-1319

†Hamilton/Avnet Electronics
4004 Beltline, Suite 200
Dallas 75234
Tel: (214) 308-8111
TWX: 910-860-5929

†Hamilton/Avnet Electronics
4850 Wright Rd., Suite 190
Stafford 77477
Tel: (713) 240-7733
TWX: 910-881-5523

†Pioneer/Standard Electronics
1826-D Kramer
Austin 78758
Tel: (512) 835-4000
FAX: 512-835-9829

†Pioneer/Standard Electronics
13710 Omega Road
Dallas 75244
Tel: (214) 386-7300
FAX: 214-490-6419

†Pioneer/Standard Electronics
10530 Rockley Road
Houston 77099
Tel: (713) 495-4700
FAX: 713-495-5642

†Wyle Distribution Group
1810 Greenville Avenue
Richardson 75081
Tel: (214) 235-9953
FAX: 214-644-5064

UTAH

Hamilton/Avnet Computer
1585 West 2100 South
Salt Lake City 84119

†Hamilton/Avnet Electronics
1585 West 2100 South
Salt Lake City 84119
Tel: (801) 972-2800
TWX: 910-925-4018

†Wyle Distribution Group
1325 West 2200 South
Suite E
West Valley 84119
Tel: (801) 974-9953

WASHINGTON

†Almac Electronics Corp.
14360 S.E. Eastgate Way
Bellevue 98007
Tel: (206) 643-9992
FAX: 206-643-9709

Hamilton/Avnet Computer
17761 Northeast 78th Place
Redmond 98052

†Hamilton/Avnet Electronics
17761 N.E. 78th Place
Redmond 98052
Tel: (206) 881-6697
FAX: 206-867-0159

Wyle Distribution Group
15385 N.E. 90th Street
Redmond 98052
Tel: (206) 881-1150
FAX: 206-881-1567

WISCONSIN

Arrow Electronics, Inc.
200 N. Patrick Blvd., Ste. 100
Brookfield 53005
Tel: (414) 792-0150
FAX: 414-792-0156

Hamilton/Avnet Computer
20875 Crossroads Circle
Suite 400
Waukesha 53186

†Hamilton/Avnet Electronics
28875 Crossroads Circle
Suite 400
Waukesha 53186
Tel: (414) 784-4510
FAX: 414-784-9509

CANADA

ALBERTA

Hamilton/Avnet Computer
2816 21st Street Northeast
Calgary T2E 6Z2

Hamilton/Avnet Electronics
2816 21st Street N.E. #3
Calgary T2E 6Z3
Tel: (403) 230-3586
FAX: 403-250-1591

Zentronics
6815 #8 Street N.E.
Suite 100
Calgary T2E 7H
Tel: (403) 295-8818
FAX: 403-295-8714

BRITISH COLUMBIA

†Hamilton/Avnet Electronics
8610 Commerce Ct.
Burnaby V5A 4N6
Tel: (604) 420-4101
FAX: 604-437-4712

Zentronics
108-11400 Bridgeport Road
Richmond V6X 1T2
Tel: (604) 273-5575
FAX: 604-273-2413

ONTARIO

Arrow Electronics, Inc.
36 Antares Dr., Unit 100
Nepean K2E 7W5
Tel: (613) 226-6903
FAX: 613-723-2018

†Arrow Electronics, Inc.
1093 Meyerside, Unit 2
Mississauga L5T 1M4
Tel: (416) 673-7769
FAX: 416-672-0849

Hamilton/Avnet Computer
Canada System Engineering Group
3688 Nashua Drive
Units 7 & 8
Mississuaga L4V 1M5

Hamilton/Avnet Computer
3688 Nashua Drive
Units 9 & 10
Mississuaga L4V 1M5

Hamilton/Avnet Computer
6845 Rexwood Road
Units 7, 8, & 9
Mississuaga L4V 1R2

Hamilton/Avnet Computer
190 Colonade Road
Nepean K2E 7J5

†Hamilton/Avnet Electronics
6845 Rexwood Road
Units 3-4-5
Mississauga L4T 1R2
Tel: (416) 677-7432
FAX: 416-677-0940

