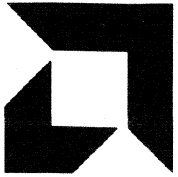


Advanced Micro Devices

MOS Microprocessors and Peripherals Data Book

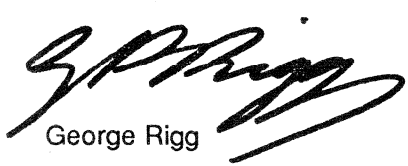
901 Thompson Place, P.O. Box 3453, Sunnyvale, California 94088
(408) 732-2400 TWX: 910-339-9280 TELEX: 34-6306

Printed in U.S.A.

The 1987 MOS Microprocessors and Peripherals Data Book presents Advanced Micro Devices' full line of MOS fixed-instruction microprocessors, peripherals and microcontrollers. Throughout our 18-year history, AMD has offered complete system solutions. This book is your guide to an array of microprocessors, their associated peripherals and a variety of microcontroller-based products.

The devices contained in this book are the building blocks needed to create innovative new designs. Our worldwide hardware and software support team of Field Applications Engineers can help you utilize these new products to complete your designs in a timely and cost-effective manner.

Today more than ever your satisfaction is essential. At Advanced Micro Devices, we bring you the quality, reliability and innovation you need from a leader in semiconductor VLSI design and manufacturing. If you have questions about any of the products described in this book or just want to know more about AMD and our complete product line, call your local Advanced Micro Devices sales office, licensed representative or franchised distributor.



George Rigg

Vice President
Processor Products Division

PREFACE

Commitment to Technology

Advanced Micro Devices' research in semiconductor manufacture and testing continues to lead to new breakthroughs. AMD has several six-inch wafer fabrication facilities which process advanced NMOS, CMOS, and Bipolar technologies. These facilities produce ICs that are more affordable and reliable than material built in less advanced fabrication areas, Innovative methods and techniques from fabrication through final test give you advanced products and the assurance of replicable processing.

Advanced Packaging Technology

AMD is also emphasizing today's new high pin-count packages; the Plastic Leaded Chip Carrier (PLCC), the Ceramic Leadless Chip Carrier (LCC), and the Pin Grid Array (PGA). Many of AMD's microprocessor and peripheral products are available in surface-mountable PLCC packages.

System Solutions

This data book contains data sheets on AMD's family of MOS microprocessors, peripherals, and microcontrollers. AMD offers a complete line of iAPX microprocessors from the 8086/8088 and 80186/80188 to the high-performance 80286. AMD maintains full compatibility with new revisions of these iAPX products. These processors are available in various packages, including PLCC as well as standard dual-in-line for the 8086/8088, and LCC for the 80186/80188/8286. In addition, AMD's technological advances provide the highest speeds available in this family, including the 10-MHz 8086/8088 and the 12.5-MHz 80286. Advanced processing techniques allow many functions to run cooler than other manufacturers' devices, giving you better overall reliability. For example, AMD's 80L286 consumes up to 50% less power than the other manufacturer's product. This translates into cooler operating temperatures leading to improved reliability. This performance enables AMD to offer the 80L286 in a lower-cost, surface-mountable 68-pin PLCC.

In addition to microprocessors, AMD provides various peripherals to support these processors. This offering includes the complete peripheral kit associated with the 8088 and 80286 CPUs. AMD also manufactures proprietary advanced peripherals, including the Am9517A/8237A DMA Controller, the Am9513A System Timing Controller, and the Am9516A Advanced DMA Controller.

Single-Chip Microprocessors

AMD's 8051 Family continues to grow. For applications where more on-chip program memory is required, the 8053 offers 8K bytes of on-chip ROM. To assist you with prototyping and development needs, AMD has the 8753-EPROM version of the 8053.

The 80C51/80C31 are CMOS versions of the popular 8051/8031. These CMOS devices are ideally suited for power-sensitive applications. The 80C51/80C31 features idle and power-down modes for further power conservation.

The 80C521/80C321 has double ROM and RAM, a watchdog timer, and a second data pointer. The watchdog timer provides protection from unexpected external events. The data pointer saves time and code space for routines such as block moves.

Originally designed for automotive applications, the 80515 and the 80535, with an on-chip A/D converter and watchdog timer, are ideally suited for most motor control applications. The 80515 also has 8K bytes of on-chip ROM, 256 bytes of RAM, and two additional I/O ports.

AMD offers several of the Zilog line of micorporcessors and associated peripherals. These include the Z8001* and Z8002* microprocessors in both commercial and military temperature ranges, and peripherals such as the industry-standard Z8530 Serial Communications Controller in several temperature, speed, and package configurations.

Customer Service

AMD's dedication to the customer extends beyond our traditional commitment to quality. We support our products with an extensive network of Applications Engineers located in manufacturing facilities and sales offices worldwide. These individuals provide you easy access to an extensive array of design tools, support literature, and personal service. For your convenience in ordering and scheduling deliveries, AMD is represented by representatives and major distributors worldwide.

*Z8001 and Z8002 are trademarks of Zilog, Inc.

CONTENTS

SECTION 1 — INDEX

SECTION 2 — ADVANCED GENERAL-PURPOSE PERIPHERALS

SECTION 3 — iAPX86 FAMILY

SECTION 4 — SINGLE-CHIP MICROCONTROLLERS

SECTION 5 — Z8000 FAMILY

SECTION 6 — GENERAL INFORMATION

INDEX

AMD LITERATURE

To obtain literature in the U.S., write or call the AMD Literature Distribution Center, 901 Thompson Place, P.O. Box 3453 — M/S 82, Sunnyvale, CA 94088; (408) 732-2400, TOLL FREE (800) 538-8450. To obtain literature from international locations, contact the nearest AMD sales office or distributor (see listings in the back of this publication).

NUMERIC INDEX

MOS Microprocessor Family

Selector Guide

PERIPHERAL FUNCTION \ PROCESSOR	80286*	80186/ 80188*	8086/ 8088*	Z8001/ Z8002**	(68000)
Clock Period	100 ns	125 ns	100 ns	165 ns	100 ns
Clock Generator	82284-10	On-Chip	8284A-1	8127	N/A
Arithmetic Processing Unit	N/A	9511A-1		9511A-4	9511A-4
Interrupt Controller	9519A-1 8259A-2	9519A 8259A	8259A-2	9519A-1	9519A
DMA Controller	9517A-5 9516A-8 8237A-5	9517A-5 9516A-8 8237A-5	9516A-8 9517A-5 8237A-5		9517A-5 9516A-8 8237A-5
Dynamic Memory Controller	2964B 2968/69/70	2964B 2968/69/70	2964B	2964B	2964B 2968/69/70
Serial I/O	8251A 8530A 7968/69	8251A 8530A 7968/69	8251A 8530A 8030A 7968/69	8030A 7968/69	8251A 8530A 7968/69
Parallel I/O	8255A 5380	8255A-5 5380	8255A 5380	5380	8255A-5 8536A 5380
Counter Timer I/O	9513A 82C54	9513A 8253-5	9513A 8253-2	9513A	9513A 8253-5
Data Ciphering Processor	8068 9568	8068 9568	8068 9568	8068	9518 8068
Error Detection and Correction	2960	2960	2960	2960	2960
Burst Error Processor	8065 9520	8065 9520	8065 9520	8065 9520	8065 9520
CRT Controller	8052	8052	8052	8052	8052
RAM I/O	N/A	N/A	N/A	N/A	N/A
Bus Control/Arbiter	82C288-10	N/A	8288	N/A	N/A
Bus Latches	29841-6	29841-6	29841-6	29841-6	29841-6
Bus Buffers	29827/28	29827/28	29827/28	29827/28	29827/28
Bus Transceivers	29861-4	29861-4	29861-4	29861-4	29861-4
EDC Buffers	2961/2	2961/2	2961/2	2961/2	2961/2
RAM Drivers	2965/6	2965/6	2965/6	2965/6	2965/6
Network Products	7960 7961 7990 7992 7996	7960 7961 7990 7992 7996	7960 7961 7990 7992 7996	7960 7961 7990 7992 7996	7960 7961 7990 7992 7996
Compression/Expression Processor	7971	7971	7971	7971	7971
Disk Controller	9580	9580	9580	9580	9580
Video Products	95C60 8150 8151 8157 8158 8159 8171/72 8175/76 8177	95C60 8150 8151 8157 8158 8159 8171/72 8175/76 8177	95C60 8150 8151 8157 8158 8159 8171/72 8175/76 8177	95C60 8150 8151 8157 8158 8159 8171/72 8175/76 8177	95C60 8150 8151 8157 8158 8159 8171/72 8175/76 8177
Sort/Search Accelerator	95C85	95C85	95C85	95C85	95C85
ISDN	79C30 79C31 79C32 79C33 7936 7938	79C30 79C31 79C32 79C33 7936 7938	79C30 79C31 79C32 79C33 7936 7938	79C30 79C31 79C32 79C33 7936 7938	79C30 79C31 79C32 79C33 7936 7938
Modem	7910 79C12	7910 79C12	7910 79C12	7910 79C12	7910 79C12

*Trademarks of Intel, Inc.
**Z8001 and Z8002 are trademarks of Zilog, Inc.

ADDITIONAL SUPPORT LITERATURE

The following is a list of AMD product literature which, in addition to the data sheets contained in this book, can be ordered from your local AMD Sales Representative or the Literature Distribution Center at (800) 538-8450, extension 5000; inside California, call (408) 749-5000.

Key: A.N. = Application Note
A.R. = Article Reprint
H.B. = Hand Book
P.D. = Product Description
R.R. = Reliability Report
T.M. = Technical Manual

DEVICE	ORDER NUMBER	DATE OF PUBLICATION	TITLE
Am9511A/8231A	615C	08/86	Floating-Point Processor (T.M.)
Am9513A	3402D	12/84	System Timing Controller (T.M.)
	8217A	06/86	Addendum to Technical Manual
Am9516A	7737A	01/86	High-Performance DMA for the VME bus (A.R.)
	4910A	10/84	Am9516/AmZ8016 Direct Memory Access Controller (T.M.)
	3334B	09/85	Am9516 Universal DMA Controller (P.D.)
Am9517A/8237A	92C	05/85	Multimode Direct Memory Access Controller (T.M.)
Am9518/AmZ8068	442B	06/85	Data Ciphering Processor (P.D.)
	1275A	12/81	Single-Chip Encrypts Data at 14 Mb/s (A.R.)
	5432A	10/86	Keep Your Secrets Secret with the Am9518 (A.R.)
	7214A	09/85	Cryptographic Chips Safeguard Computer Data (A.R.)
	4862B	05/85	Data Ciphering Processors — Am9518, Am9568, AmZ8068 (T.M.)
Am9519	5186B	05/85	Am9519A Universal Interrupt Controller (T.M.)
Am9520/Am9521/ AmZ8065	1888A	03/82	The Am9520 Burst Error Processor (P.D.)
Am9568	4862B	05/85	Data Ciphering Processors — Am9518, Am9568, AmZ8068, (T.M.)
Am9580/Am9580A	6289A	03/82	Winchester/Floppy Controller Eases Disk Interfacing (A.R.)
	7270A	10/85	Disk Drive Controller ICs (A.R.)
	8485A	08/86	Controllers Wring Peak Performance Out of Disk Drives (A.R.)
Am9582	8484A	08/86	Disk Controller IC Varies SCSI Bus Interface Designs (A.R.)
	8485A	08/86	Controllers Wring Peak Performance Out of Disk Drives (A.R.)
Am95C85	8456A	07/86	Hardware Sorting Chip Steps Up Software Pace (A.R.)
	8035A	07/86	Content Addressable Data Manager (T.M.)
	7012B	08/86	The Am95C85 Content Addressable Data Manager (P.D.)
	09491A	03/87	Chip Accelerates Sorting, Searching (A.R.)
80C31/80C51	6390A	10/86	Controller Chips Pair Up in 'S' Interface for ISDN Systems (A.R.)
Z8030/Z8530	3335A	04/84	AmZ8030/AmZ8530 Serial Communications Controller (P.D.)
	7338A	10/85	Using the AmZ8030/AmZ8530 SCC (A.N.)
	6390A	10/86	Controller Chips Pair Up in 'S' Interface for ISDN Systems (A.R.)
	7513A	03/86	AmZ8030/AmZ8530 Serial Communications Controller (T.M.)
	2188A	08/85	Peripheral Processor Interface Guide (T.M.)
	8713A	10/86	Advanced Packaging from AMD (A.R.)
	1916B	10/84	MOS Microprocessors and Peripherals (R.R.)
	9275A/0	1987	MOS Microprocessors and Peripherals Military Hand Book (H.B.)

INTERNATIONAL BIBLIOGRAPHY

For your reference we've included a listing of internationally published articles. Note: This list is for reference only; reprints of these articles are not available from AMD's Literature Distribution Center.

DEVICE	TITLE
Am9519A	"Am9519A Universal Interrupt Controller," by A.R. Shevekov (staff). Italy, Electronica Oggi, January 1983.
Am9518	"The Am9518 Data Chiphering Processor," by Dave MacMillan. Germany, Elektronik Informationen, February 1983.
Am9513	"Sophisticated Counter Brings New Power to Microprocessor," by Dave MacMillan. Italy, Electronica Oggi, May 1983.
Am9518/AmZ8068	"High-Speed Implementation of the Data Encryption Standard," by Subodh Banerjee. U.K., Communications Engineering International, June 1984.
Am9580/Am9581	"Two-Chip Set Winchester Disk Controller Optimizes Disk Interfacing," by Mark Young, Pradeep Padukone and Neil Adams. U.K., Micro Forecast, November 12, 1984. Australia, Australian Electronic Engineering, March 1985.
	"Winchester Controller Eases Disk Interfacing," by Mark Young. France, Electronique Industrielle, May 1985.
Am8052/Am8152	"CRT Controller Chip-set," (staff written), Germany, Markt & Technik, September 13, 1983.
Am9521	"Burst Error Correction Made Easy," by Robert Anderson and Barry Pelley. Germany, Elektronik Informationen, May 1984. U.K., New Electronics, March 1984. Italy, Selezione, January 1985.
Am7901/Am27512/8088	"Signal Processing Without a Multiplier," by Sid Martra and Munsi Haque. Australia, Australian Electronic Engineering, October 1985. Italy, Electronica Oggi, October 1986.
Am9580	"SCSI for the Am9580 HDC Takes Different DIrections for Different Performance Goals," by Mashe Segal. Germany, Elektronik Informationen, November 1985. Italy, Il Progettista, March 1987.
Am9518/Am9568	"Changing the Sampling Rate of a DSP System," by Kenn Lamb. Italy, Electronica Oggi, January 1986.
8751/Am9761	"Single-chip Microcomputer with 8K-Bytes of On-board EPROM Offers Important Design Advantages," by Gordon Burk and Rajesh Tanna. Germany, Elektronik Industrie, October 1985. Italy, Il Progettista, October 1986.
Am9518/Am9568	"Data Security Made Affordable," by Juergen Stelbrink. Germany, Elektronik, January 10, 1986. Italy, Electronica Oggi, April 1986.
Am95C85	"Hardware Sorting Chip Boosts Software Speed," Germany, Elektronik Industrie, December 1986. France, ETI, November 1986. Italy, Selezione, February 1987.
	"An Achievement in Memory Systems," by Robert Cremades. France, Processeurs et Systemes, November 1986.
Am9580/Am9582	"Disk Controller Board Handles ESDI or ST506," by Jochen Polster. Germany, Design & Elektronik, May 27, 1986.

INTERFACE SUPPORT PRODUCTS

INTERFACE SUPPORT PRODUCTS*

Part Number	Description
25LS2521	8-Bit Comparator
25LS2535	8-Bit Multiplexer for Control Storage
25LS2536	8-Bit Decoder with Control Storage
25LS2548	Chip Select Address Decoder
Am2960	Cascadable 16-Bit Error Detection and Correction
Am2961/2962	4-Bit Error Correction Multiple Bus Buffers
Am2964B	Dynamic Memory Controller
Am2965/2966	Octal Dynamic Memory Driver with Three-State Output
Am29806/29809	6-, 9-Bit Comparators
Am2982XX	8-, 9-, 10-Bit Registers
Am29827/29828	10-Bit Bus Drivers
Am29833/29834	Parity Bus Transceivers
Am2984X	8-, 9-, 10-Bit Latches
Am2985X/Am2986X	9-, 10-Bit Transceivers
Am8120	Octal D-Type Flip-Flop
Am8127	Z8000 Clock Generator
Am8163	Timing, Refresh and EDC Controller
Am8167	Timing, Refresh and EDC Controller
Am8212	8-Bit I/O Port
Am8216	4-Bit Parallel Bidirectional Bus Driver, Noninverting
Am8224	Clock Generator, 8080A Compatible
Am8226	4-Bit Parallel Bidirectional Inverting Bus Driver
Am8228	8080A System Controller and Bus Driver
Am8238	8080A System Controller and Bus Driver with Extended IOW and MEMW
Am8286/8287	Octal Bus Transceivers
Am8284A	Clock Generator & Driver for 8086, 8088 Processors
Am8288	Bus Controller
82C288	CMOS Bus Controller
82284	Clock Generator
STANDARD 20-PIN PAL FAMILY**	
AmPAL16XX	20-Pin IMOX PAL Elements with 6 Different Speed/Power Versions
ADVANCED PAL DEVICES**	
AmPAL22V10	24-Pin IMOX PAL with 10 Programmable Output Logic Macrocells (OLMs)
AmPAL23S8	20-Pin IMOX PAL-Based Sequencer with 4 OLMs, 4 Output Registers, & 6 Buried State Registers
AmPAL29M16	24-Pin E^2-Based CMOS PAL with 16 OLMs and 2 Clock Inputs
AmPAL29MA16	24-Pin E^2-Based Asynchronous CMOS PAL with 16 OLMs and Product Term Driven Clocks
24-PIN PAL FAMILY**	
AmPAL20XX	24-Pin PAL Family with 8 Registered/Combinatorial Outputs
AmPAL20X10	20-Input, 10-Output Combinatorial PAL
AmPAL20RPX	10 Registered/Combinatorial Outputs and Programmable Polarity
AmPAL20XRPX	20RPX with EXCLUSIVE-OR Capability

* Refer to the Bipolar Microprocessor Logic and Interface Data Book for complete product information.
** Refer to the Programmable Array Logic Handbook/Data Book for complete product information.

ADVANCED GENERAL-PURPOSE PERIPHERALS

*Note: The document included in this publication is a condensed version of a larger, stand-alone sheet. If you would like a copy of the complete data sheet, please contact the AMD Literature Distrbution Center.

AMD LITERATURE

To obtain literature in the U.S., write or call the AMD Literature Distribution Center, 901 Thompson Place, P.O. Box 3453 — M/S 82, Sunnyvale, CA 94088; (408) 732-2400, TOLL FREE (800) 538-8450. To obtain literature from international locations, contact the nearest AMD sales office or distributor (see listings in the back of this publication).

Am29000

Streamlined Instruction Processor

ADVANCE INFORMATION

DISTINCTIVE CHARACTERISTICS

- Full 32-bit, three-bus architecture
- 17 million instructions per second (MIPS) sustained
- 25-MHz operating frequency
- Efficient execution of high-level language programs
- CMOS technology
- 4-gigabyte virtual address space with demand paging
- Concurrent instruction and data accesses
- Burst-mode access support
- 192 general-purpose registers
- 512-byte Branch Target Cache
- 64-entry Memory-Management Unit
- De-multiplexed, pipelined address, instruction, and data buses
- Three-address instruction architecture

GENERAL DESCRIPTION

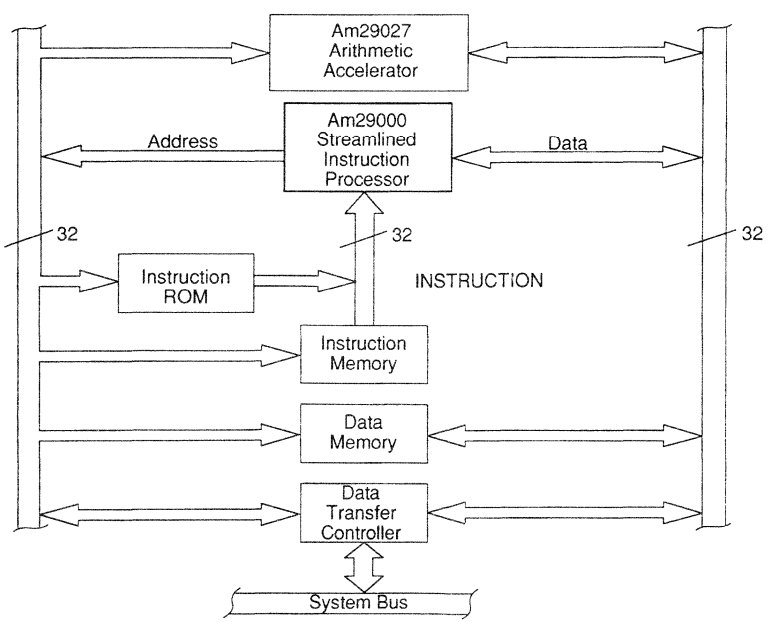
The Am29000 Streamlined Instruction Processor is a high-performance, general-purpose, 32-bit microprocessor implemented in CMOS technology. It supports a variety of applications, by virtue of a flexible architecture and rapid execution of simple instructions which are common to a wide range of tasks.

The Am29000 efficiently performs operations common to all systems, while deferring most decisions on system policies to the system architect. It is well suited for application in high-performance workstations, general-purpose super-minicomputers, high-performance real-time controllers, laser printer controllers, network protocol converters, and many other applications where high-performance, flexibility, and the ability to program using standard software tools is important.

The Am29000 instruction set has been influenced by the results of high-level-language, optimizing-compiler research. It is appropriate for a variety of languages, because it efficiently executes operations which are common to all languages. Consequently, the Am29000 is an ideal target for high-level languages such as C, Fortran, Pascal, and ADA.

The processor is packaged in a 169-terminal pin-grid-array (PGA) package, using 141 signal pins, 27 power and ground pins, and 1 alignment pin. A representative system diagram is shown below.

SIMPLIFIED SYSTEM DIAGRAM



BD007160

Publication # 09075 | Rev. A | Amendment /0
Issue Date: February 1987

Am29027

Arithmetic Accelerator

ADVANCE INFORMATION

DISTINCTIVE CHARACTERISTICS

- High-speed floating-point accelerator for the Am29000 processor
- Comprehensive floating-point and integer instruction sets
- Single-, double-, and mixed-precision operations
- Performs conversions between precisions and between data formats
- Compatible with industry-standard floating-point formats
  - IEEE P754 version 10.1
  - DEC F, DEC D, and DEC G formats
  - IBM system 370 format
- Exact IEEE compliance for denormalized numbers
- Simple interface requires no glue logic between Am29000 and Am29027
- Eight-deep register file for intermediate results and on-chip 64-bit datapath facilitate compound operations, e.g., Newton-Raphson division, sum-of-products, and transcendentals
- Supports pipelined or flow-through operation
  - Performs single- and double-precision floating-point operations at 120-ns pipelined rate
- Full compiler and assembler support
- Fabricated with Advanced Micro Devices' 1.2 micron CMOS process

GENERAL DESCRIPTION

The Am29027 Arithmetic Accelerator is a high-speed computational unit intended for use with the Am29000 Streamlined Instruction Processor (SIP). When added to a 29000-based system, the Am29027 can improve floating-point performance by an order of magnitude or more.

The Am29027 implements an extensive floating-point and integer instruction set, and can perform operations on single-, double- or mixed-precision operands. The three most popular floating-point formats (IEEE, DEC, and IBM) are supported. IEEE operations comply with standard P754, with direct implementation of special features such as gradual underflow and trap handling.

The Am29027 consists of a 64-bit ALU, a 64-bit datapath, and a control unit. The ALU has three data input ports, and can perform compound operations of the form (A * B) + C. The datapath comprises two 64-bit input operand registers, an 8-by-64-bit register file for storage of intermediate results, three operand selection multiplexers that provide for orthogonal selection of input operands, and an output multiplexer that allows access to result data, operation status, flags, or accelerator state. The control unit interprets transaction requests from the Am29000, and sequences the ALU and datapath.

Operations can be performed in either of two modes: flow-through or pipelined. In the flow-through mode, the ALU is completely combinatorial; this mode is best suited for scalar operations. Pipelined mode divides the ALU into one or two pipelined stages for use in vector operations, such as those found in graphics or signal processing.

The Am29027 connects directly to Am29000 system buses, and requires no additional interface circuitry.

Fabricated with AMD's 1.2 micron technology, the Am29027 is housed in a 169-lead pin-grid-array (PGA) package.

Publication # 09114 | Rev. A | Amendment /0
Issue Date: February 1987

Am5380

SCSI Interface Controller

PRELIMINARY

DISTINCTIVE CHARACTERISTICS

SCSI Interface

- Asynchronous interface to 1.5 megabytes per second
- Supports Initiator and Target roles
- Parity generation with optional checking
- Supports Arbitration
- Direct control of all bus signals
- High current outputs drive SCSI Bus directly

CPU Interface

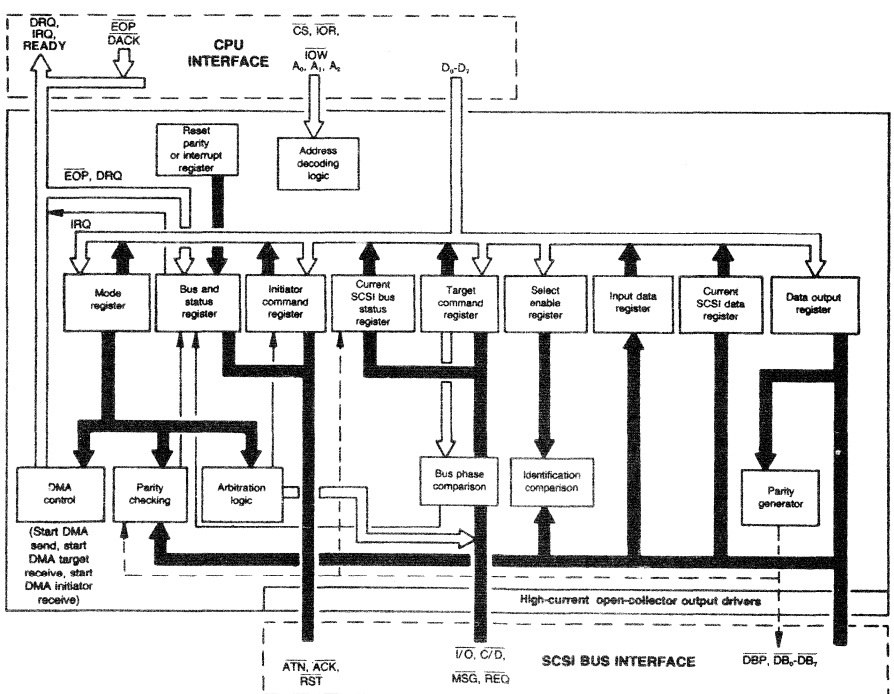
- Memory or I/O mapped interface
- DMA or programmed I/O
- Normal or Block mode DMA
- Optional CPU interrupts

GENERAL DESCRIPTION

The Am5380 Small Computer Systems Interface (SCSI) Interface Controller is a 40-pin NMOS device designed to accommodate the SCSI as defined by the ANSI X3T9.2 committee. The Am5380 operates in both the Initiator and Target roles and can, therefore, be used in host adapter, host port and formatter designs. This device supports Arbitration, including Reselection. Special high-current open-collector output drivers, capable of sinking 48 mA at 0.5 V, allow for direct connection to the SCSI Bus.

The Am5380 communicates with the system microprocessor as a peripheral device. The chip is controlled by reading and writing several internal registers which may be addressed as standard or memory-mapped I/O. Minimal processor intervention is required for DMA transfers because the Am5380 controls the necessary handshake signals. The Am5380 interrupts the CPU when it detects a bus condition that requires attention. Normal and Block mode DMA is provided to match many popular DMA controllers.

BLOCK DIAGRAM



BD006561

Publication #	Rev.	Amendment
08289	A	/0

Issue Date: July 1986

Am9511A/8231A*

Arithmetic Processor

DISTINCTIVE CHARACTERISTICS

- 2, 3 and 4MHz operation; fixed point 16-bit and 32-bit operations
- Floating point 32-bit operations; binary data formats
- Add, Subtract, Multiply and Divide; trigonometric and inverse trigonometric functions
- Square roots, logarithms, exponentiation; float to fixed and fixed to float conversions
- Stack-oriented operand storage; DMA or programmed I/O data transfers
- End signal simplifies concurrent processing; Synchronous/Asynchronous operations
- General purpose 8-bit data bus interface; standard 24-pin package
- +12 volt and +5 volt power supplies; advanced N-channel silicon gate MOS technology

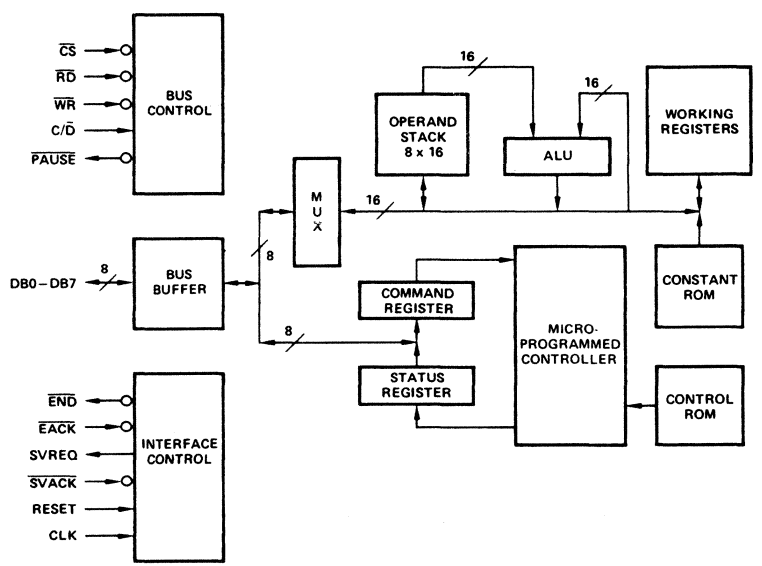
GENERAL DESCRIPTION

The Am9511A/8231A Arithmetic Processing Unit (APU) is a monolithic MOS/LSI device that provides high performance fixed and floating point arithmetic and a variety of floating point trigonometric and mathematical operations. It may be used to enhance the computational capability of a wide variety of processor-oriented systems.

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 a command is issued to perform operations on the data in the stack. Results are then available to be retrieved from the stack, or additional commands may be entered.

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.

BLOCK DIAGRAM

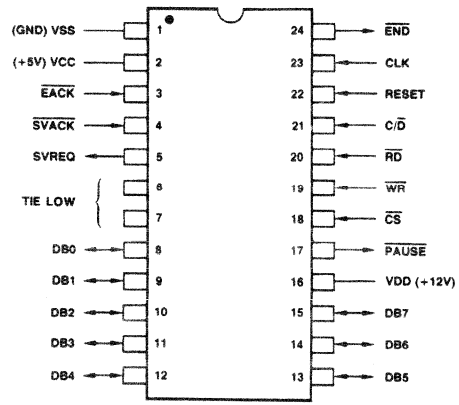


BD003340

*The 8231A is an AMD-invented device commonly referred to as the Am9511A.

Publication # 01892 | Rev. C | Amendment /0
Issue Date: May 1987

CONNECTION DIAGRAM
Top View
DIP



CD005172

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing

AM9511A -1 D C B

e. OPTIONAL PROCESSING
Blank = Standard processing
B = Burn-in

d. TEMPERATURE RANGE*
C = Commercial (0 to +70°C)
I = Industrial (−40 to +85°C)

c. PACKAGE TYPE
D = 24-Pin Ceramic DIP (CD 024)

b. SPEED OPTION
Blank = 2 MHz
−1 = 3 MHz
−4 = 4 MHz

a. DEVICE NUMBER/DESCRIPTION
Am9511A
Arithmetic Processor

Valid Combinations	
AM9511A	DC, DCB, DI,
AM9511A-1	DIB
AM9511A-4	DC, DCB

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
2	V_{CC}		+5V Power Supply.
16	V_{DD}		+12V Power Supply.
1	V_{SS}		Ground.
23	CLK	I	(Clock). An external timing source connected to the CLK input provides the necessary clocking. The CLK input can be asynchronous to the $\overline{RD}$ and $\overline{WR}$ control signals.
22	RESET	I	(Reset). A HIGH on this input causes initialization. Reset terminates any operation in progress and clears the status register to zero. The internal stack pointer is initialized, and the contents of the stack may be affected, but the command register is not affected by the reset operation. After a reset the $\overline{END}$ output will be HIGH, and the SVREQ output will be LOW. For proper initialization, the RESET input must be HIGH for at least five CLK periods following stable power supply voltages and stable clock.
21	C/$\overline{D}$	I	(Command/Data Select). The C/$\overline{D}$ input, with the $\overline{RD}$ and $\overline{WR}$ inputs, determines the type of transfer to be performed on the data bus as follows: C/$\overline{D}$ | $\overline{RD}$ | $\overline{WR}$ | Function L | H | L | Push data byte into the stack L | L | H | Pop data byte from the stack H | H | L | Enter command byte from the data bus H | L | H | Read Status X | L | L | Undefined L = LOW H = HIGH X = DON'T CARE
24	$\overline{END}$	O	(End of Execution). A LOW on this output indicates that execution of the current command is complete. This output will be cleared HIGH by activating the $\overline{EACK}$ input LOW or performing any read or write operation or device initialization using the RESET. If $\overline{EACK}$ is tied LOW, the $\overline{END}$ output will be a pulse (see $\overline{EACK}$ description). This is an open drain output and requires a pull up to +5V. Reading the status register while a command execution is in progress is allowed. However, any read or write operation clears the flip-flop that generates the $\overline{END}$ output. Thus, such continuous reading could conflict with internal logic setting the $\overline{END}$ flip-flop at the completion of command execution.
3	$\overline{EACK}$	I	(End Acknowledge). This input when LOW makes the $\overline{END}$ output go HIGH. As mentioned earlier, LOW on the $\overline{END}$ output signals completion of a command execution. The $\overline{END}$ output signal is derived from an internal flip-flop which is clocked at the completion of a command. This flip-flop is clocked to the reset state when $\overline{EACK}$ is LOW. Consequently, if the $\overline{EACK}$ is tied LOW, the $\overline{END}$ output will be a pulse that is approximately one CLK period wide.
5	SVREQ	O	(Service Request). A HIGH on this output indicates completion of a command. In this sense, this output is same as the $\overline{END}$ output. However, whether the SVREQ output will go HIGH at the completion of a command or not is determined by a service request bit in the command register. This bit must be 1 for SVREQ to go HIGH. The SVREQ can be cleared (i.e., go LOW) by activating the $\overline{SVACK}$ input LOW or initializing the device using the RESET. Also, the SVREQ will be automatically cleared after completion of any command that has the service request bit as 0.
4	$\overline{SVACK}$	I	(Service Acknowledge). A LOW on this input activates the reset input of the flip-flop generating the SVREQ output. If the $\overline{SVACK}$ input is permanently tied LOW, it will conflict with the internal setting of the flip-flop to generate the SVREQ output. Thus the SVREQ indication cannot be relied upon if the $\overline{SVACK}$ is tied LOW.
8-15	DB0-DB7	I/O	(Bidirectional Data Bus). These eight bidirectional lines are used to transfer command, status and operand information between the device and the host processor. DB0 is the least significant, and DB7 is the most significant bit position. HIGH on the data bus line corresponds to 1, and LOW corresponds to 0. When pushing operands on the stack using the data bus, the least significant byte must be pushed first and most significant byte last. When popping the stack to read the result of an operation, the most significant byte will be available on the data bus first, and the least significant byte will be the last. Moreover, for pushing operands and popping results, the number of transactions must be equal to the proper number of bytes appropriate for the chosen format. Otherwise, the internal byte pointer will not be aligned properly. The Am9511A/8231A single precision format requires 2 bytes; double precision and floating-point formats require 4 bytes.

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
18	$\overline{CS}$	I	(Chip Select). This input must be LOW to accomplish any read or write operation to the Am9511A. To perform a write operation, data is presented on DB0 through DB7 lines, C/$\overline{D}$ is driven to an appropriate level and the $\overline{CS}$ input is made LOW. However, actual writing into the Am9511A/8231A cannot start until $\overline{WR}$ is made LOW. After initiating the write operation by a $\overline{WR}$ HIGH-to-LOW transition, the $\overline{PAUSE}$ output will go LOW momentarily (TPPWW). The $\overline{WR}$ input can go HIGH after $\overline{PAUSE}$ goes HIGH. The data lines, C/$\overline{D}$ input and the $\overline{CS}$ input can change when appropriate hold time requirements are satisfied. See Write Timing diagram for details. To perform a read operation, an appropriate logic level is established on the C/$\overline{D}$ input, and $\overline{CS}$ is made LOW. The read operation does not start until the $\overline{RD}$ input goes LOW. $\overline{PAUSE}$ will go LOW for a period of TPPWR. When $\overline{PAUSE}$ goes back HIGH again, it indicates that the read operation is complete, and the required information is available on the DB0 through DB7 lines. This information will remain on the data lines as long as the $\overline{RD}$ input is LOW. The $\overline{RD}$ input can return HIGH anytime after $\overline{PAUSE}$ goes HIGH. The $\overline{CS}$ input and C/$\overline{D}$ inputs can change anytime after $\overline{RD}$ returns HIGH. See Read Timing diagram for details.
20	$\overline{RD}$	I	(Read). A LOW on this input is used to read information from an internal location and gate that information on to the data bus. The $\overline{CS}$ input must be LOW to accomplish the read operation. The C/$\overline{D}$ input determines what internal location is of interest. See C/$\overline{D}$, $\overline{CS}$ input descriptions and Read Timing diagram for details. If the END output was LOW, performing any read operation will make the $\overline{END}$ output go HIGH after the HIGH-to-LOW transition of the $\overline{RD}$ input (assuming $\overline{CS}$ is LOW).
19	$\overline{WR}$	I	(Write). A LOW on this input is used to transfer information from the data bus into an internal location. The $\overline{CS}$ must be LOW to accomplish the write operation. The C/$\overline{D}$ determines which internal location is to be written. See C/$\overline{D}$, $\overline{CS}$ input descriptions and Write Timing diagram for details. If the $\overline{END}$ output was LOW, performing any write operation will make the $\overline{END}$ output go HIGH after the LOW-to-HIGH transition of the $\overline{WR}$ input (assuming $\overline{CS}$ is LOW).
17	$\overline{PAUSE}$	O	(Pause). This output is a handshake signal used while performing read or write transactions with the Am9511A. A LOW at this output indicates that the Am9511A/8231A has not yet completed its information transfer with the host over the data bus. During a read operation, after $\overline{CS}$ went LOW, the $\overline{PAUSE}$ will become LOW shortly (TRP) after $\overline{RD}$ goes LOW. $\overline{PAUSE}$ will return high only after the data bus contains valid output data. The $\overline{CS}$ and $\overline{RD}$ should remain LOW when $\overline{PAUSE}$ is LOW. The RD may go high anytime after $\overline{PAUSE}$ goes HIGH. During a write operation, after $\overline{CS}$ went LOW, the $\overline{PAUSE}$ will be LOW for a very short duration (TPPWW) after $\overline{WR}$ goes LOW. Since the minimum of TPPWW is 0, the $\overline{PAUSE}$ may not go LOW at all for fast devices. $\overline{WR}$ may go HIGH anytime after $\overline{PAUSE}$ goes HIGH.

DETAILED DESCRIPTION

Major functional units of the Am9511A/8231A are shown in the block diagram. The Am9511A/8231A employs a microprogram controlled stack oriented architecture with 16-bit wide data paths.

The Arithmetic Logic Unit (ALU) receives one of its operands from the Operand Stack. This stack is an 8-word by 16-bit 2-port memory with last in-first out (LIFO) attributes. The second operand to the ALU is supplied by the internal 16-bit bus. In addition to supplying the second operand, this bidirectional bus also carries the results from the output of the ALU when required. Writing into the Operand Stack takes place from this internal 16-bit bus when required. Also connected to this bus are the Constant ROM and Working Registers. The ROM provides the required constants to perform the mathematical operations (Chebyshev Algorithms), while the Working Registers provide storage for the intermediate values during command execution.

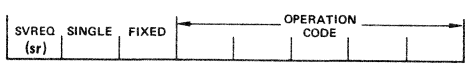
Communication between the external world and the Am9511A/8231A takes place on eight bidirectional input/output lines DB0 through DB7 (Data Bus). These signals are gated to the internal eight-bit bus through appropriate interface and buffer circuitry. Multiplexing facilities exist for bidirectional communication between the internal eight- and sixteen-bit buses. The Status Register and Command Register are also accessible via the eight-bit bus.

The Am9511A/8231A operations are controlled by the microprogram contained in the Control ROM. The Program Counter supplies the microprogram addresses and can be partially loaded from the Command Register. Associated with the Program Counter is the Subroutine Stack where return addresses are held during subroutine calls in the microprogram. The Microinstruction Register holds the current microinstruction being executed. This register facilitates pipelined microprogram execution. The Instruction Decode logic generates various internal control signals needed for the Am9511A/8231A operation.

The Interface Control logic receives several external inputs and provides handshake related outputs to facilitate interfacing the Am9511A/8231A to microprocessors.

Command Format

Each command entered into the Am9511A/8231A consists of a single 8-bit byte having the format illustrated below:



TB000063

Bits 0 – 4 select the operation to be performed as shown in the table. Bits 5 – 6 select the data format for the operation. If bit 5 is a 1, a fixed point data format is specified. If bit 5 is a 0, floating point format is specified. Bit 6 selects the precision of the data to be operated on by fixed point commands (if bit 5 = 0, bit 6 must be 0). If bit 6 is a 1, single-precision (16-bit) operands are indicated; 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 a succeeding command where bit 7 is 0. Each command issued to the Am9511A/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.

COMMAND SUMMARY

Command Code								Command Mnemonic	Command Description
7	6	5	4	3	2	1	0		
FIXED-POINT 16-BIT									
sr	1	1	0	1	1	0	0	SADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	1	1	0	1	1	0	1	SSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	1	1	0	1	1	1	0	SMUL	Multiply NOS by TOS. Lower half of result to NOS. Pop Stack.
sr	1	1	1	0	1	1	0	SMUU	Multiply NOS by TOS. Upper half of result to NOS. Pop Stack.
sr	1	1	0	1	1	1	1	SDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
FIXED-POINT 32-BIT									
sr	0	1	0	1	1	0	0	DADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	0	1	0	1	1	0	1	DSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	0	1	0	1	1	1	0	DMUL	Multiply NOS by TOS. Lower half of result to NOS. Pop Stack.
sr	0	1	1	0	1	1	0	DMUU	Multiply NOS by TOS. Upper half of result to NOS. Pop Stack.
sr	0	1	0	1	1	1	1	DDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
FLOATING-POINT 32-BIT									
sr	0	0	1	0	0	0	0	FADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	0	1	FSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	1	0	FMUL	Multiply NOS by TOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	1	1	FDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
DERIVED FLOATING-POINT FUNCTIONS									
sr	0	0	0	0	0	0	1	SQRT	Square Root of TOS. Result in TOS.
sr	0	0	0	0	0	1	0	SIN	Sine of TOS. Result in TOS.
sr	0	0	0	0	0	1	1	COS	Cosine of TOS. Result in TOS.
sr	0	0	0	0	1	0	0	TAN	Tangent of TOS. Result in TOS.
sr	0	0	0	0	1	0	1	ASIN	Inverse Sine of TOS. Result in TOS.
sr	0	0	0	0	1	1	0	ACOS	Inverse Cosine of TOS. Result in TOS.
sr	0	0	0	0	1	1	1	ATAN	Inverse Tangent of TOS. Result in TOS.
sr	0	0	0	1	0	0	0	LOG	Common Logarithm (base 10) of TOS. Result in TOS.
sr	0	0	0	1	0	0	1	LN	Natural Logarithm (base e) of TOS. Result in TOS.
sr	0	0	0	1	0	1	0	EXP	Exponential (e^X) of TOS. Result in TOS.
sr	0	0	0	1	0	1	1	PWR	NOS raised to the power in TOS. Result in NOS. Pop Stack.
DATA MANIPULATION COMMANDS									
sr	0	0	0	0	0	0	0	NOP	No Operation.
sr	0	0	1	1	1	1	1	FIXS	Convert TOS from floating-point to 16-bit fixed-point format.
sr	0	0	1	1	1	1	0	FIXD	Convert TOS from floating-point to 32-bit fixed-point format.
sr	0	0	1	1	1	0	1	FLTS	Convert TOS from 16-bit fixed-point to floating-point format.
sr	0	0	1	1	1	0	0	FLTD	Convert TOS from 32-bit fixed-point to floating-point format.
sr	1	1	1	0	1	0	0	CHSS	Change sign of 16-bit fixed-point operand on TOS.
sr	0	1	1	0	1	0	0	CHSD	Change sign of 32-bit fixed-point operand on TOS.
sr	0	0	1	0	1	0	1	CHSF	Change sign of floating-point operand on TOS.
sr	1	1	1	0	1	1	1	PTOS	Push 16-bit fixed-point operand on TOS to NOS. (Copy.)
sr	0	1	1	0	1	1	1	PTOD	Push 32-bit fixed-point operand on TOS to NOS. (Copy.)
sr	0	0	1	0	1	1	1	PTOF	Push floating-point operand on TOS to NOS. (Copy.)
sr	1	1	1	1	0	0	0	POPS	Pop 16-bit fixed-point operand from TOS. NOS becomes TOS.
sr	0	1	1	1	0	0	0	POPD	Pop 32-bit fixed-point operand from TOS. NOS becomes TOS.
sr	0	0	1	1	0	0	0	POPF	Pop floating-point operand from TOS. NOS becomes TOS.
sr	1	1	1	1	0	0	1	XCHS	Exchange 16-bit fixed-point operands TOS and NOS.
sr	0	1	1	1	0	0	1	XCHD	Exchange 32-bit fixed-point operands TOS and NOS.
sr	0	0	1	1	0	0	1	XCHF	Exchange floating-point operands TOS and NOS.
sr	0	0	1	1	0	1	0	PUPI	Push floating-point constant "π" onto TOS. Previous TOS becomes NOS.

Notes:
1. TOS means Top of Stack. NOS means Next on Stack.
2. AMD Application Brief "Algorithm Details for the Am9511A APU" provides detailed descriptions of each command function, including data ranges, accuracies, stack configurations, etc.
3. Many commands destroy one stack location (bottom of stack) during development of the result. The derived functions may destroy several stack locations. See Application Brief for details.
4. The trigonometric functions handle angles in radians, not degrees.
5. No remainder is available for the fixed-point divide functions.
6. Results will be undefined for any combination of command coding bits not specified in this table.

Command Initiation

After properly positioning the required operands on the stack, a command may be issued. The procedure for initiating a command execution is as follows:

1. Enter the appropriate command on the DB0 – DB7 lines.
2. Establish HIGH on the C/$\overline{D}$ input.
3. Establish LOW on the $\overline{CS}$ input.
4. Establish LOW on the $\overline{WR}$ input after an appropriate set-up time (see Timing diagrams).
5. Sometime after the HIGH-to-LOW level transition of $\overline{WR}$ input, the $\overline{PAUSE}$ output will become LOW. After a delay of TPPWW, it will go HIGH to acknowledge the write operation. The WR input can return to HIGH anytime after $\overline{PAUSE}$ going HIGH. The DB0 – DB7, C/$\overline{D}$ and $\overline{CS}$ inputs are allowed to change after the hold time requirements are satisfied (see Timing diagram).

An attempt to issue a new command while the current command execution is in progress is allowed. Under these circumstances, the $\overline{PAUSE}$ output will not go HIGH until the current command execution is completed.

Operand Entry

The Am9511A/8231A commands operate on the operands located at the TOS and NOS, and results are returned to the stack at NOS and then popped to TOS. The operands required for the Am9511A/8231A are one of three formats – single precision fixed-point (2 bytes), double precision fixed-point (4 bytes) or floating-point (4 bytes). The result of an operation

has the same format as the operands except for float to fix or fix to float commands.

Operands are always entered into the stack least significant byte first and most significant byte last. The following procedure must be followed to enter operands onto the stack:

1. The lower significant operand byte is established on the DB0 – DB7 lines.
2. A LOW is established on the C/$\overline{D}$ input to specify that data is to be entered into the stack.
3. The $\overline{CS}$ input is made LOW.
4. After appropriate set-up time (see Timing diagrams), the $\overline{WR}$ input is made LOW. The $\overline{PAUSE}$ output will become LOW.
5. Sometime after this event, the $\overline{PAUSE}$ will return HIGH to indicate that the write operation has been acknowledged.
6. Anytime after the $\overline{PAUSE}$ output goes HIGH, the $\overline{WR}$ input can be made HIGH. The DB0 – DB7, C/$\overline{D}$ and $\overline{CS}$ inputs can change after appropriate hold time requirements are satisfied (see Timing diagrams).

The above procedure must be repeated until all bytes of the operand are pushed onto the stack. It should be noted that for single precision fixed-point operands, 2 bytes should be pushed and 4 bytes must be pushed for double precision fixed-point or floating-point. Not pushing all the bytes of a quantity will result in byte pointer misalignment.

The Am9511A/8231A stack can accommodate 8 single precision fixed-point quantities or 4 double precision fixed-point or floating-point quantities. Pushing more quantities than the capacity of the stack will result in loss of data which is usual with any LIFO stack.

Data Removal

Result from an operation will be available at the TOS. Results can be transferred from the stack to the data bus by reading the stack. When the stack is popped for results, the most significant byte is available first and the least significant byte last. A result is always of the same precision as the operands that produced it except for format conversion commands. Thus when the result is taken from the stack, the total number of bytes popped out should agree with the precision – single precision results are 2 bytes, and double precision and floating-point results are 4 bytes. The following procedure must be used for reading the result from the stack:

1. A LOW is established on the C/$\overline{D}$ input.
2. The $\overline{CS}$ input is made LOW.
3. After appropriate set-up time (see Timing diagrams), the $\overline{RD}$ input is made LOW. The $\overline{PAUSE}$ will become LOW.
4. Sometime after this, $\overline{PAUSE}$ will return HIGH, indicating that the data is available on the DB0 – DB7 lines. This data will remain on the DB0 – DB7 lines as long as the $\overline{RD}$ input remains LOW.
5. Anytime after $\overline{PAUSE}$ goes HIGH, the $\overline{RD}$ input can return HIGH to complete transaction.
6. The $\overline{CS}$ and C/$\overline{D}$ inputs can change after appropriate hold time requirements are satisfied (see Timing diagram).
7. Repeat this procedure until all bytes appropriate for the precision of the result are popped out.

Reading of the stack does not alter its data; it only adjusts the byte pointer. If more data is popped than the capacity of the stack, the internal byte pointer will wrap around and older data will be read again, consistent with the LIFO stack.

Status Read

The Am9511A/8231A status register can be read without any regard to whether a command is in progress or not. The only implication that has to be considered is the effect this might have on the END output discussed in the signal descriptions.

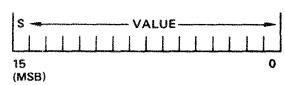
The following procedure must be followed to accomplish status register reading:

1. Establish HIGH on the C/$\overline{D}$ input.
2. Establish LOW on the $\overline{CS}$ input.
3. After appropriate set-up time (see Timing diagram) $\overline{RD}$ input is made LOW. The $\overline{PAUSE}$ will become LOW.
4. Sometime after the HIGH-to-LOW transition of $\overline{RD}$ input, the $\overline{PAUSE}$ will become HIGH, indicating that status register contents are available on the DB0 – DB7 lines. The status data will remain on DB0 – DB7 as long as $\overline{RD}$ input is LOW.
5. The $\overline{RD}$ input can be returned HIGH anytime after $\overline{PAUSE}$ goes HIGH.
6. The C/$\overline{D}$ input and $\overline{CS}$ input can change after satisfying appropriate hold time requirements (see Timing diagram).

Data Formats

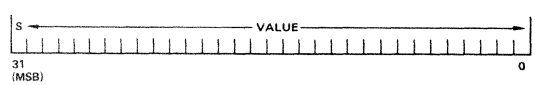
The Am9511A/8231A Arithmetic Processing Unit handles operands in both fixed-point and floating-point formats. Fixed-point operands may be represented in either single (16-bit operands) or double precision (32-bit operands), and are always represented as binary, two's complement values.

16-BIT FIXED-POINT FORMAT



TB00064M

32-BIT FIXED-POINT FORMAT



TB00065M

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 −32,767 to +32,767 for single precision and −2,147,483,647 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 if the exponent field is two digits wide and the mantissa is five digits, a range of values (positive or negative) from 1.000×10^{-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 is 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 Am9511/8231A is a binary arithmetic processor and requires that floating point data be represented by a fractional mantissa value between .5 and 1 multiplied by 2 raised to an appropriate power. This is expressed as follows:

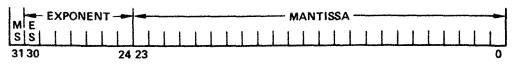
$$\text{value} = \text{mantissa} \times 2^{\text{exponent}}$$

For example, the value 100.5 expressed in this form is 0.11001001×2^{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 Am9511A/8231A is given below. The mantissa is expressed as a 24-bit (fractional) value; the exponent is expressed as an unbiased 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 to 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.