†Hamilton/Avnet Electronics
190 Colonnade Road South
Nepean K2E 7L5
Tel: (613) 226-1700
FAX: 613-226-1184

†Zentronics
1355 Meyerside Drive
Mississauga L5T 1C9
Tel: (416) 564-9600
FAX: 416-564-8320

†Zentronics
155 Colonnade Road
Unit 17
Nepean K2E 7K1
Tel: (613) 226-8840
FAX: 613-226-6352

QUEBEC

Arrow Electronics Inc.
1100 St. Regis
Dorval H9P 2T5
Tel: (514) 421-7411
FAX: 514-421-7430

Arrow Electronics, Inc.
500 Boul. St-Jean-Baptiste
Suite 280
Quebec G2E 5R9
Tel: (418) 871-7500
FAX: 418-871-6816

Hamilton/Avnet Computer
2795 Rue Halpern
St. Laurent H4S 1P8

†Hamilton/Avnet Electronics
2795 Halpern
St. Laurent H2E 7K1
Tel: (514) 335-1000
FAX: 514-335-2481

†Zentronics
520 McCaffrey
St. Laurent H4T 1N3
Tel: (514) 737-9700
FAX: 514-737-5212

†Certified Technical Distributor

EUROPEAN SALES OFFICES

FINLAND

Intel Finland OY
Ruosilantie 2
00390 Helsinki
Tel: (358) 0 544 644
TLX: 123332

FRANCE

Intel Corporation S.A.R.L.
1, Rue Edison-BP 303
78054 St. Quentin-en-Yvelines Cedex
Tel: (33) (1) 30 57 70 00
TLX: 699016

ISRAEL

Intel Semiconductor Ltd.
Atidim Industrial Park-Neve Sharet
P.O. Box 43202
Tel-Aviv 61430
Tel: (972) 03-498080
TLX: 371215

ITALY

Intel Corporation Italia S.p.A.
Milanofiori Palazzo E
20094 Assago
Milano
Tel: (39) (02) 89200950
TLX: 341286

NETHERLANDS

Intel Semiconductor B.V.
Postbus 84130
3099 CC Rotterdam
Tel: (31) 10.407.11.11
TLX: 22283

SPAIN

Intel Iberia S.A.
Zurbaran, 28
28010 Madrid
Tel: (34) (1) 308.25.52
TLX: 46880

SWEDEN

Intel Sweden A.B.
Dalvagen 24
171 36 Solna
Tel: (46) 8 734 01 00
TLX: 12261

SWITZERLAND

Intel Semiconductor A.G.
Zuerichstrasse
8185 Winkel-Rueti bei Zuerich
Tel: (41) 01/860 62 62
TLX: 825977

UNITED KINGDOM

Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon, Wiltshire SN3 1RJ
Tel: (44) (0793) 696000
TLX: 444447/8

WEST GERMANY

Intel GmbH
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen
Tel: (49) 089/90992-0
FAX: (49) 089/904/3948

Intel GmbH
Abraham Lincoln Strasse 16-18
6200 Wiesbaden
Tel: (49) 06121/7605-0
TLX: 4-186183

Intel GmbH
Zettachring 10A
7000 Stuttgart 80
Tel: (49) 0711/7287-280
TLX: 7-254826

EUROPEAN DISTRIBUTORS/REPRESENTATIVES

AUSTRIA

Bacher Electronics G.m.b.H.
Rotenmuehlgasse 26
1120 Wien
Tel: (43) (0222) 83 56 46
TLX: 31532

BELGIUM

Inelco Belgium S.A.
Av. des Croix de Guerre 94
1120 Bruxelles
Oorlogskruisenlaan, 94
1120 Brussel
Tel: (32) (02) 216 01 60
TLX: 64475 or 22090

DENMARK

ITT-Multikomponent
Naverland 29
2600 Glostrup
Tel: (45) (0) 2 45 66 45
TLX: 33 355

FINLAND

OY Fintronic AB
Melkonkatu 24A
00210 Helsinki
Tel: (358) (0) 6926022
TLX: 124224

FRANCE

Almex
Zone industrielle d'Antony
48, rue de l'Aubepine
BP 102
92164 Antony cedex
Tel: (33) (1) 46 66 21 12
TLX: 250067