TB000066

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.

Status Register

The Am9511A/8231A contains an eight bit status register with the following bit assignments:

BUSY	SIGN	ZERO	ERROR CODE				CARRY
7	6	5	4	3	2	1	0

BUSY: Indicates that Am9511A/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 error
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.

Table 1.

Command Mnemonic	Hex Code (sr = 1)	Hex Code (sr = 0)	Execution Cycles	Summary Description
16-BIT FIXED-POINT OPERATIONS				
SADD	EC	6C	16-18	Add TOS to NOS. Result to NOS. Pop Stack.
SSUB	ED	6D	30-32	Subtract TOS from NOS. Result to NOS. Pop Stack.
SMUL	EE	6E	84-94	Multiply NOS by TOS. Lower result to NOS. Pop Stack.
SMUU	F6	76	80-98	Multiply NOS by TOS. Upper result to NOS. Pop Stack.
SDIV	EF	6F	84-94	Divide NOS by TOS. Result to NOS. Pop Stack.
32-BIT FIXED-POINT OPERATIONS				
DADD	AC	2C	20-22	Add TOS to NOS. Result to NOS. Pop Stack.
DSUB	AD	2D	38-40	Subtract TOS from NOS. Result to NOS. Pop Stack.
DMUL	AE	2E	194-210	Multiply NOS by TOS. Lower result to NOS. Pop Stack.
DMUU	B6	36	182-218	Multiply NOS by TOS. Upper result to NOS. Pop Stack.
DDIV	AF	2F	196-210	Divide NOS by TOS. Result to NOS. Pop Stack.
32-BIT FLOATING-POINT PRIMARY OPERATIONS				
FADD	90	10	54-368	Add TOS to NOS. Result to NOS. Pop Stack.
FSUB	91	11	70-370	Subtract TOS from NOS. Result to NOS. Pop Stack.
FMUL	92	12	146-168	Multiply NOS by TOS. Result to NOS. Pop Stack.
FDIV	93	13	154-184	Divide NOS by TOS. Result to NOS. Pop Stack.
32-BIT FLOATING-POINT DERIVED OPERATIONS				
SQRT	81	01	782-870	Square Root of TOS. Result to TOS.
SIN	82	02	3796-4808	Sine of TOS. Result to TOS.
COS	83	03	3840-4878	Cosine of TOS. Result to TOS.
TAN	84	04	4894-5886	Tangent of TOS. Result to TOS.
ASIN	85	05	6230-7938	Inverse Sine of TOS. Result to TOS.
ACOS	86	06	6304-8284	Increase Cosine of TOS. Result to TOS.
ATAN	87	07	4992-6536	Inverse Tangent of TOS. Result to TOS.
LOG	88	08	4474-7132	Common Logarithm of TOS. Result to TOS.
LN	89	09	4296-6956	Natural Logarithm of TOS. Result to TOS.
EXP	8A	0A	3794-4878	e raised to power in TOS. Result to TOS.
PWR	8B	0B	8290-12032	NOS raised to power in TOS. Result to NOS. Pop Stack.
DATA AND STACK MANIPULATION OPERATIONS				
NOP	80	00	4	No Operation. Clear or set SVREQ.
FIXS	9F	1F	90-214	Convert TOS from floating-point format to fixed-point format.
FIXD	9E	1E	90-336	
FLTS	9D	1D	62-156	Convert TOS from fixed-point format to floating-point format.
FLTD	9C	1C	56-342	
CHSS	F4	74	22-24	Change sign of fixed-point operand on TOS.
CHSD	B4	34	26-28	
CHSF	95	15	16-20	Change sign of floating-point operandon TOS.
PTOS	F7	77	16	Push stack. Duplicate NOS in TOS.
PTOD	B7	37	20	
PTOF	97	17	20	
POPS	F8	78	10	Pop stack. Old NOS becomes new TOS. Old TOS rotates to bottom.
POPD	B8	38	12	
POPF	98	18	12	
XCHS	F9	79	18	Exchange TOS and NOS.
XCHD	B9	39	26	
XCHF	99	19	26	
PUPI	9A	1A	16	Push floating-point constant π onto TOS. Previous TOS becomes NOS.

COMMAND DESCRIPTIONS

This section contains detailed descriptions of the APU commands. They are arranged in alphabetical order by command mnemonic. In the descriptions, TOS means Top Of Stack, and NOS means Next On Stack.

All derived functions except Square Root use Chebyshev polynomial approximating algorithms. This approach is used to help minimize the internal microprogram, to minimize the maximum error values and to provide a relatively even distribution of errors over the data range. The basic arithmetic operations are used by the derived functions to compute the various Chebyshev terms. The basic operations may produce error codes in the status register as a result.

Execution times are listed in terms of clock cycles and may be converted into time values by multiplying by the clock period used. For example, an execution time of 44 clock cycles when running at a 3MHz rate translates to 14 microseconds ($44 \times 32\mu s = 14\mu s$). Variations in execution cycles reflect the data dependency of the algorithms.

In some operations exponent overflow or underflow may be possible. When this occurs, the exponent returned in the result will be 128 greater or smaller than its true value.

Many of the functions use portions of the data stack as scratch storage during development of the results. Thus, previous values in those stack locations will be lost. Scratch locations destroyed are listed in the command descriptions and shown with the crossed-out locations in the Stack Contents After diagram.

Table 1 is a summary of all the Am9511A/8231A commands. It shows the hex codes for each command, the mnemonic abbreviation, a brief description and the execution time in clock cycles. The commands are grouped by functional classes.

The command mnemonics in alphabetical order are shown below in Table 2.

Table 2.
Command Mnemonics in Alphabetical Order.

ACOS	ARCCOSINE
ASIN	ARCSINE
ATAN	ARCTANGENT
CHSD	CHANGE SIGN DOUBLE
CHSF	CHANGE SIGN FLOATING
CHSS	CHANGE SIGN SINGLE
COS	COSINE
DADD	DOUBLE ADD
DDIV	DOUBLE DIVIDE
DMUL	DOUBLE MULTIPLY LOWER
DMUU	DOUBLE MULTIPLY UPPER
DSUB	DOUBLE SUBTRACT
EXP	EXPONENTATION (e^X)
FADD	FLOATING ADD
FDIV	FLOATING DIVIDE
FIXD	FIX DOUBLE
FIXS	FIX SINGLE
FLTD	FLOAT DOUBLE
FLTS	FLOAT SINGLE
FMUL	FLOATING MULTIPLY
FSUB	FLOATING SUBTRACT
LOG	COMMON LOGARITHM
LN	NATURAL LOGARITHM
NOP	NO OPERATION
POPD	POP STACK DOUBLE
POPF	POP STACK FLOATING
POPS	POP STACK SINGLE
PTOD	PUSH STACK DOUBLE
PTOF	PUSH STACK FLOATING
PTOS	PUSH STACK SINGLE
PUPI	PUSH π
PWR	POWER (X^Y)
SADD	SINGLE ADD
SDIV	SINGLE DIVIDE
SIN	SINE
SMUL	SINGLE MULTIPLY LOWER
SMUU	SINGLE MULTIPLY UPPER
SQRT	SQUARE ROOT
SSUB	SINGLE SUBTRACT
TAN	TANGENT
XCHD	EXCHANGE OPERANDS DOUBLE
XCHF	EXCHANGE OPERANDS FLOATING
XCHS	EXCHANGE OPERANDS SINGLE

ACOS

32-BIT FLOATING-POINT INVERSE COSINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	1	0

Hex Coding: 86 with sr = 1
06 with sr = 0

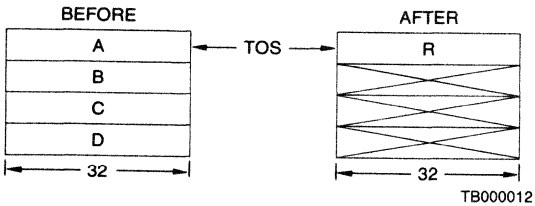
Execution Time: 6304 to 8284 k cycles clock cycles

Description:
The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point inverse cosine of A. The result R is a value in radians between 0 and π. Initial operands A, B, C and D are lost. ACOS will accept all input data values within the range of −1.0 to +1.0. Values outside this range will return an error code of 1100 in the status register.

Accuracy: ACOS exhibits a maximum relative error of 2.0×10^{-7} over the valid input data range.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000012

ATAN

32-BIT FLOATING-POINT INVERSE TANGENT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	1	1

Hex Coding: 87 with sr = 1
07 with sr = 0

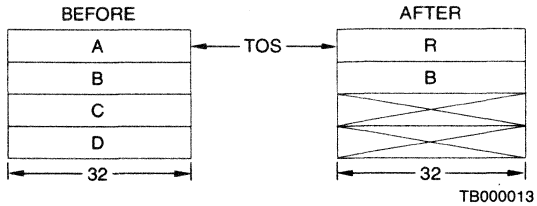
Execution Time: 4992 to 6536 clock cycles

Description:
The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point inverse tangent of A. The result R is a value in radians between $-\pi/2$ and $+\pi/2$. Initial operands A, C, and D are lost. Operand B is unchanged.
ATAN will accept all input data values that can be represented in the floating point format.

Accuracy: ATAN exhibits a maximum relative error of 3.0×10^{-7} over the input data range.

Status Affected: Sign, Zero

STACK CONTENTS



TB000013

ASIN

32-BIT FLOATING-POINT INVERSE SINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	0	1

Hex Coding: 85 with sr = 1
05 with sr = 0

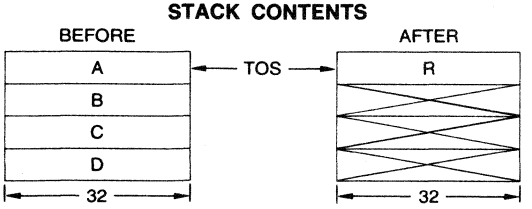
Execution Time: 6230 to 7938 k cycles clock cycles

Description:
The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point inverse sine of A. The result R is a value in radians between $-\pi/2$ and $+\pi/2$. Initial operands A, B, C, and D are lost. ASIN will accept all input data values within the range of −1.0 to +1.0. Values outside this range will return an error code of 1100 in the status register.

Accuracy: ASIN exhibits a maximum relative error of 4.0×10^{-7} over the valid input data range.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000014

CHSD

32-BIT FIXED-POINT SIGN CHANGE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	0	1	0	0

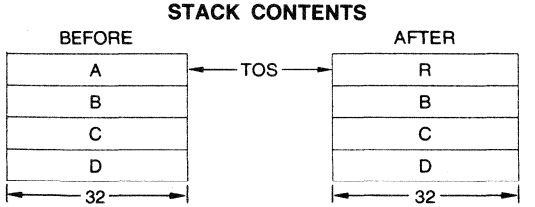
Hex Coding: B4 with sr = 1
34 with sr = 0

Execution Time: 26 to 28 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is subtracted from zero. The result R replaces A at the TOS. Other entries in the stack are not disturbed.
Overflow status will be set and the TOS will be returned unchanged when A is input as the most negative value possible in the format since no positive equivalent exists.

Status Affected: Sign, Zero, Error Field (overflow)

STACK CONTENTS



TB000047

CHSF

32-BIT FLOATING-POINT SIGN CHANGE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	1	0	1

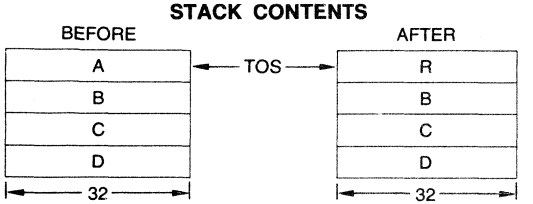
Hex Coding: 95 with sr = 1
15 with sr = 0

Execution Time: 16 to 20 clock cycles

Description:
The sign of the mantissa of the 32-bit floating-point operand A at the TOS is inverted. The result R replaces A at the TOS. Other stack entries are unchanged.
If A is input as zero (mantissa MSB = 0), no change is made.

Status Affected: Sign, Zero

STACK CONTENTS



TB000048

CHSS

16-BIT FIXED-POINT SIGN CHANGE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	0	1	0	0

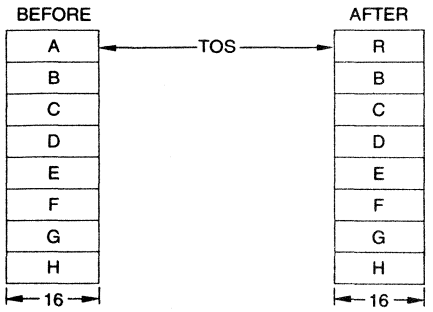
Hex Coding: F4 with sr = 1
74 with sr = 0

Execution Time: 22 to 24 clock cycles

Description:
16-bit fixed-point two's complement integer operand A at the TOS is subtracted from zero. The result R replaces A at the TOS. All other operands are unchanged.
Overflow status will be set and the TOS will be returned unchanged when A is input as the most negative value possible in the format since no positive equivalent exists.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



TB000049

COS

32-BIT FLOATING-POINT COSINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	1	1

Hex Coding: 83 with sr = 1
03 with sr = 0

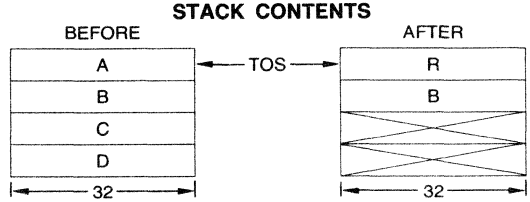
Execution Time: 3840 to 4878 clock cycles

Description:
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point cosine of A. A is assumed to be in radians. Operands A, C and D are lost. B is unchanged.
The COS function can accept any input data value that can be represented in the data format. All input values are range reduced to fall within an interval of $-\pi/2$ to $+\pi/2$ radians.

Accuracy: COS exhibits a maximum relative error of 5.0×10^{-7} for all input data values in the range of -2π to $+2\pi$ radians.

Status Affected: Sign, Zero

STACK CONTENTS



TB000015

DADD

32-BIT FIXED-POINT ADD

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	0	0

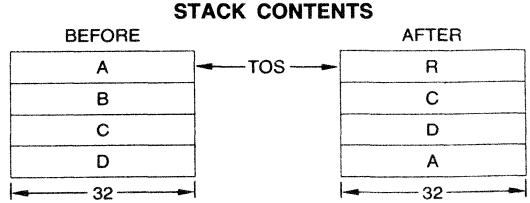
Hex Coding: AC with sr = 1
2C with sr = 0

Execution Time: 20 to 22 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is added to the 32-bit fixed-point two's complement integer operand B at the NOS. The result R replaces operand B, and the Stack is moved up so that R occupies the TOS. Operand B is lost. Operands A, C and D are unchanged. If the addition generates a carry, it is reported in the status register.
If the result is too large to be represented by the data format, the least significant 32 bits of the result are returned, and overflow status is reported.

Status Affected: Sign, Zero, Carry, Error Field

STACK CONTENTS



TB000050

DDIV

32-BIT FIXED-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	1	1

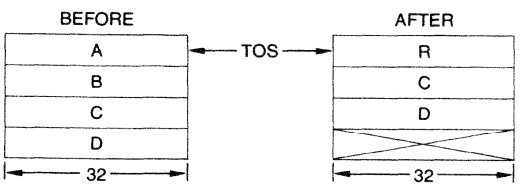
Hex Coding: AF with sr = 1
2F with sr = 0

Execution Time: 196 to 210 clock cycles when A ≠ 0
18 clock cycles when A = 0.

Description:
The 32-bit fixed-point two's complement integer operand B at NOS is divided by the 32-bit fixed-point two's complement integer operand A at the TOS. The 32-bit integer quotient R replaces B, and the stack is moved up so that R occupies the TOS. No remainder is generated. Operands A and B are lost. Operands C and D are unchanged.
If A is zero, R is set equal to B and the divided-by-zero error status will be reported. If either A or B is the most negative value possible in the format, R will be meaningless and the overflow error status will be reported.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000016

DMUU

32-BIT FIXED-POINT MULTIPLY, UPPER

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	0	1	1	0

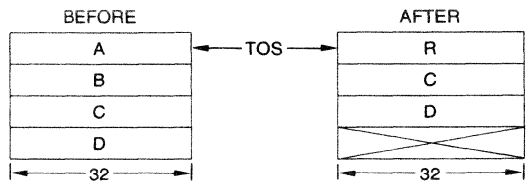
Hex Coding: B6 with sr = 1
36 with sr = 0

Execution Time: 182 to 218 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 32-bit fixed-point two's complement integer operand B at the NOS. The 32-bit most significant half of the product R replaces B and the stack is moved up so that R occupies the TOS. The least significant half of the product is lost. Operands A and B are lost. Operands C and D are unchanged.
If A or B was the most negative value possible in the format, overflow status is set and R is meaningless.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



TB000017

DMUL

32-BIT FIXED-POINT MULTIPLY, LOWER

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	1	0

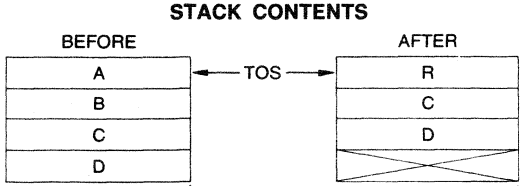
Hex Coding: AE with sr = 1
2E with sr = 0

Execution Time: 194 to 210 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 32-bit fixed-point two's complement integer operand B at the NOS. The 32-bit least significant half of the product R replaces B, and the stack is moved up so that R occupies the TOS. The most significant half of the product is lost. Operands A and B are lost. Operands C and D are unchanged.
The overflow status bit is set if the discarded upper half was non-zero. If either A or B is the most negative value that can be represented in the format, that value is returned as R and the overflow status is set.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



TB000018

DSUB

32-BIT FIXED-POINT SUBTRACT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	0	1

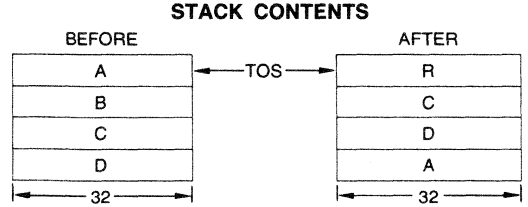
Hex Coding: AD with sr = 1
2D with sr = 0

Execution Time: 38 to 40 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is subtracted from the 32-bit fixed-point two's complement integer operand B at the NOS. The difference R replaces operand B, and the stack is moved up so that R occupies the TOS. Operand B is lost. Operands A, C and D are unchanged.
If the subtraction generates a borrow, it is reported in the carry status bit. If A is the most negative value that can be represented in the format, the overflow status is set. If the result cannot be represented in the data format range, the overflow bit is set, and the 32 least significant bits of the result are returned as R.

Status Affected: Sign, Zero, Carry, Overflow

STACK CONTENTS



TB000051

EXP

32-BIT FLOATING-POINT e^X

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	1	0

Hex Coding: 8A with sr = 1
0A with sr = 0

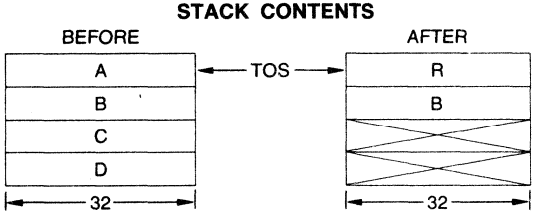
Execution Time: 3794 to 4878 clock cycles for $|A| \leq 1.0 \times 2^5$
34 clock cycles for $|A| > 1.0 \times 2^5$

Description:
The base of natural logarithms, e, is raised to an exponent value specified by the 32-bit floating-point operand A at the TOS. The result R of e^A replaces A. Operands A, C and D are lost. Operand B is unchanged.
EXP accepts all input data values within the range of $-1.0 \times 2^{+5}$ to $+1.0 \times 2^{+5}$. Input values outside this range will return a code of 1100 in the error field of the status register.

Accuracy: EXP exhibits a maximum relative error of 5.0×10^{-7} over the valid input data range.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB00019M

FADD

32-BIT FLOATING-POINT ADD

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	0	0

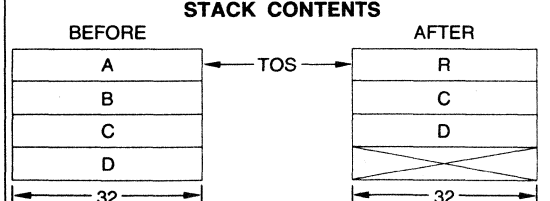
Hex Coding: 90 with sr = 1
10 with sr = 0

Execution Time: 54 to 368 clock cycles for A ≠ 0
24 clock cycles for A = 0

Description:
32-bit floating-point operand A at the TOS is added to 32-bit floating-point operand B at the NOS. The result R replaces B, and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.
Exponent alignment before the addition and normalization of the result accounts for the variation in execution time. Exponent overflow and underflow are reported in the status register, in which case the mantissa is correct and the exponent is offset by 128.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB00021M

FDIV

32-BIT FLOATING-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	1	1

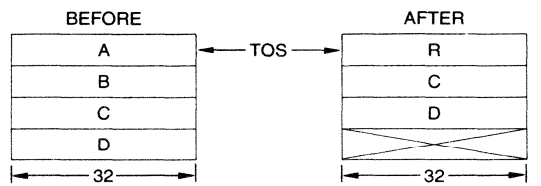
Hex Coding: 93 with sr = 1
13 with sr = 0

Execution Time: 154 to 184 clock cycles for A ≠ 0
22 clock cycles for A = 0

Description:
32-bit floating-point operand B at the NOS is divided by 32-bit floating-point operand A at the TOS. The result R replaces B, and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.
If operand A is zero, R is set equal to B and the divide-by-zero error is reported in the status register. Exponent overflow or underflow is reported in the status register, in which case the mantissa portion of the result is correct and the exponent portion is offset by 128.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000020

FIXD

32-BIT FLOATING-POINT TO 32-BIT FIXED-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	1	0

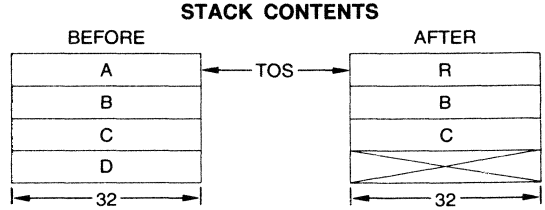
Hex Coding: 9E with sr = 1
1E with sr = 0

Execution Time: 90 to 336 clock cycles

Description:
32-bit floating-point operand A at the TOS is converted to a 32-bit fixed-point two's complement integer. The result R replaces A. Operands A and D are lost. Operands B and C are unchanged.
If the integer portion of A is larger than 31 bits when converted, the overflow status will be set and A will not be changed. Operand D, however, will still be lost.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



TB00022M

FIXS

32-BIT FLOATING-POINT TO 16-BIT FIXED-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	1	1

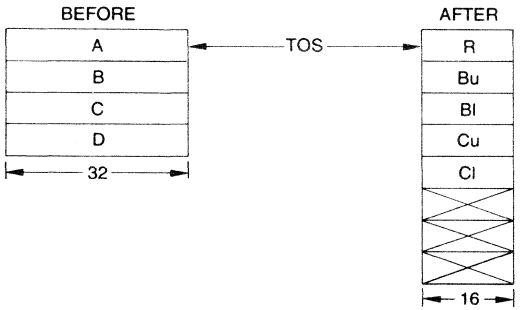
Hex Coding: 9F with sr = 1
1F with sr = 0

Execution Time: 90 to 214 clock cycles

Description:
32-bit floating-point operand A at the TOS is converted to a 16-bit fixed-point two's complement integer. The result R replaces the lower half of A, and the stack is moved up by two bytes so that R occupies the TOS. Operands A and D are lost. Operands B and C are unchanged, but appear as upper (u) and lower (l) halves on the 16-bit wide stack if they are 32-bit operands.
If the integer portion of A is larger than 15 bits when converted, the overflow status will be set and A will not be changed. Operand D, however, will still be lost.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



TB000023

FLTS

16-BIT FIXED-POINT TO 32-BIT FLOATING-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	0	1

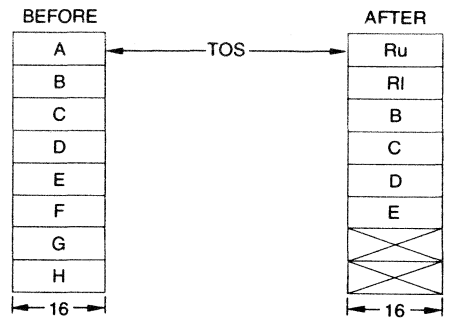
Hex Coding: 9D with sr = 1
1D with sr = 0

Execution Time: 62 to 156 clock cycles

Description:
16-bit fixed-point two's complement integer A at the TOS is converted to a 32-bit floating-point number. The lower half of the result R (Rl) replaces A; the upper half (Ru) replaces H, and the stack is moved down so that Ru occupies the TOS. Operands A, F, G and H are lost. Operands B, C, D and E are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



TB000024

FLTD

32-BIT FIXED-POINT TO 32-BIT FLOATING-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	0	0

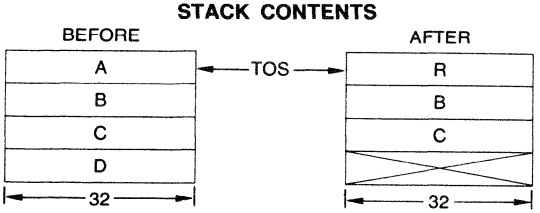
Hex Coding: 9C with sr = 1
1C with sr = 0

Execution Time: 56 to 342 clock cycles

Description:
32-bit fixed-point two's complement integer operand A at the TOS is converted to a 32-bit floating-point number. The result R replaces A at the TOS. Operands A and D are lost. Operands B and C are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



TB00025M

FMUL

32-BIT FLOATING-POINT MULTIPLY

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	1	0

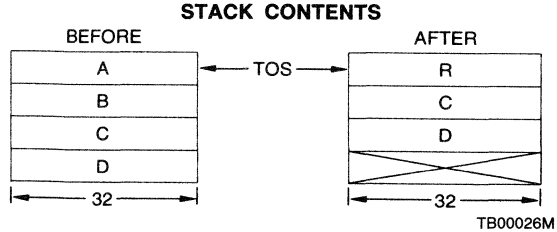
Hex Coding: 92 with sr = 1
12 with sr = 0

Execution Time: 146 to 168 clock cycles

Description:
32-bit floating-point operand A at the TOS is multiplied by the 32-bit floating-point operand B at the NOS. The normalized result R replaces B, and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.
Exponent overflow or underflow is reported in the status register, in which case the mantissa portion of the result is correct and the exponent portion is offset by 128.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB00026M

FSUB

32-BIT FLOATING-POINT SUBTRACTION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	0	1

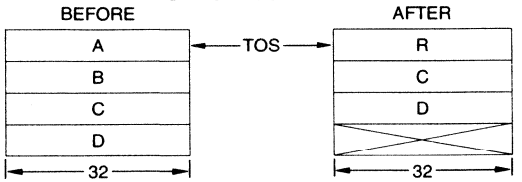
Hex Coding: 91 with sr = 1
11 with sr = 0

Execution Time: 70 to 370 clock cycles for A ≠ 0
26 clock cycles for A = 0

Description:
32-bit floating-point operand A at the TOS is subtracted from 32-bit floating-point operand B at the NOS. The normalized difference R replaces B and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.
Exponent alignment before the subtraction and normalization of the result account for the variation in execution time.
Exponent overflow or underflow is reported in the status register, in which case the mantissa portion of the result is correct and the exponent portion is offset by 128.

Status Affected: Sign, Zero, Error Field (overflow)

STACK CONTENTS



TB000027

LOG

32-BIT FLOATING-POINT COMMON LOGARITHM

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	0	0

Hex Coding: 88 with sr = 1
08 with sr = 0

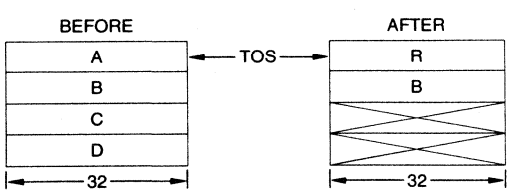
Execution Time: 4474 to 7132 clock cycles for A > 0
20 clock cycles for A ≤ 0

Description:
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point common logarithm (base 10) of A. Operands A, C and D are lost. Operand B is unchanged.
The LOG function accepts any positive input data value that can be represented by the data format. If LOG of a non-positive value is attempted, an error status of 0100 is returned.

Accuracy: LOG exhibits a maximum absolute error of 2.0×10^{-7} for the input range from 0.1 to 10, and a maximum relative error of 2.0×10^{-7} for positive values less than 0.1 or greater than 10.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000029

LN

32-BIT FLOATING-POINT NATURAL LOGARITHM

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	0	1

Hex Coding: 89 with sr = 1
09 with sr = 0

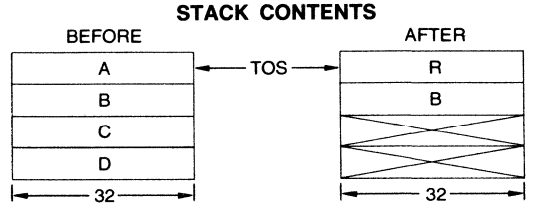
Execution Time: 4298 to 6956 clock cycles for A > 0
20 clock cycles for A ≤ 0

Description:
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point natural logarithm (base e) of A. Operands A, C and D are lost. Operand B is unchanged.
The LN function accepts all positive input data value that can be represented by the data format. If LN of a non-positive value is attempted, an error status of 0100 is returned.

Accuracy: LN exhibits a maximum absolute error of 2×10^{-7} for the input range from e^{-1} to e, and a maximum relative error of 2.0×10^{-7} for positive values less than e^{-1} or greater than e.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000028

NOP

NO OPERATION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	0	0

Hex Coding: 80 with sr = 1
00 with sr = 0

Execution Time: 4 clock cycles

Description:
The NOP command performs no internal data manipulations. It may be used to set or clear the service request interface line without changing the contents of the stack.

Status Affected: The status byte is cleared to all zeroes.

POPD

32-BIT STACK POP

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	1	0	0	0

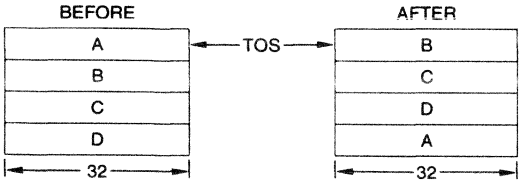
Hex Coding: B8 with sr = 1
38 with sr = 0

Execution Time: 12 clock cycles

Description:
The 32-bit stack is moved up so that the old NOS becomes the new TOS. The previous TOS rotates to the bottom of the stack. All operand values are unchanged. POPD and POPF execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



TB000082

POPS

16-BIT STACK POP

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	1	0	0	0

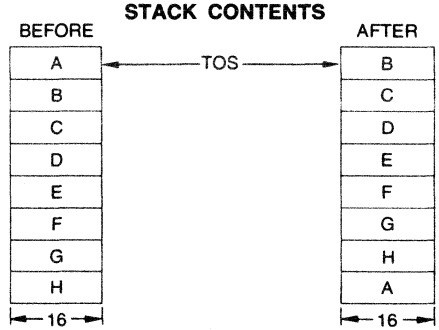
Hex Coding: F8 with sr = 1
78 with sr = 0

Execution Time: 10 clock cycles

Description:
The 16-bit stack is moved up so that the old NOS becomes the new TOS. The previous TOS rotates to the bottom of the stack. All operand values are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



TB000052

POPF

32-BIT STACK POP

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	0	0	0

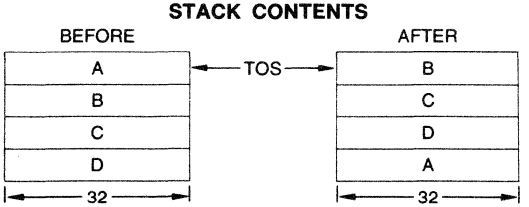
Hex Coding: 98 with sr = 1
18 with sr = 0

Execution Time: 12 clock cycles

Description:
The 32-bit stack is moved up so that the old NOS becomes the new TOS. The previous TOS rotates to the bottom of the stack. All operand values are unchanged. POPF and POPD execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



TB000053

PTOD

PUSH 32-BIT TOS ONTO STACK

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	0	1	1	1

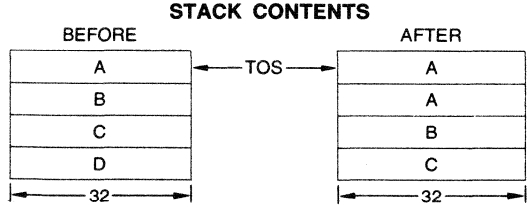
Hex Coding: B7 with sr = 1
37 with sr = 0

Execution Time: 20 clock cycles

Description:
The 32-bit stack is moved down, and the previous TOS is copied into the new TOS location. Operand D is lost. All other operand values are unchanged. PTOD and PTOF execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



TB000054

PTOF

PUSH 32-BIT TOS ONTO STACK

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	1	1	1

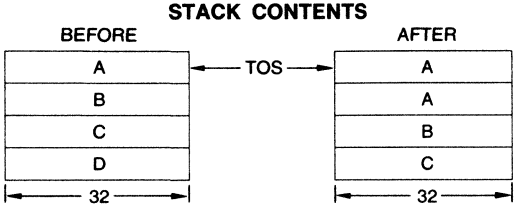
Hex Coding: 97 with sr = 1
17 with sr = 0

Execution Time: 20 clock cycles

Description:
The 32-bit stack is moved down, and the previous TOS is copied into the new TOS location. Operand D is lost. All other operand values are unchanged. PTOF and PTOD execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



TB000055

PUPI

PUSH 32-BIT FLOATING-POINT π

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	0	1	0

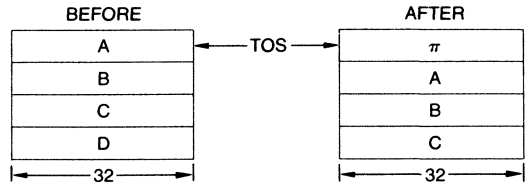
Hex Coding: 9A with sr = 1
1A with sr = 0

Execution Time: 16 clock cycles

Description:
The 32-bit stack is moved down so that the previous TOS occupies the new NOS location. 32-bit floating-point constant π is entered into the new TOS location. Operand D is lost. Operands A, B and C are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



TB000056

PTOS

PUSH 16-BIT TOS ONTO STACK

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	0	1	1	1

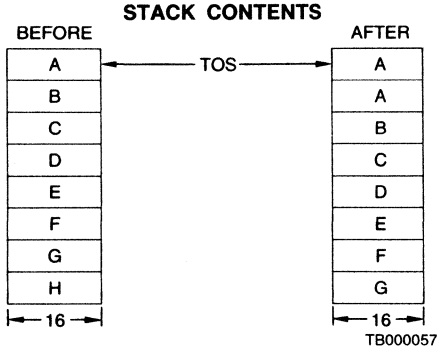
Hex Coding: F7 with sr = 1
77 with sr = 0

Execution Time: 16 clock cycles

Description:
The 16-bit stack is moved down, and the previous TOS is copied into the new TOS location. Operand H is lost, and all other operand values are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



TB000057

PWR

32-BIT FLOATING-POINT X^Y

Binary Coding:

7	6	5	4	3	2	1	0
sr	0	0	0	1	0	1	1

Hex Coding: 8B with sr = 1
0B with sr = 0

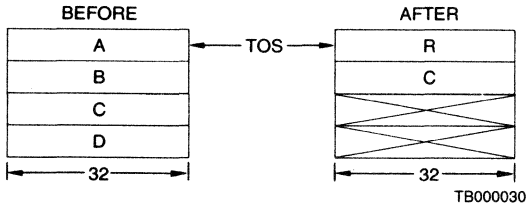
Execution Time: 8290 to 12032 clock cycles

Description:
32-bit floating-point operand B at the NOS is raised to the power specified by the 32-bit floating-point operand A at the TOS. The result R of B^A replaces B, and the stack is moved up so that R occupies the TOS. Operands A, B, and D are lost. Operand C is unchanged.
The PWR function accepts all input data values that can be represented in the data format for operand A and all positive values for operand B. If operand B is non-positive, an error status of 0100 will be returned. The EXP and LN functions are used to implement PWR using the relationship $B^A = EXP\,[A(LN\,B)]$. Thus, if the term [A(LN B)] is outside the range of $-1.0 \times 2^{+5}$, to $+1.0 \times 2^{+5}$, an error status of 1100 will be returned. Underflow and overflow conditions can occur.

Accuracy: The error performance for PWR is a function of the LN and EXP performance as expressed by:
$|(\text{Related Error})_{PWR}| = |(\text{Relative Error})_{EXP} + |A(\text{Absolute Error})_{LN}|$
The maximum relative error for PWR occurs when A is at its maximum value while [A(LN B)] is near 1.0×2^5 and the EXP error is also at its maximum. For most practical applications, the relative error for PWR will be less than 7.0×10^{-7}.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000030

SADD

16-BIT FIXED-POINT ADD

Binary Coding:

7	6	5	4	3	2	1	0
sr	1	1	0	1	1	0	0

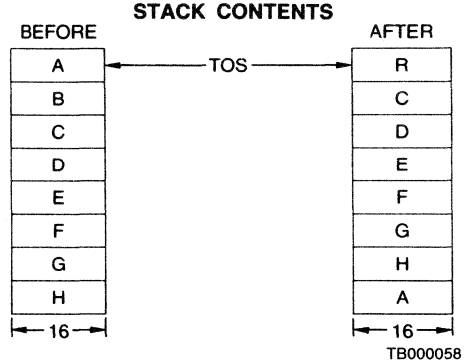
Hex Coding: EC with sr = 1
6C with sr = 0

Execution Time: 16 to 18 clock cycles

Description:
16-bit fixed-point two's complement integer operand A at the TOS is added to 16-bit fixed-point two's complement integer operand B at the NOS. The result R replaces B, and the stack is moved up so that R occupies the TOS. Operand B is lost. All other operands are unchanged.
If the addition generates a carry bit, it is reported in the status register. If an overflow occurs, it is reported in the status register, and the 16 least significant bits of the result are returned.

Status Affected: Sign, Zero, Carry, Error Field

STACK CONTENTS



TB000058

SDIV

16-BIT FIXED-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	1	1

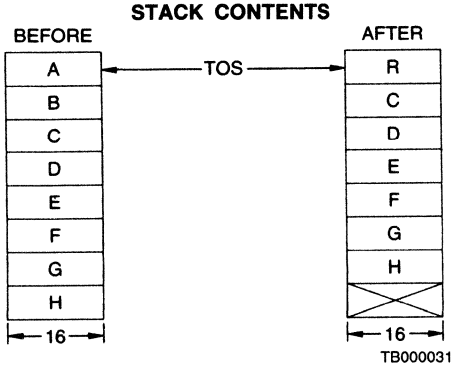
Hex Coding: EF with sr = 1
6F with sr = 0

Execution Time: 84 to 94 clock cycles for $A \neq 0$
14 clock cycles for $A = 0$

Description:
16-bit fixed-point two's complement integer operand B at the NOS is divided by 16-bit fixed-point two's complement integer operand A at the TOS. The 16-bit quotient R replaces B, and the stack is moved up so that R occupies the TOS. No remainder is generated. Operands A and B are lost. All other operands are unchanged.
If A is zero, R will be set equal to B and the divide-by-zero error status will be reported.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000031

SIN

32-BIT FLOATING-POINT SINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	1	0

Hex Coding: 82 with sr = 1
02 with sr = 0

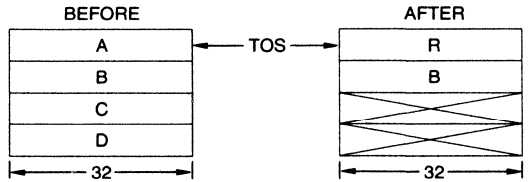
Execution Time: 3796 to 4808 clock cycles for $|A| > 2^{-12}$ radians
30 clock cycles for $|A| \leq 2^{-12}$ radians

Description:
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point sine of A. A is assumed to be in radians. Operands A, C and D are lost. Operand B is unchanged.
The SIN function will accept any input data value that can be represented by the data format. All input values are range reduced to fall within the interval $-\pi/2$ to $+\pi/2$ radians.

Accuracy: SIN exhibits a maximum relative error of 5.0×10^{-7} for input values in the range of -2π to $+2\pi$ radians.

Status Affected: Sign, Zero

STACK CONTENTS



TB000032

SMUL

16-BIT FIXED-POINT MULTIPLY, LOWER

Binary Coding:

7	6	5	4	3	2	1	0
sr	1	1	0	1	1	1	0

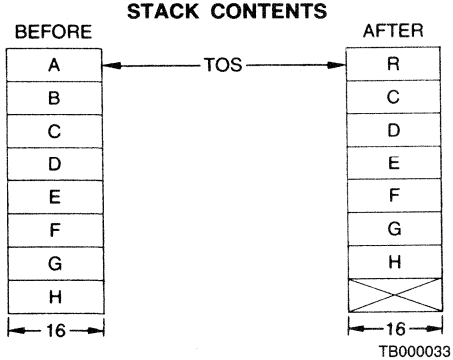
Hex Coding: EE with sr = 1
6E with sr = 0

Execution Time: 84 to 94 clock cycles

Description:
16-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 16-bit fixed-point two's complement integer operand B at the NOS. The 16-bit least significant half of the product R replaces B, and the stack is moved up so that R occupies the TOS. The most significant half of the product is lost. Operands A and B are lost. All other operands are unchanged. The overflow status bit is set if the discarded upper half was non-zero. If either A or B is the most negative value that can be represented in the format, that value is returned as R and the overflow status is set.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



SMUU

16-BIT FIXED-POINT MULTIPLY, UPPER

Binary Coding:

7	6	5	4	3	2	1	0
sr	1	1	1	0	1	1	0

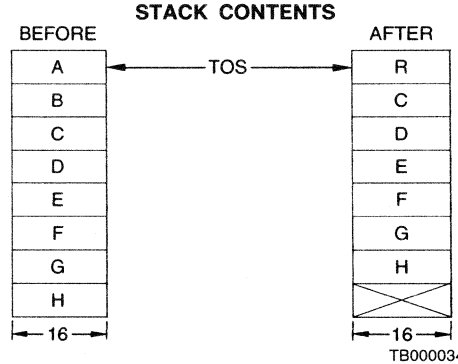
Hex Coding: F6 with sr = 1
76 with sr = 0

Execution Time: 80 to 98 clock cycles

Description:
16-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 16-bit fixed-point two's complement integer operand B at the NOS. The 16-bit most significant half of the product R replaces B, and the stack is moved up so that R occupies the TOS. The least significant half of the product is lost. Operands A and B are lost. All other operands are unchanged.
If either A or B is the most negative value that can be represented in the format, that value is returned as R and the overflow status is set.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



SQRT

32-BIT FLOATING-POINT SQUARE ROOT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	0	1

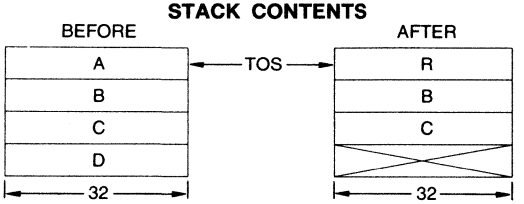
Hex Coding: 81 with sr = 1
01 with sr = 0

Execution Time: 782 to 870 clock cycles

Description:
32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point square root of A. Operands A and D are lost. Operands B and C are unchanged.
SQRT will accept any non-negative input data that can be represented by the data format. If A is negative an error code of 0100 will be returned in the status register.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000035

SSUB

16-BIT FIXED-POINT SUBTRACT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	0	1

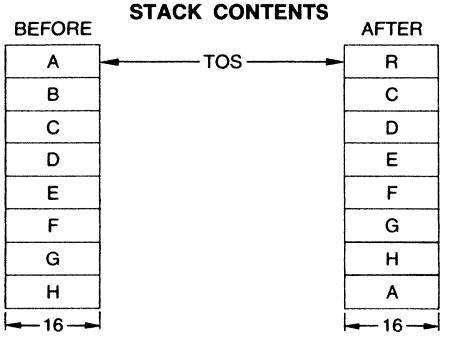
Hex Coding: ED with sr = 1
6D with sr = 0

Execution Time: 30 to 32 clock cycles

Description:
16-bit fixed-point two's complement integer operand A at the TOS is subtracted from 16-bit fixed-point two's complement integer operand B at the NOS. The result R replaces B, and the stack is moved up so that R occupies the TOS. Operand B is lost. All other operands are unchanged.
If the subtraction generates a borrow, it is reported in the carry status bit. If A is the most negative value that can be represented in the format, the overflow status is set. If the result cannot be represented in the format range, the overflow status is set and the 16 least significant bits of the result are returned as R.

Status Affected: Sign, Zero, Carry, Error Field

STACK CONTENTS



TB000059

TAN

32-BIT FLOATING-POINT TANGENT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	0	0

Hex Coding: 84 with sr = 1
04 with sr = 0

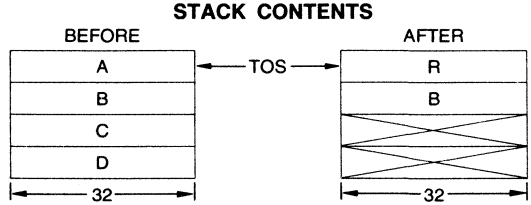
Execution Time: 4894 to 5886 clock cycles for $|A| > 2^{-12}$ radians
30 clock cycle for $|A| \leq 2^{-12}$ radians

Description:
The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point tangent of A. Operand A is assumed to be in radians. A, C and D are lost. B is unchanged.
The TAN function will accept any input data value that can be represented in the data format. All input data values are range-reduced to fall within $-\pi/4$ to $+\pi/4$ radians. TAN is unbounded for input values near odd multiples of $\pi/2$, and in such cases, the overflow bit is set in the status register. For angles smaller than 2^{-12} radians, TAN returns A as the tangent of A.

Accuracy: TAN exhibits a maximum relative error of 5.0×10^{-7} for input data values in the range of -2π to $+2\pi$ radians except for data values near odd multiples of $\pi/2$.

Status Affected: Sign, Zero, Error Field (overflow)

STACK CONTENTS



TB000036

XCHD

EXCHANGE 32-BIT STACK OPERANDS

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	1	0	0	1

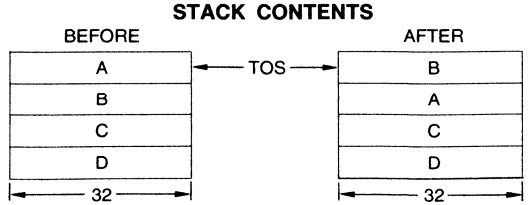
Hex Coding: B9 with sr = 1
39 with sr = 0

Execution Time: 26 clock cycles

Description:
32-bit operand A at the TOS and 32-bit operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All operands are unchanged. XCHD and XCHF execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



TB000060

XCHF

EXCHANGE 32-BIT STACK OPERANDS

Binary Coding:

7	6	5	4	3	2	1	0
sr	0	0	1	1	0	0	1

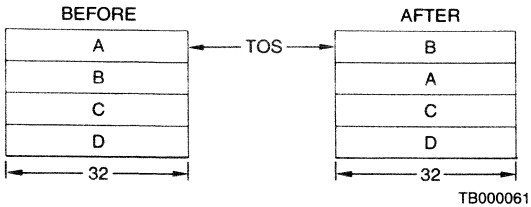
Hex Coding: 99 with sr = 1
19 with sr = 0

Execution Time: 26 clock cycles

Description:
32-bit operand A at all TOS and 32-bit operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All operands are unchanged. XCHD and XCHF execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



TB000061

XCHS

EXCHANGE 16-BIT STACK OPERANDS

Binary Coding:

7	6	5	4	3	2	1	0
sr	1	1	1	1	0	0	1

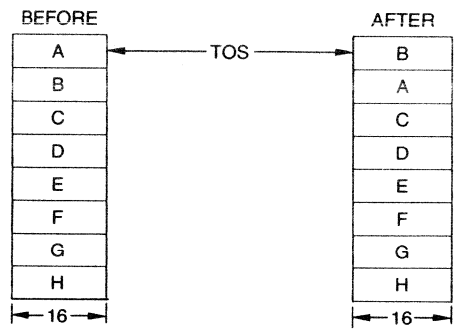
Hex Coding: F9 with sr = 1
79 with sr = 0

Execution Time: 18 clock cycles

Description:
16-bit operand A at the TOS and 16-bit operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All operands are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



TB000062

APPLICATIONS INFORMATION

The diagram in Figure 2 shows the interface connections for the Am9511A/8231A APU with operand transfers handled by an Am9517A DMA controller and CPU coordination handled by an Am9519A 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 1. The Am9511A/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.

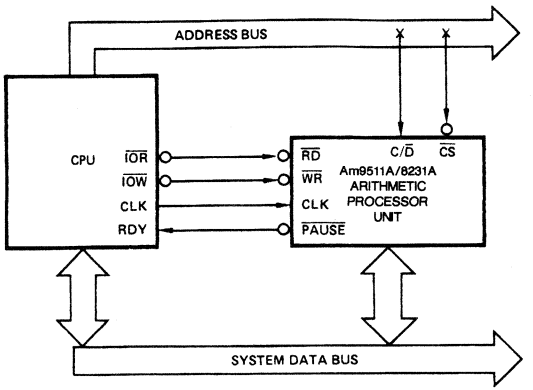


Figure 1. Am9511A/8231A Minimum Configuration Example.

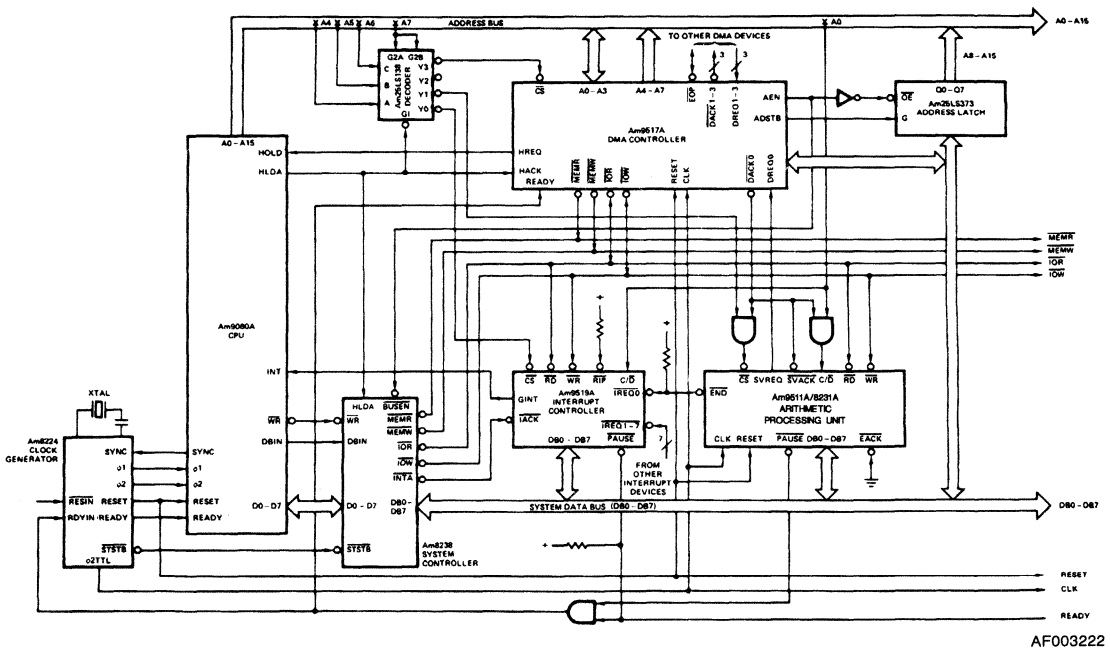


Figure 2. Am9511A/8231A High Performance Configuration Example.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
VDD with Respect to VSS ..... −0.5V to +15.0V
VCC with Respect to VSS ..... −0.5V to +7.0V
All Signal Voltages
with Respect to VSS ..... −5.0V to +7.0V
Power Dissipation (Package Limitation) 2.0W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5V ±5%
(V_{DD}) ..... 12 ±5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5V ±10%
(V_{DD}) ..... 12 ±10%

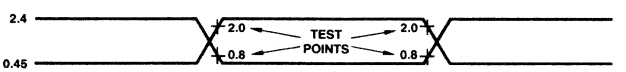
Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range (Note 1)

Parameters	Description	Test Conditions	Min	Typ	Max	Units
VOH	Output HIGH Voltage	IOH = −200μA	3.7			Volts
VOL	Output LOW Voltage	IOL = 3.2mA			0.4	Volts
VIH	Input HIGH Voltage		2.0		VCC	Volts
VIL	Input LOW Voltage		−0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VI ≤ VCC			±10	μA
IOZ	Data Bus Leakage	VO = 0.4V			10	μA
		VO = VCC			10	
ICC	VCC Supply Current	T_A = +25°C		50	90	mA
		T_A = 0°C			95	
IDD	VDD Supply Current	T_A = +25°C		50	90	mA
		T_A = 0°C			95	
CO	Output Capacitance	fc = 1.0MHz, Inputs = 0V		8	10	pF
CI	Input Capacitance			5	8	pF
CIO	I/O Capacitance			10	12	pF

Note: 1. Typical values are for T_A = 25°C, normal supply voltages and normal processing parameters.

SWITCHING TEST INPUT WAVEFORM



WF004060

See Section 6 for Thermal Characteristics Information.

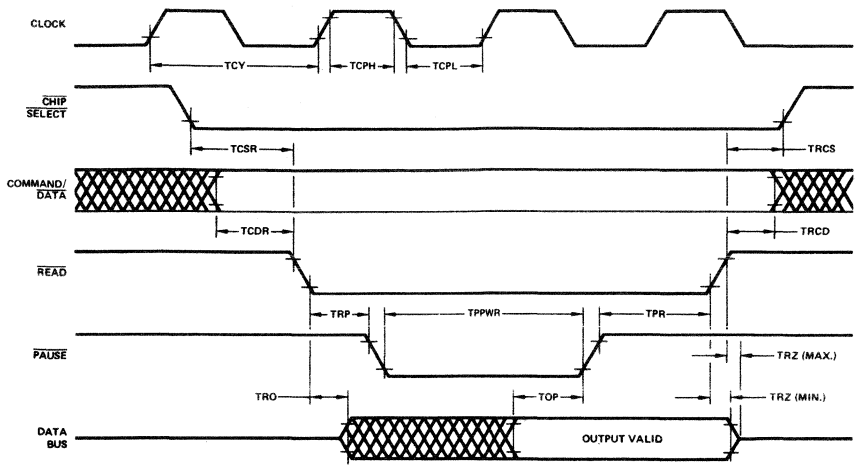
SWITCHING CHARACTERISTICS over operating range (Note 1)

Parameters	Description		Am9511A/8231A Min	Am9511A/8231A Max	Am9511A-1/8231A Min	Am9511A-1/8231A Max	Am9511A-4/8231A Min	Am9511A-4/8231A Max	Units
TAPW	$\overline{EACK}$ LOW Pulse Width		100		75		50		ns
TCDR	C/$\overline{D}$ to $\overline{RD}$ LOW Set-up Time		0		0		0		ns
TCDW	C/$\overline{D}$ to $\overline{WR}$ LOW Set-up Time		0		0		0		ns
TCPH	Clock Pulse HIGH Width		200		140		100		ns
TCPL	Clock Pulse LOW Width		240		160		120		ns
TCSR	$\overline{CS}$ LOW to $\overline{RD}$ LOW Set-up Time		0		0		0		ns
TCSW	$\overline{CS}$ LOW to $\overline{WR}$ LOW Set-up Time		0		0		0		ns
TCY	Clock Period		480	5000	320	3300	250	2500	ns
TDW	Data Bus Stable to $\overline{WR}$ HIGH Set-up Time		150		100 (Note 9)		100		ns
TEAE	$\overline{EACK}$ LOW to $\overline{END}$ HIGH Delay			200		175		150	ns
TEPW	$\overline{END}$ LOW Pulse Width (Note 4)		400		270		200		ns
TOP	Data Bus Output Valid to $\overline{PAUSE}$ HIGH Delay		0		0		0		ns
TPPWR	$\overline{PAUSE}$ LOW Pulse Width Read (Note 5)	Data	3.5TCY + 50	5.5TCY + 300	3.5TCY + 50	5.5TCY + 200	3.5TCY + 50	5.5TCY + 200	ns
		Status	1.5TCY + 50	3.5TCY + 300	1.5TCY + 50	3.5TCY + 200	1.5TCY + 50	3.5TCY + 200	
TPPWW	$\overline{PAUSE}$ LOW Pulse Width Write (Note 8)			50		50		50	ns
TPR	$\overline{PAUSE}$ HIGH to $\overline{RD}$ HIGH Hold Time		0		0		0		ns
TPW	$\overline{PAUSE}$ HIGH to $\overline{WR}$ HIGH Hold Time		0		0		0		ns
TRCD	$\overline{RD}$ HIGH to C/$\overline{D}$ Hold Time		0		0		0		ns
TRCS	$\overline{RD}$ HIGH to $\overline{CS}$ HIGH Hold Time		0		0		0		ns
TRO	$\overline{RD}$ LOW to Data Bus ON Delay		50		50		50		ns
TRP	$\overline{RD}$ LOW to $\overline{PAUSE}$ LOW Delay (Note 6)			150		100 (Note 9)		100	ns
TRZ	$\overline{RD}$ HIGH to Data Bus OFF Delay		50	200	50	150	50	120	ns
TSAPW	$\overline{SVACK}$ LOW Pulse Width		100		75		50		ns
TSAR	$\overline{SVACK}$ LOW to SVREQ LOW Delay			300		200		150	ns
TWCD	$\overline{WR}$ HIGH to C/$\overline{D}$ Hold Time		60		30		30		ns
TWCS	$\overline{WR}$ HIGH to $\overline{CS}$ HIGH Hold Time		60		30		30		ns
TWD	$\overline{WR}$ HIGH to Data Bus Hold Time		20		20		20		ns
TWI	Write Inactive Time	Command	4TCY		4TCY		4TCY		ns
		Data	5TCY		5TCY		5TCY		
TWP	$\overline{WR}$ LOW to $\overline{PAUSE}$ LOW Delay (Note 6)			150		100 (Note 9)		100	ns

Notes:
1. Typical values are for $T_A = 25°C$, nominal supply voltages and nominal processing parameters.
2. Switching parameters are listed in alphabetical order.
3. Test conditions assume transition times of 20ns or less, output loading of one TTL gate plus 100pF ±20pF and timing reference levels of 0.8V and 2.0V.
4. $\overline{END}$ low pulse width is specified for $\overline{EACK}$ tied to VSS. Otherwise TEAE applies.
5. Minimum values shown assume no previously entered command is being executed for the data access. If a previously entered command is being executed, $\overline{PAUSE}$ 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.
6. $\overline{PAUSE}$ is pulled low for both command and data operations.
7. TEX is the execution time of the current command (see the Command Execution Times table).
8. $\overline{PAUSE}$ low pulse width is less than 50ns when writing into the data port or the control port as long as the duty requirement (TWI) is observed and no previous command is being executed. TWI may be safely violated up to 500ns as long as the extended TPPWW that results is observed. If a previously entered command is being executed, $\overline{PAUSE}$ LOW Pulse Width is the time to complete execution plus the time shown.
9. 150ns for Military grade.

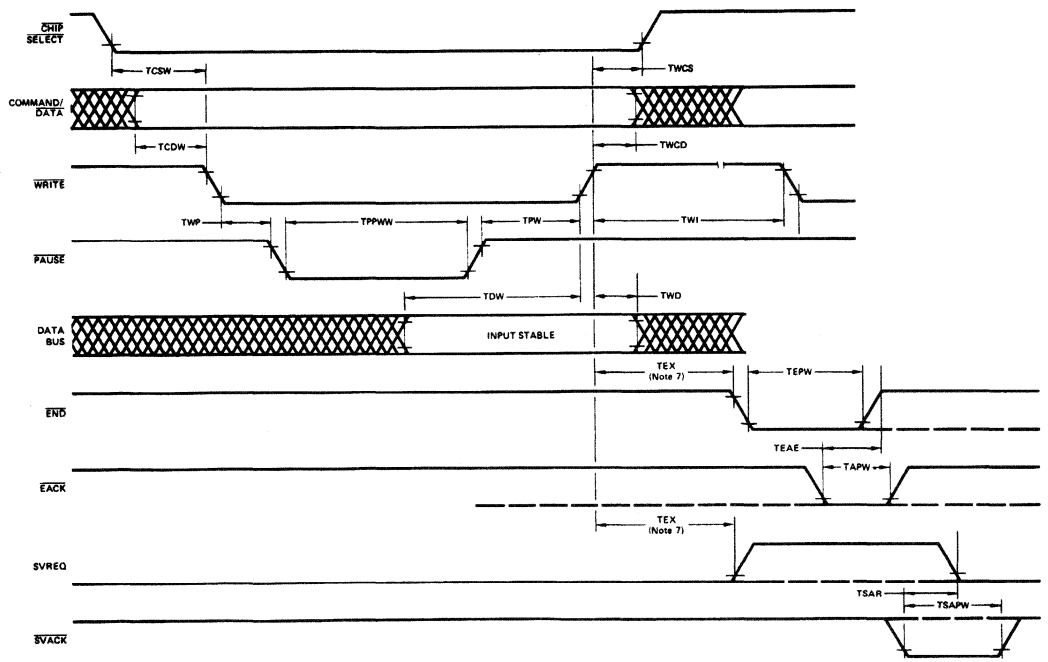
SWITCHING WAVEFORMS

READ OPERATIONS



WF004070

WRITE OPERATIONS



WF004082

Am9513A

System Timing Controller

DISTINCTIVE CHARACTERISTICS

- Five independent 16-bit counters
- High speed counting rates
- Up/down and binary/BCD counting
- Internal oscillator frequency source
- Tapped frequency scaler
- Programmable frequency output
- 8-bit or 16-bit bus interface
- Time-of-day option
- Alarm comparators on counters 1 and 2
- Complex duty cycle outputs
- One-shot or continuous outputs
- Programmable count/gate source selection
- Programmable input and output polarities
- Programmable gating functions
- Retriggering capability
- +5 volt power supply
- Standard 40-pin package

GENERAL DESCRIPTION

The Am9513A System Timing Controller is an LSI circuit designed to service many types of counting, sequencing and timing applications. It provides the capability for programmable frequency synthesis, high resolution programmable duty cycle waveforms, retriggerable digital one-shots, time-of-day clocking, coincidence alarms, complex pulse generation, high resolution baud rate generation, frequency shift keying, stop-watching timing, event count accumulation, waveform analysis, etc. A variety of programmable operating modes and control features allow the Am9513A to be personalized for particular applications as well as dynamically reconfigured under program control.

The STC includes five general-purpose 16-bit counters. A variety of internal frequency sources and external pins may be selected as inputs for individual counters with software selectable active-high or active-low input polarity. Both hardware and software gating of each counter is available. Three-state outputs for each counter provide pulses or levels and can be active-high or active-low. The counters can be programmed to count up or down in either binary or BCD. The host processor may read an accumulated count at any time without disturbing the counting process. Any of the counters may be internally concatenated to form any effective counter length up to 80 bits.

BLOCK DIAGRAM

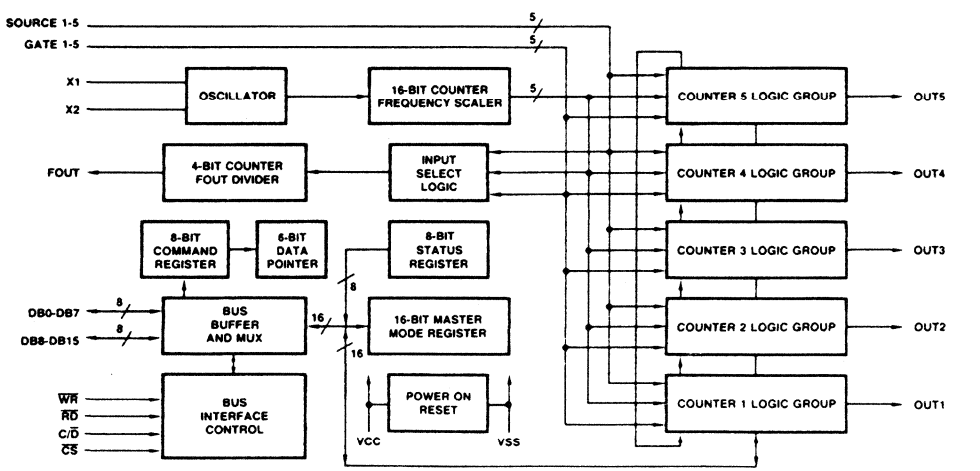
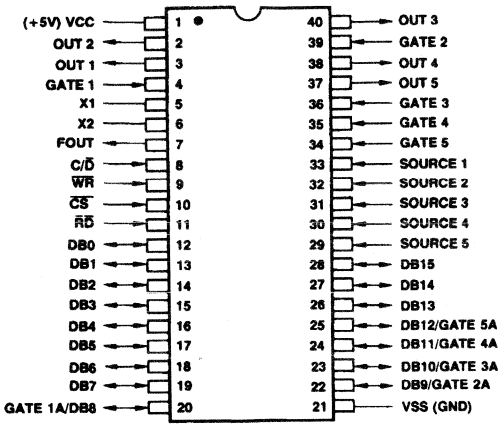


Figure 1-1.

Publication # 01731 | Rev. E | Amendment /0
Issue Date: April 1987

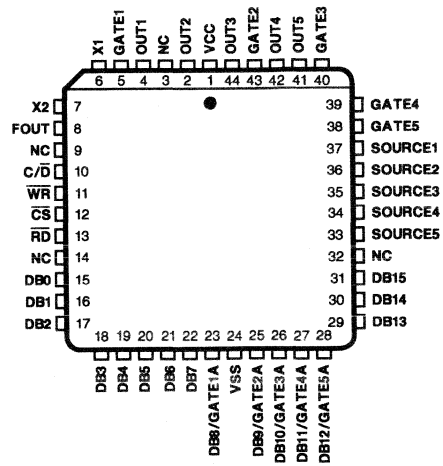
CONNECTION DIAGRAMS
Top View

DIPs



CD005212

PLCC



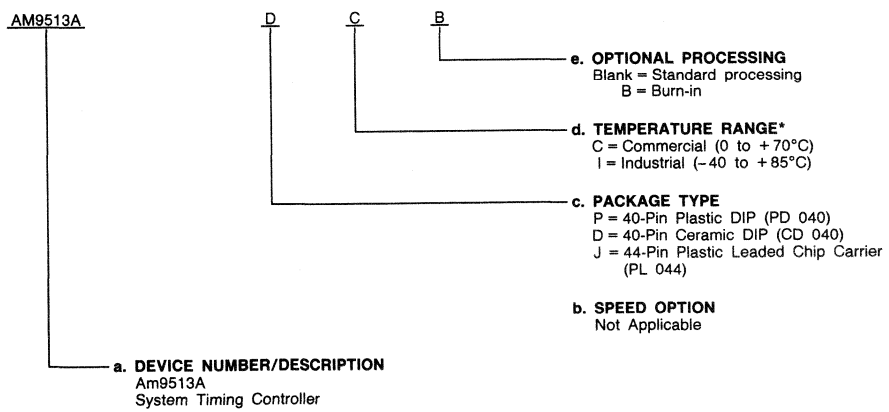
CD009912

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM9513A	PC, DC, DCB, DIB, JC

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order # 09275A/0) for electrical performance characteristics.

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
1	V_{CC}		+5V Power Supply.
21	V_{SS}		Ground.
5, 6	X1, X2	O, I	(Crystal). X1 and X2 are the connections for an external crystal used to determine the frequency of the internal oscillator. The crystal should be a parallel-resonant, fundamental-mode type. An RC or LC or other reactive network may be used instead of a crystal. For driving from an external frequency source, X1 should be left open and X2 should be connected to a TTL source and a pull-up resistor.
7	FOUT	O	(Frequency Out). The FOUT output is derived from a 4-bit counter that may be programmed to divide its input by any integer value from 1 through 16 inclusive. The input to the counter is selected from any of 15 sources, including the internal scaled oscillator frequencies. FOUT may be gated on and off under software control and when off will exhibit a low impedance to ground. Control over the various FOUT options resides in the Master Mode register. After power-up, FOUT provides a frequency that is 1/16 that of the oscillator. The input source on power-up is F1.
4, 39, 36-34	GATE1 - GATE5	I	(Gate). The Gate inputs may be used to control the operations of individual counters by determining when counting may proceed. The same Gate input may control up to three counters. Gate pins may also be selected as count sources for any of the counters and for the FOUT divider. The active polarity for a selected Gate input is programmed at each counter. Gating function options allow level-sensitive gating or edge-initiated gating. Other gating modes are available including one that allows the Gate input to select between two counter output frequencies. All gating functions may also be disabled. The active Gate input is conditioned by an auxiliary input when the unit is operating with an external 8-bit data bus. See Data Bus description. Schmitt-trigger circuitry on the GATE inputs allows slow transition times to be used.
33-29	SRC1 - SRC5	I	(Source).The Source inputs provide external signals that may be counted by any of the counters. Any Source line may be routed to any or all of the counters and the FOUT divider. The active polarity for a selected SRC input is programmed at each counter. Any duty cycle waveform will be accepted as long as the minimum pulse width is at least half the period of the maximum specified counting frequency for the part. Schmitt-trigger circuitry on the SRC inputs allows slow transition times to be used.
3, 2, 40, 38, 37	OUT1 - OUT5	O	(Counter). Each 3-state OUT signal is directly associated with a corresponding individual counter. Depending on the counter configuration, the OUT signal may be a pulse, a square wave, or a complex duty cycle waveform. OUT pulse polarities are individually programmable. The output circuitry detects the counter state that would have been all bits zero in the absence of a reinitialization. That information is used to generate the selected waveform type. An optional output mode for Counters 1 and 2 overrides the normal output mode and provides a true OUT signal when the counter contents match the contents of an Alarm register.
12-19, 20, 22-28	DB0 - DB7, DB8 - DB15	I/O	(Data Bus). The 16 bidirectional Data Bus lines are used for information exchanges with the host processor. HIGH on a Data Bus line corresponds to one and LOW corresponds to zero. These lines act as inputs when $\overline{WR}$ and $\overline{CS}$ are active and as outputs when $\overline{RD}$ and $\overline{CS}$ are active. When $\overline{CS}$ is inactive, these pins are placed in a high-impedance state. After power-up or reset, the data bus will be configured for 8-bit width and will use only DB0 through DB7. DB0 is the least significant and DB7 is the most significant bit position. The data bus may be reconfigured for 16-bit width by changing a control bit in the Master Mode register. This is accomplished by writing an 8-bit command into the low-order DB lines while holding the DB13 - DB15 lines at a logic high level. Thereafter, all 16 lines can be used, with DB0 as the least significant and DB15 as the most significant bit position. When operating in the 8-bit data bus environment, DB8 - DB15 will never be driven active by the Am9513A. DB8 through DB12 may optionally be used as additional Gate inputs (see Figure 1-3). If unused, they should be held HIGH. When pulled LOW, a GATENA signal will disable the action of the corresponding counter N gating. DB13 - DB15 should be held HIGH in 8-bit bus mode whenever $\overline{CS}$ and $\overline{WR}$ are simultaneously active.
10	$\overline{CS}$	I	(Chip Select). The active-low Chip Select input enables Read and Write operations on the data bus. When Chip Select is HIGH, the Read and Write inputs are ignored. The first Chip Select signal after power-up is used to clear the power-on reset circuitry. If Chip Select is tied to ground permanently, the power-on reset circuitry may not function. In such a configuration, the software reset command must be issued following power-up to reset the Am9513A.
11	$\overline{RD}$	I	(Read). The active-low Read signal is conditioned by Chip Select and indicates that internal information is to be transferred to the data bus. The source will be determined by the port being addressed and, for Data Port reads, by the contents of the Data Pointer register. $\overline{WR}$ and $\overline{RD}$ should be mutually exclusive.
9	$\overline{WR}$	I	(Write). The active-low Write signal is conditioned by Chip Select and indicates that data bus information is to be transferred to an internal location. The destination will be determined by the port being addressed and, for Data Port writes, by the contents of the Data Pointer register. $\overline{WR}$ and $\overline{RD}$ should be mutually exclusive.
8	C/$\overline{D}$	I	(Control/Data). The Control/Data signal selects source and destination locations for Read and Write operations on the data bus. Control Write operations load the Command register and the Data Pointer. Control Read operations output the Status register. Data Read and Data Write transfers communicate with all other internal registers. Indirect addressing at the data port is controlled internally by the Data Pointer register.

Signal	Abbreviation	Type	Pins
+ 5 Volts	VCC	Power	1
Ground	VSS	Power	1
Crystal	X1, X2	O, I	2
Read	$\overline{RD}$	Input	1
Write	$\overline{WR}$	Input	1
Chip Select	$\overline{CS}$	Input	1
Control/Data	C/$\overline{D}$	Input	1
Source N	SRC	Input	5
Gate N	GATE	Input	5
Data Bus	DB	I/O	16
Frequency Out	FOUT	Output	1
Out N	OUT	Output	5

Figure 1-2. Interface Signal Summary

Figure 1-2 summarizes the interface signals and their abbreviations for the STC.

Package Pin	Data Bus Width (MM14)	
	16 Bits	8 Bits
12	DB0	DB0
13	DB1	DB1
14	DB2	DB2
15	DB3	DB3
16	DB4	DB4
17	DB5	DB5
18	DB6	DB6
19	DB7	DB7
20	DB8	GATE 1A
22	DB9	GATE 2A
23	DB10	GATE 3A
24	DB11	GATE 4A
25	DB12	GATE 5A
26	DB13	(VIH)
27	DB14	(VIH)
28	DB15	(VIH)

Figure 1-3. Data Bus Assignments

Interface Considerations

All of the input and output signals for the Am9513A are specified with logic levels compatible with those of standard TTL circuits. In addition to providing TTL compatible voltage levels, other output conditions are specified to help configure non-standard interface circuitry. The logic level specifications take into account all worst-case combinations of the three variables that affect the logic level thresholds: ambient temperature, supply voltage and processing parameters. A change in any of these toward nominal values will improve the actual operating margins and will increase noise immunity.

Unprotected open gate inputs of high quality MOS transistors exhibit very high resistances on the order of perhaps 10^{14} ohms. It is easy, therefore, in some circumstances, for charge to enter the gate node of such an input faster than it can be discharged and consequently, for the gate voltage to rise high enough to break down the oxides and destroy the transistor.

All inputs to the Am9513A include protection networks to help prevent damaging accumulations of static charge. The protection circuitry is designed to slow the transitions of incoming current surges and to provide low-impedance discharge paths for voltages beyond the normal operating levels. Note, however, that input energy levels can nonetheless be too high to be successfully absorbed. Conventional design, storage, and handling precautions should be observed so that the protection networks themselves are not overstressed.

Within the limits of normal operation, the input protection circuitry is inactive and may be modeled as a lumped series RC as shown in Figure 1-4(a). The functionally active input connection during normal operation is the gate of a MOS transistor. No active sources or drains are connected to the inputs so that neither transient nor steady-state currents are impressed on the driving signals other than the charging or discharging of the input capacitance and the accumulated leakage associated with the protection network and the input circuit.

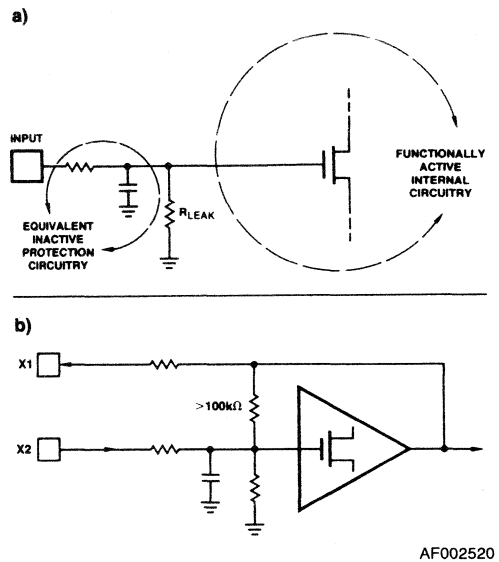


Figure 1-4. Input Circuitry

The only exception to the purely capacitive input case is the X2 crystal input. As shown in Figure 1-4(b) an internal resistor connects X1 and X2 in addition to the protection network. The resistor is a modestly high value of more than 100kohms.

Fanout from the driving circuitry into the Am9513A inputs will generally be limited by transition time considerations rather than DC current limitations when the loading is dominated by conventional MOS circuits. In an operating environment, all inputs should be terminated so they do not float and therefore will not accumulate stray static charges. Unused inputs should be tied directly to Ground or VCC, as appropriate. An input in use will have some type of logic output driving it, and termination during operation will not be a problem. Where inputs are driven from logic external to the card containing this chip, however, on-board termination should be provided to protect the chip when the board is unplugged (the input would otherwise float). A pull-up resistor or a simple inverter or gate will suffice.

DETAILED DESCRIPTION

The Am9513A System Timing Controller (STC) is a support device for processor oriented systems that is designed to enhance the available capability with respect to counting and timing operations. It provides the capability for programmable frequency synthesis, high resolution programmable duty cycle waveforms, retriggerable digital timing functions, time-of-day clocking, coincidence alarms, complex pulse generation, high resolution baud rate generation, frequency shift keying, stop-watching timing, event count accumulation, waveform analysis and many more. A variety of programmable operating modes and control features allow the Am9513A to be personalized for particular applications as well as dynamically reconfigured under program control.

The STC includes five general-purpose 16-bit counters. A variety of internal frequency sources and external pins may be selected as inputs for individual counters with software selectable active-high or active-low input polarity. Both hardware and software gating of each counter is available. Three-state outputs for each counter provide either pulses or levels. The counters can be programmed to count up or down in either binary or BCD. The accumulated count may be read without disturbing the counting process. Any of the counters may be internally concatenated to form an effective counter length of up to 80 bits.

The Am9513A block diagrams indicate the interface signals and the basic flow of information. Internal control lines and the internal data bus have been omitted. The control and data registers are all connected to a common internal 16-bit bus. The external bus may be 8- or 16-bits wide; in the 8-bit mode, the internal 16-bit information is multiplexed to the low order data bus pins DB0 through DB7.

An internal oscillator provides a convenient source of frequencies for use as counter inputs. The oscillator's frequency is controlled at the X1 and X2 interface pins by an external reactive network such as a crystal. The oscillator output is divided by the Frequency Scaler to provide several sub-frequencies. One of the scaled frequencies (or one of ten input signals) may be selected as an input to the FOUT divider and then comes out of the chip at the FOUT interface pin.

The STC is addressed by the external system as two locations: a Control port and a Data port. The Control port provides direct access to the Status and Command registers, as well as allowing the user to update the Data Pointer register. The Data port is used to communicate with all other addressable internal locations. The Data Pointer register controls the Data port addressing.

Among the registers accessible through the Data port are the Master Mode register and five Counter Mode registers, one for each counter. The Master Mode register controls the programmable options that are not controlled by the Counter Mode registers.

Each of the five general-purpose counters is 16-bits long and is independently controlled by its Counter Mode register. Through this register, a user can software select one of 16 sources as the counter input, a variety of gating and repetition modes, up or down counting in binary or BCD and active-high or active-low input and output polarities.

Associated with each counter are a Load register and a Hold register, both accessible through the Data port. The Load register is used to automatically reload the counter to any predefined value, thus controlling the effective count period. The Hold register is used to save count values without disturbing the count process, permitting the host processor to read intermediate counts. In addition, the Hold register may be used as a second Load register to generate a number of complex output waveforms.

All five counters have the same basic control logic and control registers. Counters 1 and 2 have additional Alarm registers and comparators associated with them, plus the extra logic necessary for operating in a 24-hour time-of-day mode. For real-time operation, the time-of-day logic will accept 50Hz, 60Hz or 100Hz input frequencies.

Each general counter has a single dedicated output pin. It may be turned off when the output is not of interest or may be configured in a variety of ways to drive interrupt controllers, Darlington buffers, bus drivers, etc. The counter inputs, on the other hand, are specifically not dedicated to any given interface line. Considerable versatility is available for configuring both the input and the gating of individual counters. This not only permits dynamic reassignment of inputs under software control, but also allows multiple counters to use a single input and a single gate pin to control more than one counter. Indeed, a single pin can be the gate for one counter and, at the same time, the count source for another.

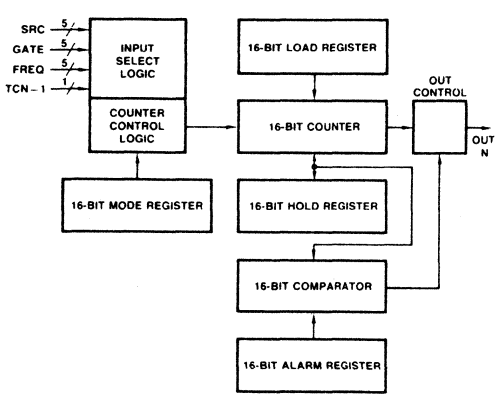


Figure 1-5. Counter Logic Groups 1 and 2

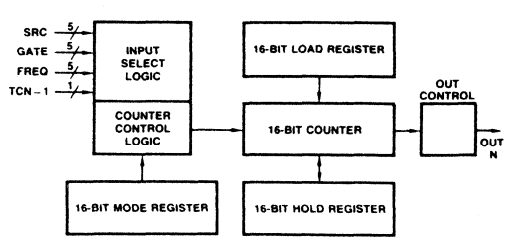


Figure 1-6. Counter Logic Groups 3, 4 and 5

A powerful command structure simplifies user interaction with the counters. A counter must be armed by one of the ARM commands before counting can commence. Once armed, the counting process may be further enabled or disabled using the hardware gating facilities. The ARM and DISARM commands permit software gating of the count process in some modes.

The LOAD command causes the counter to be reloaded with the value in either the associated Load register or the associated Hold register. It will often be used as a software retrigger or as counter initialization prior to active hardware gating.

The DISARM command disables further counting independent of any hardware gating. A disarmed counter may be reloaded using the LOAD command, may be incremented or decremented using the STEP command and may be read using the SAVE command. A count process may be resumed using an ARM command.

The SAVE command transfers the contents of a counter to its associated Hold register. This command will overwrite any previous Hold register contents. The SAVE command is designed to allow an accumulated count to be preserved so that it can be read by the host CPU at some later time.

Two combinations of the basic commands exist to either LOAD AND ARM or to DISARM AND SAVE any combination of counters. Additional commands are provided to: step an individual counter by one count; set and clear an output toggle; issue a software reset; clear and set special bits in the Master Mode register; and load the Data Pointer register.

Note: Separate LOAD and ARM commands should be used for asynchronous operations.

Power Supply

The Am9513A requires only a single 5V power supply. Maximum supply currents are specified in the electrical specification at the high end of the voltage tolerance and the low end of the temperature range. In addition, the current specifications take into account the worstcase distribution of processing parameters that may be encountered during the manufacturing life of the product. Typical supply current values, on the other hand, are specified at a nominal +5.0 volts, a nominal ambient temperature of 25°C, and nominal processing parameters. Supply current always decreases with increasing ambient temperature: thermal run-away is not a problem.

Supply current will vary somewhat from part to part, but a given unit at a given operating temperature will exhibit a nearly constant power drain. There is no functional operating region that will cause more than a few percent change in the supply current. Decoupling of VCC, then, is straightforward and will generally be used to isolate the Am9513A from VCC noise originating externally

CONTROL PORT REGISTERS

The STC is addressed by the external system as only two locations: a Control port and a Data port. Transfers at the Control port (C/$\overline{D}$ = HIGH) allow direct access to the Command register when writing and the Status register when reading. All other available internal locations are accessed for both reading and writing via the Data port (C/$\overline{D}$ = LOW). Data port transfers are executed to and from the location currently addressed by the Data Pointer register. Options available in the Master Mode register and the Data Pointer control structure allow several types of transfer sequencing to be used. See Figure 1-7.

Transfers to and from the Control port are always 8-bits wide. Each access to the Control port will transfer data between the Command register (writes) or Status register (reads) and Data Bus pins DB0-DB7, regardless of whether the Am9513A is in 8- or 16-bit bus mode. When the Am9513A is in 8-bus mode, Data Bus pins DB13-DB15 should be held at a logic high whenever $\overline{CS}$ and WR are both active.

Command Register

The Command register provides direct control over each of the five general counters and controls access through the Data port by allowing the user to update the Data Pointer register. The "Command Description" section of this data sheet explains the detailed operation of each command. A summary of all commands appears in Figure 1-20. Six of the command types are used for direct software control of the counting process. Each of these six commands contains a 5-bit S field. In a linear-select fashion, each bit in the S field corresponds to one of the five general counters (S1 = Counter 1, S2 = Counter 2, etc.). When an S bit is a one, the specified operation is performed on the counter so designated; when an S bit is a zero, no operation occurs for the corresponding counter.

Data Pointer Register

The 6-bit Data Pointer register is loaded by issuing the appropriate command through the Control port to the Command register. As shown in Figure 1-7, the contents of the Data Pointer register are used to control the Data port multiplexer, selecting which internal register is to be accessible through the Data port.

The Data Pointer consists of a 3-bit Group Pointer, a 2-bit Element Pointer and a 1-bit Byte Pointer, depicted in Figure 1-8. The Byte Pointer bit indicates which byte of a 16-bit register is to be transferred on the next access through the Data port. Whenever the Data Pointer is loaded, the Byte Pointer bit is set to one, indicating a least-significant byte is expected. The Byte Pointer toggles following each 8-bit data transfer with an 8-bit data bus (MM13 = 0), or it always remains set with the 16-bit data bus option (MM13 = 1). The Element and Group pointers are used to select which internal register is to be accessible through the Data port. Although the contents of the Element and Group Pointer in the Data Pointer register cannot be read by the host processor, the Byte Pointer is available as a bit in the Status register.

Random access to any available internal data location can be accomplished by simply loading the Data Pointer using the command shown in Figure 1-9 and then initiating a data read or data write. This procedure can be used at any time, regardless of the setting of the Data Pointer Control bit (MM14). When the 8-bit data bus configuration is being used (MM13 = 0), two bytes of data would normally be transferred following the issuing of the "Load Data Pointer" command.

To permit the host processor to rapidly access the various internal registers, automatic sequencing of the Data Pointer is provided.

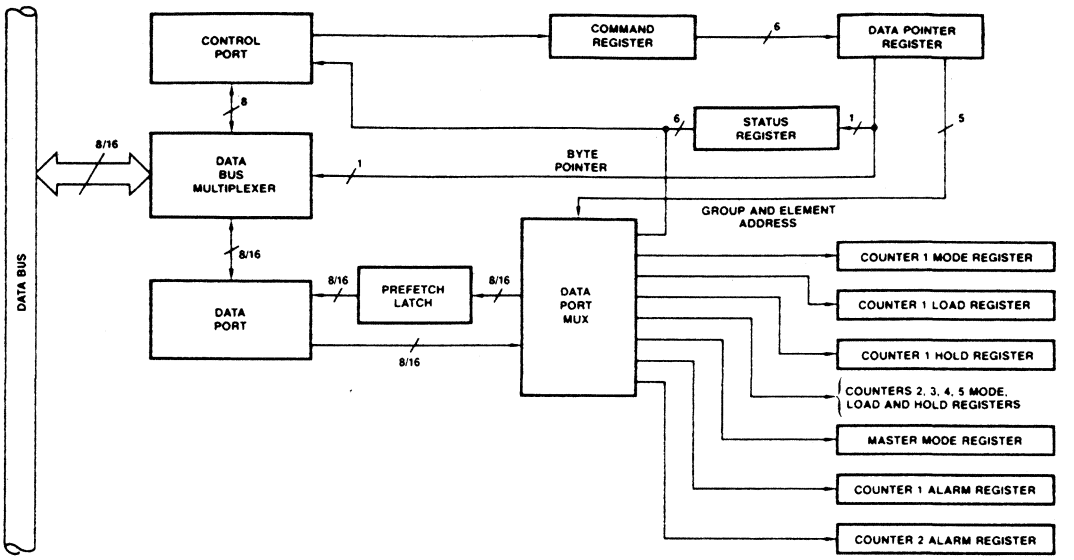


Figure 1-7. Am9513A Register Access

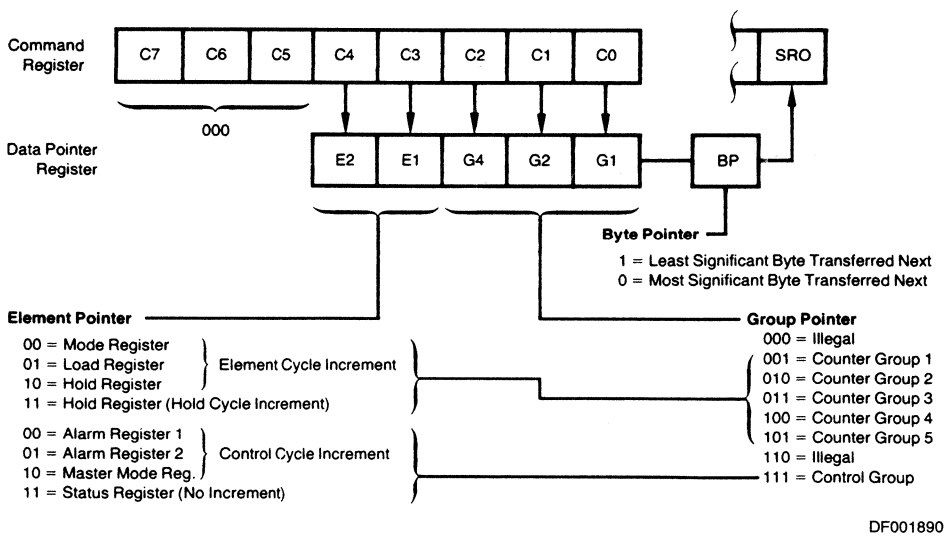


Figure 1-8. Data Pointer Register

	Element Cycle			Hold Cycle
	Mode Register	Load Register	Hold Register	Hold Register
Counter 1	FF01	FF09	FF11	FF19
Counter 2	FF02	FF0A	FF12	FF1A
Counter 3	FF03	FF0B	FF13	FF1B
Counter 4	FF04	FF0C	FF14	FF1C
Counter 5	FF05	FF0D	FF15	FF1D

Master Mode Register = FF17
Alarm 1 Register = FF07
Alarm 2 Register = FF0F
Status Register = FF1F

Notes:
1. All codes are in hex.
2. When used with an 8-bit bus, only the two low order hex digits should be written to the command port; the 'FF' prefix should be used only for a 16-bit data bus interface.

Figure 1-9. Load Data Pointer Commands

Sequencing is enabled by clearing Master Mode bit 14 (MM14) to zero. As shown in Figure 1-10 several types of sequencing are available depending on the data bus width being used and the initial Data Pointer value entered by command.

When E1 = 0 or E2 = 0 and G4, G2, G1 points to a Counter Group, the Data Pointer will proceed through the Element cycle. The Element field will automatically sequence through the three values 00, 01 and 10 starting with the value entered. When the transition from 10 to 00 occurs, the Group field will also be incremented by one. Note that the Element field in this case does not sequence to a value of 11. The Group field circulates only within the five Counter Group codes.

If E2, E1 = 11 and a Counter Group are selected, then only the Group field is sequenced. This is the Hold cycle. It allows the Hold registers to be sequentially accessed while bypassing the Mode and Load registers. The third type of sequencing is the Control cycle. If G4, G2, G1 = 111 and E2, E1 ≠ 11, the Element Pointer will be incremented through the values 00, 01 and 10, with no change to the Group Pointer.

When G4, G2, G1 = 111 and E2, E1 = 11, no incrementing takes place and only the Status register will be available through the Data port. Note that the Status register can also always be read directly through the Control port.

For all these auto-sequencing modes, if an 8-bit data bus is used, the Byte pointer will toggle after every data transfer to allow the least and most significant bytes to be transferred before the Element or Group fields are incremented.

Prefetch Circuit

To minimize the read access time to internal Am9513A registers, a prefetch circuit is used for all read operations through the Data port. Following each read or write operation through the Data port, the Data Pointer register is updated to point to the next register to be accessed. Immediately following this update, the new register data is transferred to a special prefetch latch at the interface pad logic. When the user performs a subsequent read of the Data port, the data bus drivers are enabled, outputting the prefetched data on the bus. Since the internal data register is accessed prior to the start of the read operation, its access time is transparent to the user. To keep the prefetched data consistent with the Data Pointer, prefetches are also performed after each write to the Data port and after execution of the "Load Data Pointer" command. The following rules should be kept in mind regarding Data port Transfers.

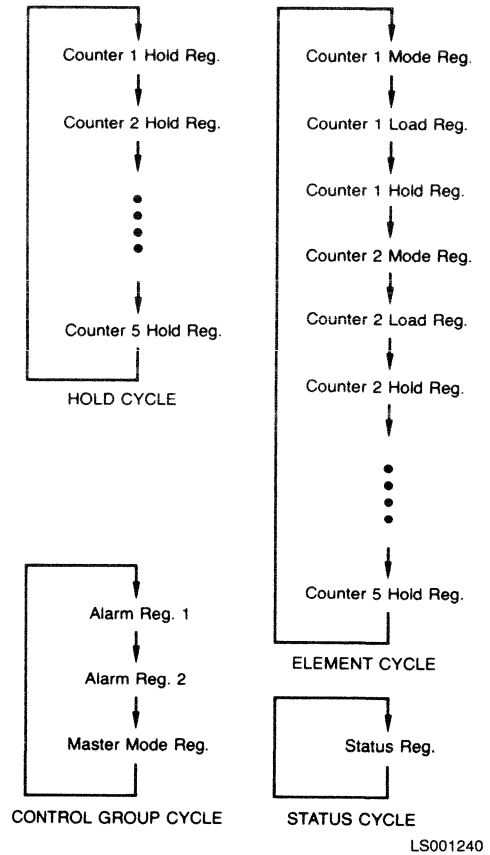


Figure 1-10. Data Pointer Sequencing

1. The Data Pointer register should always be reloaded before reading from the Data port if a command, other than "Load Data Pointer," was issued to the Am9513A following the last Data port read or write. The Data Pointer does not have to be loaded again if the first Data port transaction after a command entry is a write, since the Data port write will automatically cause a new prefetch to occur.
2. Operating modes N, O, Q, R and X allow the user to save the counter contents in the Hold register by applying an active-going gate edge. If the Data Pointer register had been pointing to the Hold register in question, the prefetched value will not correspond to the new value saved in the Hold Register. To avoid reading an incorrect value, a new "Load Data Pointer" command should be issued before attempting to read the saved data. A Data port write (to another register) will also initiate a prefetch; subsequent reads will access the recently saved Hold register data. Many systems will use the "saving" gate edge to interrupt the host CPU. In systems such as this, the interrupt service routine should issue a "Load Data Pointer" command prior to reading the saved data.

Status Register

The 8-bit read-only Status register indicates the state of the Byte Pointer bit in the Data Pointer register and the state of the

OUT signal for each of the general counters. See Figures 1-11 and 1-18. The OUT signals reported are those internal to the chip after the polarity-select logic and just before the 3-state interface buffer circuitry. Bits SR6 and SR7 may be 0 or 1.

The Status register OUT bit reflects an active-high or active-low TC output, or a TC Toggled output, as programmed in the Output Control Field of the Counter Mode register. That is, it reflects the exact state of the OUT pin. When the low-impedance to Ground Output option (CM2-CM0 = 000) is selected, the Status register will reflect an active-high TC Output. When a high-impedance Output option (CM2-CM0 = 100) is selected, the Status register will reflect an active-low TC output.

For Counters 1 and 2, the OUT pin will reflect the comparator output if the comparators are enabled. The Status register bit and OUT pin are active-high if CM2 = 0 and active-low if CM2 = 1. When the high-impedance option is selected and the comparator is enabled, the status register bit will reflect an active-high comparator output. When the low-impedance to Ground option is selected and the comparator is enabled, the status register bit will reflect an active-low comparator output.

The Status register is normally accessed by reading the Control port (see Figure 1-7) but may also be read via the Data port as part of the Control Group.

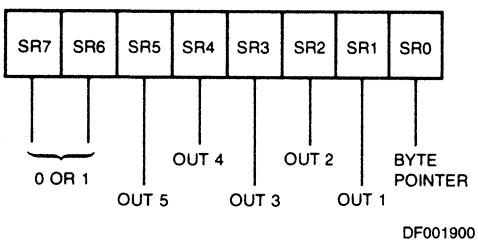


Figure 1-11. Status Register Bit Assignments

DATA PORT REGISTERS

Counter Logic Groups

As shown in Figures 1-5 and 1-6, each of the five Counter Logic Groups consists of a 16-bit general counter with associated control and output logic, a 16-bit Load register, a 16-bit Hold register and a 16-bit Mode register. In addition, Counter Groups 1 and 2 also include 16-bit Comparators and 16-bit Alarm registers. The comparator/alarm functions are controlled by the Master Mode register. The operation of the Counter Mode registers is the same for all five counters. The host CPU has both read and write access to all registers in the Counter Logic Groups through the Data port. The counter itself is never directly accessed.

Load Register

The 16-bit read/write Load register is used to control the effective length of the general counter. Any 16-bit value may be written into the Load register. That value can then be transferred into the counter each time the Terminal Count (TC) occurs. "Terminal Count" is defined as that period of time when the counter contents would have been zero if an external value had not been transferred into the counter. Thus, the terminal count frequency can be the input frequency divided by the value in the Load register. In all operating modes, either the Load or Hold register will be transferred into the counter when TC occurs. In cases where values are being accumulated in the counter, the Load register action can become transparent by filling the Load register with all zeros.

Hold Register

The 16-bit read/write Hold register is dual-purpose. It can be used in the same way as the Load register, thus offering an alternate source for module definition for the counter. The Hold register may also be used to store accumulated counter values for later transfer to the host processor. This allows the count to be sampled while the counting process proceeds without interruption. Transfer of the counter contents into the Hold register is accomplished by the hardware interface in some operating modes or by software commands at any time.

Counter Mode Register

The 16-bit read/write Counter Mode register controls the gating, counting, output and source select functions within each Counter Logic Group. The "Counter Mode Control Options" section of this document describes the detailed control options available. Figure 1-17 shows the bit assignments for the Counter Mode registers.

Alarm Registers and Comparators

Added functions are available in the Counter Logic Groups for Counters 1 and 2 (see Figure 1-5). Each contains a 16-bit Alarm register and a 16-bit Comparator. When the value in the counter reaches the value in the Alarm register, the Comparator output will go true. The Master Mode register contains control bits to individually enable/disable the comparators. When enabled, the comparator output appears on the OUT pin of the associated counter in place of the normal counter output. The output will remain true as long as the comparison is true, that is, until the next input causes the count to change. The polarity of the Comparator output will be active-high if the Output Control field of the Counter Mode register is 001 or 010 and active-low if the Output Control field is 101.

MASTER MODE CONTROL OPTIONS

The 16-bit Master Mode (MM) register is used to control those internal activities that are not controlled by the individual Counter Mode registers. This includes frequency control, Time-of-Day operation, comparator controls, data bus width and data pointer sequencing. Figure 1-12 shows the bit assignments for the Master Mode register. This section describes the use of each control field.

Master Mode register bits MM12, MM13 and MM14 can be individually set and reset using commands issued to the Command register. In addition, they can all be changed by writing directly to the Master Mode register.

After power-on reset or a Master Reset command, the Master Mode register is cleared to an all zero condition. This results in the following configuration:

Time-of-Day disabled
Both Comparators disabled
FOUT Source is frequency F1
FOUT Divider set for divide-by-16
FOUT gated on
Data Bus 8 bits wide
Data Pointer Sequencing enabled
Frequency Scaler divides in binary

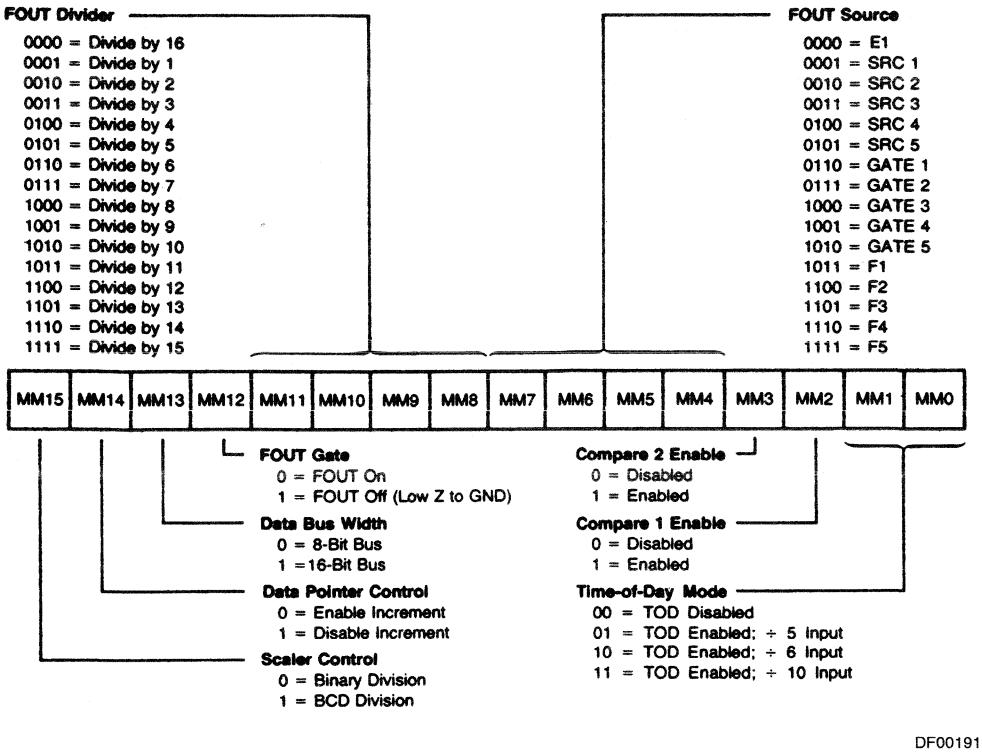


Figure 1-12. Master Mode Register Bit Assignments

Time-of-Day

Bits MM0 and MM1 of the Master Mode register specify the Time-of-Day (TOD) options. When MM0 = 0 and MM1 = 0, the special logic used to implement TOD is disabled, and Counters 1 and 2 will operate in exactly the same way as Counters 3, 4 and 5. When MM0 = 1 or MM1 = 1, additional counter decoding and control logic is enabled on Counters 1 and 2, which causes their decades to turn over at the counts that generate appropriate 24-hour TOD accumulations. For additional information, see the Time-of-Day chapter in the 9513A System timing controller technical manual.

Comparator Enable

Bits MM2 and MM3 control the Comparators associated with Counters 1 and 2. When a Comparator is enabled, its output is substituted for the normal counter output on the associated OUT1 or OUT2 pin. The comparator output will be active-high if the output control field of the Counter Mode register is 001 or 010 and active-low for a code of 101. Once the compare output is true, it will remain so until the count changes and the comparison therefore goes false.

The two Comparators can always be used individually in any operating mode. One special case occurs when the Time-of-Day option is revoked and both Comparators are enabled. The operation of Comparator 2 will then be conditioned by Comparator 1 so that a full 32-bit compare must be true in order to generate a true signal on OUT2. OUT1 will continue, as usual, to reflect the state of the 16-bit comparison between Alarm 1 and Counter 1.

FOUT Source

Master Mode bits MM4 through MM7 specify the source input for the FOUT divider. Fifteen inputs are available for selection, and they include the five Source pins, the five Gate pins and the five internal frequencies derived from the oscillator. The 16th combination of the four control bits (all zeros) is used to assure that an active frequency is available at the input to the FOUT divider following reset.

FOUT Divider

Bits MM8 through MM11 specify the dividing ratio for the FOUT Divider. The FOUT source (selected by bits MM4 through MM7) is divided by an integer value between 1 and 16, inclusive, and is then passed to the FOUT output buffer. After power-on or reset, the FOUT divider is set to divide-by-16.

FOUT Gate

Master Mode bit MM12 provides a software gating capability for the FOUT signal. When MM12 = 1, FOUT is off and in a low-impedance state to ground. MM12 may be set or cleared in conjunction with the loading of the other bits in the Master Mode register; alternatively, there are commands that allow MM12 to be individually set or cleared directly without changing any other Master Mode bits. After power-up or reset, FOUT is gated on.

When changing the FOUT divider ratio or FOUT source, transient pulses as short as half the period of the FOUT source may appear on the FOUT pin. Turning the FOUT gate on or off can also generate a transient. This should be considered when using FOUT as a system clock source.

Bus Width

Bit MM13 controls the multiplexer at the data bus interface in order to configure the part for an 8-bit or 16-bit external bus. The internal bus is always 16-bits wide. When MM13 = 1, 16-bit data is transferred directly between the internal bus and all 16 of the external bus lines. In this configuration, the Byte Pointer bit in the Data Pointer register remains set at all times. When MM13 = 0, 16-bit internal data is transferred a byte at a time to and from the eight low-order external data bus lines. The Byte Pointer bit toggles with each byte transfer in this mode.

When the Am9513A is set to operate with an 8-bit data bus width, pins DB8 through DB15 are not used for the data bus and are available for other functions. Pins DB13 through DB15 should be tied high. Pins DB8 through DB12 are used as auxiliary gating inputs and are labeled GATE1A through GATE5A respectively. The auxiliary gate pin, GATENA, is logically ANDed with the gate input to Counter N, as shown in Figure 1-13. The output of the AND gate is then used as the gating signal for Counter N.