Jermyn
60, rue des Gemeaux
Silic 580
94653 Rungis Cedex
Tel: (33) (1) 49 78 49 78
TLX: 261585

Metrologie
Tour d'Asnieres
4, av. Laurent-Cely
92606 Asnieres Cedex
Tel: (33) (1) 47 90 62 40
TLX: 611448

Tekelec-Airtronic
Cite des Bruyeres
Rue Carle Vernet - BP 2
92310 Sevres
Tel: (33) (1) 45 34 75 35
TLX: 204552

IRELAND

Micro Marketing Ltd.
Glenageary Office Park
Glenageary
Co. Dublin
Tel: (21) (353) (01) 856288
FAX: (21) (353) (01) 857364
TLX: 31584

ISRAEL

Eastronics Ltd.
11 Rozanis Street
P.O.B. 39300
Tel-Aviv 61392
Tel: (972) 03-475151
TLX: 33638

ITALY

Intesi
Divisione ITT Industries GmbH
Viale Milanofiori
Palazzo E/5
20090 Assago (MI)
Tel: (39) 02/824701
TLX: 311351

Lasi Elettronica S.p.A.
V. le Fulvio Testi, 126
20092 Cinisello Balsamo (MI)
Tel: (39) 02/2440012
TLX: 352040

Telcom S.r.l.
Via M. Civitali 75
20148 Milano
Tel: (39) 02/4049046
TLX: 335654

ITT Multicomponents
Viale Milanofiori E/5
20090 Assago (MI)
Tel: (39) 02/824701
TLX: 311351

Silverstar
Via Dei Gracchi 20
20146 Milano
Tel: (39) 02/49961
TLX: 332189

NETHERLANDS

Koning en Hartman
Elektrotechniek B.V.
Energieweg 1
2627 AP Delft
Tel: (31) (1) 15/609906
TLX: 38250

NORWAY

Nordisk Elektronikk (Norge) A/S
Postboks 123
Smedsvingen 4
1364 Hvalstad
Tel: (47) (02) 84 62 10
TLX: 77546

PORTUGAL

ATD Portugal LDA
Rua Dos Lusiados, 5 Sala B
1300 Lisboa
Tel: (35) (1) 64 80 91
TLX: 61562

Ditram
Avenida Miguel Bombarda, 133
1000 Lisboa
Tel: (35) (1) 54 53 13
TLX: 14182

SPAIN

ATD Electronica, S.A.
Plaza Ciudad de Viena, 6
28040 Madrid
Tel: (34) (1) 234 40 00
TLX: 42477

ITT-SESA
Calle Miguel Angel, 21-3
28010 Madrid
Tel: (34) (1) 419 09 57
TLX: 27461

Metrologia Iberica, S.A.
Ctra. de Fuencarral, n.80
28100 Alcobendas (Madrid)
Tel: (34) (1) 653 86 11

SWEDEN

Nordisk Elektronik AB
Torshamnsgatan 39
Box 36
164 93 Kista
Tel: (46) 08-03 46 30
TLX: 105 47

SWITZERLAND

Industrade A.G.
Hertistrasse 31
8304 Wallisellen
Tel: (41) (01) 8328111
TLX: 56788

TURKEY

EMPA Electronic
Lindwurmstrasse 95A
8000 Muenchen 2
Tel: (49) 089/53 80 570
TLX: 528573

UNITED KINGDOM

Accent Electronic Components Ltd.
Jubilee House, Jubilee Road
Letchworth, Herts SG6 1QH
Tel: (44) (0462) 670011
FAX: (44) (0462) 682467
TWX: 826505

Bytech Components Ltd.
12A Cedarwood
Chineham Business Park
Crockford Lane
Basingstoke
Hants RG24 0WD
Tel: (0256) 707107
FAX: 0256-707162

Conformix
Unit 5
A1M Business Centre
Dixons Hill Road
Welham Green
South Hatfield
Herts AL9 7JE
Tel: (07072) 73282
FAX: (07072) 61678