Data Pointer Sequencing

Bit MM14 controls the Data Pointer logic to enable or disable the automatic sequencing functions. When MM14 = 1, the contents of the Data Pointer can be changed only directly by entering a command. When MM14 = 0, several types of automatic sequencing of the Data Pointer are available. These are described in the Data Pointer register section of this document.

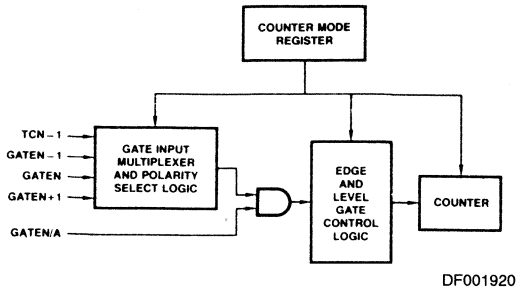
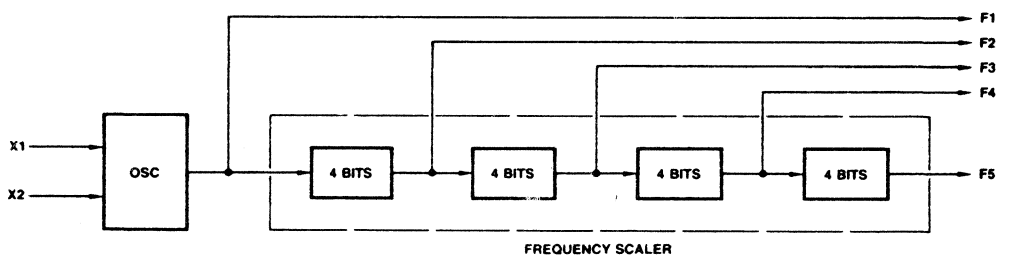


Figure 1-13. Gating Control

Thus, the host processor, by controlling MM14, may repetitively read/write a single internal location, or may sequentially read/write groups of locations. Bit MM14 can be loaded by writing to the Master Mode register or can be set or cleared by software command.

Scaler Ratios

Master Mode bit MM15 controls the counting configuration of the Frequency Scaler counter. When MM15 = 0, the Scaler divides the oscillator frequency in binary steps so that each subfrequency is 1/16 of the preceding frequency. When MM15 = 1, the Scaler divides in BCD steps so that adjacent frequencies are related by ratios of 10 instead of 16 (see Figure 1-14).



Frequency	BCD Scaling MM15 = 1	Binary Scaling MM15 = 0
F1	OSC	OSC
F2	F1 ÷ 10	F1 ÷ 16
F3	F1 ÷ 100	F1 ÷ 256
F4	F1 ÷ 1,000	F1 ÷ 4,096
F5	F1 ÷ 10,000	F1 ÷ 65,536

Figure 1-14. Frequency Scaler Ratios

Counter Mode	A	B	C	D	E	F	G	H	I	J	K	L
Special Gate (CM7)	0	0	0	0	0	0	0	0	0	0	0	0
Reload Source (CM6)	0	0	0	0	0	0	1	1	1	1	1	1
Repetition (CM5)	0	0	0	1	1	1	0	0	0	1	1	1
Gate Control (CM15-CM13)	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE
Count to TC once, then disarm	X	X	X									
Count to TC twice, then disarm							X	X	X			
Count to TC repeatedly without disarming				X	X	X				X	X	X
Gate input does not gate counter input	X			X			X			X		
Count only during active gate level		X			X			X			X	
Start count on active gate edge and stop count on next TC			X			X						
Start count on active gate edge and stop count on second TC									X			X
No hardware retriggering	X	X	X	X	X	X	X	X	X	X	X	X
Reload counter from Load register on TC	X	X	X	X	X	X						
Reload counter on each TC, alternating reload source between Load and Hold registers							X	X	X	X	X	X
Transfer Load register into counter on each TC that gate is LOW, transfer Hold register into counter on each TC that gate is HIGH.												
On active gate edge transfer counter into Hold register and then reload counter from Load register												

Counter Mode	M	N	O	P	Q	R	S	T	U	V	W	X
Special Gate (CM7)	1	1	1	1	1	1	1	1	1	1	1	1
Reload Source (CM6)	0	0	0	0	0	0	1	1	1	1	1	1
Repetition (CM5)	0	0	0	1	1	1	0	0	0	1	1	1
Gate Control (CM15-CM13)	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE
Count to TC once, then disarm		X	X									
Count to TC twice, then disarm							X					
Count to TC repeatedly without disarming					X	X				X		X
Gate input does not gate counter input							X			X		
Count only during active gate level		X			X							
Start count on active gate edge and stop count on next TC			X			X						X
Start count on active gate edge and stop count on second TC												
No hardware retriggering							X			X		X
Reload counter from Load register on TC		X	X		X	X						X
Reload counter on each TC, alternating reload source between Load and Hold registers.												
Transfer Load register into counter on each TC that gate is LOW, transfer Hold register into counter on each TC that gate is HIGH.							X			X		
On active gate edge transfer counter into Hold register and then reload counter from Load register		X	X		X	X						
On active gate edge transfer counter into Hold register, but counting continues												X

Notes: 1. Counter modes M, P, T, U and W are reserved and should not be used.
2. Mode X is available for Am9513A only.

Figure 1-15 Counter Mode Operating Summary

COUNTER MODE DESCRIPTIONS

Counter Mode register bits CM15-CM13 and CM7-CM5 select the operating mode for each counter (see Figure 1-15). To simplify references to a particular mode, each mode is assigned a letter from A through X. Representative waveforms for the counter modes are illustrated in Figures 1-16a through 1-16v. (Because the letter suffix in the figure number is keyed to the mode, Figures 1-16m, 1-16p, 1-16t, 1-16u and 1-16w do not exist.) The figures assume down counting on rising source edges. Those modes which automatically disarm the counter (CM5 = 0) are shown with the $\overline{WR}$ plus entering the required ARM command; for modes which count repetitively (CM5 = 1), the ARM command is omitted. The retriggering modes (N, O, Q and R) are shown with one retrigger operation. Both a TC output waveform and a TC Toggled output waveform are shown for each mode. The symbols L and H are used to represent count values equal to the Load and Hold register contents, respectively. The symbols K and N represent arbitrary count values. For each mode, the required bit pattern in the Counter Mode register is shown; "don't care" bits are marked "X." These figures are designed to clarify the mode descriptions; the Am9513A Electrical Specification should be used as the authoritative reference for timing relationships between signals.

To keep the following mode descriptions concise and to the point, the phrase "source edges" is used to refer to active-going source edges only, not to inactive-going edges. Similarly, the phrase "gate edges" refers only to active-going gate edges. Also, again to avoid verbosity and euphuism, the descriptions of some modes state that a counter is stopped or disarmed "on a TC, inhibiting further counting." As is fully explained in the TC section of this document, for these modes the counter is actually stopped or disarmed following the active-going source edge which drives the counter out of TC. In other words, since a counter in the TC state always counts, irrespective of its gating or arming status, the stopping or disarming of the count sequence is delayed until TC is terminated.

MODE A

Software-Triggered Strobe with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode A, shown in Figure 1-16a, is one of the simplest operating modes. The counter will be available for counting source edges when it is issued an ARM command. On each TC, the counter will reload from the Load register and automatically disarm itself, inhibiting further counting. Counting will resume when a new ARM command is issued.

MODE B

Software-Triggered Strobe with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode B, shown in Figure 1-16b, is identical to Mode A except that source edges are counted only when the assigned Gate is active. The counter must be armed before counting can occur. Once armed, the counter will count all source edges which occur while the Gate is active and disregard those edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. On each TC the counter will reload from the Load register and automatically disarm itself, inhibiting further counting unitl a new ARM command is issued.

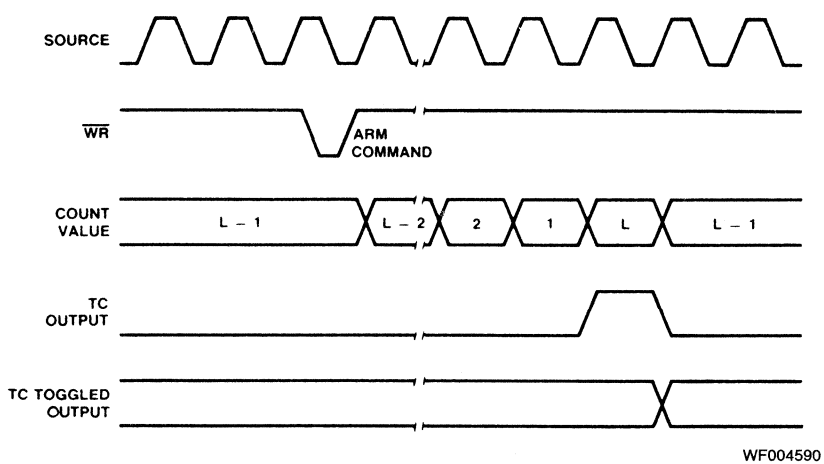
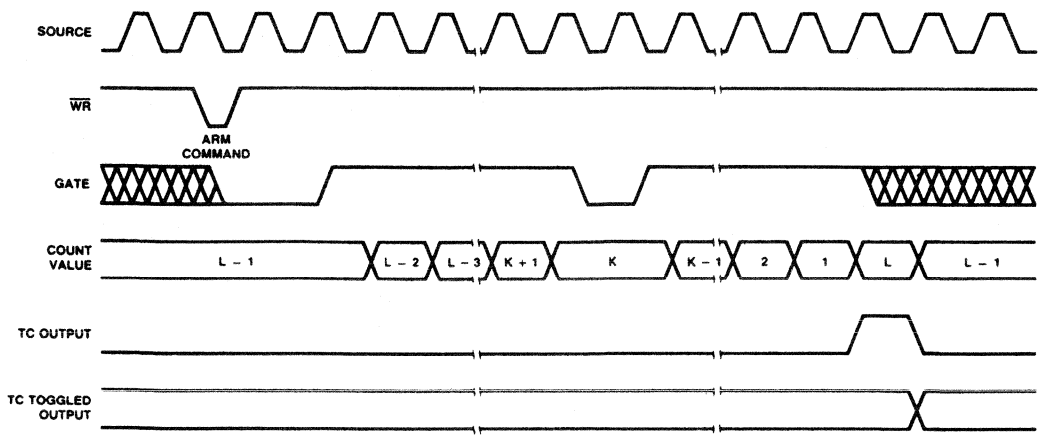


Figure 1-16a. Mode A Waveforms



WF004600

Figure 1-16b. Mode B Waveforms

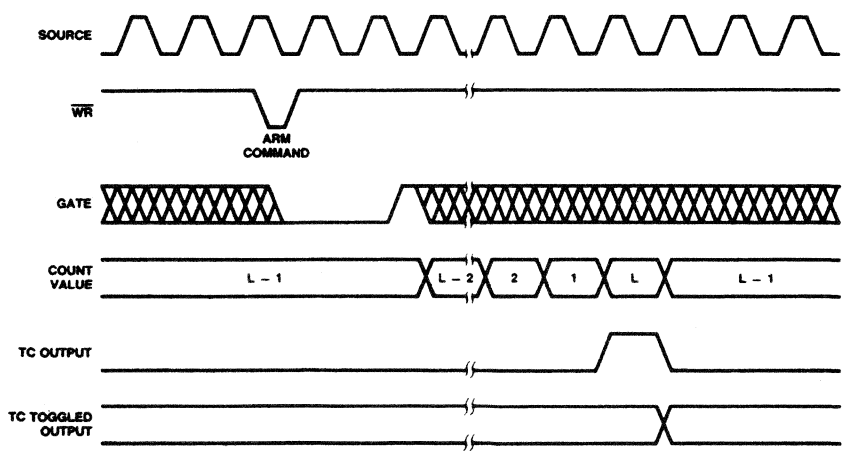
MODE C

Hardware-Triggered Strobe

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode C, shown in Figure 1-16c, is identical to Mode A, except that counting will not begin until a Gate edge is applied to the armed counter. The counter must be armed before application of the triggered Gate edge; Gate edges applied to a disarmed counter are disregarded. The counter will start counting on the first source edge after the triggering Gate edge and will continue counting until TC. At TC, the counter will reload from the Load register and automatically disarm itself. Counting will then remain inhibited until a new ARM command and a new Gate edge are applied in that order. Note that after application of a triggered Gate edge, the Gate input will be disregarded for the remainder of the count cycle. This differs from Mode B, where the Gate can be modulated throughout the count cycle to stop and start the counter.



WF004610

Figure 1-16c. Mode C Waveforms

MODE D

Rate Generator with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode D, shown in Figure 1-16d, is typically used in frequency generation applications. In this mode, the Gate input does not affect counter operation. Once armed, the counter will count to TC repetitively. On each TC, the counter will reload itself from the Load register; hence, the Load register value determines the time between TCs. A square wave rate generator may be obtained by specifying the TC Toggled output mode in the Counter Mode register.

MODE E

Rate Generator with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode E, shown in Figure 1-16e, is identical to Mode D, except the counter will only count those source edges which occur while the Gate input is active. This feature allows the counting process to be enabled and disabled under hardware control. A square wave rate generator may be obtained by specifying the TC Toggled output mode.

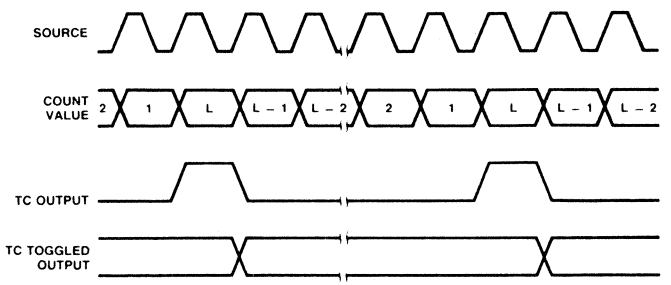


Figure 1-16d. Mode D Waveforms

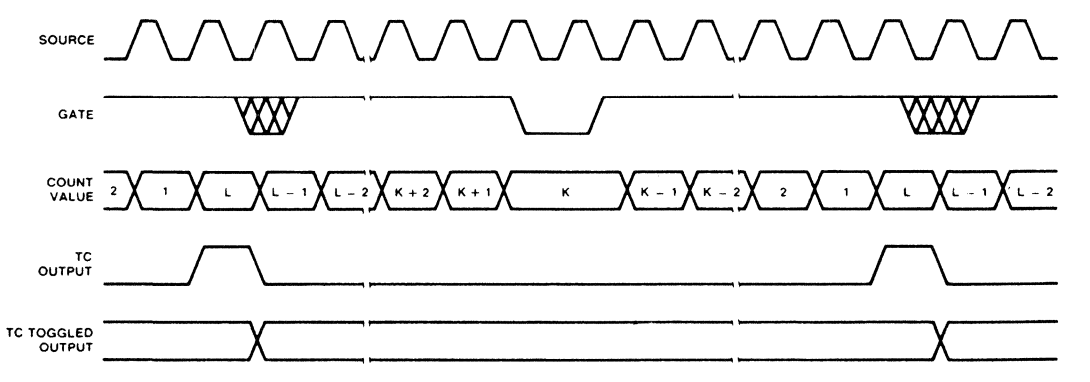


Figure 1-16e. Mode E Waveforms

MODE F

Non-Retriggerable One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode F, shown in Figure 1-16f, provides a non-retriggerable one-shot timing function. The counter must be armed before it will function. Application of a Gate edge to the armed counter will enable counting. When the counter reaches TC, it will reload itself from the Load register. The counter will then stop counting, awaiting a new Gate edge. Note that unlike Mode C, a new ARM command is not needed after TC, only a new Gate edge. After application of a triggering Gate edge, the Gate input is disregarded until TC.

MODE G

Software-Triggered Delayed Pulse One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

In Mode G, the Gate does not affect the counter's operation. Once armed, the counter will count to TC twice and then automatically disarm itself. For most applications, the counter will initially be loaded from the Load register either by a LOAD command or by the last TC of an earlier timing cycle. Upon counting to the first TC, the counter will reload itself from the Hold register. Counting will proceed until the second TC, when the counter will reload itself from the Load register and automatically disarm itself, inhibiting further counting. Counting can be resumed by issuing a new ARM command. A software-triggered delayed pulse one-shot may be generated by specifying the TC Toggled output mode in the Counter Mode register. The initial counter contents control the delay from the ARM command until the output pulse starts. The Hold register contents control the pulse duration. Mode G is shown in Figure 1-16g.

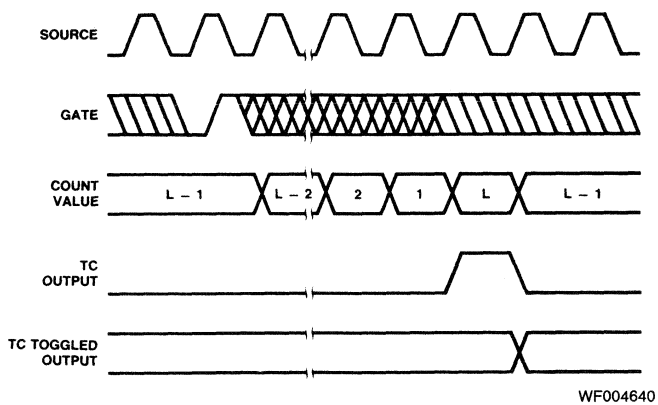


Figure 1-16f. Mode F Waveforms

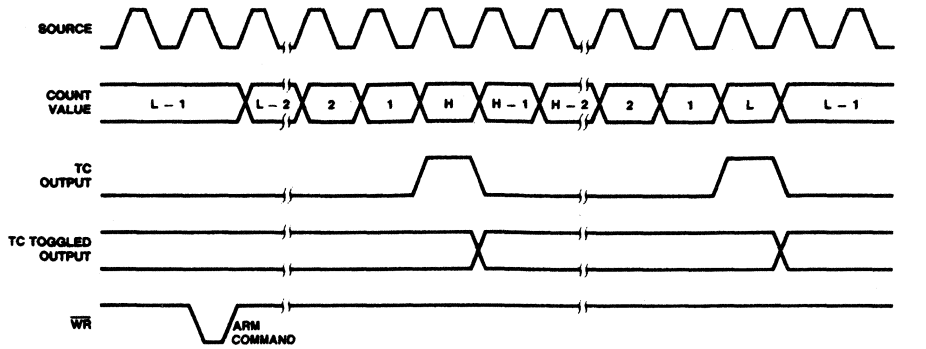


Figure 1-16g. Mode G Waveforms

MODE H

Software-Triggered Delayed Pulse One-Shot with Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

Mode H, shown in Figure 1-16h, is identical to Mode G except that the Gate input is used to qualify which source edges are to be counted. The counter must be armed for counting to occur. Once armed, the counter will count all source edges that occur while the Gate is inactive. This permits the Gate to turn the count process on and off. As with Mode G, the counter will be reloaded from the Hold register on the first TC and reloaded from the Load register and disarmed on the second TC. This mode allows the Gate to control the extension of both the initial output delay time and the pulse width.

MODE I

Hardware-Triggered Delayed Pulse Strobe

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

Mode I, shown in Figure 1-16i, is identical to Mode G, except that counting will not begin until a Gate edge is applied to an armed counter. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. An armed counter will start counting on the first source edge after the triggering Gate edge. Counting will then proceed in the same manner as in Mode G. After the second TC, the counter will disarm itself. An ARM command and Gate edge must be issued in this order to restart counting. Note that after application of a triggering Gate edge, the Gate input will be disregarded until the second TC. This differs from Mode H, where the Gate can be modulated throughout the count cycle to stop and start the counter.

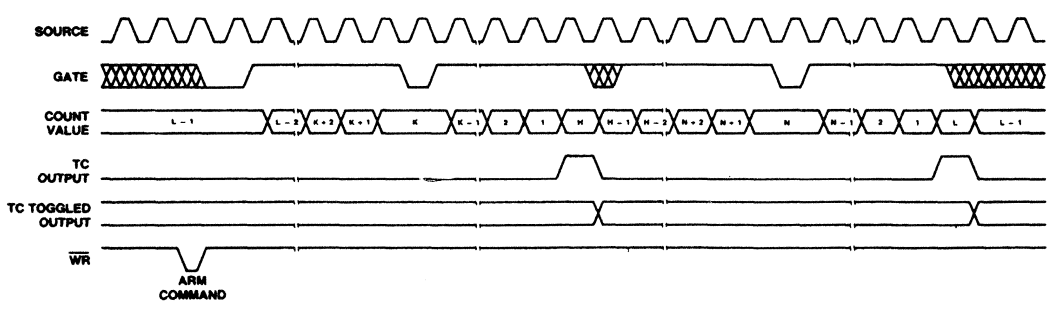


Figure 1-16h. Mode H Waveforms

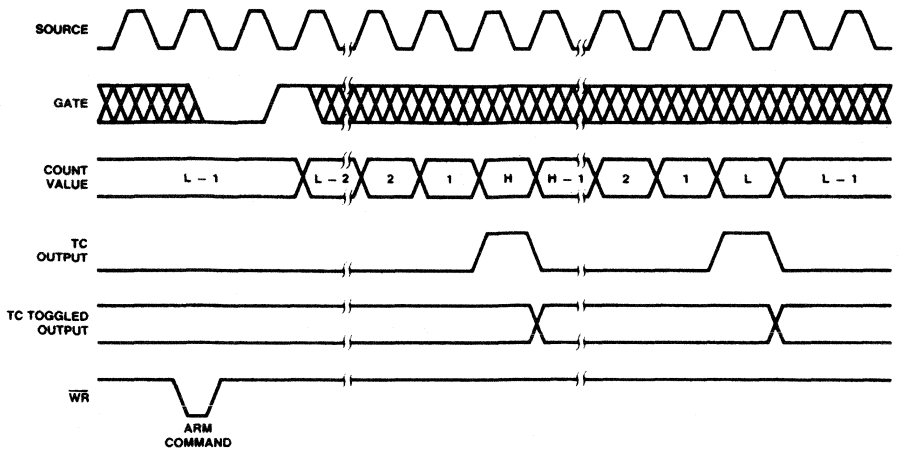


Figure 1-16i. Mode I Waveforms

MODE J

Variable Duty Cycle Rate Generator with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode J, shown in Figure 1-16i, will find the greatest usage in frequency generation applications with variable duty cycle requirements. Once armed, the counter will count continuously until it is issued a DISARM command. On the first TC, the counter will be reloaded from the Hold register. Counting will then proceed until the second TC at which time the counter will be reloaded from the Load register. Counting will continue, with the reload source alternating on each TC, until a DISARM command is issued to the counter. (The third TC reloads from the Hold register, the fourth TC reloads from the Load register, etc.) A variable duty cycle output can be generated by specifying the TC Toggled output in the Counter Mode register. The Load and Hold values then directly control the output duty cycle, with high resolution available when relatively high count values are used.

MODE K

Variable Duty Cycle Rate Generator with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode K, shown in Figure 1-16k, is identical to Mode J, except that source edges are only counted when the Gate is active. The counter must be armed for counting to occur. Once armed, the counter will count all source edges which occur while Gate is active and disregard those source edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. As with Mode J, the reload source used will alternate on each TC, starting with the Hold register on the first TC after any ARM command. When the TC Toggled output is used, this mode allows the Gate to modulate the duty cycle of the output waveform. It can affect both the HIGH and LOW portions of the output waveform.

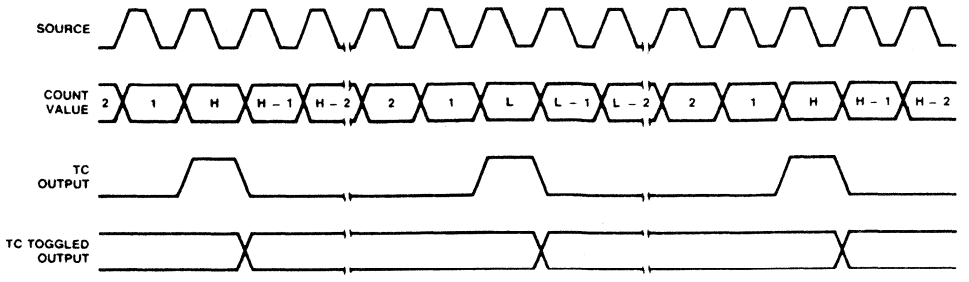


Figure 1-16j. Mode J Waveforms

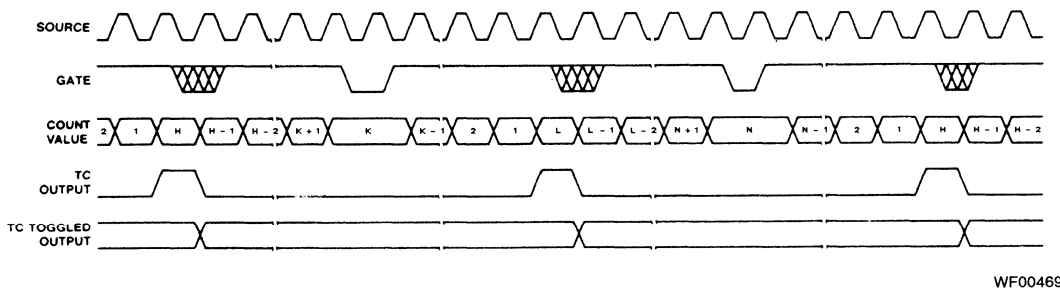


Figure 1-16k. Mode K Waveforms

MODE L

Hardware-Triggered Delayed Pulse One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode L, shown in Figure 1-16l, is similar to Mode J except that counting will not begin until a Gate edge is applied to an armed counter. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. The counter will start counting source edges after the triggering Gate edge, and counting will proceed until the second TC. Note that after application of a triggering Gate edge, the Gate input will be disregarded for the remainder of the count cycle. This differs from Mode K, where the gate can be modulated throughout the count cycle to stop and start the counter. On the first TC after application of the triggering Gate edge, the counter will be reloaded from the Hold register. On the second TC, the counter will be reloaded from the Load register, and counting will stop until a new gate edge is issued to the counter. Note that unlike Mode K, new Gate edges are required after every second TC to continue counting.

MODE N

Software-Triggered Strobe with Level Gating and Hardware Retriggering

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	0	X	X	X	X	X

Mode N, shown in Figure 1-16n, provides a software-triggered strobe with level gating that is also hardware retriggerable. The counter must be issued an ARM command before counting can occur. Once armed, the counter will count all source edges which occur while the gate is active and disregard those source edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. After the issuance of the ARM command and the application of an active Gate, the counter will count to TC. Upon reaching TC, the counter will reload from the Load register and automatically disarm itself, inhibiting further counting. Counting will resume upon the issuance of a new ARM command. All active-going Gate edges issued to an armed counter will cause a retrigger operation. Upon application of the Gate edge, the counter contents will be saved in the Hold register. On the first qualified source edge after application of the retriggering gate edge, the contents of the Load register will be transferred into the counter. Counting will resume on the second qualified source edge after the retriggering Gate edge. Qualified source edges are active-going edges which occur while the Gate is active.

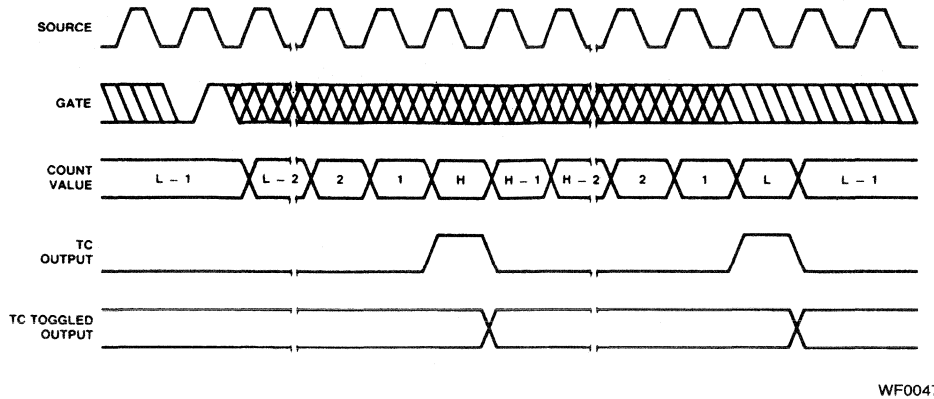


Figure 1-16l. Mode L Waveforms

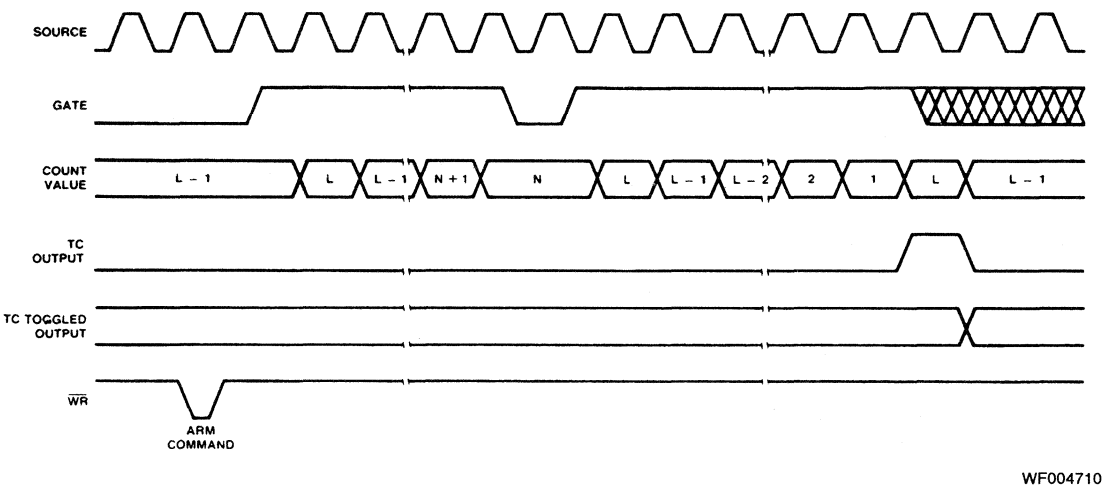


Figure 1-16n. Mode N Waveforms

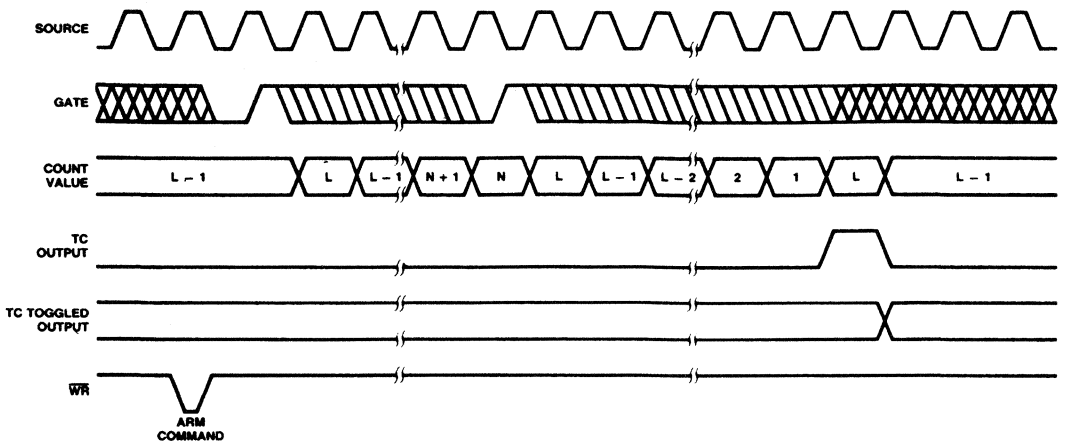
MODE O

Software-Triggered Strobe with Edge Gating and Hardware Retriggering

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	0	X	X	X	X	X

Mode O, shown in Figure 1-16o, is similar Mode N, except that counting will not begin until an active-going Gate edge is applied to an armed counter and the Gate level is not used to modulate counting. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. Irrespective of the Gate level, the counter will count all source edges after the triggering Gate edge until the first TC. On the first TC,the counter will be reloaded from the Load register and disarmed. A new ARM command and a new Gate edge must be applied in that order to initiate a new counting cycle. Unlike Modes C, F, I and L, which disregard the Gate input once counting starts, in Mode O the count process will be retriggered on all active-going Gate edges, including the first Gate edge used to start the counter. On each retriggering Gate edge, the counter contents will be transferred into the Hold register. On the first source edge after the retriggering Gate edge, the Load register contents will be transferred into the counter. Counting will resume on the second-source edge after a retrigger.



WF004720

Figure 1-16o. Mode O Waveforms

MODE Q

Rate Generator with Synchronization (Event Counter with Auto-Read/Reset)

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	1	X	X	X	X	X

Mode Q, shown in Figure 1-16q, provides a rate generator with synchronization or an event counter with auto-read/reset. The counter must first be issued an ARM command before counting can occur. Once armed, the counter will count all source edges which occur while the Gate is active and disregard those edges which cccur while the Gate is inactive. This permits the Gate to turn the count process on and off. After the issuance of an ARM command and the application of an active Gate, the counter will count to TC repetitively. On each TC,the counter will reload itself from the Load register. The counter may be retriggered at any time by presenting an active-going Gate edge to the Gate input. The retriggering Gate edge will transfer the contents of the counter into the Hold register. The first qualified source edge after the retriggering Gate edge will transfer the contents of the Load register into the Counter. Counting will resume on the second qualified source edge after the retriggering Gate edge. Qualified source edges are active-going edges which occur while the Gate is active.

MODE R

Retriggerable One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	1	X	X	X	X	X

Mode R, shown in Figure 1-16r, is similar to Mode Q, except that edge gating rather than level gating is used. In other words, rather than use the Gate level to qualify which source edges to count, Gate edges are used to start the counting operation. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. After application of a Gate edge, an armed counter will count all source edges until TC, irrespective of the Gate level. On the first TC, the counter will be reloaded from the Load register and stopped. Subsequent counting will not occur until a new Gate edge is applied. All Gate edges applied to the counter, including the first used to trigger counting, initiate a retrigger operation. Upon application of a Gate edge, the counter contents are saved in the Hold register. On the first source edge after the retriggering Gate edge, the Load register contents will be transferred into the counter. Counting will resume on the second source edge after the retriggering Gate edge.

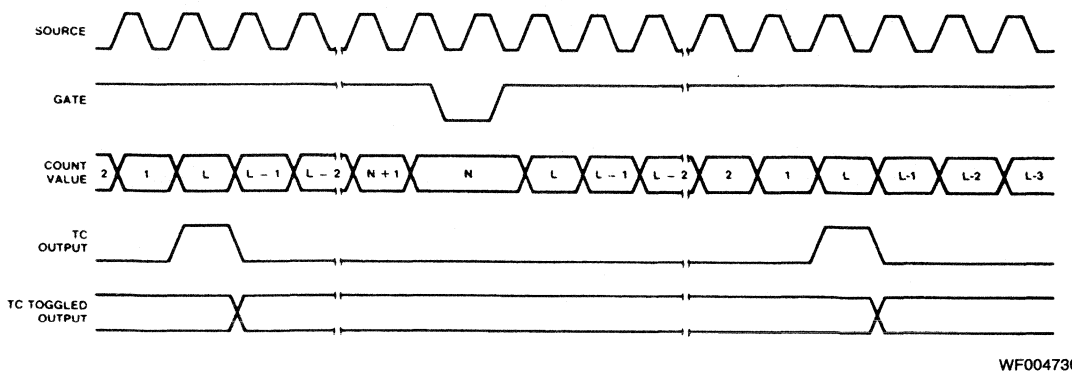


Figure 1-16q. Mode Q Waveforms

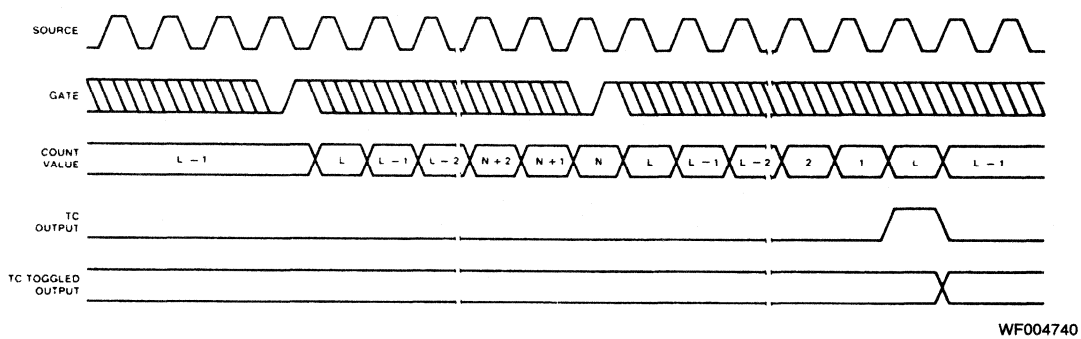


Figure 1-16r. Mode R Waveforms

MODE S

RELOAD SOURCE

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	0	X	X	X	X	X

In this mode, the reload source for LOAD commands (irrespective of whether the counter is armed or disarmed) and for TC-initiated reloads is determined by the Gate input. The Gate input in Mode S is used only to select the reload source, not to start or modulate counting. When the Gate is Low, the Load register is used; when the Gate is High, the Hold register is used. Note the Low-Load, High-Hold mnemonic convention. Once armed, the counter will count to TC twice and then disarm itself. On each TC, the counter will be reloaded from the reload source selected by the Gate. Following the second TC, an ARM command is required to start a new counting cycle. Mode S is shown in Figure 1-16s.

MODE V

Frequency-Shift Keying

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	1	X	X	X	X	X

Mode V, shown in Figure 1-16v, provides frequency-shift keying modulation capability. Gate operation in this mode is identical to that in Mode S. If the Gate is Low, a LOAD command or a TC-induced reload will reload the counter from the Load register. If the Gate is HIGH, LOADs and reloads will occur from the Hold register. The polarity of the Gate only selects the reload source; it does not start or modulate counting. Once armed, the counter will count repetitively to TC. On each TC, the counter will reload itself from the register determined by the polarity of the Gate. Counting will continue in this manner until a DISARM command is issued to the counter. Frequency shift keying may be obtained by specifying a TC Toggled output mode in the Counter Mode register. The switching of frequencies is achieved by modulating the Gate.

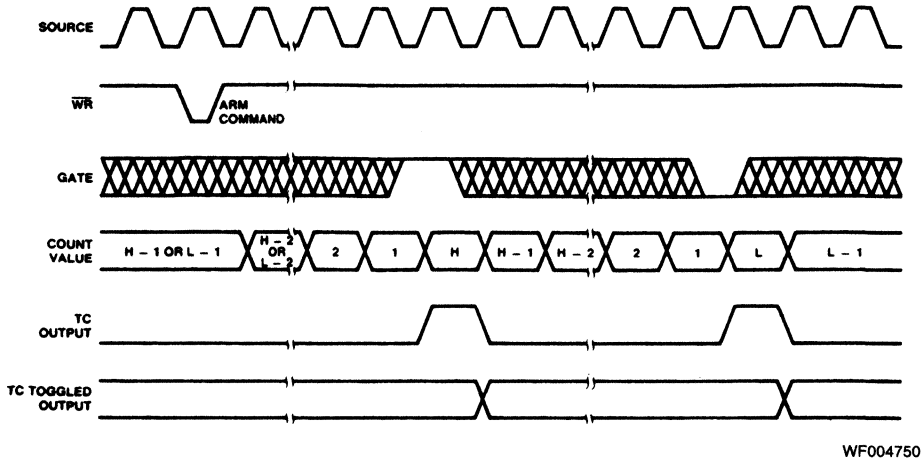


Figure 1-16s. Mode S Waveforms

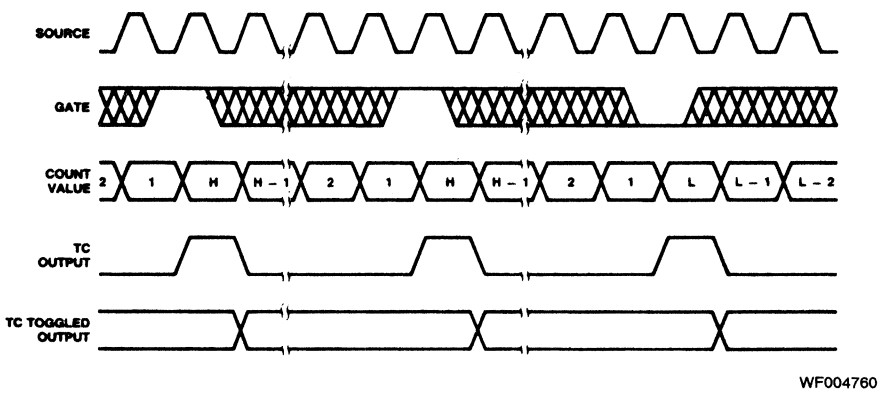


Figure 1-16v. Mode V Waveforms

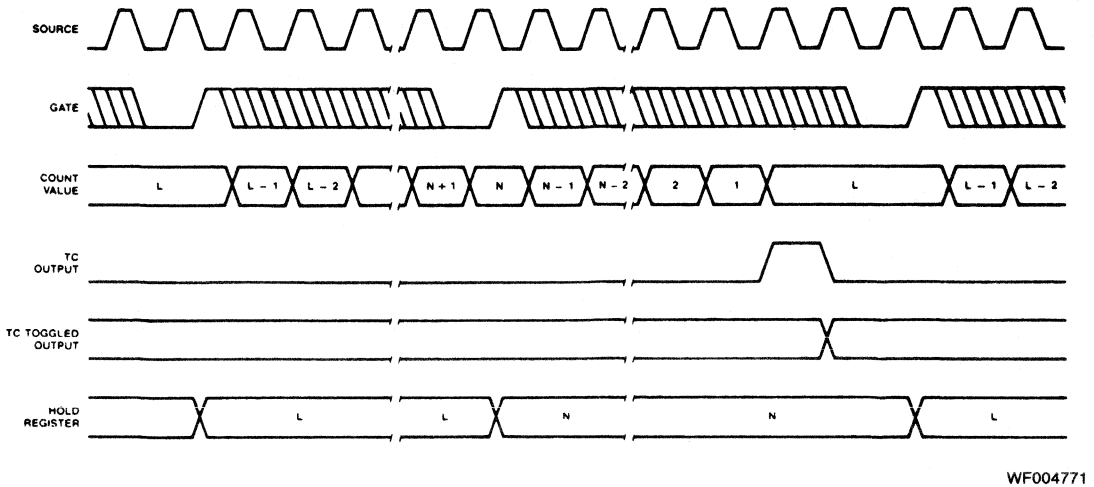


Figure 1-16x. Mode X Waveforms

MODE X

Hardware Save (available in Am9513A only)

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
Edge			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	1	X	X	X	X	X

Mode X, as shown in Figure 1-16x, provides a hardware sampling of the counter contents without interrupting the count. A LOAD AND ARM command or a LOAD command followed by an ARM command is required to initialize the counter. Once armed, a Gate edge starts the counting operation; Gate edges applied to a disarmed counter are disregarded. After application of the Triggering Gate edge, the counter will count all qualified source edges until the first TC, irrespective of the gate level. All gate edges applied during the counting sequence will store the current count in the Hold register, but they will not interrupt the counting sequence. On each TC, the counter will be reloaded from the Load register and stopped. Subsequent counting requires a new triggering Gate edge; counting resumes on the first source edge following the triggering Gate edge.

Note: Mode X is only available in the Am9513'A' devices.

COUNTER MODE CONTROL OPTIONS

Each Counter Logic Group includes a 16-bit Counter Mode (CM) register used to control all of the individual options available with its associated general counter. These options include output configuration, count control, count source and gating control. Figure 1-17 shows the bit assignments for the Counter Mode registers. This section describes the control options in detail. Note that generally each counter is independently configured and does not depend on information outside its Counter Logic Group. The Counter Mode register should be loaded only when the counter is Disarmed. Attempts to load the Counter Mode register when the counter is armed may result in erratic counter operation.

After power-on reset or a Master Reset command, the Counter Mode registers are initialized to a preset condition. The value entered is 0B00 hex and results in the following control configuration:

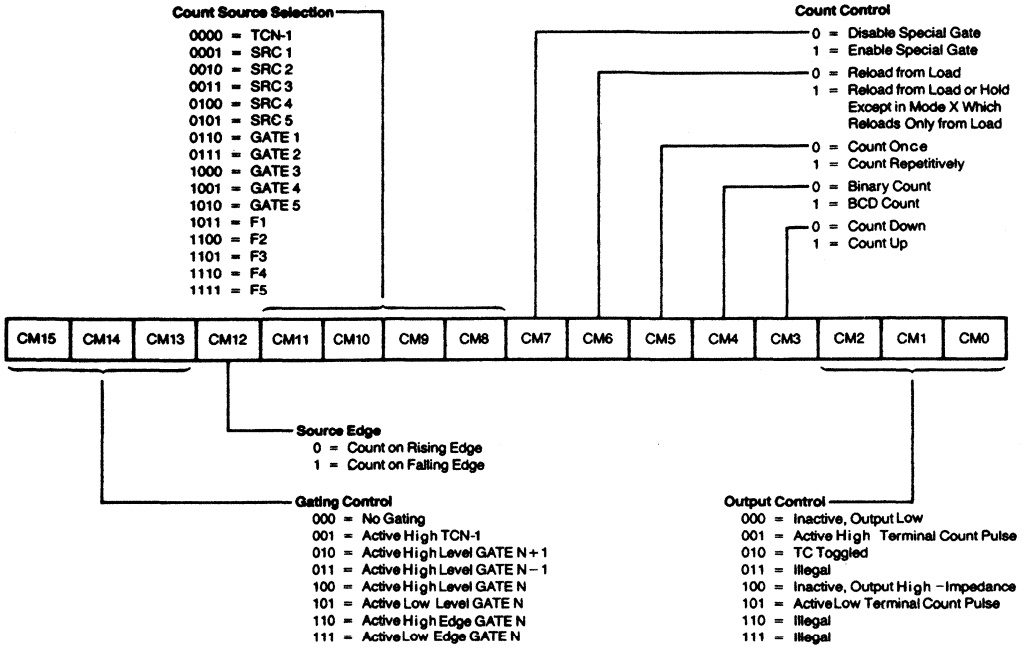
Output low-impedance to ground
Count down
Count binary
Count once
Load register selected
No retriggering
F1 input source selected
Positive-true input polarity
No gating

Output Control

Counter mode bits CM0 through CM2 specify the output control configuration. Figure 1-18 shows a schematic representation of the output control logic. The OUT pin may be off (a high-impedance state), or it may be inactive with a low-impedance to ground. The three remaining valid combinations represent the active-high, active-low or TC Toggle output waveforms.

One output form available is called Terminal Count (TC) and represents the period in time that the counter reaches an equivalent value of zero. TC will occur on the next count when the counter is at 0001 for down counting, at 9999 (BCD) for BCD up counting or at FFFF (hex) for binary up counting. Figure 1-19 shows a Terminal Count pulse and an example context that generated it. The TC width is determined by the period of the counting source. Regardless of any gating input or whether the counter is Armed or Disarmed, the terminal count will go active for only one clock cycle. Figure 1-19 assumes active-high source polarity, counter armed, counter decrementing and an external reload value of K.

The counter will always be loaded from an external location when TC occurs; the user can choose the source location and the value. If a non-zero value is picked, the counter will never really attain a zero state, and TC will indicate the counter state that would have been zero had no parallel transfer occurred.



Note: See Figure 1-16 for restrictions on Count Control and Gating Control bit combinations.

Figure 1-17. Counter Mode Register Bit Assignments

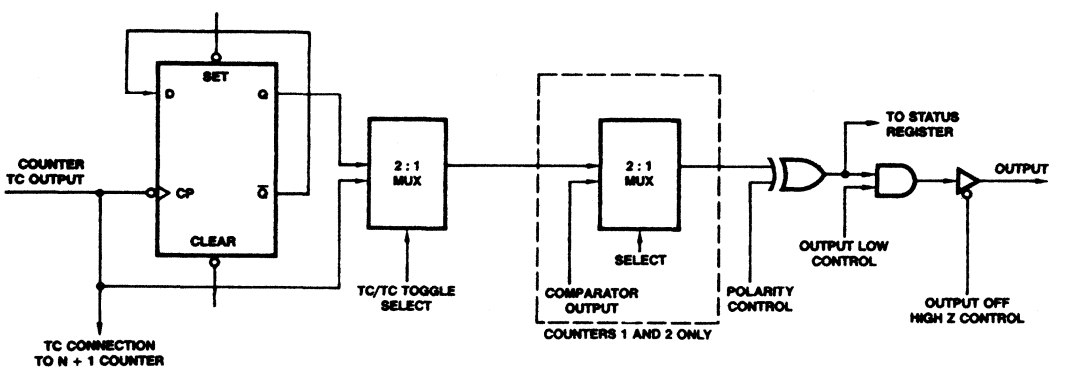


Figure 1-18. Output Control Logic

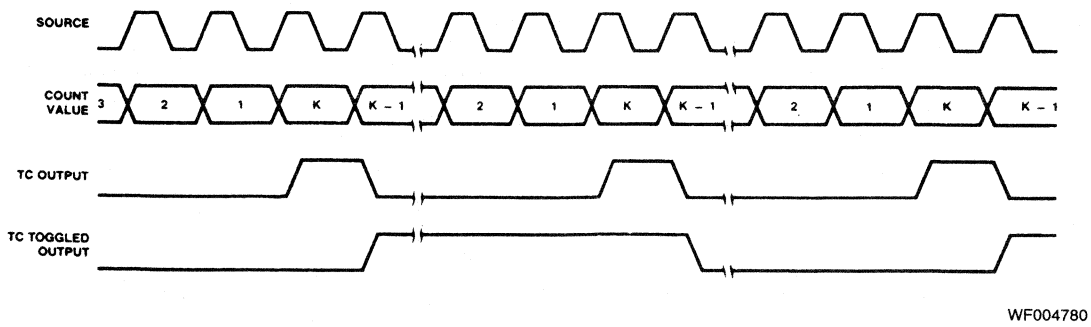


Figure 1-19. Counter Output Waveforms

The other output form, TC Toggled, uses the trailing edge of TC to toggle a flip-flop to generate an output level instead of a pulse. The toggle output is 1/2 the frequency of TC. The TC Toggled output will frequently be used to generate variable duty-cycle square waves in Operating Modes G through K.

In Mode L the TC Toggled output can be used to generate a one-shot function, with the delay to the start of the output pulse and the width of the output pulse separately programmable. With selection of the minimum delay to the start of the pulse, the output will toggle on the second source pulse following application of the triggering Gate edge.

Note that the TC Toggled output form contains no implication about whether the output is active-high or active-low. Unlike the TC output, which generates a transient pulse which can clearly be active-high or active-low, the TC Toggled output waveform only flips the state of the output on each TC. The sole criterion of whether the TC Toggled output is active-high or active-low is the level of the output at the start of the count cycle. This can be controlled by the Set and Clear Output commands. (See Figure 1-20.)

TC (Terminal Count)

On each Terminal Count (TC), the counter will reload itself from the Load or Hold register. TC is defined as that period of time when the counter contents would have been zero had no reload occurred. Some special conditions apply to counter operation immediately before and during TC.

1. In the clock cycle before TC, an internal signal is generated that commits the counter to go to TC on the next count, and retriggering by a hardware Gate edge (Modes N, O, Q and R) or a software LOAD or LOAD AND ARM command will not extend the time to TC. Note that the "next count" driving the counter to TC can be caused by the application of a count source edge (in level gating modes, the edge must occur while the gate is active, or it will be disregarded), by the application of a LOAD or LOAD AND ARM command (see 2 below) or by the application of a STEP command.
2. If a LOAD or LOAD AND ARM command is executed during the cycle preceding TC, the counter will immediately go to TC. If these commands are issued during TC, the TC state will immediately terminate.
3. When TC is active, the counter will always count the next source edge issued to it, even if it is disarmed or gated off during TC. This means that TC will never be active for longer than one count period and it may, in fact, be shorter if a STEP command or a LOAD or LOAD AND ARM command is applied during TC (see item 2 above). This also means that a counter that is disarmed or stopped on TC is actually disarmed/stopped immediately following TC.

This may cause count sequences different from what a user might expect. Since the counter is always reloaded at the start of TC and since it always counts at the end of TC, the counter contents following TC will differ by one from the reloaded value, irrespective of the operating mode used.

If the reloaded value was 0001 for down counting, 9999 (BCD) for BCD up counting or FFFF (hex) for binary up counting, the count at the end of TC will drive the counter into TC again regardless of whether the counter is gated off or disarmed. As long as these values are reloaded, the TC output will stay active. If a TC Toggled output is selected, it will toggle on each count. Execution of a LOAD, LOAD AND ARM or STEP command with these counter contents will act the same as application of a source pulse, causing TC to remain active and a TC Toggled output to toggle.

Count Control

Counter Mode bits CM3 through CM7 specify the various options available for direct control of the counting process. CM3 and CM4 operate independently of the others and control up/down and BCD/binary counting. They may be combined freely with other control bits to form many types of counting configurations. The other three bits and the Gating Control field interact in complex ways. Bit CM5 controls the repetition of the count process. When CM5 = 1, counting will proceed in the specified mode until the counter is disarmed. When CM5 = 0, the count process will proceed only until one full cycle of operation occurs. This may occur after one or two TC events. The counter is then disarmed automatically. The single or double TC requirement will depend on the state of other control bits. Note that even if the counter is automatically disarmed upon a TC, it always counts the count source edge which generates the trailing TC edge.

When TC occurs, the counter is always reloaded with a value from either the Load register of the Hold register. Bit CM6 specifies the source options for reloading the counter. When CM6 = 0, the contents of the Load register will be transferred into the counter at every occurrence of TC. When CM6 = 1, the counter reload location will be either the Load or Hold Register. The reload location in this case may be controlled externally by using a Gate pin (Modes S and V) or may alternate on each TC (Modes G through L). With alternating sources and with the TC Toggled output selected, the duty cycle of the output waveform is controlled by the relative Load and Hold values and very fine resolution of duty cycles ratios may be achieved.

Bit CM7 controls the special gating functions that allow retriggering and the selection of Load or Hold sources for counter reloading. The use and definition of CM7 will depend

on the status of the Gating Control field and bits CM5 and CM6.

Hardware Retriggering

Whenever hardware retriggering is enabled (Modes N, O, Q, and R), all active-going Gate edges initiate retrigger operations. On application of the Gate edge, the counter contents will be transferred to the Hold register. On the first qualified source edge after application of the retriggering Gate edge, the Load register contents will be transferred into the counter. (Qualified source edges are edges which occur while the counter is gated on and Armed.)

This means that, if level gating is used, the edge occurring on active-going gate transitions will initiate a retrigger. Similarly, when edge gating is enabled, an edge used to start the counter will also initiate a retrigger. The first count source edge applied after the Gate edge will not increment/decrement the counter but retrigger it.

If a LOAD, LOAD AND ARM, or a STEP Command occurs between the retriggering Gate edge and the first qualified source edge, it will be interpreted as a source edge and transfer the Load register contents into the counter. Thereafter, the counter will count all qualified source edges.

When some form of Gating is specified, CM7 controls hardware retriggering. In this case, when CM7 = 0, hardware retriggering does not occur; when CM7 = 1, the counter is retriggered any time an active-going Gate edge occurs. Retriggering causes the counter value to be saved in the Hold register and the Load register contents to be transferred into the counter.

When No Gating is specified, the definition of CM7 changes. In this case, when CM7 = 0, the Gate input has no effect on the counting; when CM7 = 1, the Gate input specifies the source (selecting either the Load or Hold register) used to reload the counter when TC occurs. Figure 1-15 shows the various available control combinations for these interrelated bits.

Count Source Selection

Counter Mode bits CM8 through CM12 specify the source used as input to the counter and the active edge that is counted. Bit CM12 controls the polarity for all the sources; logic zero counts rising edges and logic one counts falling edges. Bits CM8 through CM11 select 1 of 16 counting sources to route to the counter input. Five of the available inputs are internal frequencies derived from the internal oscillator (see Figure 1-14 for frequency assignments). Ten of the available inputs are interface pins; five are labeled SRC and five are labeled GATE.

The 16th available input is the TC output from the adjacent lower-numbered counter. (The Counter 5 TC wraps around to the Counter 1 input.) This option allows internal concatenating that permits very long counts to be accumulated. Since all five counters may be concatenated, it is possible to configure a counter that is 80-bits long on one Am9513A chip. When TCN-1 is the source, the count ripples between the connected counters. External connections can also be made, and can use the toggle bit for even longer counts. This is easily accomplished by selecting a TC Toggled output mode and wiring OUTN to one of the SRC inputs.

Gating Control

Counter Mode bits CM15, CM14, CM13 specify the hardware gating options. When "no gating" is selected (000),the counter will proceed unconditionally as long as it is armed. For any other gating mode, the count process is conditioned by the specified gating configuration.

For a code of 100 in this field, counting can proceed only when the pin labeled GATEN associated with Counter N is at a logic high level. When it goes LOW, counting is simply suspended until the Gate goes HIGH again. A code of 101 performs the same function with an opposite active polarity. Codes 010 and 011 offer the same function as 100, but specify alternate input pins as Gating Sources. This allows any of three interface pins to be used as gates for a given counter. On Counter 4, for example, pin 34, pin 35 or pin 36 may be used to perform the gating function. This also allows a single Gate pin to simultaneously control up to three counters. Counters 1 and 5 are considered adjacent when using TCN - 1 (001), Gate N +1 (010) and Gate N –1 (011) controls.

For codes of 110 or 111 in this field, counting proceeds after the specified active Gate edge until one or two TC events occur. Within this interval, the Gate input is ignored, except for the retriggering option. When repetition is selected, a cycle will be repeated as soon as another Gate edge occurs. With repetition selected, any Gate edge applied after TC goes active will start a new count cycle. Edge gating is useful when implementing a digital single-shot since the gate can serve as a convenient firing trigger.

A 001 code in this field selects the TC (not TOGGLE) output from the adjacent lower-numbered counter as the gate. This is useful for synchronous counting when adjacent counters are concatenated.

COMMAND DESCRIPTIONS

The command set for the Am9513A allows the host processor to customize and manage the operating modes and features for particular applications, to initialize and update both the internal data and control information, and to manipulate operating bits during operation. Commands are entered directly into the 8-bit Command register by writing into the Control port (see Figure 1-7).

All available commands are described in the following text. Figure 1-20 summarizes the command codes and includes a brief description of each function. Figure 1-21 shows all the unused code combinations; unused codes should not be entered into the Command register since undefined activities may occur.

Six of the command types are used for direct software control of the counting process and they each contain a 5-bit S field. In a linear-select fashion, each bit in the S field corresponds to one of five general counters (S1 = Counter 1, S2 = Counter 2, etc.). When an S bit is a one, the specified operation is performed on the counter so designated; when an S bit is a zero, no operation occurs for the corresponding counter. This type of command format has three basic advantages. It saves host software by allowing any combination of counters to be acted on by a single command. It allows simultaneous action on multiple counters where synchronization of commands is important. It allows counter-specific service routines to control individual counters without needing to be aware of the operating context of other counters.

Three of the commands use a 3-bit binary code (N4, N2, N1) to identify the affected counter (a 001 programs counter 1, etc.). Unlike the previously mentioned commands, these commands allow you to program only one counter at a time.

Command Code								Command Description
C7	C6	C5	C4	C3	C2	C1	C0	
0	0	0	E2	E1	G4	G2	G1	Load Data Pointer register with contents of E and G fields. (G ≠ 000, G ≠ 110)
0	0	1	S5	S4	S3	S2	S1	Arm counting for all selected counters
0	1	0	S5	S4	S3	S2	S1	Load contents of specified source into all selected counters
0	1	1	S5	S4	S3	S2	S1	Load and Arm all selected counters*
1	0	0	S5	S4	S3	S2	S1	Disarm and Save all selected counters
1	0	1	S5	S4	S3	S2	S1	Save all selected counters in Hold register
1	1	0	S5	S4	S3	S2	S1	Disarm all selected counters
1	1	1	0	1	N4	N2	N1	Set Toggle out (HIGH) for counter N (001 ≤ N ≤ 101)
1	1	1	0	0	N4	N2	N1	Clear Toggle out (LOW) for counter N (001 ≤ N ≤ 101)
1	1	1	1	0	N4	N2	N1	Step counter N (001 ≤ N ≤ 101)
1	1	1	0	1	0	0	0	Set MM14 (Disable Data Pointer Sequencing)
1	1	1	0	1	1	1	0	Set MM12 (Gate off FOUT)
1	1	1	0	1	1	1	1	Set MM13 (Enter 16-bit bus mode)
1	1	1	0	0	0	0	0	Clear MM14 (Enable Data Pointer Sequencing)
1	1	1	0	0	1	1	0	Clear MM12 (Gate on FOUT)
1	1	1	0	0	1	1	1	Clear MM13 (Enter 8-bit bus mode)
1	1	1	1	1	0	0	0	Enable Prefetch for Write operations (Am9513'A' only)
1	1	1	1	1	0	0	1	Disable Prefetch for Write operations (Am9513'A' only)
1	1	1	1	1	1	1	1	Master reset

*Not to be used for asynchronous operations.

Figure 1-20. Am9513A Command Summary

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	0	0	0	0
1	1	1	1	0	1	1	0
1	1	1	1	0	1	1	1
0	0	0	X	X	1	1	0
0	0	0	X	X	0	0	0
*1	1	1	1	1	X	X	X

*Unused except when XXX = 111, 001 or 000.

Figure 1-21. Am9513A Unused Command Codes

Arm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	0	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be enabled for counting. A counter must be armed before counting can commerce. Once armed, the counting process may be further enabled or disabled using the hardware gating facilities. This command can only arm or do nothing for a given counter; a zero in the S field does not disarm the counter.

ARM and DISARM commands can be used to gate counter operation on and off under software control. DISARM commands entered while a counter is in the TC state will not take effect until the counter leaves TC. This ensures that the counter never latches up in a TC state. (The counter may leave the TC state because of application of a count source edge, execution of a LOAD or LOAD AND ARM command, or execution of a STEP command.)

In modes which alternate reload sources (Modes G-L), the ARMing operation is used as a reset for the logic which determines which reload source to use on the upcoming TC. Following each ARM or LOAD AND ARM command, a counter in one of these modes will reload from the Hold register on the first TC and alternate reload sources thereafter (reload from the Load register on the second TC, the Hold register on the third, etc.).

Load Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	1	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified in the S field, will be loaded with previously entered values. The source of information for each counter will be either the associated Load register or the associated Hold register, as determined by the operating configuration in the Mode register. The Load/Hold contents are not changed. This command will cause a transfer independent of any current operating configuration for the counter. It will often be used as a software retrigger or as counter initialization prior to active hardware gating.

If a LOAD or LOAD AND ARM command is executed during the cycle preceding TC, the counter will go immediately to TC. This occurs because the LOAD operation is performed by generating a pseudo-count pulse internal to the Am9513A, and the Am9513A is expecting to go into TC on the next count pulse. The reload source used to reload the counter will be the same as that which would have been used if the TC were generated by a source edge rather than by the LOAD operation.

Execution of a LOAD or LOAD AND ARM command while the counter is in TC will cause the TC to end. For Armed counters in all modes except S or V, the LOAD source used will be that to be used for the upcoming TC. (The LOADing operation will not alter the selection of reload source for the upcoming TC.) For Disarmed counters in modes except S or V, the reload sources used will be the LOAD register. For modes S or V, the reload source will be selected by the GATE input, regardless of whether the counter is Armed or Disarmed.

Special considerations apply when modes with alternating reload sources are used (Modes G-L). If a LOAD command drives the counter to TC in these modes, the reload source for the next TC will be from the opposite reload location. In other words, the LOAD-generated TC will cause the reload sources to alternate just as a TC generated by a source edge would. Note that if a second LOAD command is issued during the LOAD-generated TC (or during any other TC, for that matter), the second LOAD command will terminate the TC and cause a reload from the source designated for use with the next TC. The second LOAD will not alter the reload source for the next TC since the second LOAD does not generate a TC; reload sources alternate on TCs only, not on LOAD commands.

Load and Arm Counters*

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	1	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified in the S field, will be first loaded and then armed. This command is equivalent to issuing a LOAD command and then an ARM command.

A LOAD AND ARM command which drives a counter to TC generates the same sequence of operations as execution of a LOAD command and then an ARM command. In modes which disarm on TC (Modes A-C and N-O, and Modes G-I and S if the current TC is the second in the cycle), the ARM part of the LOAD AND ARM command will re-enable counting for another cycle. In modes which alternate reload sources (Modes G-L), the ARMing operation will cause the next TC to reload from the HOLD register, irrespective of which reload source the current TC used.

*This command should not be used during asynchronous operations.

Disarm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be disabled from counting. A disarmed counter will cease all counting independent of other conditions. The only exception to this is that a counter in the TC state will always count once, in order to leave TC, before DISARMing. This count may be generated by a source edge, by a LOAD or LOAD AND ARM command (the LOAD AND ARM command will negate the DISARM command) or by a STEP command. A disarmed counter may be updated using the LOAD command and may be read using the SAVE command. A count process may be resumed using an ARM command. See the ARM command description for further details.

Save Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	0	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by S field, will have their contents transferred into their associated Hold register. The transfer takes place without interfering with any counting that may be underway. This command will overwrite any previous Hold register contents. The SAVE command is designed to allow an accumulated count to be preserved so that it can be read by the host CPU at some later time.

Disarm and Save Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	0	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be disarmed, and the contents of the counter will be transferred into the associated Hold registers. This command is identical to issuing a DISARM command followed by a SAVE command.

Set TC Toggle Output

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	N4	N2	N1

$(001 \leqslant N \leqslant 101)$

Description: The initial output level for TC Toggle mode is set (HIGH) for counter N selected by N4, N2, N1 = 001 (Counter 1) thru 101 (Counter 5) respectively. This command conditions the TC Toggle flip-flop (see Figure 1-18), but does not appear at the counter output unless TC Toggle mode (CM2, CM1, CM0 = 010) is selected.

Clear TC Toggle Output

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	N4	N2	N1

$(001 \leqslant N \leqslant 101)$

Description: The initial output level for TC Toggle mode is Cleared (LOW) for counter N selected by N4, N2, N1 = 001 (Counter 1) thru 101 (Counter 5) respectively. This command conditions the TC Toggle flip-flop (see Figure 1-18, but does not appear at the counter output unless TC Toggle mode (CM2, CM1, CM0 = 010) is selected.

Step Counter

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	0	N4	N2	N1

$(001 \leqslant N \leqslant 101)$

Description: Counter N is incremented or decremented by one, depending on its operating configuration. If the Counter Mode register associated with the selected counter has its CM3 bit cleared to zero, this command will cause the counter to decrement by one. If CM3 is set to a logic high, this command will increment the counter by one. The STEP command will take effect even on a disarmed counter.

Load Data Pointer Register

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	0	0	E2	E1	G4	G2	G1

(G4, G2, G1 $\neq$ 000, $\neq$ 110)

Description: Bits in the E and G fields will be transferred into the corresponding Element and Group fields of the Data Pointer register as shown in Figure 1-8. The Byte Pointer bit in the Data Pointer register is set. Transfers into the Data Pointer only occur for G field values of 001, 010, 011, 100, 101 and 111. Values of 000 and 110 for G should not be used. See the "Setting the Data Pointer Register" section of this document for additional details.

Disable Data Pointer Sequencing

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	0	0	0

Description: This command sets Master Mode bit 14 without affecting other bits in the Master Mode register. MM14 controls the automatic sequencing of the Data Pointer register. Disabling the sequencing allows repetitive host processor access to a given internal location without repetitive updating of the Data Pointer. MM14 may also be controlled by loading a full word into the Master Mode register.

Enable Data Pointer Sequencing

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	0	0	0

Description: This command clears Master Mode bit 14 without affecting other bits in the Master Mode register. MM14 controls the automatic sequencing of the Data Pointer register. Enabling the sequencing allows sequential host processor access to several internal locations without repetitive updating of the Data Pointer. MM14 may also be controlled by loading a full word into the Master Mode register. See the "Data Pointer Register" section of this document for additional information on Data Pointer sequencing

Enable 16-Bit Data Bus

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	1	1	1

Description: This command sets Master Mode bit 13 without affecting other bits in the Master Mode register. MM13 controls the multiplexer in the data bus buffer. When MM13 is set, no multiplexing takes place and all 16 external data bus lines are used to transfer information into and out of the STC. MM13 may also be controlled by loading the full Master Mode register in parallel.

Enable 8-Bit Data Bus

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	1	1	1

Description: This command clears Master Mode bit 13 without affecting other bits in the Master Mode register. MM13 controls the multiplexer in the data bus buffer. When MM13 is cleared, the multiplexer is enabled and 16-bit internal information is transferred eight bits at a time to the eight low-order external data bus lines. MM13 may also be controlled by loading the full Master Mode register in parallel.

Gate Off FOUT

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	1	1	0

Description: This command sets Master Mode bit 12 without affecting other bits in the Master Mode register. MM12 controls the output state of the FOUT signal. When gated off, the FOUT line will exhibit a low-impedance to ground. MM12 may also be controlled by loading the full Master Mode register in parallel.

Gate On FOUT

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	1	1	0

Description: This command clears Master Mode bit 12 without affecting other bits in the Master Mode register. MM12 controls the output status of the FOUT signal. When MM12 is cleared, FOUT will become active and will drive out the selected and divided FOUT signal. MM12 may also be controlled by loading the full Master Mode register in parallel. When FOUT is gated on or off, a transient pulse may be generated on the FOUT signal.

Disable Prefetch for Write Operations

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	0	0	1

Description: This command disables the prefetch circuitry during Write operations (if does not affect Read operations). This reduces the write recovery time and allows the user to use block move instructions for initialization of the Am9513A registers. Once prefetch is disabled for writing, an Enable Prefetch for Write or a Reset command is necessary to re-enable the prefetch circuitry for writing. Note: This command is only available in Am9513'A' devices; it is an illegal command in the "non-A Am9513" device.

Enable Prefetch for Write Operations

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	0	0	0

Description: This command re-enables the prefetch circuitry for Write operations. It is used only to terminate the Disable Prefetch Command. Note: This command is only available in Am9513'A' devices; it is an illegal command in the "non-A Am9513" device.

Master Reset

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	1	1	1

Description: The Master Reset command duplicates the action of the power-on reset circuitry. It disarms all counters, enters 0000 in the Master Mode, Load and Hold registers and enters 0B00 (hex) in the Counter Mode registers.

Following either a power-up or software reset, the LOAD command should be applied to all the counters to clear any that may be in a TC state. The Data Pointer register should also be set to a legal value, since reset does not initialize it. A complete reset operation follows.

1. Using the procedure given in the "Command Initiation" section of this document, enter the FF (hex) command to perform a software reset.
2. Using the "Command Initiation" procedure, enter the LOAD command for all counters, opcode 5F (hex).
3. Using the procedure given in the "Setting the Data Pointer Register" section of this document, set the Data Pointer to a valid code. The legal Data Pointer codes are given in Figure 1-9.

The Master Mode, Counter Mode, Load and Hold registers can now be initialized to the desired values.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
VCC with Respect to VSS ..... −0.5 V to +7.0 V
All Signal Voltages
with Respect to VSS ..... −0.5 V to +7.0 V
Power Dissipitation (Package Limitation) ..... 1.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

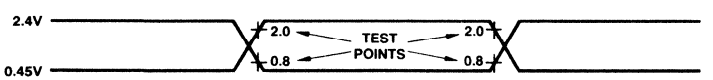
Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified.

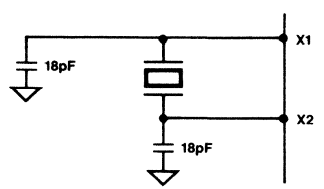
Parameters	Description		Test Conditions	Min	Max	Units
VIL	Input Low Voltage	All Inputs Except X2		VSS − 0.5	0.8	Volts
		X2 Input		VSS − 0.5	0.8	
VIH	Input High Voltage	All Input Except X2		2.2 V	VCC	Volts
		X2 Input		3.8	VCC	
VITH	Input Hysteresis (SRC and GATE Inputs Only)			0.2		Volts
VOL	Output Low Voltage		IOL = 3.2 mA		0.4	Volts
VOH	Output High Voltage		IOH = −200 μA	2.4		Volts
IIX	Input Load Current (Except X2)		VSS ≤ VIN ≤ VCC		±10	μA
IIX	Input Load Current X2		VSS ≤ VIN ≤ VCC		±100	μA
IOZ	Output Leakage Current (Except X1)		VSS + 0.4 ≤ VOUT ≤ VCC High-Impedance State		±25	μA
ICC	VCC Supply Current (Steady State)				255	mA
CIN	Input Capacitance		f = 1 MHz, T_A = +25°C. All pins not under test at 0 V.		10*	pF
COUT	Output Capacitance				15*	
CIO	IN/OUT Capacitance				20*	

* Guaranteed by design.

SWITCHING TEST INPUT/OUTPUT WAVEFORMS



WF004810



TC002000

Crystal is fundamental mode parallel resonant 32 pF load capacitance less than 100 Ω ESR C_o less than 100 pF.

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified (Note 1)

Parameters	Description		Figure	Am9513A Min	Am9513A Max	Units
TAVRL	C/$\overline{D}$ Valid to Read Low		23	25		ns
TAVWH	C/$\overline{D}$ Valid to Write High		23	170		ns
TCHCH	X2 High to X2 High (X2 Period) (Note 13)		24	145		ns
TCHCL	X2 High to X2 Low (X2 High Pulse Width) (Note 13)		24	70		ns
TCLCH	X2 Low to X2 High (X2 Low Pulse Width) (Note 13)		24	70		ns
TDVWH	Data In Valid to Write High		23	80		ns
TEHEH	Count Source High to Count Source High (Source Cycle Time) (Note 7)		24	145		ns
TEHEL TELEH	Count Source Pulse Duration (Note 7)		24	70		ns
TEHFV	Count Source High to FOUT Valid (Note 7)		24		500	ns
TEHGV	Count Source High to Gate Valid (Level Gating Hold Time) (Notes 7, 9, 10)		24	10		ns
TEHRL	Count Source High to Read Low (Set-up Time) (Notes 2, 7)		23	190		ns
TEHWH	Count Source High to Write High (Set-up Time) (Notes 3, 7)		23	−100		ns
TEHYV	Count Source High to Out Valid (Note 7)	TC Output	24		300	ns
		Immediate or Delayed Toggle Output	24		300	
		Comparator Output	24		350	
TFN	FN High to FN + 1 Valid (Note 11)		24		75	ns
TGVEH	Gate Valid to Count Source High (Level Gating Set-up Time) (Notes 7, 9, 10)		24	100		ns
TGVGV	Gate Valid to Gate Valid (Gate Pulse Duration) (Notes 8, 10)		24	145		ns
TGVWH	Gate Valid to Write High (Notes 3, 10)		23	−100		ns
TRHAX	Read High to C/$\overline{D}$ Don't Care		23	0		ns
TRHEH	Read High to Count Source High (Notes 4, 7)		23	0		ns
TRHQX	Read High to Data Out Invalid		23	10		ns
TRHQZ	Read High to Data Out at High-Impedance (Data Bus Release Time)		23		85	ns
TRHRL	Read High to Read Low (Read Recovery Time)		23	1000		ns
TRHSH	Read High to $\overline{CS}$ High (Note 12)		23	0		ns
TRHWL	Read High to Write Low (Read Recovery Time)		23	1000		ns
TRLQV	Read Low to Data Out Valid		23		110	ns
TRLQX	Read Low to Data Bus Driven (Data Bus Drive Time)		23	20		ns
TRLRH	Read Low to Read High (Read Pulse Duration) (Note 12)		23	160		ns
TSLRL	$\overline{CS}$ Low to Read Low (Note 12)		23	20		ns
TSLWH	$\overline{CS}$ Low to Write High (Note 12)		23	170		ns
TWHAX	Write High to C/$\overline{D}$ Don't Care		23	20		ns
TWHDX	Write High to Data In Don't Care		23	20		ns
TWHEH	Write High to Count Source High (Notes 5, 7, 14, 15)		23	550		ns
TWHGV	Write High to Gate Valid (Notes 5, 10, 14)		23	475		ns
TWHRL	Write High to Read Low (Write Recovery Time) (Note 16)		23	1500*		ns
TWHSH	Write High to $\overline{CS}$ High (Note 12)		23	20		ns
TWHWL	Write High to Write Low (Write Recovery Time) (Note 16)		23	1500*		ns
TWHYV	Write High to Out Valid (Notes 6, 14)		23		650	ns
TWLWH	Write Low to Write High (Write Pulse Duration) (Note 12)		23	150		ns
TGVEH2	Gate Valid to Count Source High (Special Gate) (Notes 10, 13, 17)		24	200		ns
TEHGV2	Count Source High to Gate Valid (Special Gate) (Notes 10, 13, 18)		24	80		ns

Notes:

1. Abbreviations used for the switching parameter symbols are given as the letter T followed by four or five characters. The first and third characters represent the signal names on which the measurements start and end. Signal abbreviations used are:

   A (Address) = C/$\overline{D}$
   C (Clock) = X2
   D (Data In) = DB0-DB15
   E (Enabled counter source input) = SRC1-SRC5, GATE1-GATE5, F1-F5,TCN-1
   F = FOUT
   G (Counter gate input) = GATE1-GATE5, TCN-1
   Q (Data Out) = DB0-DB15
   R (Read) = $\overline{RD}$
   S (Chip Select) = $\overline{CS}$
   W (Write) = $\overline{WR}$
   Y (Output) = OUT1-OUT5

The second and fourth letters designate the reference states of the signals named in the first and third letters respectively, using the following abbreviations.

H = HIGH
L = LOW
V = VALID
X = Unknown or Don't care
Z = High-Impedance

2. Any input transition that occurs before this minimum setup requirement will be reflected in the contents read from the status register.

3. Any input transition that occurs before this minimum setup requirement will act on the counter before the execution of the operation initiated by the write and the counter may be off by one count.

4. Any input transition that occurs after this minimum hold time is guaranteed to not influence the contents read from the status register on the current read operation.

5. Any input transition that occurs after this minimum hold time is guaranteed to be seen by the counter as occurring after the action initiated by the write operation and the counter may be off by one count.

6. This parameter applies to cases where the write operation causes a change in the output bit.

7. The enabled count source is one of F1-F5, TCN-1 SRC1-SRC5 or GATE1-GATE 5, as selected in the applicable Counter Mode register. The timing diagram assumes the counter counts on rising source edges. The timing specifications are the same for falling-edge counting.

8. This parameter applies to edge gating (CM15-CM13 = 110 or 111) and gating when both CM7 = 1 and CM15-CM13 ≠ 000. This parameter represents the minimum GATE pulse width needed to ensure that the pulse initiates counting or counter reloading.

9. This parameter applies to both edge and level gating (CM15-CM13 = 001 through 111 and CM7 = 0). This parameter represents the minimum setup or hold times to ensure that the Gate input is seen at the intended level on the active source edge and the counter may be off by one count.

10. This parameter assumes that the GATENA input is unused (16-bit bus mode) or is tied high. In cases where the GATENA input is used, this timing specification must be met by both the GATE and GATENA inputs.

11. Signals F1-F5 cannot be directly monitored by the user. The phase difference between these signals will manifest itself by causing counters using two different F signals to count at different times on nominally simultaneous transitions in the F signals. F1 = X2.

12. This timing specification assumes that $\overline{CS}$ is active whenever $\overline{RD}$ or $\overline{WR}$ are active. $\overline{CS}$ may be held active indefinitely.

13. This parameter assumes X2 is driven from an external gate with a square wave.

14. This parameter assumes that the write operation is to the command register.

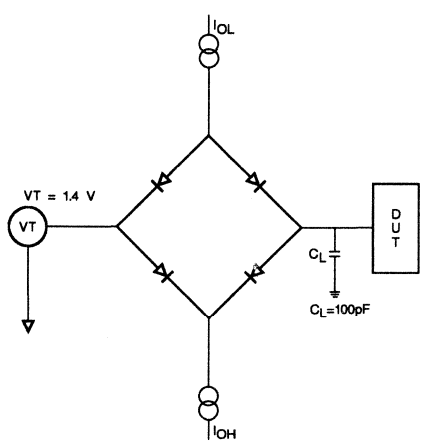
15. This timing specification applies to single-action commands only (e.g., LOAD, ARM, SAVE, etc.). For double-action commands such as LOAD AND ARM and DISARM AND SAVE, TWHEH minimum = 700 ns.

16. In short data write mode, TWHRL and TWHWL minimum = 1000 ns.

17. This parameter applies to the hardware retrigger/save modes N, O, Q, R, and X (CM7 = 1 and CM15 — CM13 < > 000). This parameter ensures that the gating pulse initiates a hardware retrigger/save operation.

18. This parameter applies to hardware load source select modes S and V (CM7 = 1 and CM15 — CM13 = 000). This parameter represents the minimum hold time to ensure that the GATE input selects the correct load source on the active source edge.

SWITCHING TEST CIRCUIT

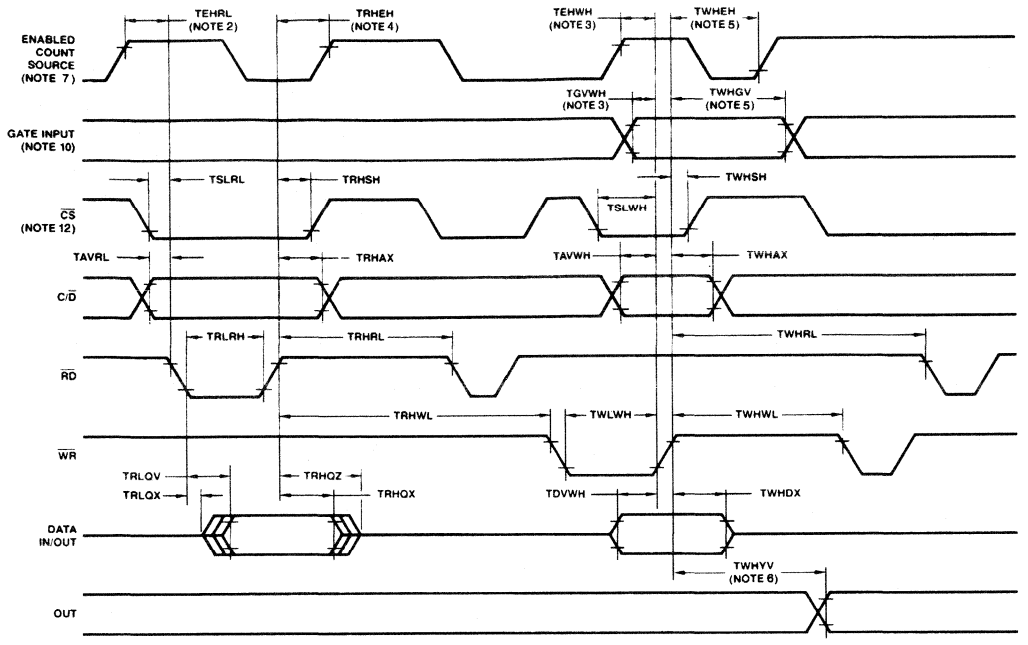


TC003852

This test circuit is the dynamic load of a Teradyne J941.

SWITCHING WAVEFORMS

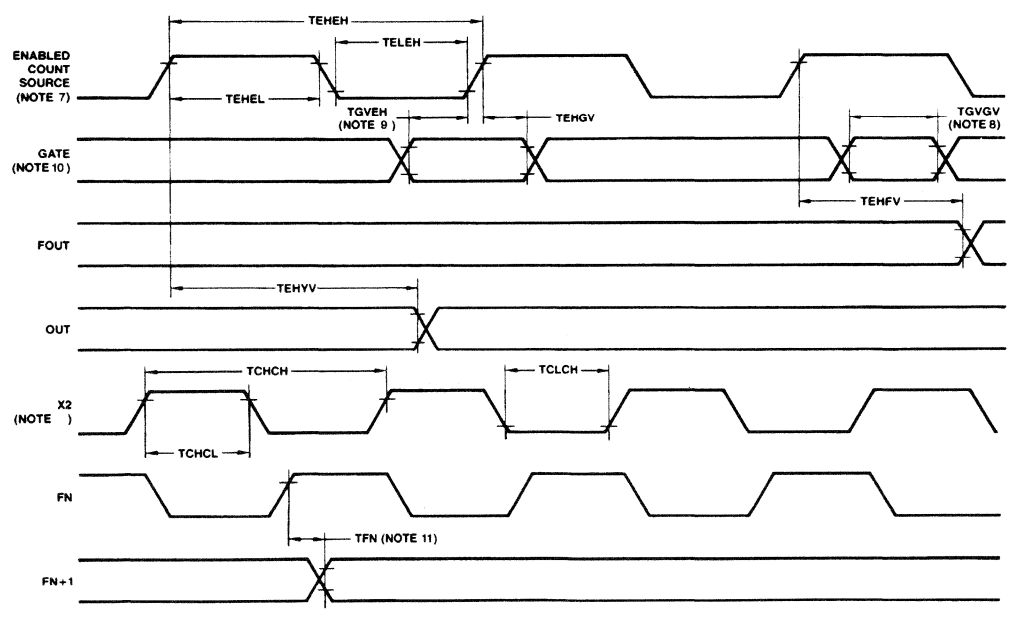
BUS TRANSFER SWITCHING WAVEFORMS



WF004791

Figure 1-23.

COUNTER SWITCHING WAVEFORMS



WF004801

Figure 1-24.

APPENDIX A

Design Hints

1) When a crystal is not being used, X_1 and X_2 should be connected as shown for TTL input (Figure 1) and no input (Figure 2).

2) Recommended oscillator capacitor values are 18 pF on X_1 and X_2.

3) Unused inputs should be tied to V_{CC}.

4) The TC output can glitch when the counter is loaded. For this reason this output should not be connected to edge sensitive interrupts. The counter output should be set or cleared after the LOAD command.

5. The two most significant bits of the status register are not specified. They may be zero or one.

6. The mode register should not be modified when the counter is armed.

7. The LOAD and HOLD registers should not be changed during TC.

8. When using the different clocks for different counters be aware that there is a 75 ns skew between F1, F2, F3, F4 and F5.

9. The TC output will remain inactive if programmed to be in the TC TOGGLE mode and the step command is used to increment or decrement the counter. The output will go into TC if programmed to be in the active High or active Low terminal count modes. The only two ways out of TC in this case are:

   —Arming the counter and having an active source connected to it.

   —Issuing another step command.

10) Timing parameters TEHWH and TGVWH are specified as negative. The diagrams in Figure 3 show the relationship between these signals.

11) In mode X the counter will count all qualified source edges until the second (not the first) TC and then stop.

12) A TC can occur when the counters are loaded if the counter was stopped at $FFFF_H$ or 9999_{10} in the count up mode or at count 0001 when counting down. This is because an internal TC is generated which forces TC to be generated on the next count pulse.

13) In modes that alternate the reload source between the load and the hold registers (e.g., mode J), if the counter is disarmed at 0001_H for down counting or 9999_{10} for BCD up counting or $FFFF_H$ for binary up counting and rearmed, the reload source after the first TC will be the load register instead of the hold register. To avoid this, issue a software "dummy" load to the counter immediately after the disarm command.

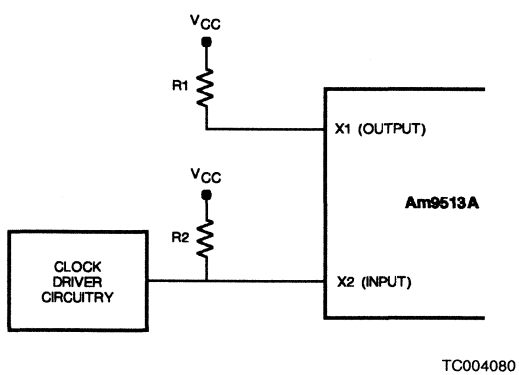
14) In the down counting mode of the Am9513A, if a 0001 is loaded into the counter and another LOAD COUNTER command is issued, the TC of that counter will go active. If the load register contents are subsequently changed, and the counter armed, the first clock edge will cause the new load register contents to transfer into the counter and the next clock edge will decrement the counter and make it go out of TC.

15) Glitches on $\overline{CS}$ just before the $\overline{RD}$ or $\overline{WR}$ pulse may cause the part to behave incorrectly.

16) Timing parameters TGVEH & TEHGV must not be violated; Figure 4 shows a method.

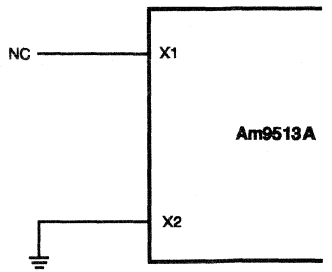
Troubleshooting (Symptom: Solution)

1) Registers not being programmed correctly: Check READ or WRITE recovery time.
2) Setup and hold problems observed in synchronous systems: Try switching from positive edge to negative edge triggering.



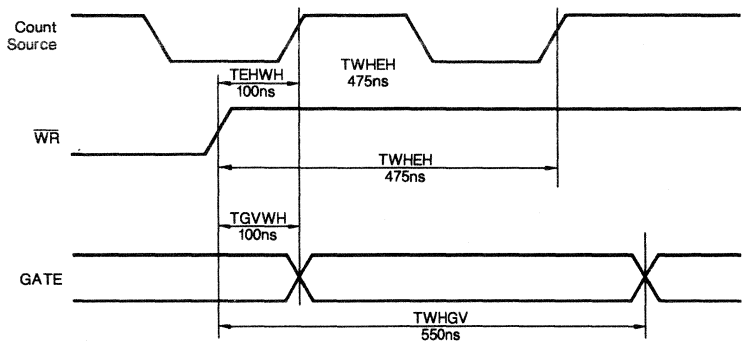
R1 = 6.8 kΩ ±10%
R2 = Function of Driver Circuitry
X1 V_{IH} = 3.8 V
X2 V_{IL} = 0.8 V

Figure A1. Crystal Input Configuration



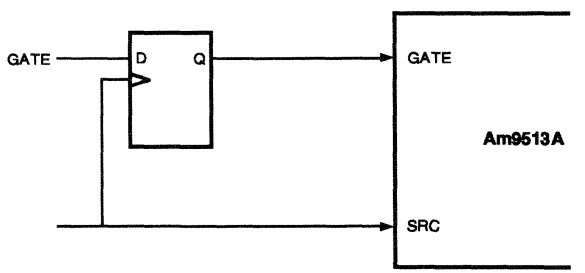
TC004090

Figure A2. Crystal Input Configuration



WF023981

Figure A3. TEHWH/TGVWH Timing Diagram



TC004100

Figure A4. GATE/SRC Configuration Suggestion

Am9516A

Universal DMA Controller (UDC)

DISTINCTIVE CHARACTERISTICS

- Transfer Modes: Single, demand dedicated with bus hold, demand dedicated with bus release, demand interleave
- 16 Megabyte physical addressing range
- Automatic loading/reloading of control parameters by each channel
- Optional automatic chaining of operations
- Channel interleave operations
- Interleave operations with system bus
- Masked data pattern matching for search operations
- Vectored interrupts on selected transfer conditions
- Software DMA request
- Software or hardware wait state insertion
- Transfer up to 6.66 Mbytes/second at 10MHz clock

BLOCK DIAGRAM

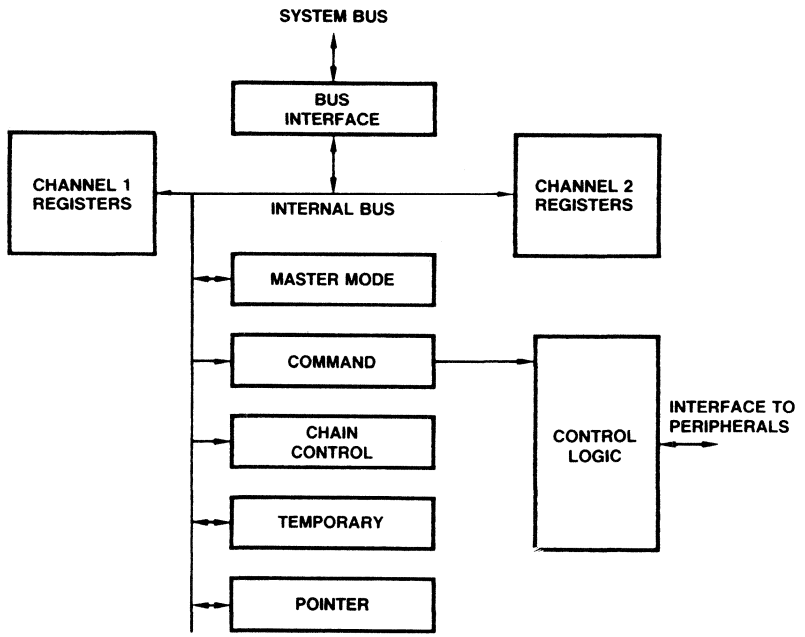


Figure 1.

Publication # 03242 Rev. D Amendment /0
Issue Date: April 1987

GENERAL DESCRIPTION

The Am9516A Universal DMA Controller (UDC) is a high performance peripheral interface circuit for 8086 and 68000 CPUs. In addition to providing data block transfer capability between memory and peripherals, each of the UDC's two channels can perform peripheral-to-peripheral as well as memory-to-memory transfer. A special Search Mode of Operation compares data read from a memory or peripheral source to the content of a pattern register.

For all DMA operations (search, transfer, and transfer-and-search), the UDC can operate with either byte or word data sizes. In some system configurations it may be necessary to transfer between word-organized memory and a byte-oriented peripheral. The UDC provides a byte packing/unpacking capability through its byte-word funneling transfer or transfer-and-search option. Some DMA applications may continuously transfer data between the same two memory areas. These applications may not require the flexibility inherent in reloading registers from memory tables. To service these repetitive DMA operations, base registers are provided on each channel which re-initialize the current source and destination Address and Operation Count registers. To change the data transfer direction under CPU control, provision is made for reassigning the source address as a destination and the destination as a source, eliminating the need for actual reloading of these address registers.

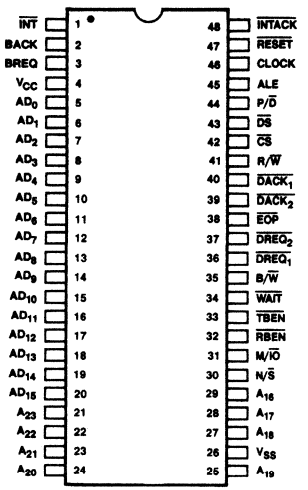
Frequently DMA devices must interface to slow peripherals or slow memory. In addition to providing a hardware $\overline{\text{WAIT}}$ input, the Am9516A UDC allows the user to select independently for both source and destination addresses and automatic insertion of 0, 1, 2 or 4 wait states. The user may even disable the $\overline{\text{WAIT}}$ input pin function altogether and use these software programmed wait states exclusively.

High throughput and powerful transfer options are of limited usefulness if a DMA requires frequent reloading by the host CPU. The Am9516A UDC minimizes CPU interactions by allowing each channel to load its control parameters from memory into the channel's control registers. The only action required of the CPU is to load the address of the control parameter table into the channel's Chain Address register and then issue a "Start Chain" Command to start the register loading operation.

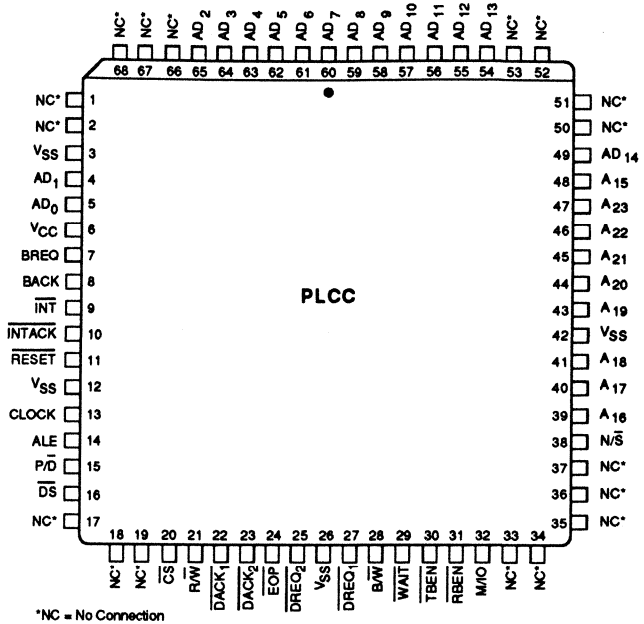
The Am9516A UDC is packaged in a 48-pin DIP and uses a single +5V Power Supply.

CONNECTION DIAGRAMS
Top View

DIPs



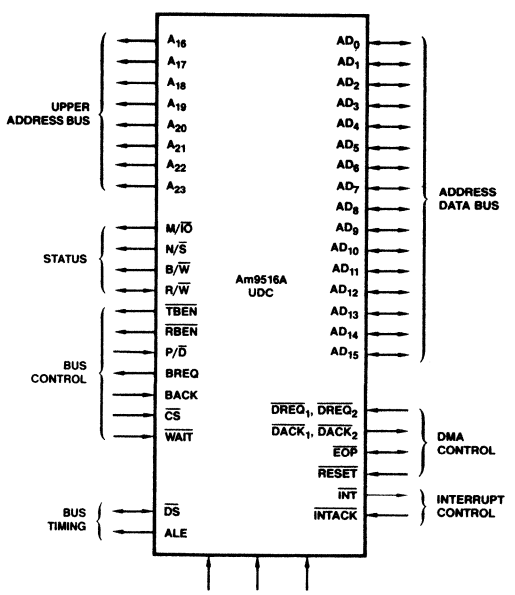
CD005592



*NC = No Connection

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL

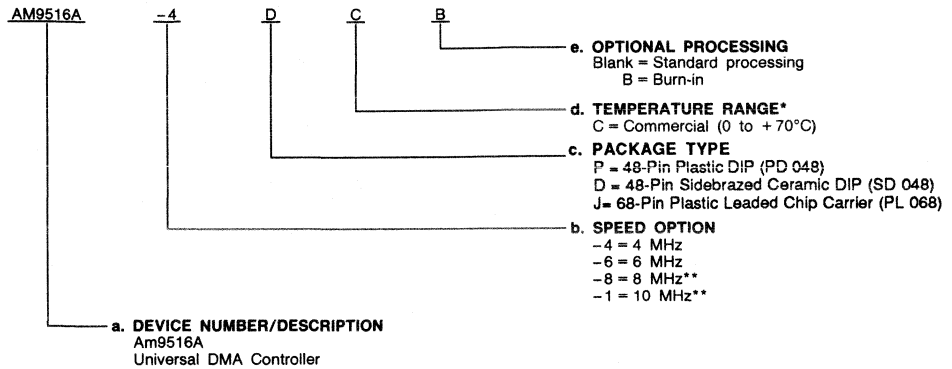


LS001331

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM9516A-4	DC,DCB,PC,JC
AM9516A-6	
AM9516A-8	
AM9516A-1	DC

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

**Preliminary; to be announced.

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
4	V_{CC}		+5V Power Supply.
26	V_{SS}		Ground.
46	CLOCK	I	Clock.
46	CLOCK	I	(Clock). The Clock signal controls the internal operations and the rates of data transfers. It is usually derived from a master system clock or the associated CPU clock. The Clock input requires a high voltage input signal. Many UDC input signals can make transitions independent of the UDC clock; these signals can be asynchronous to the UDC clock. On other signals, such as $\overline{WAIT}$ inputs, transitions must meet setup and hold requirements relative to the UDC clock. See the Timing diagrams for details.
5-20	$AD_0 - AD_{15}$	I/O	(Address-Data Bus, Three-State). The Address Data Bus is a time-mulitiplexed, bidirectional, active-high, three-state bus used for all I/O and memory transactions. HIGH on the bus corresponds to 1 and LOW corresponds to 0. AD_0 is the least significant bit position and AD_{15} is the most significant. The presence of addresses is defined by the timing edge of ALE, and the asserted or requested presence of data is defined by the $\overline{DS}$ signal. The status output lines M/$\overline{IO}$ and N/$\overline{S}$ indicate the type of transaction, either memory or I/O. The R/$\overline{W}$ line indicates the direction of the transaction. When the UDC is in control of the system bus, it dominates the AD Bus; when the UDC is not in control of the system bus, the CPU or other external devices dominate the AD Bus. The presence of address of data on the $AD_0 - AD_{15}$ bus is defined only by ALE and $\overline{DS}$. When the UDC is not in control of the bus, there is no required relation between the presence of address or data and the UDC clock. This allows the UDC to be used with a system bus which does not have a bussed clock signal.
43	$\overline{DS}$	I/O	(Data Strobe, Three-State). Data Strobe is a bidirectional, active-low, three-state signal. A LOW on this signal indicates that the AD_0-AD_{15} bus is being used for data transfer. When the UDC is not in control of the system bus and the external system is transferring information to or from the UDC, $\overline{DS}$ is a timing input used by the UDC to move data to or from the $AD_0 - AD_{15}$ bus. Data is written into the UDC by the external ststem on the LOW-to-HIGH $\overline{DS}$ transition. Data is read from the UDC by the external system while $\overline{DS}$ is LOW. There are no timing requirements between $\overline{DS}$ as an input and the UDC clock; this allows use of the UDC with a system bus which does not have a bussed clock. During a DMA operation when the UDC is in control of the ststem, $\overline{DS}$ is an output generated by the UDC and used by the system to move data to or from the $AD_0 - AD_{15}$ bus. When the UDC has bus control, it writes to the external system by placing data on the $AD_0 - AD_{15}$ bus before the HIGH-to LOW $\overline{DS}$ transition and holding the data stable until after the LOW-to-HIGH $\overline{DS}$ transition; while reading from the external system, the LOW-to-HIGH transition of $\overline{DS}$ inputs data from the $AD_0 - AD_{15}$ bus into the UDC (see Timing diagram).
41	R/$\overline{W}$	I/O	(Read/Write, Three-State). Read/Write is a bidirectional, three-state signal. Read polarity is HIGH and write polarity is LOW. R/$\overline{W}$ indicates the data direction of the current bus transaction, and is stable starting when ALE is HIGH until the bus transaction ends (see Timing diagram). When the UDC is not in control of the system bus and the external system is transferring information to or from the UDC, R/$\overline{W}$ is a status input used by the UDC to determine if data is entering or leaving on the $AD_0 - AD_{15}$ bus during $\overline{DS}$ time. In such a case, Read (HIGH) indicates that the system is requesting data from the UDC, and Write (LOW) indicates that the system is presenting data to the UDC. There are no timing requirements between R/$\overline{W}$ as an input and the UDC clock; transitions on R/$\overline{W}$ as an input are only defined relative to $\overline{DS}$. When the UDC is in control of the system bus, R/$\overline{W}$ is an output generated by the UDC, with Read indicating that data is being requested from the addressed location or device. the addressed location or device and Write indicating that data is being presented to the addressed location or device. Flyby DMA operations are a special case where R/$\overline{W}$ is valid for the normally addressed memory or peripheral locations and must be interpreted in reverse by the "Flyby" peripheral that uses it.
33	$\overline{TBEN}$	O	(Transmit Buffer Enable, Open Drain). Transmit Buffer Enable is an active-low, open drain output. When UDC is a bus master, a LOW on this output indicates that the data is being transferred from the UDC to the data bus lines through the buffer. The purpose of this signal is to eliminate bus contention. When UDC is not in control of the system bus, these pins float to three-state OFF.
32	$\overline{RBEN}$	O	(Receive Buffer Enable, Open Drain). Receive Buffer Enable is an active-low, open drain output. When UDC is in control of system bus, a LOW on this output indicates that the data is being transferred from the data bus lines to the UDC through the buffer. The purpose of this signal is to eliminate bus contention. This pin floats to three-state OFF when the UDC is not in control of the system bus.
45	ALE	O	(Address Latch Enable). This active HIGH signal is provided by the UDC to latch the address signals $AD_0 - AD_{15}$ into the address latch. This pin is never floated.
44	P/$\overline{D}$	I	(Pointer/Data). Pointer/Data is an input signal to indicate the information is on the $AD_0 - AD_{15}$ bus only when the UDC is the bus slave. A HIGH on this signal indicates the information is on the AD bus is an address of the internal register to be accessed. The data on the AD bus is loaded into the Pointer register of UDC. A LOW on this signal indicates that a data transfer is taking place between the bus and the internal register designated by the Pointer register. Note that if a transaction is carried out with R/$\overline{W}$ HIGH and P/$\overline{D}$ HIGH, the contents of the Pointer register will be read.
31	M/$\overline{IO}$	O	(Memory/Input-Output, Three-State). This signal specifies the type of transaction. A HIGH on this pin indicates a memory transaction. A LOW on this pin indicates an I/O transaction. It floats to three-state OFF when UDC is not in control of the ststem bus.
30	N/$\overline{S}$	O	(Normal/System, Three-State). This output is a three-state signal activated only when the UDC is the bus master. Normal is indicated when N/$\overline{S}$ is HIGH, and System is indicated when N/$\overline{S}$ is LOW. This signal supplements the M/$\overline{IO}$ line and is used to indicate which memory or I/O space is being accessed.
35	B/$\overline{W}$	O	(Byte/Word, Three-State). This output indicates the size of data transferred on the $AD_0 - AD_{15}$ bus. HIGH indicates a byte (8-bit) and LOW indicates a word (16-bit) transfer. This output is activated when ALE is HIGH and remains valid for the duration of the whole transaction (see Timing diagram). All word-sized data are word-aligned and must be addressed by even addresses ($A_0 = 0$). When addressing byte read transactions, the least significant address bit determines which byte is needed; an even address specifies the most significant byte ($AD_8 - AD_{15}$), and an odd address specifies the least significant byte ($AD_0 - AD_7$). (Note that the higher address specifies the least significant byte!) This addressing mechanism applies to memory accesses as well as I/O accesses. When the UDC is a slave, it ignores the B/$\overline{W}$ signal and this pin floats to three-state OFF.