Bytech Systems
3 The Western Centre
Western Road
Bracknell RG12 1RW
Tel: (44) (0344) 55333
FAX: (44) (0344) 867270
TWX: 849624

Jermyn
Vestry Estate
Otford Road
Sevenoaks
Kent TN14 5EU
Tel: (44) (0732) 450144
FAX: (44) (0732) 451251
TWX: 95142

MMD Ltd.
3 Bennet Court
Bennet Road
Reading
Berkshire RG2 0QX
Tel: (44) (0734) 313232
FAX: (44) (0734) 313255
TWX: 846669

Rapid Recall, Ltd.
Rapid House
Oxford Road
High Wycombe
Buckinghamshire HP11 2EE
Tel: (44) (0494) 26271
FAX: (44) (0494) 21860
TWX: 837931

Rapid Recall, Ltd.
28 High Street
Nantwich
Cheshire CW5 5AS
Tel: (0270) 627505
FAX: (0270) 629883
TWX: 36329

WEST GERMANY

Electronic 2000 AG
Stahlgruberring 12
8000 Muenchen 82
Tel: (49) 089/42001-0
TLX: 522561

ITT Multikomponent GmbH
Postfach 1265
Bahnhofstrasse 44
7141 Moeglingen
Tel: (49) 07141/4879
TLX: 7264472

Jermyn GmbH
Im Dachsstueck 9
6250 Limburg
Tel: (49) 06431/508-0
TLX: 415257-0

Metrologie GmbH
Meglingerstrasse 49
8000 Muenchen 71
Tel: (49) 089/78042-0
TLX: 5213189

Proelectron Vertriebs GmbH
Max Planck Strasse 1-3
6072 Dreieich
Tel: (49) 06103/30434-3
TLX: 417903

YUGOSLAVIA

H.R. Microelectronics Corp.
2005 de la Cruz Blvd., Ste. 223
Santa Clara, CA 95050
U.S.A.
Tel: (1) (408) 988-0286
TLX: 387452

Rapido Electronic Components S.p.a.
Via C. Beccaria, 8
34133 Trieste
Italia
Tel: (39) 040/360555
TLX: 460461

INTERNATIONAL SALES OFFICES

AUSTRALIA

Intel Australia Pty. Ltd.
Unit 13
Allambie Grove Business Park
25 Frenchs Forest Road East
Frenchs Forest, NSW, 2086
Tel: 61-2975-3300
FAX: 61-2975-3375

BRAZIL

Intel Semicondutores do Brazil LTDA
Av. Paulista, 1159-CJS 404/405
01311 - Sao Paulo - S.P.
Tel: 55-11-287-5899
TLX: 3911153146 ISDB
FAX: 55-11-287-5119

CHINA/HONG KONG

Intel PRC Corporation
15/F, Office 1, Citic Bldg.
Jian Guo Men Wai Street
Beijing, PRC
Tel: (1) 500-4850
TLX: 22947 INTEL CN
FAX: (1) 500-2953

Intel Semiconductor Ltd.*
10/F East Tower
Bond Center
Queensway, Central
Hong Kong
Tel: (852) 844-4555
FAX: (852) 868-1989

INDIA

Intel Asia Electronics, Inc.
4/2, Samrah Plaza
St. Mark's Road
Bangalore 560001
Tel: 011-91-812-215065
TLX: 953-845-2646 INTL IN
FAX: 091-812-215067

JAPAN

Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26
Tel: 0298-47-8511
TLX: 3656-160
FAX: 0298-47-8450

Intel Japan K.K.*
Daiichi Mitsugi Bldg.
1-8889 Fuchu-cho
Fuchu-shi, Tokyo 183
Tel: 0423-60-7871
FAX: 0423-60-0315

Intel Japan K.K.*
Bldg. Kumagaya
2-69 Hon-cho
Kumagaya-shi, Saitama 360
Tel: 0485-24-6871
FAX: 0485-24-7518

Intel Japan K.K.*
Kawa-asa Bldg.
2-11-5 Shin-Yokohama
Kohoku-ku, Yokohama-shi
Kanagawa, 222
Tel: 045-474-7661
FAX: 045-471-4394