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
42	$\overline{CS}$	I	(Chip Select). This pin is an active-low input. A CPU or other external device uses $\overline{CS}$ to activate the UDC for reading and writing of its internal registers. There are no timing requirements between the $\overline{CS}$ input and the UDC clock; the $\overline{CS}$ input timing requirements are only defined relative to $\overline{DS}$. This pin is ignored when UDC is in control of system bus.
34	$\overline{WAIT}$	I	(WAIT). This pin is an active-low input. Slow memories and peripheral devices may use $\overline{WAIT}$ to extend $\overline{DS}$ and $\overline{RBEN}$ or $\overline{TBEN}$ during operation. Unlike the $\overline{CS}$ input, transitions on the $\overline{WAIT}$ input must meet certain timing requirements relative to the UDC clock. See Timing Diagram 4 for details. The Wait function may be disabled using a control bit in the Master Mode register (MM2).
3	BREQ	O	(Bus Request). Bus Request is an active-HIGH signalsignal used by the UDC to obtain control of the bus from the CPU. BREQ lines from multiple devices are connected to a priority encoder.
2	BACK	I	(Bus Acknowledge). BACK is an active-HIGH, asynchronchronous input, indicating that the CPU has relinquished the bus and that no higher priority device has assumed bus control. Since BACK is internally synchronized by the UDC before being used, transitions on BACK do not have to be synchronous with the UDC clock. The BACK input is usually connected to the HLDA line from the CPU or to the output of a priority decoder.
1	$\overline{INT}$	O	(Interrupt Request, Open Drain). Interrupt Request is an active-low output used to interrupt the CPU. It is driven LOW whenever the IP and CIE bits of the Status Register are set. It is cleared by UDC after receiving a clear IP command.
48	$\overline{INTACK}$	I	(Interrupt Acknowledge). Interrupt Acknowledge is an active-low input indicating that the request for interrupt has been granted. The UDC will place a vector onto the AD bus if the No Vector or Interrupt bit (MM3) is reset.
47	$\overline{RESET}$	I	(Reset). Reset is an active-low input to disable the UDC and clear its Master Mode register.
36, 37	$\overline{DREQ_1}$, $\overline{DREQ_2}$	I	(DMA Request). The DMA Request lines are two active-low inputs, one per channel. They may make transitions independent of the UDC clock and are used by external logic to initiate and control DMA operations performed by the UDC.
40, 39	$\overline{DACK_1}$, $\overline{DACK_2}$	O	(DMA Acknowledge). The DMA Acknowledge lines are active-low outputs, one per channel, which indicate that the channel is performing a DMA operation. $\overline{DACK}$ is pulsed, held active or held inactive during DMA operations as programmed in the Channel Mode register. For Flowthru operations, the peripheral is fully addressed using the conventional I/O addressing protocols and therefore may choose to ignore $\overline{DACK}$. $\overline{DACK}$ is always output as programmed in the Channel Mode register for a DMA operation, even when the operation is intiated by a CPU software request command or as a result of chaining. $\overline{DACK}$ is not output during the chaining operations.
38	$\overline{EOP}$	I/O	(End of Process). $\overline{EOP}$ is an active-low, open-drain, bidirectional signal. It must be pulled up with an external resistor of 1.8kohm or more. The UDC emits an output pulse on $\overline{EOP}$ when a TC or MC termination occurs, as defined later. An external source may terminate a DMA operation in progress by driving $\overline{EOP}$ low. $\overline{EOP}$ always applies to the active channel; if no channel is active, $\overline{EOP}$ is ignored.
29-27 25-21	$A_{16}-A_{23}$	O	(Upper Address Bus, Three-state). The $A_{16}-A_{23}$ address lines are three-state outputs activated only when the UDC is controlling the system bus. Combined with the lower 16 address bits appearing on AD_0 through AD_{15} respectively, this 24-bit linear address allows the UDC to access anywhere within 16 Megabytes of memory.

Note: All inputs to the UDC, except the clock are directly TTL compatible.

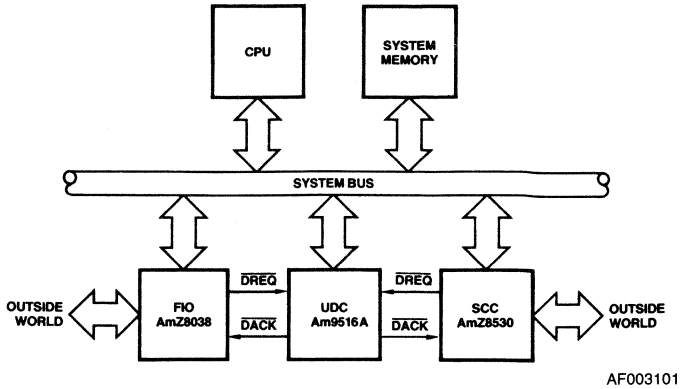


Figure 2. UDC Configurations

PRODUCT OVERVIEW

Register Description

The Am9516A UDC block diagram illustrates the internal registers. Figure 3 lists each register along with its size and read/write access restrictions. Registers which can be read by the CPU are either fast (F) or slow (S) readable. Fast registers can be read by a normal CPU I/O operation without additional wait states. Reading slow registers requires multiple wait states. Registers can be written to by the host CPU (W) and/or can be loaded by the DMA channel itself during chaining (C). All reads or writes must be word accesses since the UDC ignores the B/$\overline{W}$ line in slave mode. It is the responsibility of the user to supply the necessary external logic if slow readable registers are to be read.

The UDC registers can be categorized into chip-level registers, which control the overall operation and configuration of the UDC, and channel-level registers which are duplicated for each channel. The five chip-level registers are the Master Mode register, the Command register, the Chain Control register, the Pointer register, and the Temporary register. The Master Mode register selects the way the UDC chip interfaces to the system. The Command register is written to by the host CPU to initiate certain operations within the UDC chip, such as resetting the unit. The Chain Control register is used by a channel while it is reloading its channel-level registers from memory. The Pointer register is written to by the host CPU when the P/$\overline{D}$ input is HIGH. The data in Pointer register is the address of the internal register to be accessed. The Temporary register is used to hold data for Flowthru Transfer/Transfer-and-Searches.

The channel-level registers can be divided into two subcategories: general purpose registers, which would be found on most DMA chips, and special purpose registers, which provide additional features and functionality. The general purpose registers are the Base and Current Operation Count registers, the Base and Current Address registers A and B, and the Channel Mode register. The special purpose registers are the Pattern and Mask registers, the Status register, the Interrupt Vector register, the Interrupt Save register, and the Chain Address register.

The internal registers are read or written in two steps. First, the address of the register to be accessed is written to the Pointer register, when the P/$\overline{D}$ input is HIGH. Then, the data is read from or written into the desired register, which is indicated by the Pointer register, when P/$\overline{D}$ input is LOW. Note that a read with P/$\overline{D}$ HIGH causes the contents of the Pointer register to be read on AD_1 through AD_6.

Master Mode Register

The 4-bit Master Mode register, shown in Figure 4, controls the chip-level interfaces. It can be read from and written to by the host CPU without wait states through pins $AD_0 - AD_3$, but it is not loadable by chaining. On a reset, the Master Mode register is cleared to all zeroes. The function of each of the Master Mode bits is described in the following paragraphs.

The Chip Enable bit CE = 1 enables the UDC to request the bus. When enabled, the UDC can perform DMA Operations and reload registers. It can always issue interrupts and respond to interrupt acknowledges. When the Chip Enable bit is cleared, the UDC is inhibited from requesting control of the system bus and, therefore, inhibited from performing chaining or DMA operations.

The CPU Interleave bit enables interleaving between the CPU and the UDC.

The Wait Line Enable bit is used to enable sampling of the $\overline{WAIT}$ line during Memory and I/O transactions. Because the UDC provides the ability to insert software programmable wait states, many users may disable sampling of the $\overline{WAIT}$ pin to eliminate the logic driving this pin. The Wait Line Enable bit provides this flexibility. See the "Wait States" section of this document for details on wait state insertion.

The "No Vector on Interrupt" bit selects whether the UDC channel or a peripheral returns a vector during interrupt acknowledge cycles. When this bit is cleared, a channel receiving an interrupt acknowledge will drive the contents of its Interrupt Save register onto the $AD_0 - AD_{15}$ data bus while $\overline{INTACK}$ is LOW. If this bit is set, interrupts are serviced in an identical manner, but the $AD_0 - AD_{15}$ data bus remains in a high-impedance state throughout the acknowledge cycle.

Pointer Register

The Pointer register contains the address of the internal register to be accessed. It can be read from or written to by the CPU when the P/$\overline{D}$ line is HIGH.

Name	Size	Number	Access Type	Port Address CH-1/CH-2
Master Mode Register	4 bits	1	FW	38
Pointer Register	6 bits	1	FW	
Chain-Control Register	10 bits	1	C	
Temporary Register	16 bits	1	D	
Command Register	8 bits	1	W	2E/2C*
Current Address Register – A:				
Up-Addr/Tag field	14 bits	2	CFW	1A/18
Lower Address field	16 bits	2	CFW	0A/08
Current Address Register – B:				
Up-Addr/Tag field	14 bits	2	CFW	12/10
Lower Address field	16 bits	2	CFW	02/00
Base Address Register – A:				
Up-Addr/Tag field	14 bits	2	CFW	1E/1C
Lower Address field	16 bits	2	CFW	0E/0C
Base Address Register – B:				
Up-Addr/Tag field	14 bits	2	CFW	16/14
Lower Address field	16 bits	2	CFW	06/04
Current Operation Count	16 bits	2	CFW	32/30
Base Operation Count	16 bits	2	CFW	36/34
Pattern Register	16 bits	2	CSW	4A/48
Mask Register	16 bits	2	CSW	4E/4C
Status Register	16 bits	2	F	2E/2C
Interrupt Save Register	16 bits	2	F	2A/28
Interrupt Vector Register	8 bits	2	CSW	5A/58
Channel Mode Register – HIGH	5 bits	2	CS	56/54
Channel Mode Register – LOW	16 bits	2	CSW	52/50
Chain Address Register:				
Up-Addr/Tag field	10 bits	2	CFW	26/24
Lower Address field	16 bits	2	CFW	22/20

Access Codes: C = Chain Loadable
D = Accessible by UDC channel
F = Fast Readable
S = Slow Readable
W = Writable by CPU

Note: The address of the register to be accessed is stored in the Pointer register.

*Port addresses of the Command register can be used alternately for both channels except when issuing a "set or clear IP" command.

Figure 3. UDC Internal Register

Chain Control Register

When a channel starts a chaining operation, it fetches a Reload word from the memory location pointed to by the Chain Address register (Figure 11). This word is then stored in the Chain Control register. The Chain Control register cannot be written to or read from by the CPU. Once a channel starts a chain operation, the channel will not relinquish bus control until all registers specified in the Reload word are reloaded unless an $\overline{EOP}$ signal is issued to the chip. Issuing an $\overline{EOP}$ to a channel during chaining will prevent the chain operation from resuming and the contents of the Reload Word register can be discarded.

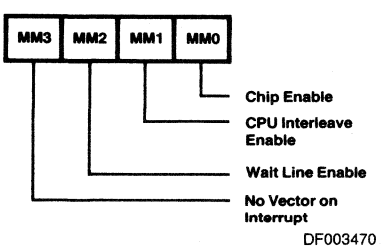


Figure 4. Master Mode Register

Temporary Register

The Temporary register is used to stage data during Flowthru transfers and to hold data being compared during a Search or a Transfer-and-Search. The temporary register cannot be written to or read from by the CPU. In byte-word funneling, data may be loaded into or from the Temporary register on a byte-by-byte basis, with bytes sometimes moving between the low byte of the data bus and the high byte of the Temporary register or vice-versa. See the "Transfer" section for details.

Command Register

The UDC Command register (Figure 20) is an 8-bit write-only register written to by the host CPU. The Command register is loaded from the data on $AD_7 - AD_0$; the data on $AD_{15} - AD_8$ is disregarded. A complete discussion of the commands is given in the "Command Descriptions" section.

Current and Base Address Registers A and B

The Current Address registers A and B (Current ARA and ARB) are used to point to the source and destination addresses for DMA operations. The contents of the Base ARA and ARB registers are loaded into the Current ARA and ARB registers at the end of a DMA operation if the user enables Base-to-Current reloading in the Completion Field of the Channel Mode register. This facilitates DMA operations without reloading of the Current registers. The ARA and ARB registers can be loaded during chaining, can be written to by the host CPU without wait states and can be read by the CPU.

Each of the Base and Current ARA and ARB registers consists of two words organized as a 6-bit Tag Field and an 8-bit Upper Address in one word and a 16-bit Lower Address in the other. See Figure 5. The Tag Field selects whether the address is to be incremented, decremented or left unchanged, and the status codes associated with the address. The Tag field also allows the user to insert 0, 1, 2 or 4 wait states into memory or I/O accesses addressed by the offset and segment fields.

The Address Reference Select Field in the Tag field selects whether the address pertains to memory space or I/O space. Note that the N/$\overline{S}$ output pin may be either HIGH (indicating Normal) or LOW (indicating System) for space. At the end of each iteration of a DMA Operation, the user may select to leave the address unchanged or to increment it or to decrement it. I/O addresses, if changed, are always incremented/decremented by 2. Memory addresses are changed by 1 if the address points to a byte operand (as programmed in the Channel Mode register's Operation field) and by 2 if the address points to a word operand. Note that, if an I/O or memory address is used to point to a word operand, the address must be even to avoid unpredictable results. An address used to point to a byte operand may be even or odd. Since memory byte operand addresses will increment/decrement by 1, they will toggle between even and odd values. Since I/O byte operand addresses will increment/decrement by 2, once programmed to an even or odd value, they will remain even or odd, allowing consecutive I/O operations to access the same half of the data bus. High bus is for even address and low bus for odd.

Current and Base Operation Count Registers

Both the Current and Base Operation Count registers may be loaded during chaining, and may be written to and read from by the host CPU.

The 16-bit Current Operation Count register is used to specify the number of words or bytes to be transferred, searched or transferred-and-searched. For word-to-word operations and byte-word funneling, the Current Operation Count register must be programmed with the number of words to be transferred or searched.

Each time a datum is transferred or searched, the Operation Count register is decremented by 1. Once all of the data is transferred or searched, the transfer or search operation will stop, the Current Operation Count register will contain all zeroes, and the TC bit in Status Register will be "1." If the transfer or search stops before the Current Operation Count register reaches 0, the contents of the register will indicate the number of bytes or words remaining to be transferred or searched. This allows a channel which had been stopped prematurely to be restarted where it left off without requiring reloading of the Current Operation Count register.

For the byte-to-byte operations, the Current Operation Count register should specify the number of bytes to be transferred or searched. The maximum number of bytes which can be specified is 64K bytes by setting the Current Operation Count register to 0000.

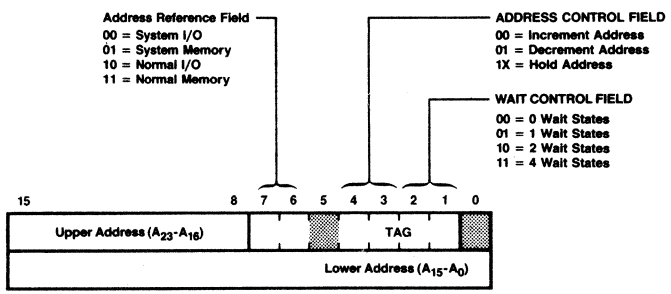


Figure 5. Address Registers A and B

Pattern and Mask Registers

The 16-bit Pattern and Mask registers are used in Search and Transfer-and-Search operations. Both the Pattern and Mask registers may be loaded by chaining, may be written to by the host CPU, and may be read from by the host CPU, provided wait states are inserted, since these registers are slow readable. The Pattern register contains the pattern that the read data is compared to. Setting a Mask register bit to "1" specifies that the bit always matches. See the "Search" and "Transfer-and-Search" sections for further details.

Status Register

The two 16-bit Status registers, depicted in Figure 6, are read-only registers which can be read by the CPU without wait states. Each of these registers reports on the status of its associated channel.

The Interrupt Status Field in the Status register contains the Channel Interrupt Enable (CIE) and Interrupt Pending (IP) bits. These bits are described in detail in the "Interrupt" section of this document.

The UDC status field contains the current channel status. The "channel initialized and waiting for request" status is not explicitly stated – it is reflected by Status register bits ST_{12} through ST_9 being all zero. The "Waiting for Bus" (WFB) status will cause bit ST_{10} to be set and indicates that the channel wants bus control to perform a DMA operation. The channel may or may not actually be asserting BREQ HIGH, depending on the programming of the Master Mode Chip Enable bit (MMO) when the channel decided it wanted the bus. See the "Bus Request/Grant" section for details. If a channel completes a DMA operation and neither Base-to-Current reloading nor auto-chaining were enabled, the No Auto-Reload or Chaining (NAC) bit will be set. The NAC bit will be reset when the channel receives a "Start Chain Command." If two interrupts are queued, the Second Interrupt Pending bit (SIP) will be set and the channel will be inhibited from further activity until an interrupt acknowledge occurs. See the "Interrupt" section for details. Finally, if the channel is issued an $\overline{EOP}$ during chaining, the Chaining Abort (CA) and the NAC will be set. These bits are also set when a "reset" is issued to the UDC. The CA bit holds the NAC bit in the set state. The CA bit is cleared when a new Chain Upper Address and Tag word or Lower Address word is loaded into the channel.

The Hardware Interface Field provides a Hardware Request (HRQ) bit which provides a means of monitoring the channels $\overline{DREQ}$ input pin. When the $\overline{DREQ}$ pin is LOW, the HRQ bit will be "1" and vice-versa. The Hardware Mask (HM) bit, when set, prevents the UDC from responding to a LOW on $\overline{DREQ}$. Note, however, that the Hardware Request bit always reports the true (unmasked) status of $\overline{DREQ}$ regardless of the setting of the HM bit.

The Completion Field stores data at the end of each DMA operation. This data indicates why the DMA operation ended. When the next DMA operation ends, new data is loaded into these bits overwriting, thereby erasing the old setting. Three bits indicate whether the DMA operation ended as a result of a TC, MC or $\overline{EOP}$ termination. The TC bit will be "1" if the Operation Count reaching zero ended the DMA operation. The MC bit will be "1" if an MC termination occurred regardless of whether Stop-on-Match or Stop-on-no-Match was selected. The EOP bit is set only when an external $\overline{EOP}$ ends a DMA transfer; it is not set for $\overline{EOP}$ issued during chaining. Note that two or even all three of MC, TC and EOP may be set if multiple reasons exist for ending the DMA operation. The MCH and MCL bits report on the match state of the upper and lower comparator bytes, respectively. These bits are set when the associated comparator byte has a match and are reset otherwise, regardless of whether Stop-on-Match or Stop-on-no-Match is programmed. Regardless of the DMA operation performed, these bits will reflect the comparator status at the end of the DMA operation. These two bits are provided to help determine which byte matched or did not match when using 8-bit matches with word searches and transfer-and-searches. The three reserved bits return zeroes during reads.

Interrupt Vector and Interrupt Save Registers

Each channel has an Interrupt Vector register and an Interrupt Save register. The Interrupt Vector is 8-bit wide and is written to and read from on $AD_0 - AD_7$. The Interrupt Save register may be read by the CPU without wait states. The Interrupt Vector register contains the vector or identifier to be output during an Interrupt Acknowledge cycle. When an interrupt occurs (IP = 1), either because a DMA operation terminated or because $\overline{EOP}$ was driven LOW during chaining, the contents of the Interrupt Vector register and part of the Channel Status register are stored in the 16-bit Interrupt Save register (See Figure 7).

Because the vector and status are stored, a new vector can be loaded into the Interrupt Vector register during chaining, and a new DMA operation can be performed before an interrupt acknowledge cycle occurs. If another interrupt occurs on the channel before the first is acknowledged, further channel activity is suspended.

As soon as the first clear IP command is issued, the status and vector for the second interrupt are loaded into the Interrupt Save register and Channel Operation resumes. The UDC can retain only two interrupts for each channel; a third operation cannot be initiated until the first interrupt has been cleared. See the "Interrupt" section for further details.

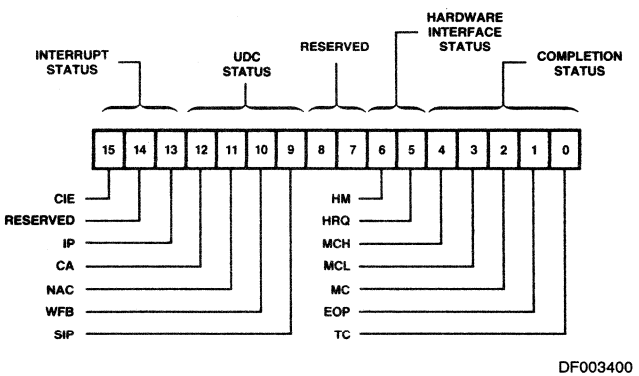


Figure 6. Status Register

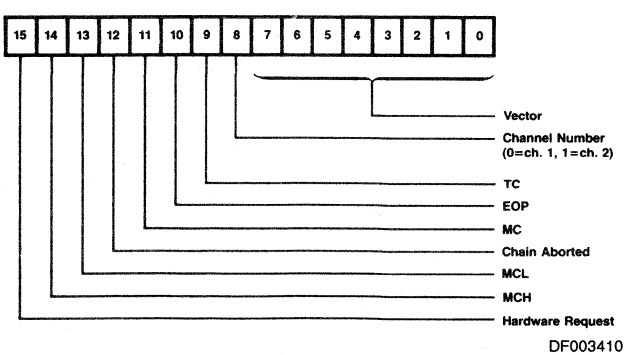


Figure 7. Interrupt Save Register

Channel Mode Register

The Channel Mode registers are two words wide. There are 21 bits defined in each Channel Mode register; the other 11 bits are unused. See Figure 8. The Channel Mode registers may be loaded during chaining and may be read by the host CPU. CPU reads of the Channel Mode register are slow reads and require insertion of multiple wait states. The Channel Mode Low word (bits 0 – 15) may be written to directly by the host CPU. The Channel Mode register selects what type of DMA operation the channel is to perform, how the operation is to be executed, and what action, if any, is to be taken when the channel finishes.

The Data Operation Field and the Transfer Type field select the type of operation the channel is to perform. It also selects the operand size of bytes or words (see Figure 9 for code-definition). The different types of operations are described in detail in the "DMA Operations" section. The Flip bit is used to select whether the Current ARA register points to the source and the Current ARB register points to the destination or vice-versa.

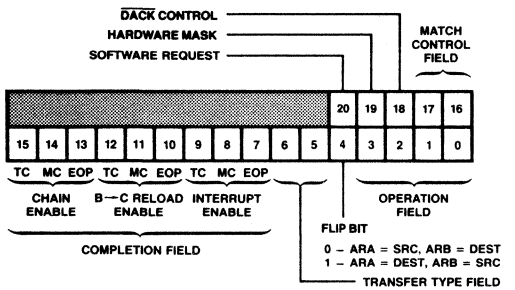


Figure 8. Channel Mode Register

The Completion Field is used to program the action taken by the channel at the end of a DMA operation. This field is discussed in the "Completion Options" section. The 2-bit Match Control field selects whether matches use an 8-bit or 16-bit pattern and whether the channel is to stop-on-match or stop-on-no-match. See Figure 9 and the "Search" section for details. The Software Request bit and Hardware Mask bit can be set and cleared by software command in addition to being loaded in parallel with other Channel Mode bits. These bits are described in detail in the "Initiating DMA Operations" section.

The $\overline{\text{DACK}}$ Control bit is used to specify when the $\overline{\text{DACK}}$ pin is driven active. When this bit is cleared, the channel's $\overline{\text{DACK}}$ pin will be active whenever the channel is performing a DMA Operation, regardless of the type of transaction. Note that the pin will not be active while the channel is chaining. If this bit is set, the $\overline{\text{DACK}}$ pin will be inactive during chaining, during both Flowthru Transfers and Flowthru Transfer-and-Searches, and during Searches, but $\overline{\text{DACK}}$ will be pulsed active during Flyby Transfers and Flyby Transfers-and-Searches at the time necessary to strobe data into or out of the Flyby peripheral. Flyby operations are discussed in detail in the "Flyby Transactions" section.

DATA OPERATION FIELD			
	Operand Size		Transaction Type
Code/Operation	ARA	ARB	
Transfer			
0001	Byte	Byte	Flowthru
100X	Byte	Word	Flowthru
0000	Word	Word	Flowthru
0011	Byte	Byte	Flyby
0010	Word	Word	Flyby
Transfer-and-Search			
0101	Byte	Byte	Flowthru
110X	Byte	Word	Flowthru
0100	Word	Word	Flowthru
0111	Byte	Byte	Flyby
0110	Word	Word	Flyby
Search			
1111	Byte	Byte	N/A
1110	Word	Word	N/A
101X	Illegal		

TRANSFER TYPE FIELD AND MATCH CONTROL FIELD		
Transfer Type	Code	Match Control
Single Transfer	00	Stop on No Match
Demand Dedicated/Bus Hold	01	Stop on No Match
Demand Dedicated/Bus Release	10	Stop on Word Match
Demand Interleave	11	Stop on Byte Match

Figure 9. Channel Mode Coding

Chain Address Register

Each channel has a Chain Address register which points to the chain control table in memory containing data to be loaded into the channel's registers. The Chain Address register, as shown in Figure 10, is two words long. The first word consists of an Upper Address and Tag field. The second word contains the 16-bit Lower Address portion of the memory address. The Tag field contains 2 bits used to designate the number of wait states to be inserted during accesses to the Chain Control Table.

The Chain Address register may be loaded during chaining and may be read from and written to by the host CPU without wait states. If an EOP is issued to the UDC during chaining, the Chain Address register holds the old address. This is true even if the access failure occurred while new Chain Address data was being loaded, since the old data is restored unless both words of the new data are successfully read. Note, however, that EOPs that occur when chaining and while loading a new Chain Address cause the new data to be lost.

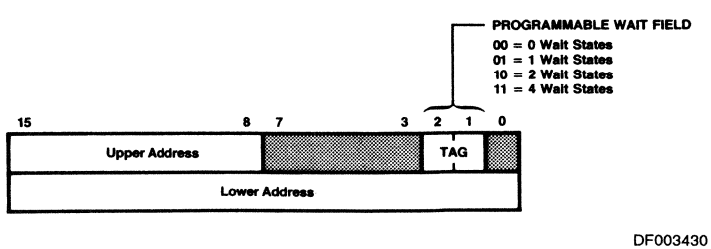


Figure 10. Chain Address Register

DETAILED DESCRIPTION

Any given DMA operation, be it a Transfer, a Search, or a Transfer-and-Search operation, consists of three phases. In the first phase, the channel's registers are initialized to specify and control the desired DMA operation. In the second phase, the DMA operation itself is started and performed. The final phase involves terminating the DMA operation and performing any actions selected to occur on termination. Each of these different phases is described in detail in the following sections.

Reset

The UDC can be reset either by hardware or software. The software reset command is described in the "Commands" section. Hardware resets are applied by pulling $\overline{\text{RESET}}$ LOW. The UDC may be in control of the bus when a reset is applied. BACK is removed internally causing the outputs to go tri-state. If BACK remains HIGH after reset, the UDC will not drive the bus unless BREQ is active. As soon as BACK goes inactive, the UDC places the $AD_0 - AD_{15}$, $A_{16} - AD_{23}$, R/$\overline{\text{W}}$, $\overline{\text{DS}}$, N/$\overline{\text{S}}$, M/$\overline{\text{IO}}$ B/$\overline{\text{W}}$, $\overline{\text{TBEN}}$ and $\overline{\text{RBEN}}$ signals in the high-impedance state.

Both software and hardware resets clear the Master Mode register, clear the CIE, IP and SIP bits, and set the CA and NAC bits in each Channel's Status register. The contents of all other UDC registers will be unchanged for a software reset. Since a hardware reset may have been applied partway through a DMA operation being performed by a UDC channel, the channel's registers should be assumed to contain indeterminate data following a hardware reset.

The Master Mode register contains all zeroes after a reset. The UDC is disabled, and the CPU interleave and hardware wait are inhibited.

Because the CA and NAC bits in the Status register are set by a reset, the channel will be prevented from starting a DMA operation until its Chain Address register's Segment, Tag and Offset fields are programmed and the channel is issued a "Start Chain" Command.

Channel Initialization

The philosophy behind the Am9516A UDC design is that the UDC should be able to operate with a minimum of interaction with the host CPU. This goal is achieved by having the UDC load its own control parameters from memory into each channel. The CPU has to program only the Master Mode register and each Channel's Chain Address register. All other registers are loaded by the channels themselves from a table located in the System memory space and pointed to by the Chain Address register. This reloading operation is called chaining, and the table is called the Chain Control Table.

The Upper and Lower Address fields of the Chain Address register form a 24-bit address which points to a location in system memory space. Chaining is performed by repetitively reading words from memory. Note that the Chain Address register should always be loaded with an even Address; loading an odd Address will cause unpredictable results. The 2-bit Tag field facilitates interfacing to slow memory by allowing the user to select 0, 1, 2 or 4 programmable wait states. The UDC will automatically insert the programmed number of wait states in each memory access during chaining.

The Chain Address register points to the first word in the Chain Control Table. This word is called the Reload Word. See Figure 11. The purpose of the Reload Word is to specify which registers in the channel are to be reloaded. Reload Word bits 10 – 15 are undefined and may be 0 or 1. Each of bits 0 through 9 in the Reload Word correspond to either one or two registers in the channel (see Figure 12). When a Reload Word bit is "1," it means that the register or registers corresponding to that bit are to be reloaded. If a Reload Word bit is "0," the register or registers corresponding to that bit are not to be reloaded. The data to be loaded into the selected register(s) follow the Reload Word in memory (i.e., the data are stored at successively larger memory addresses). The Chain Control Table is a variable length table. Only the data to be loaded are in the table, and the data are packed together.

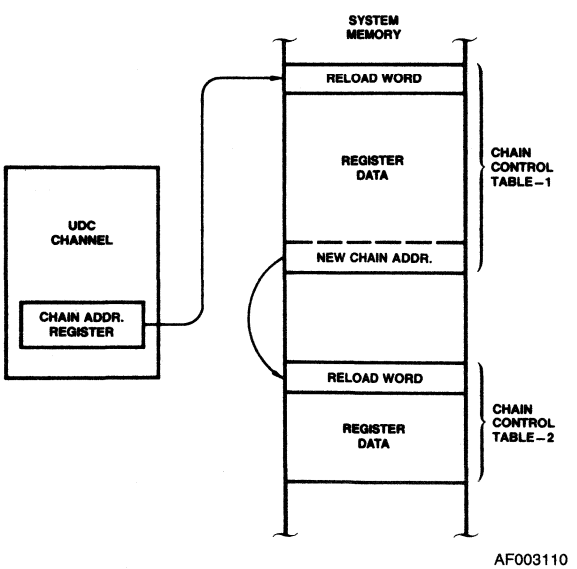


Figure 11. Chaining and Chain Control Tables

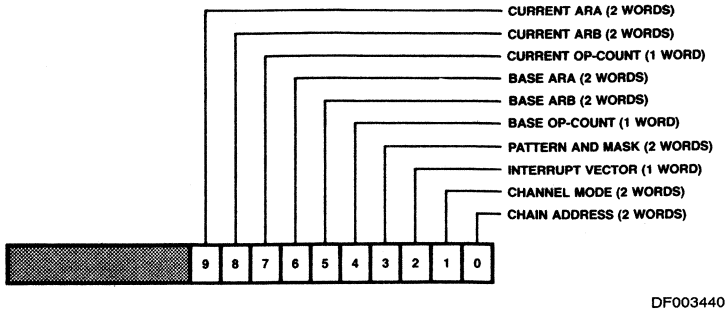


Figure 12. Reload Word/Chain Control Register

When the channel is to reload itself, it first uses the Chain Address register contents to load the Reload Word into the UDC's Chain Control register. Next, the Chain Address register contents are incremented by two to point to the next word in memory. The channel than scans the Reload Word register from bit 9 down to bit 0 to see which registers are to be reloaded. If no registers are specified (bits 9 – 0 are all 0), no registers will be reloaded. If at least one of bits 9 – 0 is set to "1," the register(s) corresponding to the set bit are reloaded, the bit is cleared and the Chain Address register is incremented by 2. The channel continues this operation of scanning the bits from the most significant to least significant bit position, clearing each set bit after reloading its associated registers and incrementing the Chain Address register by 2. If all of bits 9 to 0 are set, all the registers will be reloaded in the order: Current ARA, Current ARB, Current Operation Count,. . . Channel Mode and Chain Address. Figure 13 shows two examples of Chain Control Tables. Example 1 shows the ordering of data when all registers are to be reloaded. In example 2 only some registers are reloaded. Once the channel is reloaded, it is ready to perform a DMA operation. Note when loading address registers, the Upper Address and Tag word are loaded first, then the Lower Address word. Also, the Pattern register is loaded before the Mask register.

Initiating DMA Operations

DMA operations can be initiated in one of three ways – by software request, by hardware request and by loading a set software request bit into the Channel Mode register during Chaining.

Starting After Chaining

If the software request bit of the Channel Mode register is loaded with a "1" during chaining, the channel will perform the programmed DMA operation at the end of chaining. If the channel is programmed for Single Operation or Demand, it will perform the operation immediately. The channel will give up the bus after chaining and before the operation if the CPU interleave bit in the Master Mode register is set. See the "Channel Response" section for details. Note that once a channel starts a chaining operation by fetching a Reload Word, it retains bus control at least until chaining of the last register's data is performed.

Software Requests

The CPU can issue Software Request commands to start DMA operations on a channel. This will cause the channel to request the bus and perform transfers. See the description of the software request command for details.

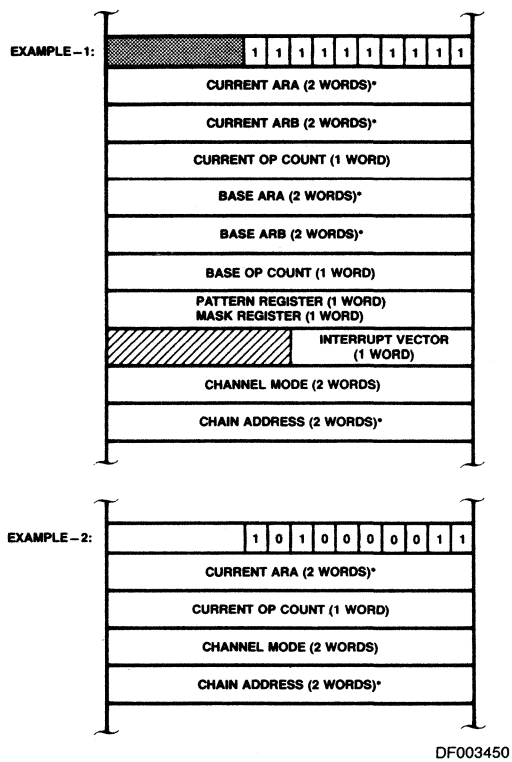


Figure 13. Examples of Chain Control Table

*Load the Upper Address and Tag Word first, then the Lower Address Word.

Hardware Requests

DMA operations will often be started by applying a LOW on the channel's $\overline{\text{DREQ}}$ input. The "Channel Response" section describes when the LOW $\overline{\text{DREQ}}$ signals are sampled and when the $\overline{\text{DREQ}}$ requests can be applied to start the next DMA operation after chaining (see Timing Diagrams 1 and 2).

Bus Request/Grant

Before the UDC can perform a DMA Operation, it must gain control of the system bus. The BREQ and BACK interface pins provide connections between the UDC and the host CPU and other devices, if present, to arbitrate which device has control of the system bus. When the UDC wants to gain bus control, it drives BREQ HIGH.

Some period of time after the UDC drives BREQ HIGH, the CPU will relinquish bus control and drive its HLDA signal HIGH. When the UDC's BACK input goes HIGH, it may begin performing operations on the system bus. When the UDC finishes its operation, it stops driving BREQ HIGH.

When more than one device is used, a priority encoder and a priority decoder are used to decide the bus grant priority.

DMA Operations

There are three types of DMA operations: Transfer, Search, and Transfer-and-Search. Transfers move data from a source location to a destination location. Two types of transfers are provided: Flowthru and Flyby. Searches read data from a source and compare the read data to the contents of the Pattern register. A Mask register allows the user to declare "don't care" bits.

The user can program that the search is to stop either when the read data matches the masked pattern or when the read data fails to match the masked pattern. This capability is called Stop-on-Match and Stop-on-no-Match. Transfer-and-Search combines the two functions to facilitate the transferring of variable length data blocks. Like Transfer, Transfer-and-Search can be performed in either Flowthru or Flyby mode.

Transfers

Transfers use four of the Channel registers to control the transfer operation: the Current ARA and ARB registers; the Current Operation Count register; and the Channel Mode register. Channel Mode register bit CM_4 is called the Flip bit and is used to select whether ARA is to point to the source and ARB is to point to the destination or vice-versa. The Current Operation Count register specifies the number of words or bytes to be transferred.

Bits $CM_3 - M_0$ in the Channel Mode register program whether a Flowthru or Flyby transfer is to be performed. Flowthru transfers are performed in either two or three steps. First, the channel outputs the address of the source and reads the source data into the UDC's Temporary register. In two-step Flowthru Transfer, the channel will then address the destination and write the Temporary register data to the destination location. The three-step Flowthru operation (i.e., the byte-word funneling) is described later in this section. The source and destination for Flowthru Transfers can both be memory locations or both peripheral devices, or one may be a memory location and the other a peripheral device. The $\overline{DACK}$ output for the transferring channel may be programmed to be inactive throughout the transfer or active during the transfer. This is controlled by bit CM_{18} in the Channel Mode register.

Flyby transfers provide improved transfer throughput over Flowthru but are restricted to transfers between memory and peripherals or between two peripherals. Flyby operations are described in detail in the "Flyby Transactions" section.

Transfers can use both byte- and word-sized data. Flowthru byte-to-byte transfers are performed by reading a byte from the source and writing a byte to the destination. The Current Operation Count register must be loaded with the number of bytes to be transferred. Both the Current ARA and Current ARB registers, if programmed to increment/decrement, will change by ±1 if the register points to a memory space ($TG_6 = 2$) and by ±2 if the register points to an I/O space ($TG_6 = 0$).

Flowthru word-to-word transfers require that the Current Operation Count specify the number of words to be transferred. Both the Current ARA and Current ARB registers, if programmed to increment/decrement, will change by ±2 regardless of whether the register points to memory or I/O space.

Byte-word funneling provides packing and unpacking of byte data to facilitate high speed transfers between byte and word peripherals and/or memory. This funneling option can only be used in Flowthru mode. Funneled Flowthru transfers are performed in three steps. For transfers from a byte source to a word destination, two consecutive byte reads are performed from the source address. The data read is assembled into the UDC's Temporary register. In the third step, the Temporary register data is written to the destination address in a word transfer. Funneled transfers from a word source to a byte destination are performed by first loading a word from the source into the UDC's Temporary register. The word is then written out to the destination in two byte writes. For funnel operations, the byte-oriented address must be in the Current ARA register, and the word-oriented address must be in the Current ARB register. The Flip bit (CM_4) in the Channel Mode register is used to specify which address is the source and which is the destination. When the byte address is to be incremented or decremented, the increment/decrement operation occurs after each of the two reads or writes. The Current Operation Count Register must be loaded with the number of words to be transferred.

In byte-to-word funneling operations, it is necessary to specify which half of the Temporary register (upper or lower byte) is loaded with the first byte of data. Similarly, for word-to-byte funneling operations, it is necessary to define which half of the Temporary register is written out first. Figure 14 summarizes these characteristics for both byte-to-word and word-to-byte funneling operations. The criterion used to determine the packing/unpacking order is based on whether the Current ARB register is programmed for incrementing or decrementing of the address. Note that if the address is to remain unchanged (i.e., if bit TG_4 in the Tag Field of the Current ARB register is 1), the increment/decrement bit (bit TG_3) still specifies the packing order.

Search

Searches use five of the Channel registers to control the operation: either the Current ARA or ARB, the Operation Count, the Pattern and Mask registers, and the Channel Mode register. Channel Mode register bit CM_4 is called the Flip bit and is used to select either Current ARA or ARB as the register specifying the source for the search. Only one of the Current Address registers is used for search operations since there is no destination address required. The Current Operation Count register specifies the maximum number of words or bytes to be searched.

Search operations involve repetitive reads from the peripheral or memory until the specified match condition is met. The search then stops. This is called a Match Condition or MC termination. Each time a read is performed, the Source address, if so programmed, is incremented or decremented and the Operation Count is decremented by 1. If the match condition has not been met by the time the Operation Count reaches zero, the zero value will force a TC termination, ending the search. Searches can also stop due to a LOW being applied to the $\overline{EOP}$ interface pin. During a Search operation, the channel's $\overline{DACK}$ output will be either inactive or active throughout the search. This is controlled by bit CM_{18} in the Channel Mode register. The reads from the peripheral or

memory performed during Search follow the timing sequences described in the "Flowthru Transactions" sections.

On each read during a Search operation, the UDC's Temporary register is loaded with data and compared to the Pattern register. The user can select that the Search is to stop when the Pattern and Temporary register contents match or when they don't match. This Stop-On-Match/Stop-On-No-Match feature is programmed in bit CM_{17} of the Channel Mode register. CM_2 is an enable for the output of the comparator and allows the MC signal to be generated. A Mask register allows the user to exclude or mask selected Temporary register bits from the comparison by setting the corresponding Mask register bit to "1." The masked bits are defined to always match. Thus, in Stop-On-Match, successful matching of the unmasked bits, in conjunction with the always-matched masked bits, will cause the search to stop. For Stop-On-No-Match, the always-matched masked bits are by definition excluded from not matching and therefore excluded from stopping the search.

For word reads the user may select either 8-bit or 16-bit compares through the Channel Mode register bit CM_{16}. In an 8-bit, Stop-On-Match, word-read operation, successful matching of either the upper or lower byte of unmasked Pattern and Temporary registers bits will stop the search. Both bytes do not have to match. In 16-bit Stop-On-Match with word reads, all unmasked Pattern and Temporary register bits must match to stop the search. In an 8-bit or 16-bit, Stop-On-No-Match, word-read Search operation, failure of any bit to match will terminate the Search operation.

In an 8-bit Stop-On-Match with byte-reads, the Search will stop if either the upper or lower byte of unmasked Pattern and Temporary register bits match. For an 8-bit Stop-On-No-Match with byte reads, failure of matching in any unmasked Pattern and Temporary register bit will cause the Search to stop. For 8-bit searches, the upper and lower bytes of the Pattern and Mask register should usually be programmed with the same data. Failure to set the upper and lower bytes of the Pattern and Mask registers to identical values will result in different comparison criteria being used for the upper and lower bytes of the Temporary register. Users failing to program identical values for the upper and lower bytes can predict the results by recognizing that in 8-bit Stop-On-Match, the search will end if all the unmasked bits in either the upper or lower bytes match, and for 8-bit Stop-On-No-Match, the failure of any unmasked bit to match will end the Search. For accurate predictions, it is also necessary to know that for word reads the Temporary register high and low bytes are loaded from $AD_{15} - AD_8$ and $AD_7 - AD_0$ respectively. In byte reads, the read byte is duplicated in both halves of the Temporary register except in funneling.

Funneling Direction	Current ARB Tag Field		Increment/Decrement and Packing/Unpacking Rules
	TG_4	TG_3	
Word-to-Byte ($CM_4 = 1$)	0	0	Increment ARB, Write High Byte First
	0	1	Decrement ARB, Write Low Byte First
	1	0	Hold ARB, Write High Byte First
	1	1	Hold ARB, Write Low Byte First
Byte-to-Word ($CM_4 = 0$)	0	0	Increment ARB, Read High Half of Word First
	0	1	Decrement ARB, Read Low Half of Word First
	1	0	Hold ARB, Read High Half of Word First
	1	1	Hold ARB, Read Low Half of Word First

Figure 14. Byte/Word Funneling

Transfer-and-Search

Transfer-and-Search combines the operations of the Transfer and the Search functions. The registers used to control Transfer-and-Searches are the Current ARA and ARB registers, the Operation Count register, the Pattern and Mask registers, and the Channel Mode register.

A Transfer-and-Search operation will end when the data transferred meets the match condition specified in Channel Mode register bits $CM_{17} - CM_{16}$. The Mask and Pattern registers indicate those bits being compared with the Temporary register contents. Like Transfers and Searches, Transfers-and-Searches will also be terminated if the operation count goes to zero or if a LOW is applied to the $\overline{EOP}$ pin. Regardless of whether Transfer-and-Search stops because of a TC, MC or EOP, it will always complete the iteration by writing to the destination address before ending (writing twice for word-to-byte funneling).

In Flowthru mode, the Transfer-and-Search timing is identical to Flowthru Transfer. While the data is in the Temporary register, it is masked by the Mask register and compared to the Pattern register. For word Transfer and Transfer-and-Search, the high and low bytes of the Temporary register are always written to and read from $AD_{15} - AD_8$ and $AD_7 - AD_0$ respectively. For byte Transfer and Transfer-and-Search, the byte read is always loaded into both halves of the Temporary register, and the entire register is driven directly out onto the $AD_{15} - AD_0$ bus. Transfer-and-Search can also be used with byte word funneling. In funneling, the match is an 8-bit match or 16-bit match as determined by the setting of bit CM_{16}.

Flyby Transfer-and-Search can be used to increase throughput for transfer between two peripherals or between memory and a peripheral. Memory-to-Memory Flyby is not supported. Also, in Flyby, the operand sizes of the source and destination must be the same, funneling is not supported. A complete discussion of Flyby timing is given the "Flyby Transactions" section. During a Flyby Transfer-and-Search, data is loaded into the Temporary register to facilitate the comparison operation, and at the same time, data is transferred from the source to the destination. When byte operands are used, data is loaded into both bytes of the Temporary register, from the $AD_{15} - AD_8$ bus if the Current ARA register is even and from $AD_7 - AD_0$ line if the Current ARA register is odd. This will alternate for memory bytes so the user must drive both halves of the bus to use the search. When word operands are used, data is loaded directly from $AD_{15} - AD_8$ and $AD_7 - AD_0$ into the Temporary register's high and low bytes respectively.

Channel Response

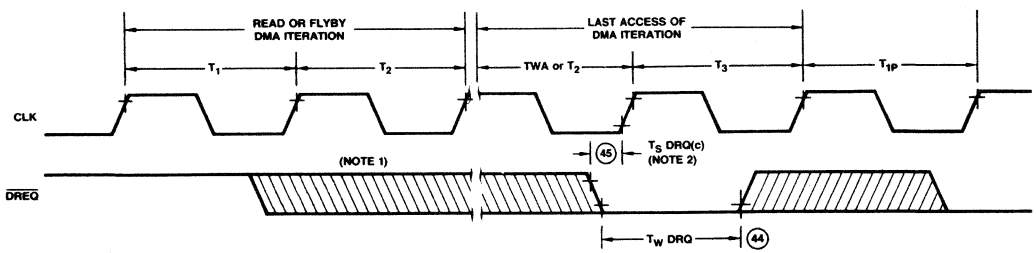
Channel Mode register bits $CM_6 - CM_5$ select the channel's response to the request to start a DMA operation. The response falls into either of two types: Single Operation or Demand. There are three subtypes for Demand operations: Demand Dedicated with Bus Hold, Demand Dedicated with Bus Release, and Demand Interleave. To make discussions clear, it is necessary to define the term "single iteration of a

DMA operation." For Search operations, one iteration consists of a single read operation and a comparison of the read data to the unmasked Pattern register bits. The Operation count will be decremented by 1, and the Current Address register used incremented or decremented if so programmed. For Transfer and Transfer-and-Search operations, a single iteration comprises reading a datum from the source, writing it to the destination, comparing the read datum to the unmasked Pattern register bits (Transfer-and-Search only), decrementing the Operation count by 1 and incrementing/decrementing the Current ARA and ARB registers if so programmed. In byte-word funneling, a single iteration consists of two reads followed by a write (Byte-to-Word funneling) or one read followed by two writes (Word-to-Byte funneling). In all Transfer and Transfer-and-Search cases, the iteration will not stop until the data in the Temporary register is written to the destination. See Appendix B for flowchart.

Single Operation

The Single Operation response is intended for use with peripherals which transfer single bytes or words at irregular intervals. Each application of a Software request command will cause the channel to perform a single iteration of the DMA operation. Similarly, if the Software request bit is set by chaining, at the end of chaining the channel will perform a single iteration of the DMA operation. Each application of a HIGH-to-LOW transition on the $\overline{DREQ}$ input will also cause a single iteration of the DMA operation. If the Hardware mask bit is set when the transition is made, the iteration will be performed when the mask is cleared, providing the DMA operation has not terminated. See the Set/Clear Hardware mask bit command for details. Each time a Single Operation ends, the channel will give up control of the bus unless a new transition has occurred on $\overline{DREQ}$. The new transition can occur anytime after the HIGH-to-LOW ALE transition of a read or Flyby memory or I/O access of the DMA iteration. Timing Diagram 1 shows the times after which a new transition can be applied and recognized to avoid giving up the us at the end of the current iteration.

*TIMING DIAGRAM 1. Sampling $\overline{DREQ}$ During Single Transfer DMA Operations



WF007460

Notes: 1. HIGH-to-LOW $\overline{DREQ}$ transitions will only be recognized after the HIGH-to-LOW transition of the clock during T_1 of a read or flyby DMA iteration.

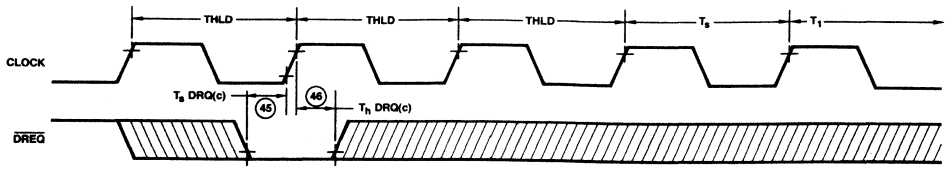
2. A HIGH-to-LOW $\overline{DREQ}$ transition must meet the conditions in Note 1 and must occur TsDRQ(c) before state T_3 of the last access of the DMA iteration if the channel is to retain bus control and immediately start the next iteration. $\overline{DREQ}$ may go HIGH before TsDRQ(c) if it has met the TwDRQ parameter.

3. Flyby and Search transactions have only a single access; parameter TsDRQ(c) should be referenced to the start of T_3 of the access. All other operations will always have two or three accesses per iteration.

*See Appendix D for timing parameters.

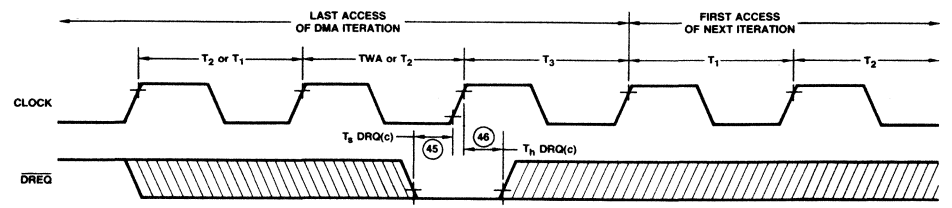
TIMING DIAGRAM 2. $\overline{DREQ}$ Sampling in Demand Mode

(a) Sampling of $\overline{DREQ}$ while in Bus Hold Mode



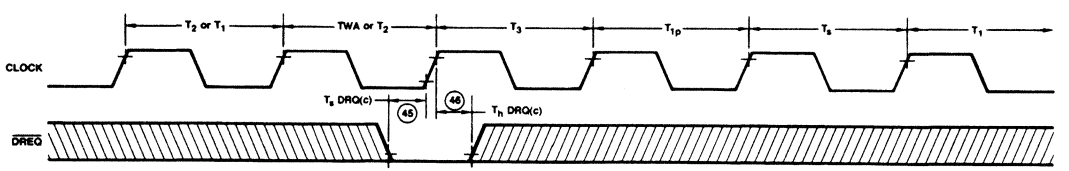
WF007480

(b) $\overline{DREQ}$ Sampling in Demand Mode During DMA Operations



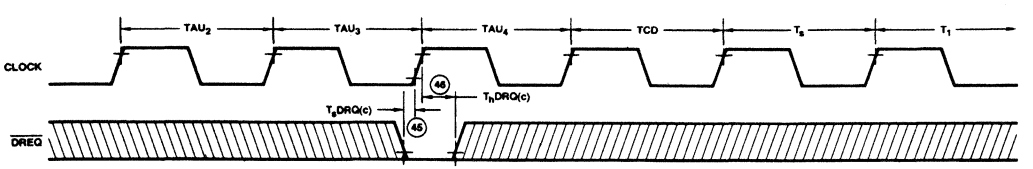
WF007490

(c) Sampling $\overline{DREQ}$ at the End of Chaining



WF007500

(d) Sampling $\overline{DREQ}$ at the End of Base-to-Current Reloading



WF007510

Notes: 1. $\overline{DREQ}$ must be LOW from the start of TsDRQ(c) to the end of ThDREQ(c) to ensure that the request is recognized.
2. Failure to meet this setup time will result in the channel releasing the bus.
3. Ts is a setup state, generated before entering DMA operation cycle.
4. TAU_2, TAU_3 and TAU_4 are auto-reload states, followed by TCD (chain decision) state.

Demand Dedicated With Bus Hold

In Demand Dedicated with Bus Hold (abbreviated Bus Hold), the application of a Software request command or the setting of the software request bit during chaining or applying a LOW level on the $\overline{DREQ}$ input will cause the channel to acquire bus control.

If $\overline{DACK}$ is programmed as a level output ($CM_{18} = 0$), $\overline{DACK}$ will be active from when the channel acquires bus control to when it relinquishes control. A Software Request will cause the channel to request the bus and perform the DMA operations until TC, MC or EOP.

Once the channel gains bus control due to a LOW $\overline{DREQ}$ level, it samples $\overline{DREQ}$ as shown in Timing Diagram 2. If $\overline{DREQ}$ is LOW, an iteration of the DMA operation is performed. If $\overline{DREQ}$ is HIGH, the channel retains bus control and continues to drive all bus control signals active or inactive, but performs no DMA operation. Thus the user can start or stop execution of DMA operations by modulating $\overline{DREQ}$. Once TC, MC or EOP occurs, the channel will either release the bus or, if chaining or Base-to-Current reloading is to occur, perform the desired operation. After chaining or Base-to-Current reloading, if the channel is still in Bus Hold mode and does not have a set software request bit (set either by chaining or command), the channel will relinquish bus control unless a LOW $\overline{DREQ}$ level occurs within the time limits.

Demand Dedicated With Bus Release

In Demand Dedicated with Bus Release (abbreviated Bus Release), the application of a Software Request will cause the channel to request the bus and perform the programmed DMA operation until TC, MC or EOP. If the channel was programmed for Bus Release and the software request bit was set during chaining, the channel will start the DMA operation as soon as chaining ends, without releasing the bus and will continue performing the operation until TC, MC or EOP.

When an active LOW $\overline{DREQ}$ is applied to a channel programmed for Bus Release, the channel will acquire the bus and perform DMA operations: (a) until TC, MC or EOP or (b) until $\overline{DREQ}$ goes inactive. Timing Diagram 2 shows when $\overline{DREQ}$ is sampled to determine if the channel should perform another cycle or release the bus. Note that this sampling also occurs on the last cycle of a chaining operation. If a channel has an active $\overline{DREQ}$ at the end of chaining, it will begin performing DMA operations immediately, without releasing the bus. When a TC, MC or EOP occurs, terminating a Bus Release mode operation, the channel, if enabled for chaining and/or Base-to-Current reloading, will perform reloading and/or chaining (assuming the Status register's SIP bit is clear) without releasing the bus.

If the SIP bit in the Channel Status register is set when a DMA termination occurs, the channel will relinquish the bus control until an Interrupt Acknowledge has been received and the SIP bit is cleared. After an interrupt has been serviced, the channel will perform the Base-to-Current reloading and/or chaining if enabled for the termination.

If an active request is not applied and the channel is in Demand Dedicated with Bus Hold, the channel will go into state THLD (see Timing Diagram 2(a)). If an active request is not applied and the channel is in Demand Dedicated with Bus Release or Demand Interleave mode, it will release the bus. Note that even if an active request is applied in Demand Interleave, the channel may still release the bus. The request for Demand Interleave should continue to be applied to ensure that the channel eventually responds to the request by acquiring the bus (i.e., the request is not latched by the channel).

Demand Interleave

Demand Interleave behaves in different ways depending on the setting of Master Mode register bit MM. If MM is set, the UDC will always relinquish bus control and then re-request it after each DMA iteration. This permits the CPU and other devices to gain bus control. If MM is clear, control can pass from one UDC channel to the other without requiring the UDC to release bus control. If both channels have active requests, control will pass to the channel which did not just have control. For instance if MM is clear and both channels have active requests and are in Demand Interleave mode, control will toggle between the channels after each DMA operation iteration and the UDC will retain bus control until both channels are finished with the bus. If MM is set and both channels have active requests and are in Demand Interleave mode, each channel will relinquish control to the CPU after each iteration resulting in the following control sequence: channel 1, CPU, channel 2, CPU, etc. Note that if there are other devices on the bus, they may gain control during the part of the sequence labelled CPU. See Appendix B for flowchart.

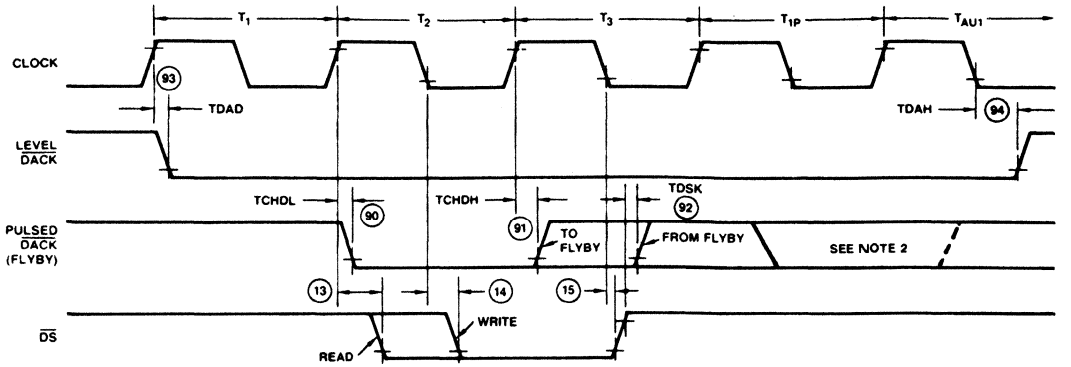
A software or hardware request will cause a channel programmed for Demand Interleave to perform interleaved DMA operations until TC, MC or EOP. If the Software request bit is set during chaining, the channel will retain the bus after chaining and will immediately start performing a DMA iteration and will interleave all DMA iterations after the first. If $\overline{DREQ}$ is LOW on the last cycle during chaining, the channel will perform a single iteration immediately after chaining and interleave thereafter until: (a) TC, MC or EOP or (b) $\overline{DREQ}$ goes HIGH. If (b) occurs, the channel will relinquish the bus until $\overline{DREQ}$ goes LOW again and the channel again starts performing interleaved operations. If (a) occurs, the channel will not interleave before first performing chaining and/or Base-to-Current reloading (assuming SIP is cleared).

The waveform of $\overline{DACK}$ is programmed in Channel Mode Register (CM_{18}). The Pulsed $\overline{DACK}$ is for flyby transaction only. See Timing Diagram 3. Note: This figure shows a single Search or Flyby iteration. State TWA is optionally inserted if programmed. For more than one iteration, the level $\overline{DACK}$ output would stay active during the time the channel had bus control. When CM_{18} is set, the $\overline{DACK}$ output will be inactive for all nonflyby modes.

Wait States

The number of wait states to be added to the memory or I/O transfer can be programmed by the user as 0, 1, 2 or 4 and can be separately programmed for the Current Address registers A and B and for the Chain Address register. This allows different speed memories and peripherals to be associated with each of these addresses. The Base Address registers A and B also have a Tag Field which is loaded into the Current ARA and ARB registers during Base-to-Current reloading. Because many users utilizing the software programmable wait states will not need the ability to generate hardware wait states through the $\overline{WAIT}$ pin, the wait function can be disabled by clearing the Wait Line Enable bit (MM_2) in the Master Mode register.

TIMING DIAGRAM 3. $\overline{DACK}$ Timing



WF007521

Notes: 1. Level $\overline{DACK}$ RE occurs as shown if auto-reloading is not programmed; otherwise, it stays LOW for three additional clocks.

2. This extra $\overline{DACK}$ pulse occurs only at EOP. It should be used to distinguish which channel got the EOP.

During DMA transactions, the $\overline{WAIT}$ input is sampled in the middle of the T_2 state. If $\overline{WAIT}$ is HIGH, and if no programmable wait states are selected, the UDC will proceed to state T_3. Otherwise, at least one wait state will be inserted. The $\overline{WAIT}$ line is then sampled in the middle of state TWA. If $\overline{WAIT}$ is HIGH, the UDC will proceed to state T_3. Otherwise additional wait states will be inserted. (See Timing Diagram 4.)

Consider what happens in a transaction when both hardware and software wait states are inserted. Each time the $\overline{WAIT}$ line is sampled, if it is LOW, a hardware wait state will be inserted in the next cycle. The software wait state insertion will be suspended until $\overline{WAIT}$ is sampled and is HIGH. The hardware wait states may be inserted anytime during the software wait state sequence. It is important to know that hardware wait states are served consecutively rather than concurrently with software wait states. For example, assume for a Flowthru I/O Transaction that a user has programmed 4 software wait states. Driving a LOW on the $\overline{WAIT}$ input during T_2 for 2 cycles would insert 2 hardware wait states. Driving $\overline{WAIT}$ HIGH for 3 cycles would allow insertion of three of the four software wait states. Driving $\overline{WAIT}$ LOW for 2 more cycles would insert 2 more hardware wait states. Finally, driving $\overline{WAIT}$ HIGH would allow the final software wait state to be inserted. During this last software wait state, the $\overline{WAIT}$ pin would be sampled for the last time. If it is HIGH, the channel will proceed to state T_3. If the pin is LOW, the channel will insert hardware wait states until the pin goes HIGH and the channel would then enter state T_3 to complete the I/O transaction.

DMA Transactions

There are three types of transactions performed by the Am9516A UDC: Flowthru, Flyby and Search. Figures 15 and 16 show the configurations of Flowthru and Flyby Transactions.

Flowthru Transactions

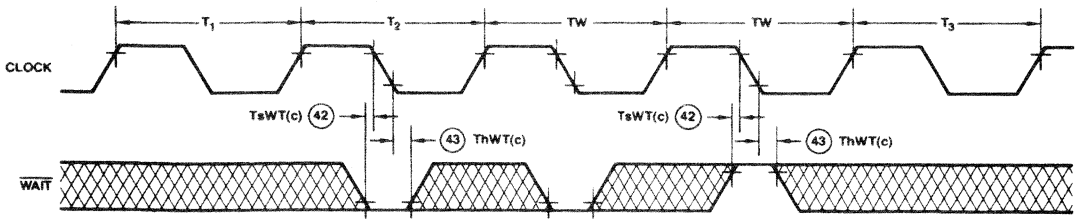
A Flowthru Transaction consists of Read and Write cycles. Each cycle consists of three states: T_1, T_2, and T_3 as shown in Timing Diagram 5. The user may select to insert software wait states through the Tag fields of the Current ARA and ARB registers. In addition, if Master Mode register bit $MM_2 = 1$, hardware wait states may be inserted by driving a LOW signal on the $\overline{WAIT}$ pin.

The M/$\overline{IO}$ and N/$\overline{S}$ lines will reflect the appropriate level for the current cycle early in T_1. The TG_6 and TG_7 bits of the current ARA and ARB registers should be programmed properly. The ALE output will be pulsed HIGH to mark the beginning of the cycle. The offset portion of the address for the peripheral being accessed will appear on $AD_0 - AD_{15}$ during T_1. The R/$\overline{W}$ and B/$\overline{W}$ lines will select a read or write operation for bytes or words. The R/$\overline{W}$, N/$\overline{S}$, M/$\overline{IO}$ and B/$\overline{W}$ lines will become stable during T_1 and will remain stable until after T_3.

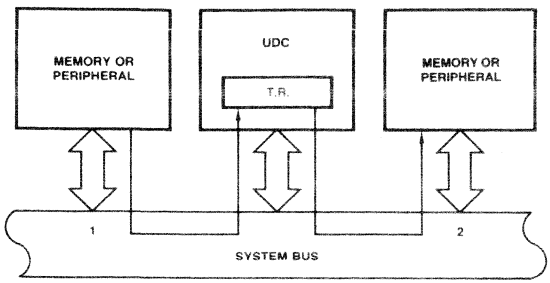
I/O address space is byte-addressed, but both 8- and 16-bit data sizes are supported. During I/O transactions the B/$\overline{W}$ output signal will be HIGH for byte transactions and LOW for word transactions. For I/O transactions, both even and odd addresses can be output; hence, the address bit output on AD_0 may be 0 or 1.

The channel can perform both I/O read and I/O write operations; the M/$\overline{IO}$ line will be LOW. During an I/O read, the $AD_0 - AD_{15}$ bus will be placed in the high-impedance state by the UDC during T_2. The UDC will drive the $\overline{DS}$ output LOW to signal the peripheral that data can be gated onto the bus. The UDC will strobe the data into its Temporary register during T_3. $\overline{DS}$ will be driven HIGH to signal the end of the I/O transaction. During I/O write, the UDC will drive the contents of the Temporary register onto the $AD_0 - AD_{15}$ bus and shortly after will drive the $\overline{DS}$ output LOW until T_3. Peripherals may strobe the data on AD bus into their internal registers on either the falling or rising edge. If the peripheral is to be accessed in a Flyby transaction also, data should be written on the rising edge of $\overline{DS}$ only.

TIMING DIAGRAM 4. $\overline{WAIT}$ Timing

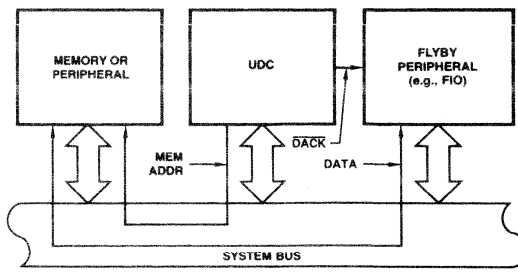


WF007680



AF003120

Figure 15. Configuration of Flowthru Transaction



AF003130

Figure 16. Configuration of Flyby Transaction

For byte I/O writes, the channel will drive the same data on data bus lines $AD_0 - AD_7$ and $AD_8 - AD_{15}$. During byte I/O reads when the address bit on AD_0 is 0, the UDC will strobe data in from data lines $AD_8 - AD_{15}$. During byte I/O reads when the address bit on AD_0 is 1, the UDC will strobe data in from data lines $AD_0 - AD_7$. Thus, when an 8-bit peripheral is connected to the bus, its internal registers will typically be mapped at all even or all odd addresses. To simplify accesses to 8-bit peripherals, byte oriented I/O addresses are incremented/decremented by 2.

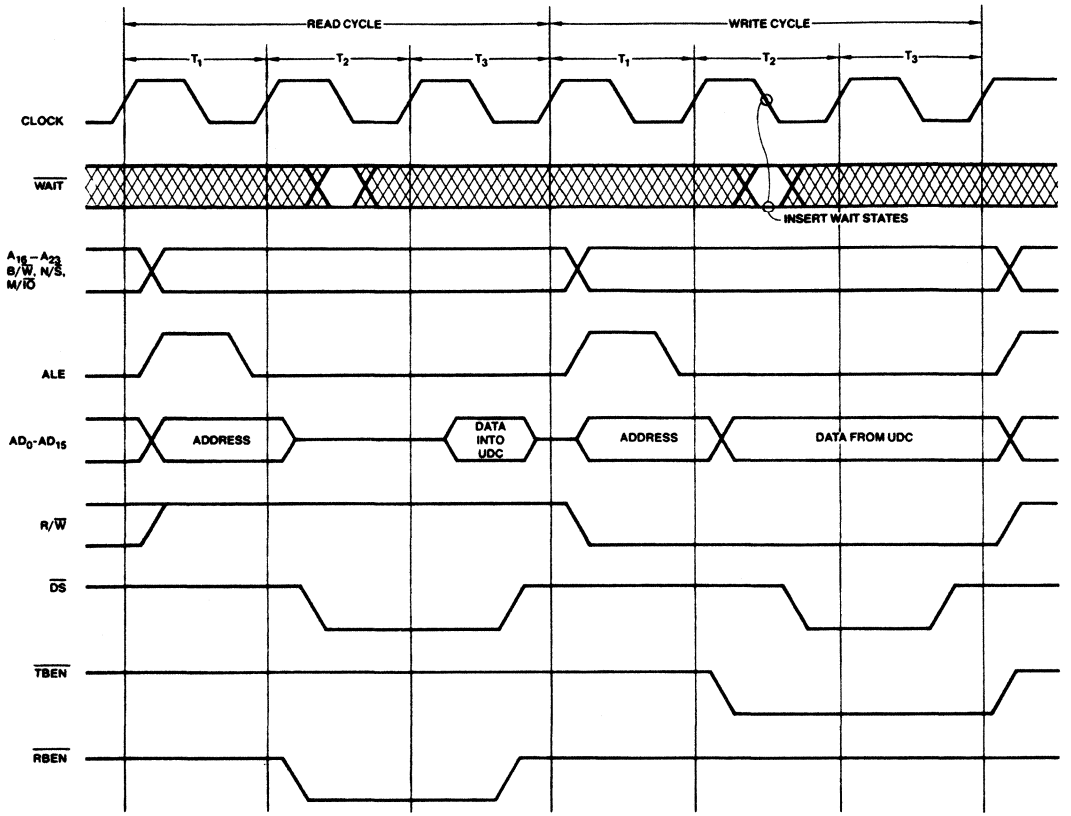
The channel can perform the I/O read and memory write operation, the memory read and I/O write operation, and the memory read and memory write operation, also. The timing for all Flowthru transactions is the same.

During chaining operations the UDC reads words from an address in System memory pointed to by the active channel's Chain Address register. Those chaining operations are performed identically to the Flowthru memory read transactions, except that the data is loaded into an internal UDC channel register rather than the Temporary register. Note that chaining never causes a write or a byte read; thus, all memory writes or all byte accesses are due to DMA operations. A typical memory operation consists of three states: T_1, T_2, and T_3, as shown in Timing Diagram 5. The user may select to insert 1, 2 or 4 software wait states after state T_2 and before state T_3 by programming the Tag field of the Current Address register or the Chain Address register. If the Wait Line Enable bit in the Master Mode register is set, the user may also insert hardware wait states after state T_2 and before state T_3 by driving a LOW on the $\overline{WAIT}$ line. The operation of Flowthru memory transactions is performed identically to the Flowthru I/O transactions. (See Timing Diagram 5.)

Flyby Transactions

Flyby Transfer and Flyby Transfer-and-Search operations are performed in a single cycle, providing a transfer rate significantly faster than that available from Flowthrus. In Flyby, operations can only be performed between memory and peripheral or between peripheral and peripheral. Memory-to-Memory operations cannot be performed in Flyby mode; these must be done using Flowthru.

TIMING DIAGRAM 5. Flowthru Transactions



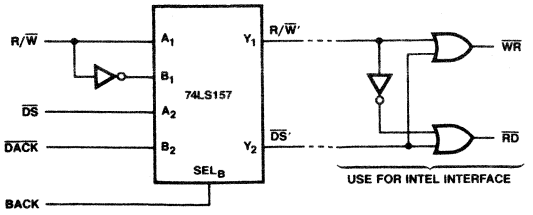
WF007540

The Flyby Transaction can only be used with peripherals having a special Flyby signal input or with external logic. This Flyby input is connected to the channel's $\overline{DACK}$ output. For memory-peripheral Flyby, the address of the source memory location must be programmed in the Current ARA register. The Current ARB register must be programmed with the destination memory location for peripheral-memory Flyby. For Flyby peripheral-to-peripheral transaction, if both peripherals have a Flyby input, only one (called "flyby peripheral") should be connected to $\overline{DACK}$; the other peripheral's Flyby input should be held high during the Flyby operation. The address of the peripheral (called "non-flyby peripheral") not connected to the channel's $\overline{DACK}$ output should be programmed in the Current ARB register when it is a destination. When the non-flyby peripheral is a source, its address should be programmed in the current ARA register. Note that a set Flip bit ($CM_4 = 1$) is for Flyby peripheral to Non-Flyby peripheral or Memory Write transaction (defined as "From Flyby Transaction"), and a clear Flip bit ($CM_4 = 0$) is for the memory or non-flyby peripheral read to Flyby peripheral transaction (defined as "To Flyby Transaction").

Transaction	CM_4	R/$\overline{W}$	Address of Memory or Non-Flyby Peripheral
To Flyby	0	HIGH	ARA
From Flyby	1	LOW	ARB

A Flyby operation is performed using three states: T_1, T_2, and T_3. During T_1 the channel pulses ALE and outputs the address information. See Timing Diagram 6. The R/$\overline{W}$ line is HIGH for "To Flyby" Transaction, and the R/$\overline{W}$ line is LOW for "From Flyby" Transaction.

The channel's M/$\overline{IO}$ and N/$\overline{S}$ lines are coded as specified by the Current ARA or ARB Tag field. The B/$\overline{W}$ line indicates the operand size programmed in the Channel Mode register Operation field. During state T_1 the channel drives R/$\overline{W}$ line to indicate the transaction direction. During state T_2 the channel drives both $\overline{DS}$ and $\overline{DACK}$ active. The Flyby Peripheral connected to $\overline{DACK}$ inverts the R/$\overline{W}$ signal to determine whether it is being read from or written to (see Figure 17).



$$R/\overline{W}' = \overline{BACK \cdot R/\overline{W} + \overline{BACK} \cdot \overline{R/W}}$$
$$\overline{DS}' = \overline{BACK \cdot DACK + \overline{BACK} \cdot DS}$$
$$\overline{RD} = \overline{DACK \cdot \overline{R/W} \cdot BACK + DS \cdot R/\overline{W} \cdot \overline{BACK}}$$
$$\overline{WR} = \overline{DACK \cdot R/W \cdot BACK + \overline{DS} \cdot R/W \cdot \overline{BACK}}$$

AF003140

Figure 17. Flyby Peripheral Interface

The pulsed $\overline{DACK}$ input serves two purposes: to select the peripheral for the Read/Write, and to provide timing information on when to drive data onto or input data from the $AD_0 - AD_{15}$ bus. Note that because the "Flyby Peripheral" never gets explicitly addressed by $AD_0 - AD_{15}$, it must know which internal register is to be loaded from or driven onto the $AD_0 - AD_{15}$ bus. On state T_3, the $\overline{DS}$ and $\overline{DACK}$ lines are driven inactive to conclude the transfer. In Transfer-and-Search mode, data is loaded into the UDC's Temporary register on the LOW-to-HIGH $\overline{DS}$ transition to perform the Search function.

To provide adequate data setup time, the rising edge of $\overline{DS}$ or $\overline{DACK}$ should be the edge used to perform the write to the transfer destination. To extend the active time of $\overline{DS}$ and $\overline{DACK}$, wait states can be inserted between T_2 and T_3. Software wait states can be inserted by programming the appropriate code in the Tag field of the Current ARA or ARB registers. Hardware wait states can be inserted by pulling $\overline{WAIT}$ LOW if the Wait Line Enable bit in the Master Mode register is set. The $\overline{WAIT}$ line is sampled in the middle of the T_2 or TWA state.

Termination

There are three ways a Transfer-and-Search or Search operation can end and two ways a Transfer operation can end. When a channel's Current Operation count goes to 0, the DMA operation being performed will end. This is called a TC or Terminal Count termination. A DMA operation can also be stopped by driving the $\overline{EOP}$ pin LOW with external logic. This is called an EOP termination. Search and Transfer-and-Search operations have a third method of terminating called Match Condition or MC termination. An MC termination occurs when the data being Transferred-and-Searched or Searched meets the match condition programmed in Channel Mode register bits $CM_{17} - CM_{16}$. These bits allow the user to stop when a match occurs between the unmasked Pattern register bits and the data read from the source, or when a no-match occurs. Both byte and word matches are supported. MC terminations do not apply to Transfer operations since the pattern matching logic is disabled in Transfer mode.

End-of-Process

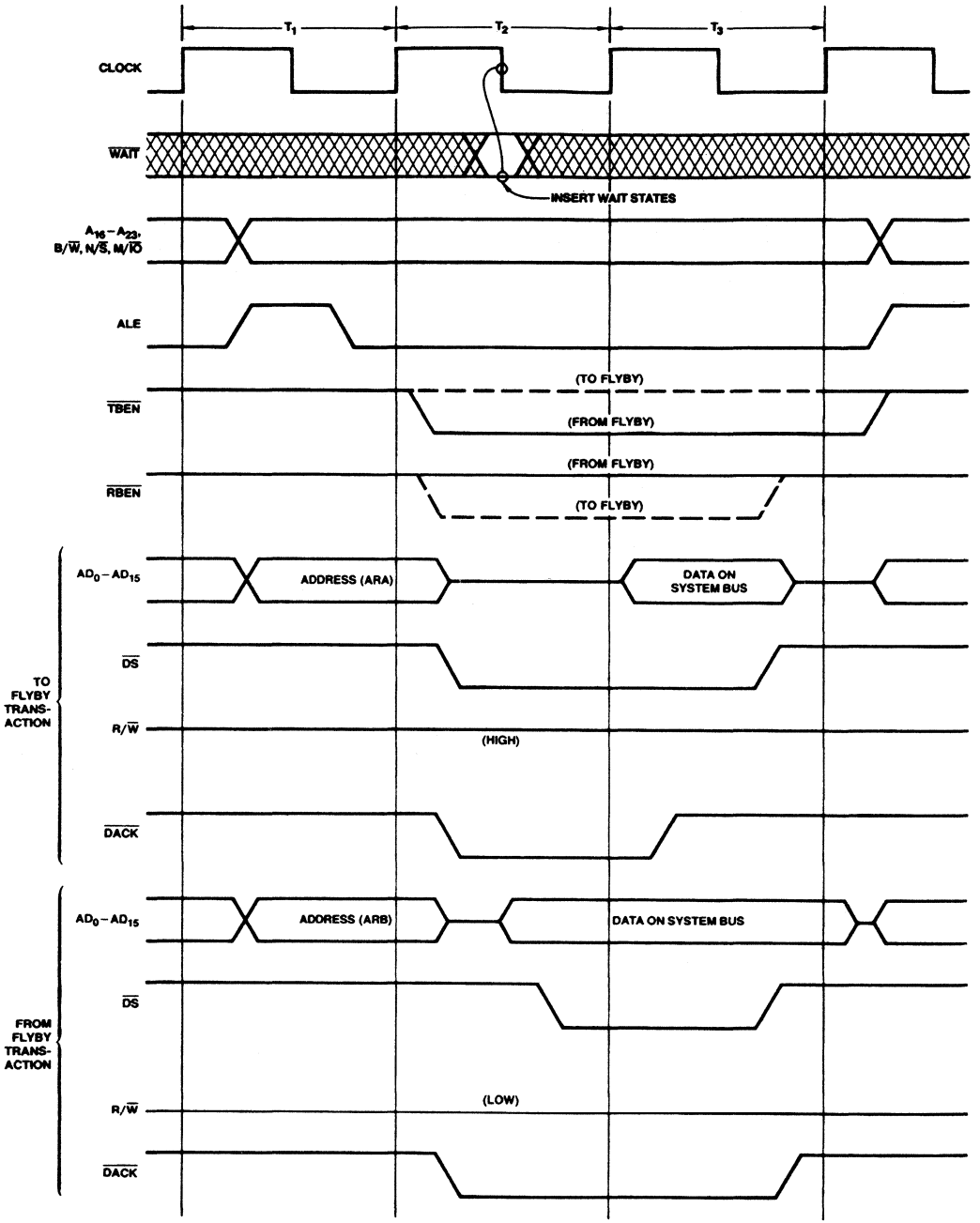
The End-of-Process (EOP) interface pin is a bi-directional signal. Whenever a TC, MC or EOP termination occurs, the UDC will drive the $\overline{EOP}$ pin LOW. During DMA operations, the $\overline{EOP}$ pin is sampled by the UDC to determine if the $\overline{EOP}$ is being driven LOW by external logic. Timing Diagram 7 shows when internal EOPs are generated marking termination of all Transfers. These figures also show the point during the DMA iteration when the EOP pin is sampled. The generation of internal EOPs and sampling of external EOPs for Transfer-and-Searches follow the same timing used for Transfers. Since there is a single $\overline{EOP}$ pin for both channels, $\overline{EOP}$ should only be driven LOW by a channel while that channel is being serviced. This can be accomplished by selecting a level $\overline{DACK}$ output ($CMR_{18} = 0$) and gating each channel's $\overline{EOP}$ request with $\overline{DACK}$, as shown in Figure 18.

If an EOP is detected while the channel is trying to reload the Chain Address register, the new Chain Address Offset and Segment are discarded and the old address + 2 is preserved to allow inspection of the erroneous address.

Programming Completion Options

When a channel ends a DMA operation, the reason for ending is stored in the Completion Status Field of the channel's Status register. See Figure 6. This information is retained until the next DMA operation ends at which time the Status register is updated to reflect the reason(s) for the latest termination. Note that it is conceivable that more than one bit in the Completion Field could be set. An as extreme example, if a channel decremented its Current Operation count to zero, causing a TC termination; input data from the source generated a match causing an MC termination; and there was a LOW on the $\overline{EOP}$ pin resulting in an EOP termination, all three of the channel's Status register completion bits would be set.

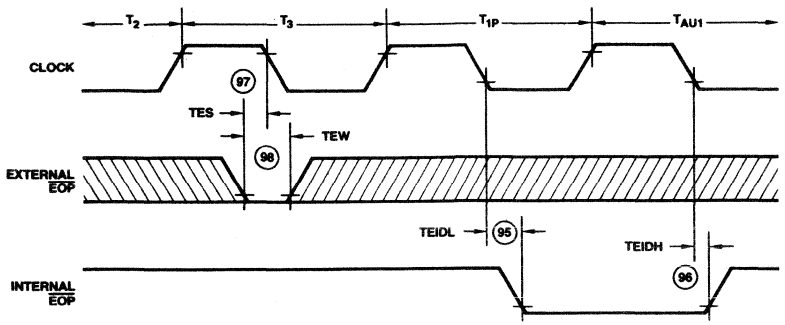
TIMING DIAGRAM 6. Flyby Transactions



WF007550

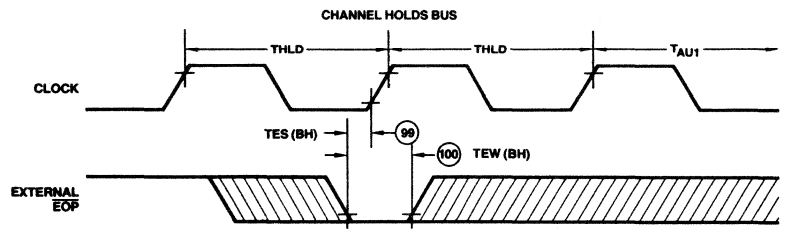
TIMING DIAGRAM 7. $\overline{EOP}$ Timing

(a) $\overline{EOP}$ Sampling and Generation During DMA Operations



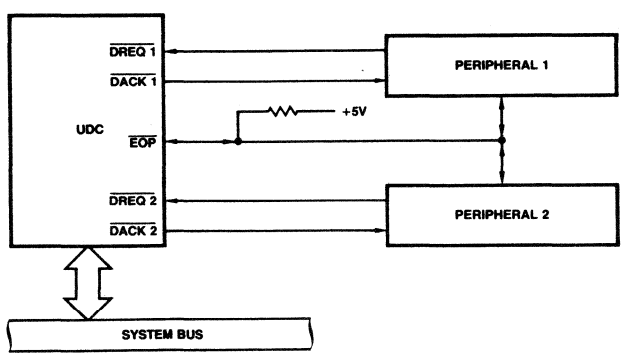
WF007560

(b) Sampling of $\overline{EOP}$ During Bus Hold



WF007570

Notes: 1. The diagram lists state names for both I/O and memory accesses. Sampling of $\overline{EOP}$ will occur on the falling edge of state T_3.
2. State T_{1P} is a pseudo-T_1 state, generated following termination of any DMA operation.
3. T_{AU1} is an auto-intialization state, generated following the TC, MC or $\overline{EOP}$ termination.



AF003150

Figure 18. $\overline{EOP}$ Connection

When a DMA operation ends, the channel can:

(a) Issue an Interrupt request (i.e., setting the IP or SIP bit of the channel's Status register);

(b) Perform Base-to-Current reloading;

(c) Chain reload the next DMA operation;

(d) Perform any combination of the above; or

(e) None of the above.

The user selects the action to be performed by the channel in the Completion option field of the Channel Mode register. For each type of termination (TC, MC or $\overline{EOP}$), the user can choose which action or actions are to be taken. If no reloading is selected for the type of termination that occurred, the NAC bit in the Status register will be set.

More than one action can occur when a DMA operation ends. This may arise because more than one action was programmed for the applicable termination. The priorities of those actions are Interrupt request first, Base-to-Current reloading second, and then chaining. The Interrupt cannot be serviced unless the UDC has relinquished the bus.

Interrupts

To allow the UDC to start executing a new DMA operation after issuing an Interrupt, but before an Interrupt acknowledge is received, a two-deep Interrupt queue is implemented on each channel. The following discussion will describe the standard Interrupt structure and then elaborate on the additional Interrupt queuing capability of the UDC.

A complete Interrupt cycle consists of an Interrupt request followed by an Interrupt-acknowledge transaction. The request, which consists of $\overline{INT}$ being pulled LOW, notifies the CPU that an Interrupt is pending. The Interrupt-acknowledge transaction, which is initiated by the CPU as a result of the request, performs two functions: it selects the peripheral whose Interrupt is to be acknowledged, and it obtains a vector that identifies the selected device and operation – the cause of the Interrupt.

A peripheral can have one or more sources of Interrupt. Each Interrupt source has two bits that control how it generates Interrupts. These bits are a Channel Interrupt Enable bit (CIE) and an Interrupt Pending bit (IP). On the UDC, each channel is an Interrupt source. The two Interrupt control bits are located in bits CM_{15} and CM_{13} of each channel's Status register.

Each channel has its own vector register for identifying the source of the Interrupt during an Interrupt acknowledge transaction. There is one bit (MM_3) in the Master Mode register used for controlling Interrupt behavior for the whole device.

Once a channel issues an Interrupt, it is desirable to allow the channel to proceed with the next DMA operation before the Interrupt is acknowledged. This could lead to problems if the UDC channel attempted to chain reload the Vector register contents. In such a situation, it may not be clear whether the old or new vector would be returned during the acknowledge. This dilemma is resolved in the UDC by providing each channel with an Interrupt Save register. When the channel sets IP as part of the procedure followed to issue an Interrupt, the contents of the Vector register and some of the Status register bits are saved in an Interrupt Save register. See Figure 8. When an Interrupt Acknowledge cycle is performed, the contents of the Interrupt Save register are driven onto the bus. Although the use of an Interrupt Save register allows the channel to proceed with a new task, problems can still potentially arise if a second Interrupt is to be issued by the channel before the first Interrupt is acknowledged. To avoid conflicts between the first and second Interrupt, each channel has a Second Interrupt Pending (SIP) bit in its Status register. When a second Interrupt is to be issued before the first Interrupt is acknowledged, the SIP bit is set and the channel relinquishes the bus until an acknowledge occurs. For compatibility with polled Interrupt schemes, the Interrupt save register can be read by the host CPU without wait states. As an aid to debugging a system's Interrupt logic, whenever IP is set, the Interrupt Save register is loaded from the Vector and Status registers.

Note that the SIP bit is transferred to the IP bit when IP is cleared by the host CPU. Whenever CIE is set, $\overline{INT}$ will go LOW as soon as IP is set.

Base-to-Current Reloading

When a channel finishes a DMA operation, the user may select to perform a Base-to-Current reload. (Base-to-Current reloading is also referred to as Auto-reloading in this document.) In this type of reload, the Current Address registers A and B are loaded with the data in the Base Address registers A and B respectively, and the Current Operation Count register is loaded with the data in the Base Operation Count. The Base-to-Current reload operation facilitates repetitive DMA operations without the multiple memory accesses required by chaining. Although the channel must have bus control to perform Base-to-Current reloading, the complete reloading operation occurs in four clock cycles (i.e., TAU_1 through TAU_4). Note that if the channel had to relinquish the bus because two unacknowledged interrupts were queued, it will have to regain bus control to perform any Base-to-Current reloading (or chaining, for that matter). In this case it acquires the system bus once an Interrupt acknowledge is received, even if it immediately afterward will relinquish the bus because no hardware/software request is present.

Chaining

If the channel is programmed to chain at the end of a DMA operation, it will use the Chain Address register to point to a Chain Control Table in memory. The first word in the table is a Reload word, specifying the register(s) to be loaded. Following the Reload word are the data values to be transferred into the register(s). Chaining is described in detail in the "Channel Initialization" section.

Because chaining occurs after Base-to-Current reloading, it is possible to reset the Current Address registers A and B and the Current Operation Count register to the values used for previous DMA operations, then chain reload one or two of these registers to some special value to be used, perhaps, for this DMA operation only. If the Base values are not reloaded during chaining, the channel can revert back to the Base values at a later cycle.

If an all zero Reload word is fetched during chaining, the chain operation will not reload any registers, but in all other respects, it will perform like any other chaining operation. Thus, the Chain Address will be incremented by 2 to point to the next word in memory, and at the end of the all Zero-Reload word chain operation, the channel will be ready to perform a DMA operation. All Zero-Reload words are useful as "Stubs" to start or terminate linked lists of DMA operations traversed by chaining. On the other hand, care must be taken in their use since the channel may perform an erroneous operation if it is unintentionally started after the chaining operation.

COMMAND DESCRIPTIONS

Figure 19 shows a list of UDC commands. The commands are executed immediately after being written by the host CPU into the UDC's Command register (Figure 20). A description of each command follows.

Reset (00)

This command causes the UDC to be set to the same state generated by a Hardware Reset. The Master Mode register is set to all zeros; the CIE, IP and SIP bits are cleared; the NAC and CA bits in each channel's Status register are set; and the channel activity is forbidden. The Chain Address must be programmed since its state may be indeterminate after a Reset. The lockout preventing channel activity is cleared by issuing a Start Chain command.

Start Chain Channel 1/Channel 2 (A_0/A_1)

This command causes the selected channel to clear the No Auto-Reload or Chain (NAC) bit in the channel's Status register and to start a chain reload operation of the channel's registers, as described in the "Channel Initialization" section. These effects will take place even if the fetched Reload word is all zeros. This command will only be honored if the Chain Abort (CA) bit and the Second Interrupt Pending (SIP) bit in the channel's Status register are clear. If either the CA or SIP bit is set, this command is disregarded.

When the Waiting For Bus (WFB) bit of Status register is set, if the "Start Chain" command is issued, the channel will honor the command after one DMA iteration. It is nearly impossible for the CPU to issue a command when WFB = 1 and the UDC is enabled.

Command	Opcode Bits 7654	Opcode Bits 3210	Example Code HEX
Reset	000X	XXXX	00
Start Chain Channel 1	101X	XXX0	A0
Start Chain Channel 2	101X	XXX1	A1
Set Software Request Channel 1	010X	XX10	42
Set Software Request Channel 2	010X	XX11	43
Clear Software Request Channel 1	010X	XX00	40
Clear Software Request Channel 2	010X	XX01	41
Set Hardware Mask Channel 1	100X	XX10	82
Set Hardware Mask Channel 2	100X	XX11	83
Clear Hardware Mask Channel 1	100X	XX00	80
Clear Hardware Mask Channel 2	100X	XX01	81
Set CIE, or, IP Channel 1	001E	XP10	32*
Set CIE, or, IP Channel 2	001E	XP11	33*
Clear CIE, or, IP Channel 1	001E	XP00	30*
Clear CIE, or, IP Channel 2	001E	XP01	31*
Set Flip Bit Channel 1	011X	XX10	62
Set Flip Bit Channel 2	011X	XX11	63
Clear Flip Bit Channel 1	011X	XX00	60
Clear Flip Bit Channel 2	011X	XX01	61

*Notes: 1. E = Set to 1 to perform set/clear on CIE; Clear to 0 for no effect on CIE.
2. P = Set to 1 to perform set/clear on IP; Clear to 0 for no effect on IP.
3. X = "don't care" bit. This bit is not decoded and may be 0 or 1.

Figure 19. UDC Command Summary

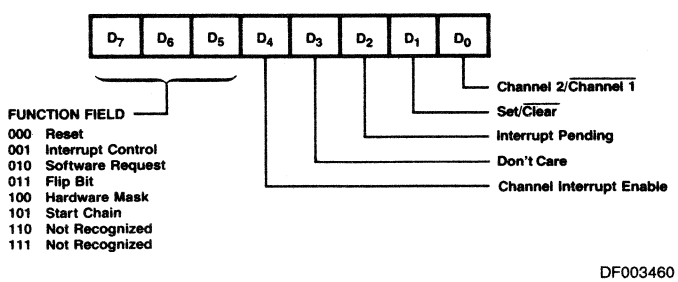


Figure 20. Command Register

Software Request Channel 1/Channel 2 (Set: 42/43, Clear: 40/41)

This command sets or clears the software request bit in the selected channel's Mode register. If the Second Interrupt Pending (SIP) bit and No Auto-Reload or Chain (NAC) bit in the channel's Status register are both cleared, the channel will start executing the programmed DMA operation. If either the SIP or NAC bit is set, the channel will not start executing a DMA operation until both bits are cleared. The SIP bit will clear when the channel receives an Interrupt acknowledge. One way to clear the NAC bit is to issue a Start Chain command to the channel. If the fetched Reload Word is all zeros, the channel's registers will remain unchanged and the software request bit, if set earlier by command, will cause the programmed DMA operation to start immediately. If during chaining new information is loaded into the Channel Mode register, this new information will, of course, overwrite the software request bit.

Set/Clear Hardware Mask 1/Mask 2 (Set: 82/83; Clear: 80/81)

This command sets or clears the Hardware Mask bit in the selected channel's Mode register. This command always takes effect. The Hardware Mask bit inhibits recognition of an active signal on the channel's $\overline{DREQ}$ input; this bit does not affect recognition of a software request. If the channel is in single transfer mode, it performs DMA operations upon receipt of a transition on $\overline{DREQ}$ rather than in response to a $\overline{DREQ}$ level. Transitions occurring while the Hardware Mask bit is set will be stored and serviced when the Hardware Mask is cleared, assuming the Channel has not chained. The UDC will request the system bus 1 1/2 to 2 clocks after the receipt of any $\overline{DREQ}$, after which a minimum of one DMA iteration is unavoidable. $\overline{DREQ}$ transitions are only stored for the current DMA operation. If the channel performs a chain operation of single transfer mode, any $\overline{DREQ}$ transition stored for later service is cleared.

Timing Diagrams 1 and 2 show the minimum times when a new $\overline{DREQ}$ can be applied if it is to be serviced by the new DMA operation. Note in Diagram 1 the notation of First iteration and Last iteration. This means, for example, $\overline{DREQ}$ may be asserted during the write cycle T_1 of a Flowthru transaction, but may never be asserted during T_1 of a Flyby transaction because Flyby is done in one iteration.

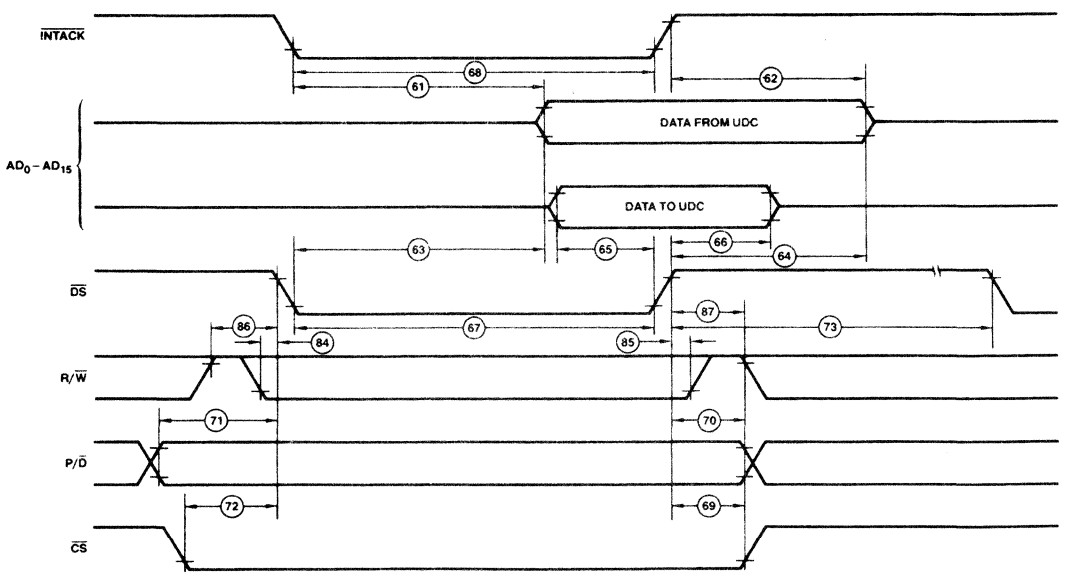
Set/Clear CIE, and IP Channel 1/Channel 2 (see Figure 19)

This command allows the user to either set or clear any combination of the CIE and IP bits in the selected channel's Status register. These bits control the operation of the channel's Interrupt structure and are described in detail in the "Interrupts" section. Setting the IP bit causes the Interrupt Save register to be loaded with the current Vector and Status. The IP bit is cleared to facilitate an efficient conclusion to the processing of an interrupt.

Set/Clear Flip Bit Channel 1/Channel 2 (Set: 62/63; Clear: 60/61)

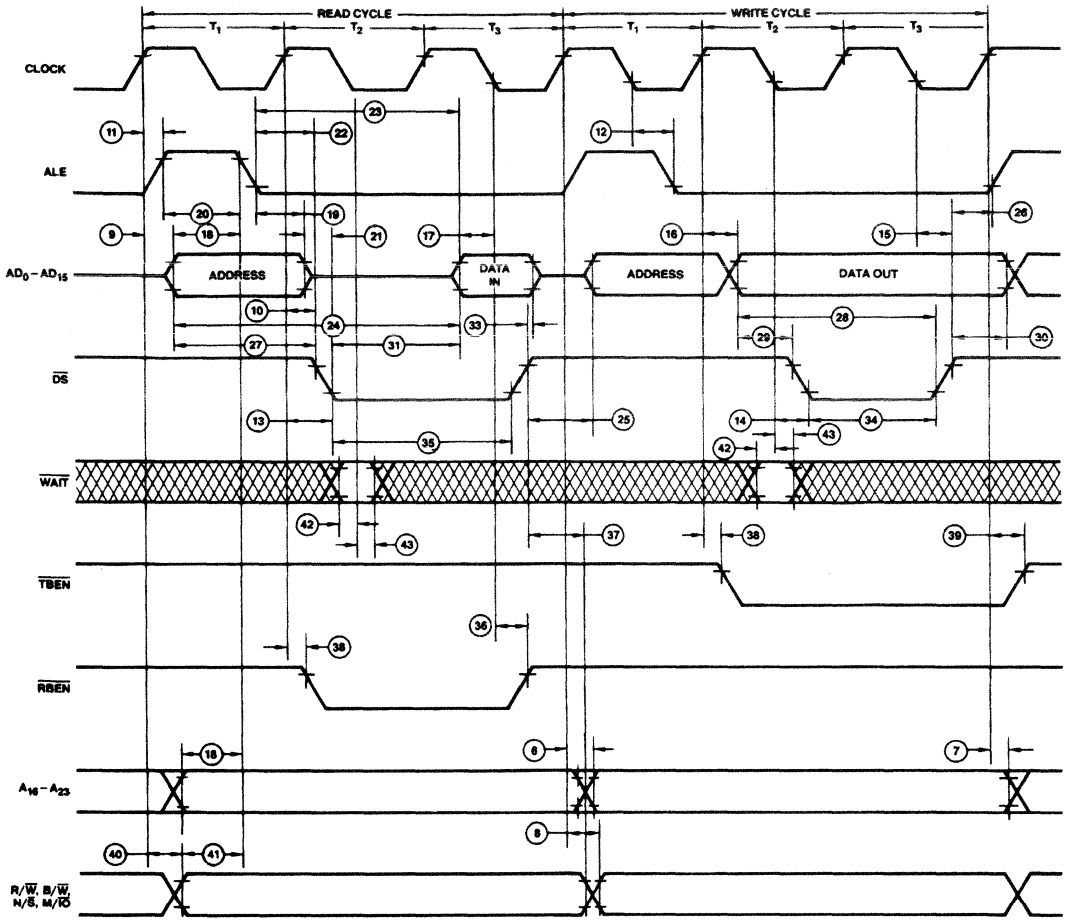
The Flip Bit in the selected channel's Mode register can be cleared and set by this command. This allows the user to reverse the source and destination and thereby reverse the data transfer direction without reprogramming the channel. This command will be most useful when repetitive DMA operations are being performed by the channel, using Base-to-Current reloading for channel reinitialization and using this command to control the direction of transfer. Chaining new information into the Channel Mode register will, of course, overwrite the Flip bit.

TIMING DIAGRAM 8. AC Timing when UDC is a Bus Slave



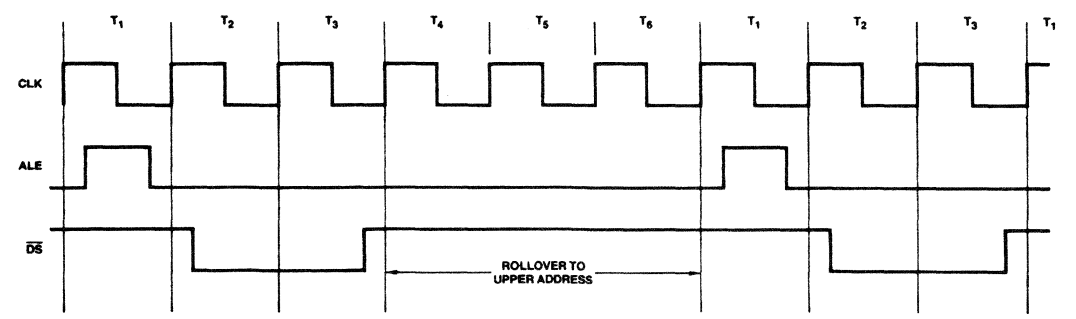
WF007580

TIMING DIAGRAM 9. AC Timing when UDC is a Bus Master



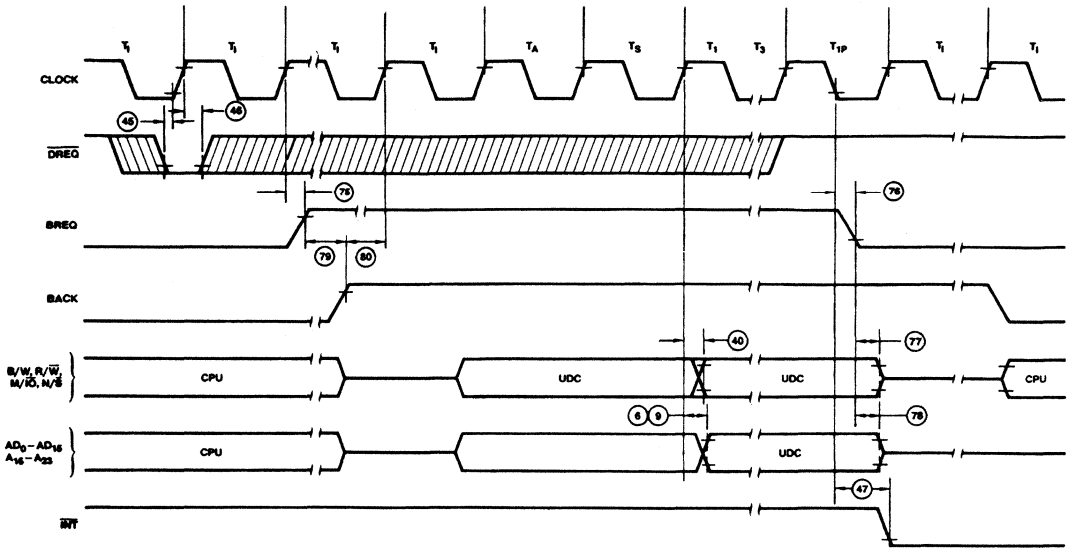
WF007591

TIMING DIAGRAM 10. Upper Address Rollover Timing



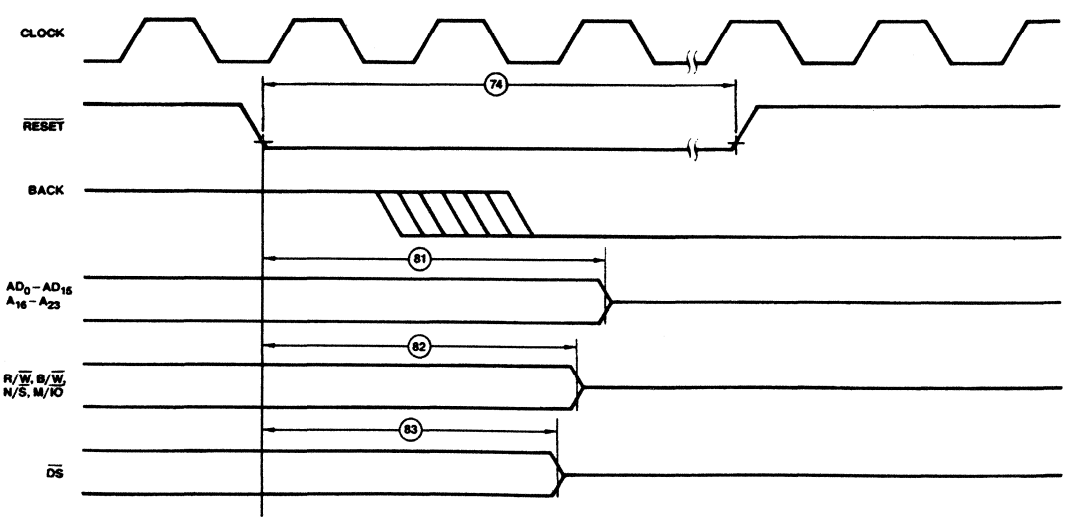
WF007600

TIMING DIAGRAM 11. Bus Exchange Timing

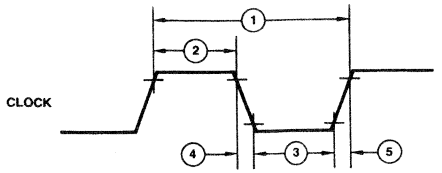


Notes: 1. Under no circumstance can BACK be removed prior to BREQ.
2. One extra ALE occurs each time the 9516 releases the bus. No $\overline{DS}$ accompanies it, so this should not present a problem.