Intel Japan K.K.*
Ryokuchi-Eki Bldg.
2-4-1 Terauchi
Toyonaka-shi, Osaka 560
Tel: 06-863-1091
FAX: 06-863-1084

Intel Japan K.K.
Shinmaru Bldg.
1-5-1 Marunouchi
Chiyoda-ku, Tokyo 100
Tel: 03-201-3621
FAX: 03-201-6850

Intel Japan K.K.
Green Bldg.
1-16-20 Nishiki
Naka-ku, Nagoya-shi
Aichi 450
Tel: 052-204-1261
FAX: 052-204-1285

KOREA

Intel Technology Asia, Ltd.
16th Floor, Life Bldg.
61 Yoido-dong, Youngdeungpo-Ku
Seoul 150-010
Tel: (2) 784-8186, 8286, 8386
TLX: K29312 INTELKO
FAX: (2) 784-8096

SINGAPORE

Intel Singapore Technology, Ltd.
101 Thomson Road #21-05/06
United Square
Singapore 1130
Tel: 250-7811
TLX: 39921 INTEL
FAX: 250-9256

TAIWAN

Intel Technology Far East Ltd.
8th Floor, No. 205
Bank Tower Bldg.
Tung Hua N. Road
Taipei
Tel: 886-2-716-9660
FAX: 886-2-717-2455

INTERNATIONAL DISTRIBUTORS/REPRESENTATIVES

ARGENTINA

Dafsys S.R.L.
Chacabuco, 90-6 Piso
1069-Buenos Aires
Tel: 54-1-334-7726
FAX: 54-1-334-1871

AUSTRALIA

Email Electronics
15-17 Hume Street
Huntingdale, 3166
Tel: 011-61-3-544-8244
TLX: AA 30895
FAX: 011-61-3-543-8179

NSD-Australia
205 Middleborough Rd.
Box Hill, Victoria 3128
Tel: 03 8900970
FAX: 03 8990819

BRAZIL

Elebra Componentes
Rua Geraldo Flausina Gomes, 78
7 Andar
04575 - Sao Paulo - S.P.
Tel: 55-11-534-9641
TLX: 55-11-54593/54591
FAX: 55-11-534-9424

CHINA/HONG KONG

Novel Precision Machinery Co., Ltd.
Room 728 Trade Square
681 Cheung Sha Wan Road
Kowloon, Hong Kong
Tel: (852) 360-8999
TWX: 32032 NVTNL HX
FAX: (852) 725-3695

INDIA

Micronic Devices
Arun Complex
No. 65 D.V.G. Road
Basavanagudi
Bangalore 560 004
Tel: 011-91-812-600-631
011-91-812-611-365
TLX: 9538458332 MDBG

Micronic Devices
No. 516 5th Floor
Swastik Chambers
Sion, Trombay Road
Chembur
Bombay 400 071
TLX: 9531 171447 MDEV

Micronic Devices
25/8, 1st Floor
Bada Bazaar Marg
Old Rajinder Nagar
New Delhi 110 060
Tel: 011-91-11-5723509
011-91-11-589771
TLX: 031-63253 MDND IN

Micronic Devices
6-3-348/12A Dwarakapuri Colony
Hyderabad 500 482
Tel: 011-91-842-226748

S&S Corporation
1587 Kooser Road
San Jose, CA 95118
Tel: (408) 978-6216
TLX: 820281
FAX: (408) 978-8635

JAPAN

Asahi Electronics Co. Ltd.
KMM Bldg. 2-14-1 Asano
Kokurakita-ku
Kitakyushu-shi 802
Tel: 093-511-6471
FAX: 093-551-7861

CTC Components Systems Co., Ltd.
4-8-1 Dobashi, Miyamae-ku
Kawasaki-shi, Kanagawa 213
Tel: 044-852-5121
FAX: 044-877-4268

Dia Semicon Systems, Inc.
Flower Hill Shinmachi Higashi-kan
1-23-9 Shinmachi, Setagaya-ku
Tokyo 154
Tel: 03-439-1600
FAX: 03-439-1601

Okaya Koki
2-4-18 Sakae
Naka-ku, Nagoya-shi 460
Tel: 052-204-2916
FAX: 052-204-2901

Ryoyo Electro Corp.
Konwa Bldg.
1-12-22 Tsukiji
Chuo-ku, Tokyo 104
Tel: 03-546-5011
FAX: 03-546-5044

KOREA

J-Tek Corporation
Dong Sung Bldg. 9/F
158-24, Samsung-Dong, Kangnam-Ku
Seoul 135-090
Tel: (822) 557-8039
FAX: (822) 557-8304

Samsung Electronics
Samsung Main Bldg.
150 Taepyung-Ro-2KA, Chung-Ku
Seoul 100-102
C.P.O. Box 8780
Tel: (822) 751-3680
TWX: KORSST K 27970
FAX: (822) 753-9065

MEXICO

SSB Electronics, Inc.
675 Palomar Street, Bldg. 4, Suite A
Chula Vista, CA 92011
Tel: (619) 585-3253
TLX: 287751 CBALL UR
FAX: (619) 585-8322

Dicopel S.A.
Tochtli 368 Fracc. Ind. San Antonio
Azcapotzalco
C.P. 02760-Mexico, D.F.
Tel: 52-5-561-3211
TLX: 177 3790 Dicome
FAX: 52-5-561-1279

PHI S.A. de C.V.
Fco. Villa esq. Ajusco s/n
Cuernavaca—Morelos
Tel: 52-73-13-9412
FAX: 52-73-17-5333

NEW ZEALAND

Email Electronics
36 Olive Road
Penrose, Auckland
Tel: 011-64-9-591-155
FAX: 011-64-9-592-681

SINGAPORE

Electronic Resources Pte, Ltd.
17 Harvey Road
#03-01 Singapore 1336
Tel: (65) 283-0888
TWX: RS 56541 ERS
FAX: (65) 289-5327

SOUTH AFRICA

Electronic Building Elements
178 Erasmus St. (off Watermeyet St.)
Meyerspark, Pretoria, 0184
Tel: 011-2712-803-7680
FAX: 011-2712-803-8294

TAIWAN

Micro Electronics Corporation
12th Floor, Section 3
285 Nanking East Road
Taipei, R.O.C.
Tel: (886) 2-7198419
FAX: (886) 2-7197916

Acer Sertek Inc.
15th Floor, Section 2
Chien Kuo North Rd.
Taipei 18479 R.O.C.
Tel: 886-2-501-0055
TWX: 23756 SERTEK
FAX: (886) 2-5012521

*Field Application Location

DOMESTIC SERVICE OFFICES

ALASKA

Intel Corp.
c/o TransAlaska Network
1515 Lore Rd.
Anchorage 99507
Tel: (907) 522-1776

Intel Corp.
c/o TransAlaska Data Systems
c/o GCI Operations
520 Fifth Ave., Suite 407
Fairbanks 99701
Tel: (907) 452-6264

ARIZONA

*Intel Corp.
410 North 44th Street
Suite 500
Phoenix 85008
Tel: (602) 231-0386
FAX: (602) 244-0446

*Intel Corp.
500 E. Fry Blvd., Suite M-15
Sierra Vista 85635
Tel: (602) 459-5010

ARKANSAS

Intel Corp.
c/o Federal Express
1500 West Park Drive
Little Rock 72204

CALIFORNIA

*Intel Corp.
21515 Vanowen St., Ste. 116
Canoga Park 91303
Tel: (818) 704-8500

*Intel Corp.
300 N. Continental Blvd.
Suite 100
El Segundo 90245
Tel: (213) 640-6040

*Intel Corp.
1900 Prairie City Rd.
Folsom 95630-9597
Tel: (916) 351-6143

*Intel Corp.
9665 Chesapeake Dr., Suite 325
San Diego 92123
Tel: (619) 292-8086

**Intel Corp.
400 N. Tustin Avenue
Suite 450
Santa Ana 92705
Tel: (714) 835-9642

**Intel Corp.
2700 San Tomas Exp., 1st Floor
Santa Clara 95051
Tel: (408) 970-1747

COLORADO

*Intel Corp.
600 S. Cherry St., Suite 700
Denver 80222
Tel: (303) 321-8086

CONNECTICUT

*Intel Corp.
301 Lee Farm Corporate Park
83 Wooster Heights Rd.
Danbury 06811
Tel: (203) 748-3130

FLORIDA

**Intel Corp.
800 Fairway Dr., Suite 160
Deerfield Beach 33441
Tel: (305) 421-0506
FAX: (305) 421-2444

*Intel Corp.
5850 T.G. Lee Blvd., Ste. 340
Orlando 32822
Tel: (407) 240-8000

GEORGIA

*Intel Corp.
20 Technology Park, Suite 150
Norcross 30092
Tel: (404) 449-0541

5523 Theresa Street
Columbus 31907

HAWAII

**Intel Corp.
Honolulu 96820
Tel: (808) 847-6738

ILLINOIS

**†Intel Corp.
Woodfield Corp. Center III
300 N. Martingale Rd., Ste. 400
Schaumburg 60173
Tel: (708) 605-8031

INDIANA

*Intel Corp.
8910 Purdue Rd., Ste. 350
Indianapolis 46268
Tel: (317) 875-0623

KANSAS

*Intel Corp.
10985 Cody, Suite 140
Overland Park 66210
Tel: (913) 345-2727

KENTUCKY

Intel Corp.
133 Walton Ave., Office 1A
Lexington 40508
Tel: (606) 255-2957

Intel Corp.
896 Hillcrest Road, Apt. A
Radcliff 40160 (Louisville)

LOUISIANA

Hammond 70401
(serviced from Jackson, MS)

MARYLAND

**Intel Corp.
10010 Junction Dr., Suite 200
Annapolis Junction 20701
Tel: (301) 206-2860

MASSACHUSETTS

**Intel Corp.
Westford Corp. Center
3 Carlisle Rd., 2nd Floor
Westford 01886
Tel: (508) 692-0960

MICHIGAN

*Intel Corp.
7071 Orchard Lake Rd., Ste. 100
West Bloomfield 48322
Tel: (313) 851-8905

MINNESOTA

*Intel Corp.
3500 W. 80th St., Suite 360
Bloomington 55431
Tel: (612) 835-6722

MISSISSIPPI

Intel Corp.
c/o Compu-Care
2001 Airport Road, Suite 205F
Jackson 39208
Tel: (601) 932-6275

MISSOURI

*Intel Corp.
4203 Earth City Exp., Ste. 131
Earth City 63045
Tel: (314) 291-1990

Intel Corp.
Route 2, Box 221
Smithville 64089
Tel: (913) 345-2727

NEW JERSEY

**Intel Corp.
300 Sylvan Avenue
Englewood Cliffs 07632
Tel: (201) 567-0821

*Intel Corp.
Parkway 109 Office Center
328 Newman Springs Road
Red Bank 07701
Tel: (201) 747-2233

NEW MEXICO

Intel Corp.
Rio Rancho 1
4100 Sara Road
Rio Rancho 87124-1025
(near Albuquerque)
Tel: (505) 893-7000

NEW YORK

*Intel Corp.
2950 Expressway Dr. South
Suite 130
Islandia 11722
Tel: (516) 231-3300

Intel Corp.
Westage Business Center
Bldg. 300, Route 9
Fishkill 12524
Tel: (914) 897-3860

Intel Corp.
5858 East Molloy Road
Syracuse 13211
Tel: (315) 454-0576

NORTH CAROLINA

*Intel Corp.
5800 Executive Center Drive
Suite 105
Charlotte 28212
Tel: (704) 568-8966

**Intel Corp.
5540 Centerview Dr., Suite 215
Raleigh 27606
Tel: (919) 851-9537

OHIO

**Intel Corp.
3401 Park Center Dr., Ste. 220
Dayton 45414
Tel: (513) 890-5350

*Intel Corp.
25700 Science Park Dr., Ste. 100
Beachwood 44122
Tel: (216) 464-2736

OREGON

**Intel Corp.
15254 N.W. Greenbrier Parkway
Building B
Beaverton 97005
Tel: (503) 645-8051

PENNSYLVANIA

*†Intel Corp.
925 Harvest Drive
Suite 200
Blue Bell 19422
Tel: (215) 641-1000
1-800-468-3548
FAX: (215) 641-0785

**†Intel Corp.
400 Penn Center Blvd., Ste. 610
Pittsburgh 15235
Tel: (412) 823-4970

*Intel Corp.
1513 Cedar Cliff Dr.
Camp Hill 17011
Tel: (717) 761-0860

PUERTO RICO

Intel Corp.
South Industrial Park
P.O. Box 910
Las Piedras 00671
Tel: (809) 733-8616

TEXAS

**Intel Corp.
Westech 360, Suite 4230
8911 Capitol of Texas Hwy.
Austin 78752-1239
Tel : (512) 794-8086

**†Intel Corp.
12000 Ford Rd., Suite 401
Dallas 75234
Tel: (214) 241-8087

**Intel Corp.
7322 SW Freeway, Suite 1490
Houston 77074
Tel: (713) 988-8086

UTAH

Intel Corp.
428 East 6400 South
Suite 104
Murray 84107
Tel: (801) 263-8051
FAX: (801) 268-1457

VIRGINIA

*Intel Corp.
1504 Santa Rosa Rd., Ste. 108
Richmond 23288
Tel: (804) 282-5668

WASHINGTON

**Intel Corp.
155 108th Avenue N.E., Ste. 386
Bellevue 98004
Tel: (206) 453-8086

CANADA

ONTARIO

**Intel Semiconductor of
Canada, Ltd.
2650 Queensview Dr., Ste. 250
Ottawa K2B 8H6
Tel: (613) 829-9714

**Intel Semiconductor of
Canada, Ltd.
190 Attwell Dr., Ste. 102
Rexdale (Toronto) M9W 6H8
Tel: (416) 675-2105

QUEBEC

**Intel Semiconductor of
Canada, Ltd.
1 Rue Holiday
Suite 115
Tour East
Pt. Claire H9R 5N3
Tel: (514) 694-9130
FAX: 514-694-0064

CUSTOMER TRAINING CENTERS

CALIFORNIA

2700 San Tomas Expressway
Santa Clara 95051
Tel: 1-800-328-0386

MARYLAND

10010 Junction Dr.
Suite 200
Annapolis Junction 20701
Tel: 1-800-328-0386

SYSTEMS ENGINEERING OFFICES

MINNESOTA

3500 W. 80th Street
Suite 360
Bloomington 55431
Tel: (612) 835-6722

NEW YORK

2950 Expressway Dr., South
Islandia 11722
Tel: (506) 231-3300

*Carry-in locations
**Carry-in/mail-in locations

UNITED STATES
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26

FRANCE
Intel Corporation S.A.R.L.
1, Rue Edison, BP 303
78054 Saint-Quentin-en-Yvelines Cedex

UNITED KINGDOM
Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon
Wiltshire, England SN3 1RJ

GERMANY
Intel GmbH
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen

HONG KONG
Intel Semiconductor Ltd.
10/F East Tower
Bond Center
Queensway, Central

CANADA
Intel Semiconductor of Canada, Ltd.
190 Attwell Drive, Suite 500
Rexdale, Ontario M9W 6H8

intel®

Peripheral Components

Intel's peripheral components surround the microprocessor and help unleash its power. Consider the advantages of using components designed by the same people who have designed Intel's industry-leading microprocessors. Built-in functions and standard Intel microprocessor/peripheral interfaces deliver real-time and performance advantages to the designer.

From the world's most complete line of 32-bit bus chip sets supporting Extended Industry Standard Architecture (EISA) and Micro Channel* to Intel's direct memory access and floppy disk controllers, Intel offers a complete line of peripheral components to complement its microprocessors.

This handbook contains data sheets which include comprehensive charts covering symbols and functions, block diagrams, operational and functional descriptions. Application notes provide diagrams for interface considerations, specific designs, and hardware information. In addition, there are technical briefs on all of Intel's peripheral components, including helpful information for developing leading-edge 32-bit systems.

*Micro Channel is a registered trademark of IBM Corp.

Printed in USA/0191/50K/RRD/LK
Peripheral Components

ISBN# 1-55512-127-6