TIMING DIAGRAM 12. Reset Timing

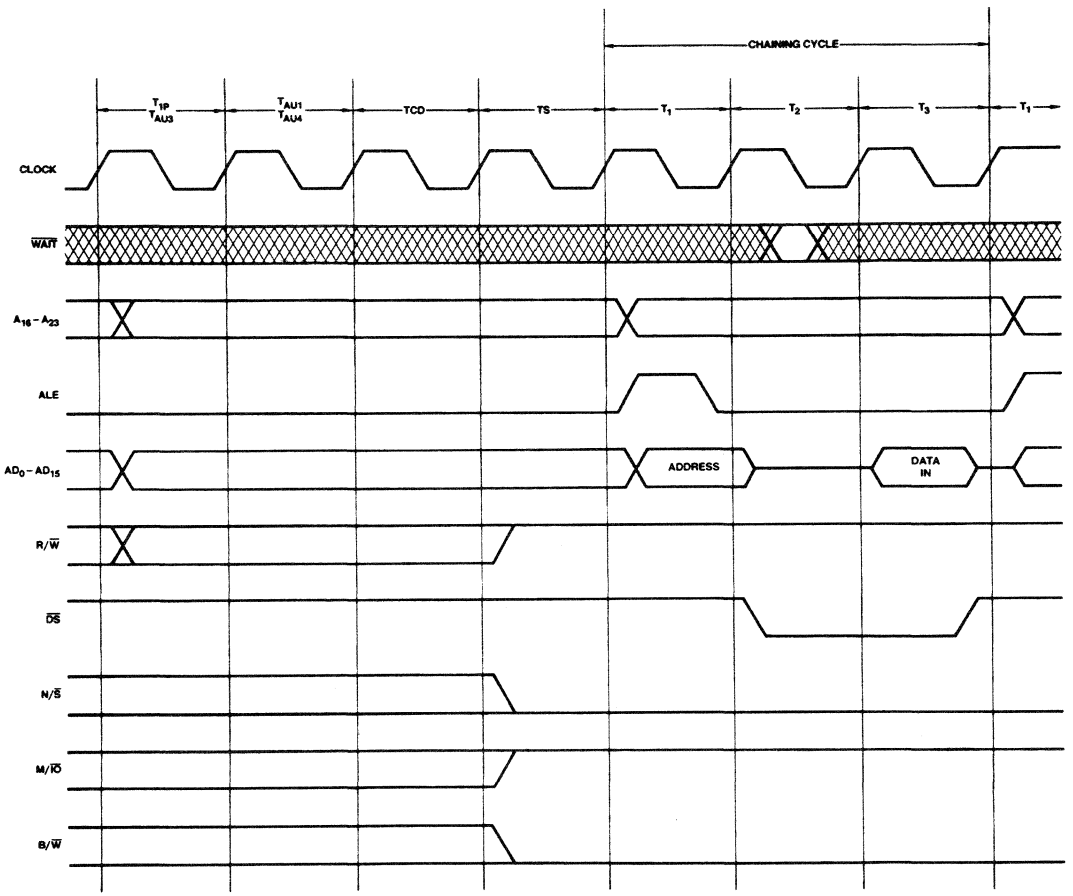


TIMING DIAGRAM 13. Clock Waveform



WF007630

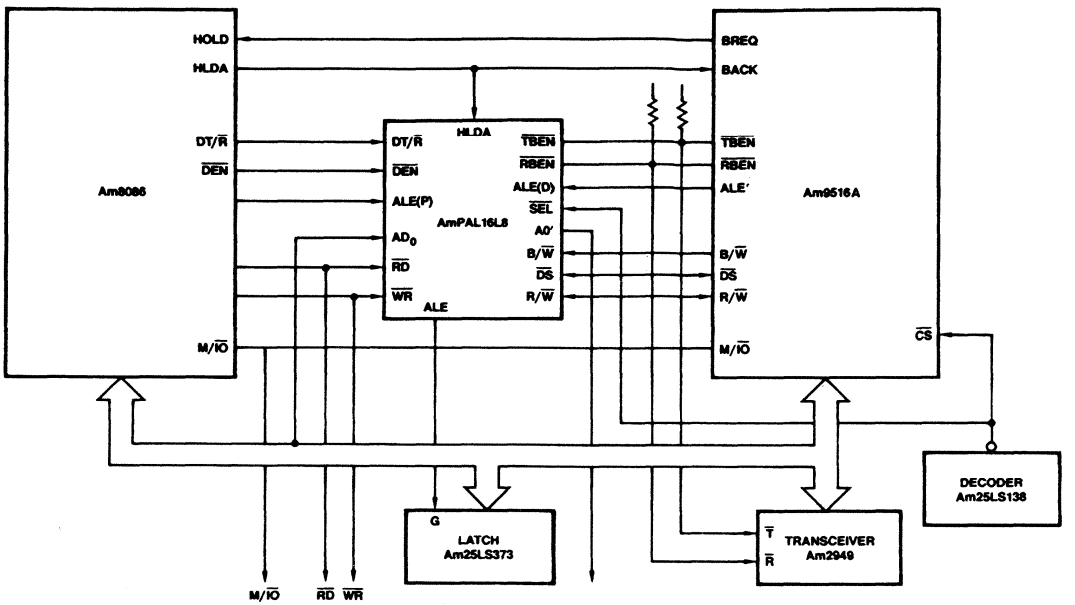
TIMING DIAGRAM 14. Timing During Chaining



WF007640

APPLICATIONS INFORMATION

Figures 21(a) and 21(b) show the configuration of an Am9516A UDC and an Am8086 microprocessor on the same board. Figure 22 shows a configuration for them when the Am9516A UDC is on a different board. The configuration of an Am9516A UDC to 68000 CPU interface is shown in Figure 23. An example of an Am8086 initialization program is shown in Figure 24. Figure 25 shows the reload table for chaining. The details of the Programmable Array Logic (PAL*) for those interfaces are described in Appendix B.



AF003161

Figure 21(a). Am9516A UDC to Am8086 CPU Interface (Minimum Mode)

AmPAL16L8 PALASM FILE

PAL16L8

Pat 001

Am9516A to Am8086 min mode interface chip

Advanced Micro Devices

NC ALED ALEP HLDA BW AD_0 DT/DEN/SEL GND

NC/RBEN/RD ALE A_0/RW/DS/WR/TBEN V_{CC}

If (/HLDA) DS = RD + WR

If (/HLDA) RW = DT

If (/HLDA) TBEN = /DT*/SEL*DEN

If (/HLDA) RBEN = DT*/SEL*DEN

If (HLDA) RD = /RW * DS

If (HLDA) WR = RW * DS

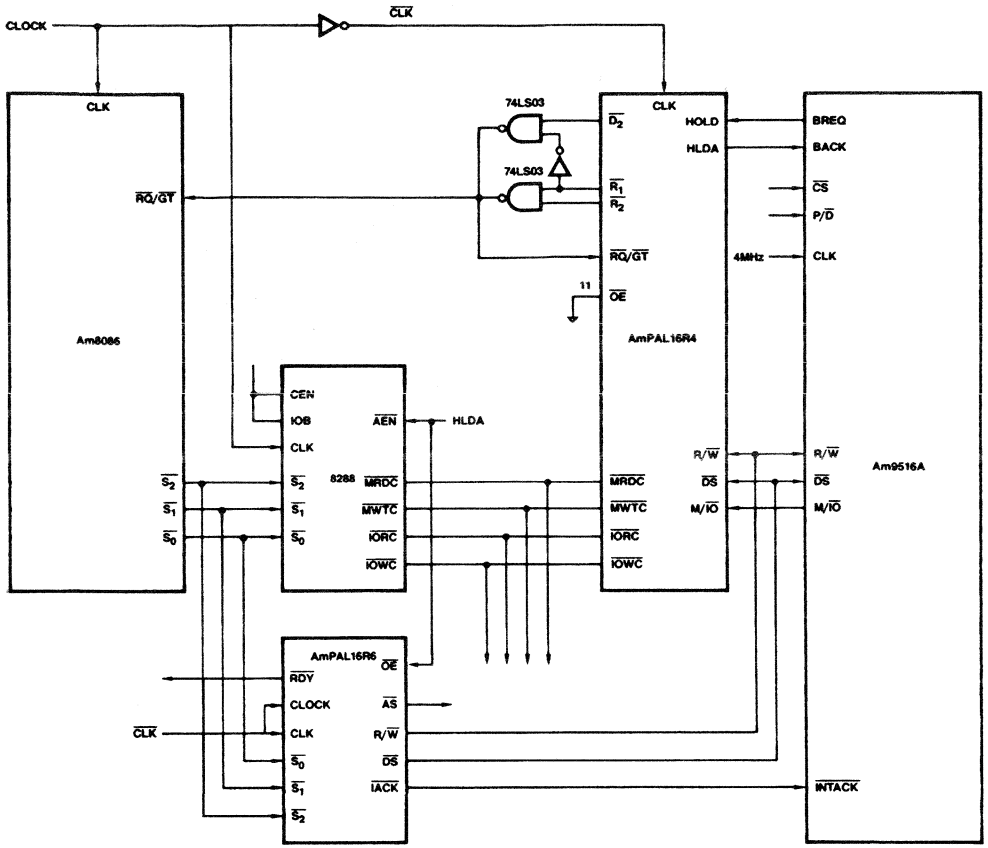
ALE = /ALEP * /ALED

A_0 = /AD_0*/BW*HLDA*ALED +
/AD_0*BW*HLDA*ALED +
/AD_0*/HLAD*ALEP + A_0*/ALEP + A_0*/ALED

DESCRIPTION

This PAL converts the control signals to interface the Am8086 in min mode to the Am9516A DMA controller. Another example shows how this is done in max mode.

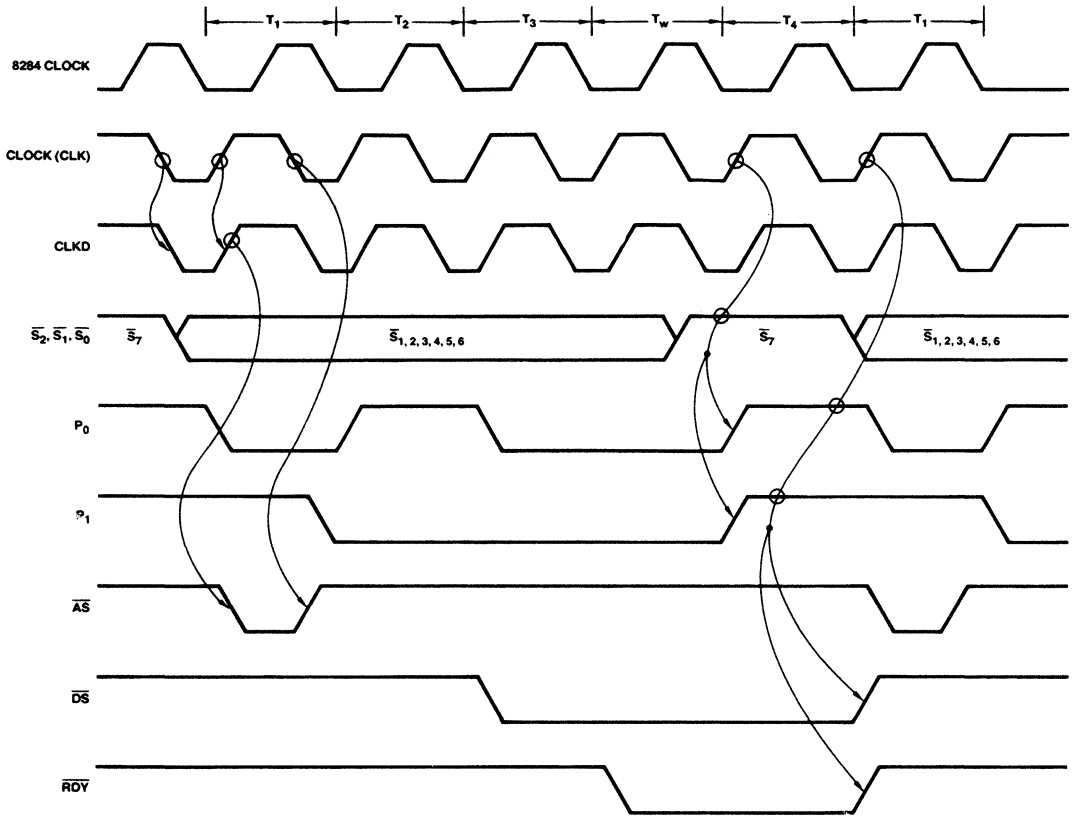
*PAL is a trademark of Monolithic Memories, Inc.



AF003171

Figure 21(b). Am9516A UDC to 8086 CPU Interface (Maximum Mode)

Timing Diagram of AmPAL16R6



WF007650

AmPAL16R6 PALASM FILE

```
AmPAL16R6
PAT003
Am8086 to AmZ85XX Peripheral Interface
Advanced Micro Devices

CLOCK RESET CLK/S0/S1/S2 NC NC NC GND
/OE/AS/P1/RW/DS/PO/LACK/RDY CLKD VCC

P0: = /RESET*S0*/P0*/P1  +
      /RESET*S1*/P0*/P1  +
      /RESET*S2*/P0*/P1  +
      /RESET*S0*P1       +
      /RESET*S1*P1       +
      /RESET*S2*P1

P1: = /RESET*P0*/P1      +
      /RESET*P1*S0       +
      /RESET*P1*S1       +
      /RESET*P1*S2

DS: = /IACK*/P0*P1*S0*/S1*S2 +
      /IACK*/P0*P1*/S0*S1*S2 +
      IACK*S0*S1*S2      +
      DS*P0*P1

RW: = S0*/S1

IACK: = /RESET*S0*S1*S2 + IACK*P0*P1*/DS +
        IACK*/P0*/P1

RDY: = /RESET*S0*/S1*S2*P0*P1 +
       /RESET*/S0*S1*S2*P0*P1 +
       /RESET*DS*RDY*P0*P1

/CLKD = CLK

AS = /CLKD*P0*/P1*/IACK*CLK
```

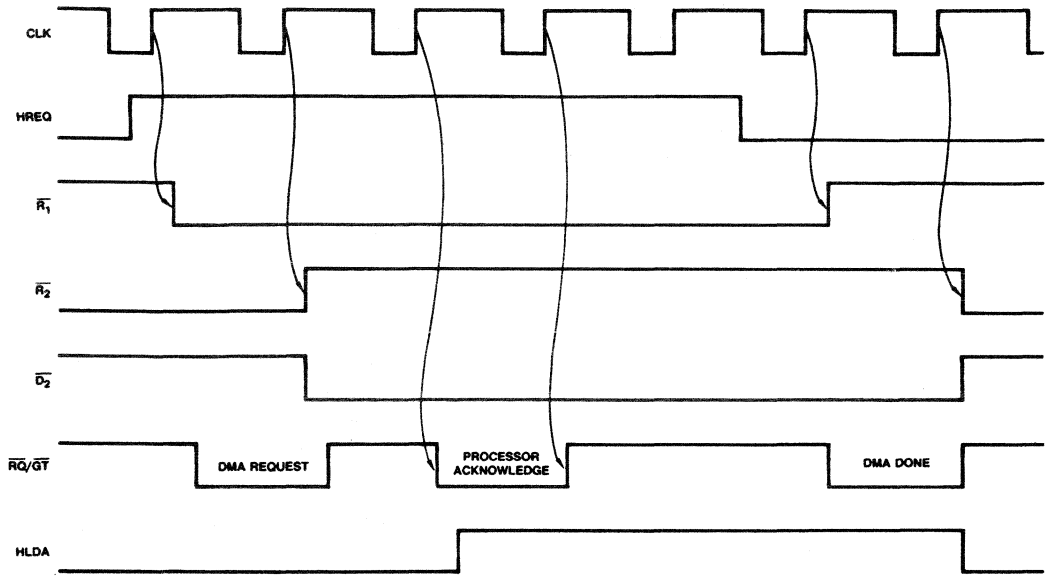
DESCRIPTION

This PAL translates Am8086 bus signals into compatible signals for the Am9516A. It is also applicable to AmZ85XX peripherals by altering/RW and /DS to/RD and /WR. One flip-flop is available to give the necessary delay to the falling edge of/WR.

Note: The CLK signal must be externally inverted for this design.

A >

AmPAL16R4 Timing



WF007660

AmPAL16R4 PALASM FILE

B > Type Am9516A PAL
PAL16R4

Am8086 to Am9516A interface

Advanced Micro Devices

CLK/RQGT HOLD NC NC NC/RW/DS MIO GND

/OE/MWTC/MRDC HLDA/D_2/R_2/R_1/IOWC/IORC V_{CC}

If (HLDA) IORC = /MIO*DS*/RW

If (HLDA) IOWC = /MIO*DS*RW

If (HLDA) MRDC = MIO*DS*/RW

If (HLDA) MWTC = MIO*DS*RW

R_1: = HOLD

R_2: = /R_1

D_2: = R_1

/HLDA: = /R_1 + /D_2*/HLDA + /RQGT*/HLDA

DESCRIPTION

This device converts the min mode signals HOLD and HLDA to the max mode /RQGT protocol. Additionally, it generates the 8288 equivalent control outputs /MRDC, /MWTC, /IORC, and /IOWC. This PAL was used to connect the Am9516A to the Am8086 in max mode.

B >

Note: If HOLD is taken away prior to grant pulse, design will not work correctly because the release pulse will overlap the grant pulse.

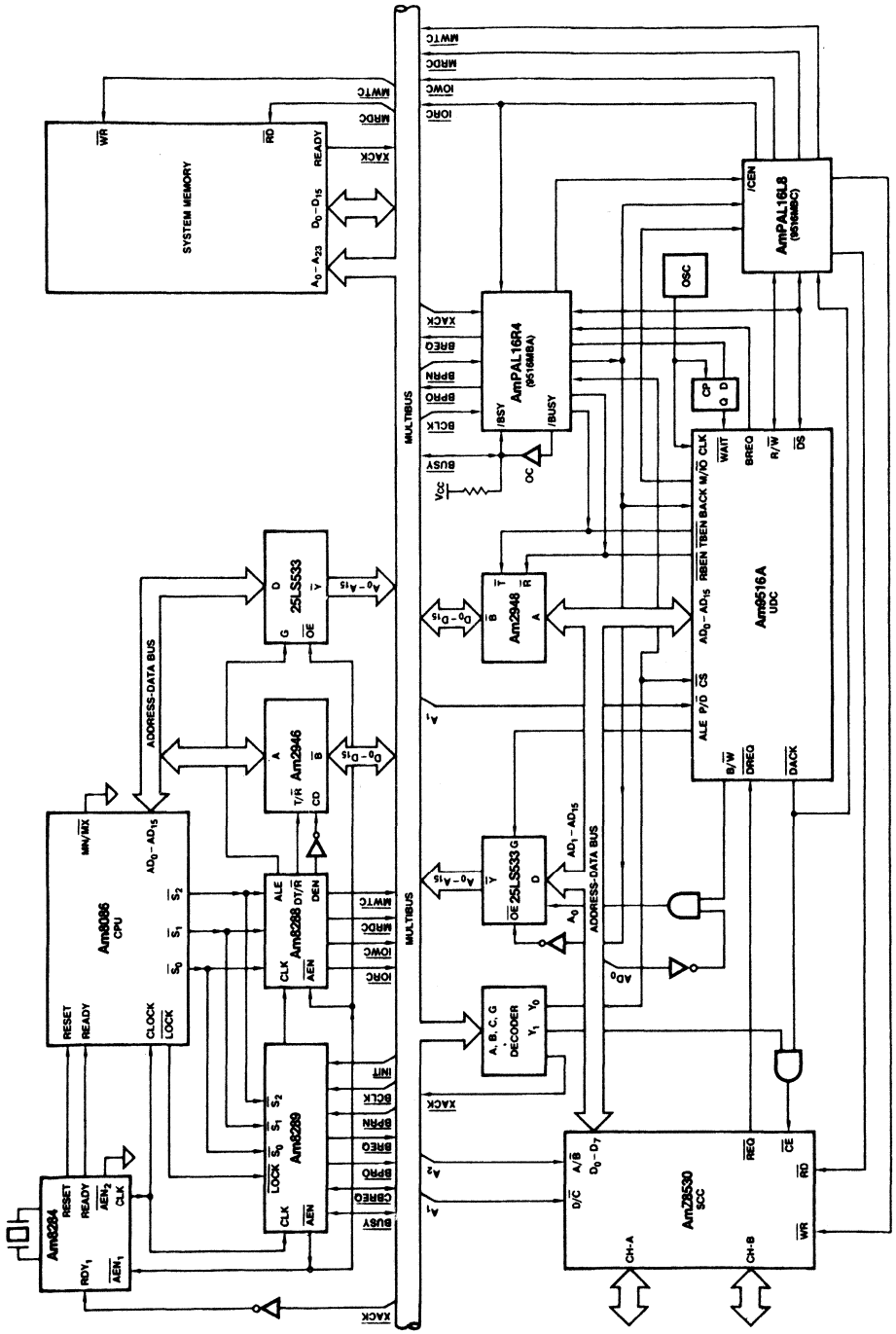
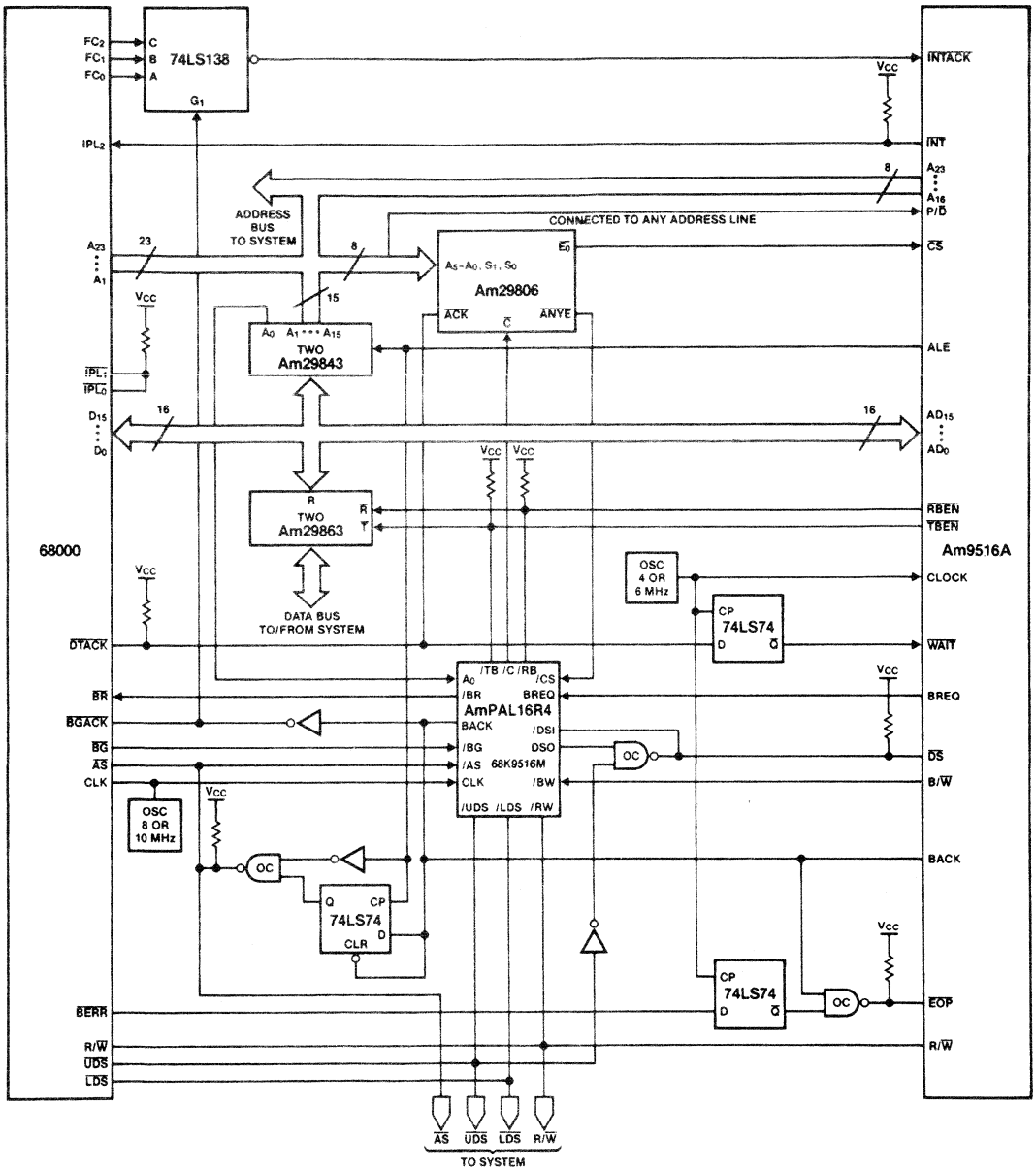


Figure 22. Am9516A to 8086 in a MULTIBUS Environment



BD003941

Figure 23. The Am9516A UDC to 68000 CPU Interface

PAL16R4 PAL DESIGN SPECIFICATION
PAT006 JOE BRCICH 9/01/83
68000 TO Am9516A INTERFACE WITH DATA HOLD CORRECTION
ADVANCED MICRO DEVICES

```
CLK   RW    AO    BREQ   /BG    /DSI    /AS    /BW   /CS
/OE   /LDS  /UDS  DSO    /C     BACK    /BR    /TB   /RB

IF (/BACK) RB  =  /CS * RW * UDS      +
                  /CS * RW * LDS
IF (/BACK) TB  =  /CS * /RW
IF (BACK) UDS  =  DSI * /A0 * /BW     +
                  BW * DSI
IF (BACK) LDS  =  DSI * A0 * /BW      +
                  BW * DSI
BR := BREQ * BG * BR * AS             +
      BREQ * /BG * /BACK
/BACK :=  /BREQ                         +
          /BREQ * /BG                   +
          /BREQ * AS                    +
          /BREQ * /BACK                 +
          /BG * /BACK                   +
          AS * /BACK
C := UDS * /BACK                        +
     LDS * /BACK
/DSO := BACK                            +
        /BACK * /RW * C
```

DESCRIPTION

IF BREQ*BACK IS TRUE THE Am9516A HAS THE BUS, OTHERWISE THE 68000 HAS THE BUS. THIS PAL CONNECTS THE Am9516A TO THE 68000 WITH ONE WAIT STATE DURING WRITES WHILE SHORTENING /DS TO ACHIEVE PROPER DATA HOLD TIME. IT ALSO CONVERTS THE BUS EXCHANGE PROTOCOL INTO 68000 FORMAT. THIS DESIGN ASSUMES NO OTHER BUS MASTERS IN THE SYSTEM. /RB AND /TB CONTROL THE TRANCEIVERS WHEN CPU IS BUS MASTER. /CS MUST BE FUNCTION OF ALL DEVICES CONNECTED TO THE CPU BUS NOT JUST THE Am9516A /CS AS SHOWN HERE.

The /CS to /DS set-up time of 30ns is met in the following ways:

(1) During a read cycle the only effect from not meeting this set-up time is that the data valid access time from the Am9516A will be delayed by a proportional amount. Since the minimum /DS Low width from the 10-MHz 68000 (during a read) is 193ns and the minimum /DS Low width to the Am9516A is 150ns, we have 43ns margin not counting gate delays which will further increase this margin.

(2) During a write cycle this is not an issue since the /DS comes later and is stretched longer due to the Wait state.

AmPAL16R4 68K9516M PALASM File

```
PAL16L8                    PAL DESIGN SPECIFICATION 9516MBC
PAT 003                          JOE BRCICH 26 JULY 84
MULTIBUS CONTROL FOR Am9516A
ADVANCED MICRO DEVICES

BACK  MIO   NC    NC    /DACK  NC    NC    NC    /CEN   GND
NC    /RD   /IORC /DS   /MWTC  /MRDC /IOWC /RW   /WR    VCC

IF (BACK) IORC  = /MIO * DS * /RW * CEN

IF (BACK) IOWC  = /MIO * DS * RW * CEN

IF (BACK) MRDC  = MIO * DS * /RW * CEN

IF (BACK) MWTC  = MIO * DS * RW * CEN

RD = DACK * RW * BACK +
     IORC * /BACK

WR = DACK * /RW * BACK +
     IOWC * /BACK

IF (/BACK) DS = IORC + IOWC

IF (/BACK) RW = IOWC
```

DESCRIPTION

THIS PAL CONVERTS MULTIBUS SIGNALS INTO Am9516A COMPATIBLE SIGNALS AND VICE VERSA. IT ALSO SUPPORTS THE 8530 IN FLYBY MODE.

MULTIBUS Control for Am9516A (AmPAL16L8)

```
PAL16R4                    PAL DESIGN SPECIFICATION 9516MBA
PAT 004                          JOE BRCICH 30 July 84
MULTIBUS ARBITER FOR Am9516A
ADVANCED MICRO DEVICES

/BCLK  /XACK  BRQ   /BSY  /BPRN  /DS   NC    /IORC  /CS   GND
/OE    /RBEN  /TBEN BACK  /CEN   /BREQ /BUSY /BPRO  /WAIT VCC

IF (/BACK) TBEN  = IORC * CS

IF (/BACK) RBEN  = /IORC * CS

WAIT  = /XACK * BACK

BREQ := BRQ

BPRO  = /BRQ * BPRN

/BACK := /BUSY

BUSY := BREQ * BPRN * /BSY * /BUSY  +
        BREQ * BUSY * BPRN          +
        BREQ * BUSY

CEN := BACK
```

DESCRIPTION

/CEN DELAYS THE COMMANDS TO MEET THE MULTIBUS REQUIREMENT THAT ADDRESS AND DATA BE VALID AT LEAST 50NS PRIOR TO CONTROL ACTIVE. /IOWC WAS NOT USED SINCE USING /IORC IMPROVES HOLD TIME. THIS DESIGN DOES NOT SUPPORT THE /CBRQ FUNCTION.

MULTIBUS Arbiter for Am9516A (AmPAL16R4)

```
•                   •
•                   •
•                   •
B0 38               MOV     AL,38H      ;LOADING POINTER OF MASTER
E6 12               OUT     12H         ;MODE REGISTER
B8 07 00            MOV     AX,007H     ;LOADING MMR CODE
E7 10               OUTW    10H         ;
B0 26               MOV     AL,26H      ;LOADING POINTER OF CHAIN
E6 12               OUT     12H         ;ADDRESS REGISTER'S SEGMENT
B8 00 00            MOV     AX,0000H    ;LOADING SEGMENT OF CAR-1
E7 10               OUTW    10H         ;
B0 22               MOV     AL,22H      ;LOADING POINTER OF CHAIN
E6 12               OUT     12H         ;ADDRESS REGISTER'S OFFSET
B8 20 10            MOV     AX,1020H    ;LOADING OFFSET OF CAR-1
E7 10               OUTW    10H         ;
B0 2C               MOV     AL,2CH      ;LOADING POINTER OF COMMAND
E6 12               OUT     12H         ;REGISTER
B0 A0               MOV     AL,A0H      ;LOADING "START CHAIN" COMMAND
E6 10               OUT     10H         ;ISSUING "START CHAIN" COMMAND
•                   •
•                   •
•                   •
```

Figure 24. Initialization Program for 8086 CPU

Notes: The P/$\overline{D}$ input is connected to A1 line; $\overline{CS}$ is decoded from A7 through A4 (all O).

ADDRESS	0	2	4	6	8	A	C	E
1000	0000	1020	0000	1020	0007	0005	0006	0005
1010	0002	AAAA	0009	00A0	0004	0042	0042	0001
1020	03FF	0000	1F00	0000	1060	0010	0000	1F00
1030	0000	1080	0012	0000	FFFF	0001	0000	8020
1040	0000	1020	1111	1111	0000	FFFF	2004	0000
1050	0010	0000	0000	1020	0018	1020	2222	1007
1060	CACA	CACA	CACA	CACA	CACA	CACA	CACA	CACA
1070	CACA	CACA	CACA	CACA	CACA	CACA	CACA	CACA

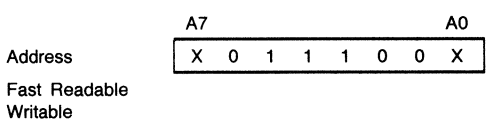
Reload Word

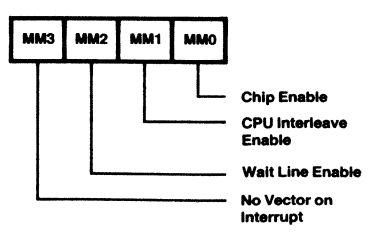
TB000084

Figure 25. Reload Table for Chaining

APPENDIX A
UDC REGISTER SUMMARY

Master Mode Register





DF003470

Miscellaneous Registers

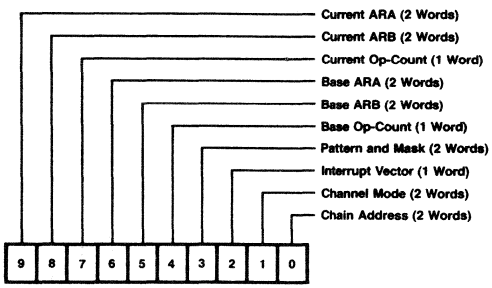
Address	A7							A0		
	X	0	1	1	0	0	1	X	Current Operation Count	CH1
	X	0	1	1	0	0	0	X	Current Operation Count	CH2
	X	0	1	1	0	1	1	X	Base Operation Count	CH1
	X	0	1	1	0	1	0	X	Base Operation Count	CH2
	X	1	0	0	1	0	1	X	Pattern	CH1
	X	1	0	0	1	0	0	X	Pattern	CH2
	X	1	0	0	1	1	1	X	Mask	CH1
	X	1	0	0	1	1	0	X	Mask	CH2

Chain Loadable
Writable
Pattern and Mask – Slow Readable
Operation Count – Fast Readable

D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

Chain Control Register

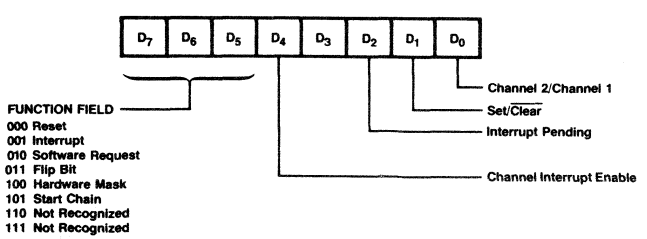
Chain Loadable Only



DF003480

Command Register

	A7							A0	
Address	X	0	1	0	1	1	1	X	CH1
Writable Only	X	0	1	0	1	1	0	X	CH2

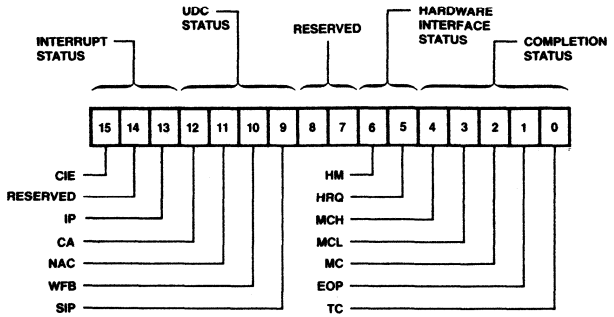


DF003490

Status Register

	A7						A0	
Address	X	0	1	0	1	1	X	CH1
	X	0	1	0	1	1	0	X	CH2

Fast Readable

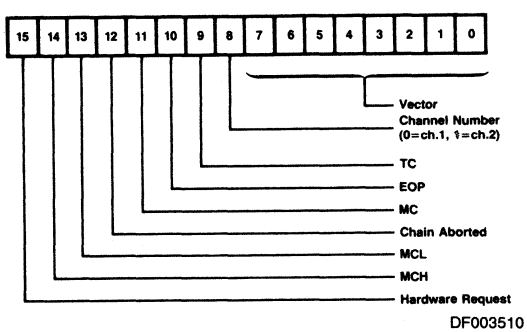


DF003500

Interrupt Save Register

	A7						A0	
Address	X	0	1	0	1	0	1	X	CH1
	X	0	1	0	1	0	0	X	CH2

Fast Readable

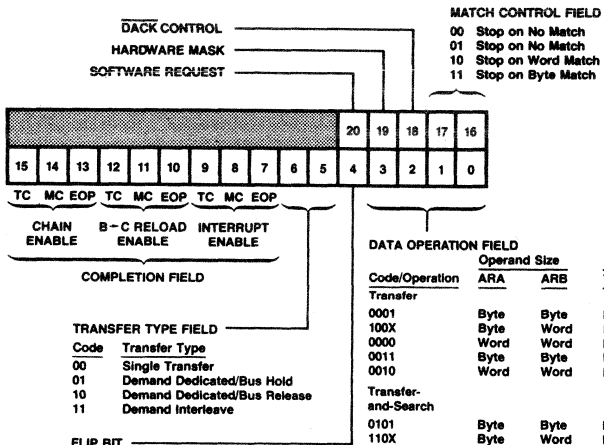


DF003510

Channel Mode Register

A7							A0	
X	1	0	1	0	1	1	X	High CH1
X	1	0	1	0	1	0	X	High CH2
X	1	0	1	0	0	1	X	Low CH1
X	1	0	1	0	0	0	X	Low CH2

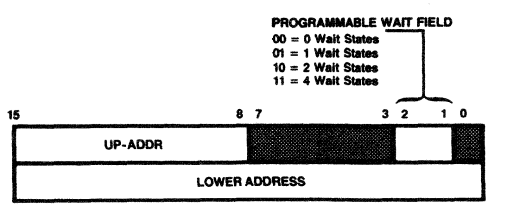
Chain Loadable
Writable (Lower 16 bits)
Slow Readable



DF003520

Chain Address Register

	A7							A0	
Address	X	0	1	0	0	1	1	X	Up-Addr CH1
	X	0	1	0	0	1	0	X	Up-Addr CH2
	X	0	1	0	0	0	1	X	Low-Addr CH1
	X	0	1	0	0	0	0	X	Low-Addr CH2

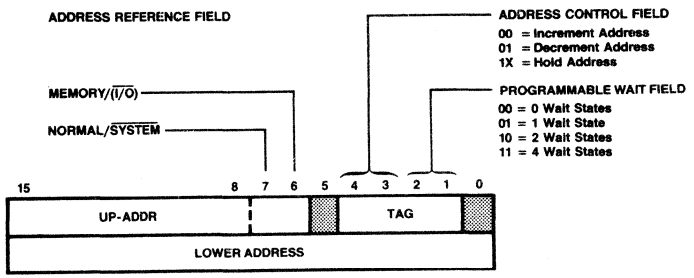


DF003530

Address Registers

Address	A7							A0		
	X	0	0	1	1	0	1	X	Current ARA Up-Addr/Tag	CH1
	X	0	0	1	1	0	0	X	Current ARA Up-Addr/Tag	CH2
	X	0	0	0	1	0	1	X	Current ARA Low-Addr	CH1
	X	0	0	0	1	0	0	X	Current ARA Low-Addr	CH2
	X	0	0	1	0	0	1	X	Current ARB Up-Addr/Tag	CH1
	X	0	0	1	0	0	0	X	Current ARB Up-Addr/Tag	CH2
	X	0	0	0	0	0	1	X	Current ARB Low-Addr	CH1
	X	0	0	0	0	0	0	X	Current ARB Low-Addr	CH2
	X	0	0	1	1	1	1	X	Base ARA Up-Addr/Tag	CH1
	X	0	0	1	1	1	0	X	Base ARA Up-Addr/Tag	CH2
	X	0	0	0	1	1	1	X	Base ARA Low-Addr	CH1
	X	0	0	0	1	1	0	X	Base ARA Low-Addr	CH2
	X	0	0	1	0	1	1	X	Base ARB Up-Addr/Tag	CH1
	X	0	0	1	0	1	0	X	Base ARB Up-Addr/Tag	CH2
	X	0	0	0	0	1	1	X	Base ARB Low-Addr	CH1
	X	0	0	0	0	1	0	X	Base ARB Low-Addr	CH2

Chain Loadable
Fast Readable and Writable

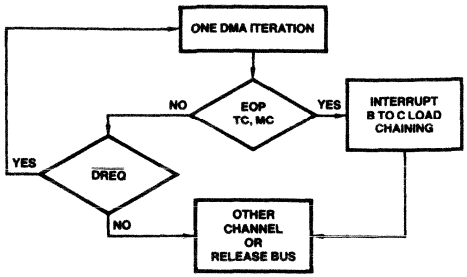


DF003540

APPENDIX B

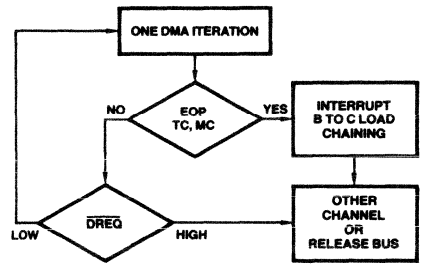
Flow Charts of DMA Operations:
Figure B-1 shows the basic DMA operations with software or hardware request. The Demand Interleave operations are shown in Figure B-2.

(a) Single Operation



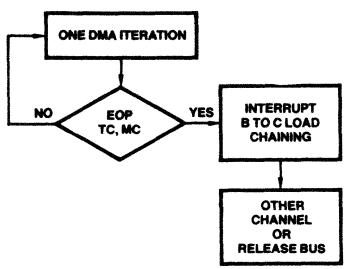
PF001250

(c) Demand Dedicated with Bus Release (Hardware Request)



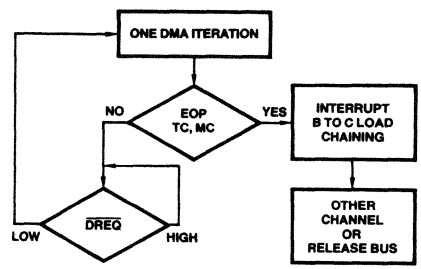
PF001260

(b) Demand Operation when Software Requesting



PF001270

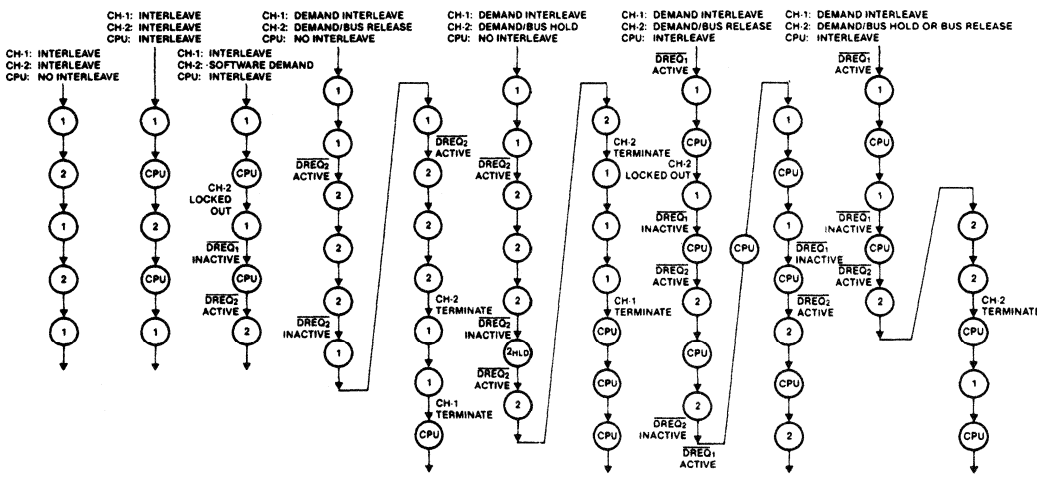
(d) Demand Dedicated with Bus Hold (Hardware Request)



PF001280

Figure B-1. Basic DMA Operations of Am9516A UDC

NTMultibus is a trademark of Intel Corp.**

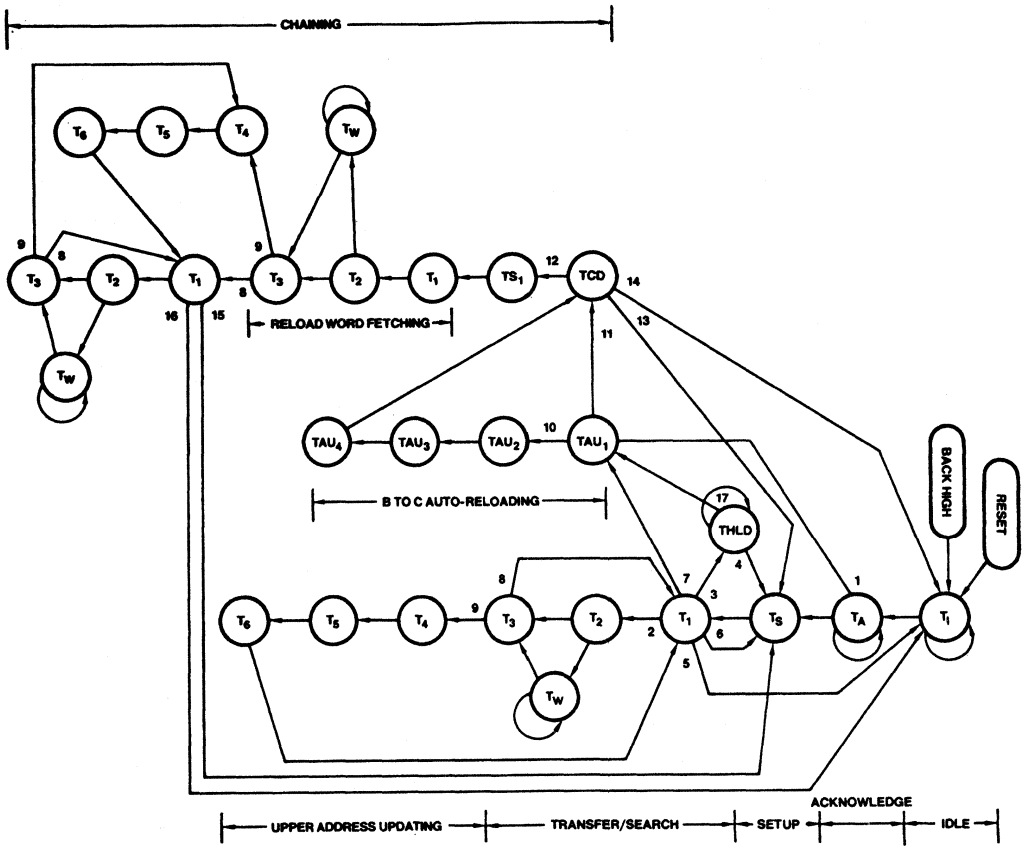


PF001300

Figure B-2. Demand Interleave Operations of Am9516A UDC

APPENDIX C

Am9516A STATE DIAGRAM



AF003180

Am9516A INTERNAL OPERATION ROUTINES

1. "Start Chain" command issued or start updating routine* after an interrupt has been served.**
2. Normal DMA operation.
3. Demand with Bus hold while DREQ is inactive.
4. DREQ is active while bus held.
5. Single transfer, CPU interleave enabled, or demand with bus release while current DREQ is inactive and no DMA request is pending.
6. Single Transfer or Demand/Bus release while current DREQ is inactive, but the other DMA request is pending.
7. TC, MC or EOP termination occurs.
8. One DMA or chain transaction is done and the upper address is not changed.
9. One DMA or chain transaction is done and the upper address is changed.
10. Base-to-current auto-reloading is enabled.
11. Base-to-current auto-reloading is disabled.
12. Chaining is enabled.
13. Chaining is disabled and another DMA request is pending.
14. Chaining is disabled and no DMA request is pending.
15. Chaining ends and another DMA request is pending.
16. Chaining ends and no DMA request is pending.
17. EOP termination of Bus Hold.

*Updating routine includes base-to-current auto-reloading and chaining.

**When a second interrupt is to be issued before the first interrupt is acknowledged, the SIP bit of a Status register is set and the channel relinquishes the bus until the first interrupt has been served. If the channel was to perform the updating routine, once the SIP bit is cleared, DTC will reacquire the bus and perform the appropriate operation (i.e., 1).

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
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

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ± 5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

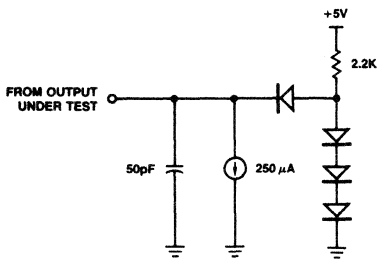
Parameters	Description	Test Conditions	Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator	3.8	V_{CC} +0.3	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator	−0.5	0.45	Volts
V_{IH1}	Input High Voltage	All Pins Except 2, 36, 37, 38, 47, 48	2.0	V_{CC} +0.3	Volts
V_{IH2}	Input High Voltage	Pins 2, 36, 37, 38, 47, 48	2.2	V_{CC} +0.3	Volts
V_{IL}	Input Low Voltage		−0.5	0.8	Volts
V_{OH1}	Output High Voltage	I_{OH} = −250μA (Except Pins 1, 32, 33, 38)	2.4		Volts
V_{OH2}	Output High Voltage	I_{OH} = −200 μA, Pins 1, 32, 33, 38	2.0		Volts
V_{OL}	Output Low Voltage	I_{OL} = 3.2 mA		0.45	Volts
I_{IL}	Input Leakage	$V_{SS} \leq V_{IN} \leq V_{CC}$		±10	μA
I_{OL}	Output Leakage	$V_{SS} \leq V_{OUT} \leq V_{CC}$		±10	μA
I_{CC}	V_{CC} Supply Current	T_A = 0°C		350	mA
		T_A = 70°C		200	mA
CCLK	C	Unmeasured pins returned to ground. f = 1 MHz over specified temperature range.		25	pF
C_{IN}	Input Capacitance (Except Pin 46)	Unmeasured pins returned to ground. f = 1MHz over specified temperature range.		10	pF
C_{OUT}	Output Capacitance			15	pF
$C_{I/O}$	Bidirectional Capacitance			20	pF

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

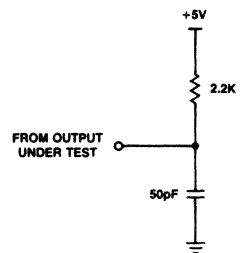
+4.75 ≤ V<CC> ≤ +5.25V
GND = 0V
0°C ≤ T_A ≤ +70°C

Standard Test Load



TC002010

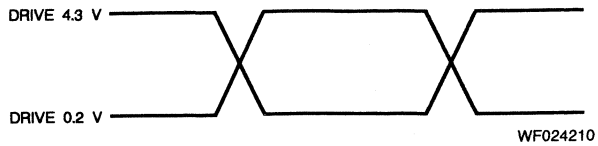
Open-Drain Test Load



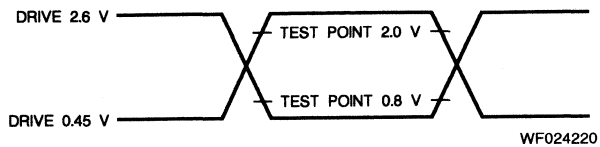
TC002020

See Section 6 for Thermal Characteristics Information.

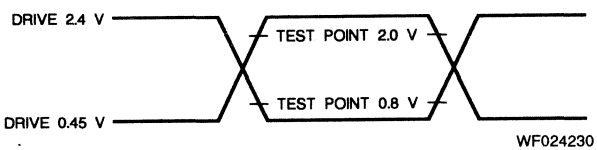
SWITCHING TEST WAVEFORMS



A. External CLOCK Generator



B. BACK, DREQ1, DREQ2, RESET, INTACK, and EOP only



C. ALL pins <u>except</u> BACK, DREQ1, DREQ2, RESET, INTACK, and EOP.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
TIMING FOR UDC AS BUS MASTER

			Preliminary								
			4MHz		6MHz		8MHz		10MHz		
Number	Parameters	Description	Min	Max	Min	Max	Min	Max	Min	Max	Units
1	TcC	Clock Cycle Time	250	2000	165	2000	125	2000	100	2000	ns
2	TwCh	Clock Width (HIGH)	105	1000	70	1000	55		45		ns
3	TwCl	Clock Width (LOW)	105		70		55		45		ns
4	TfC	Clock Fall Time		20		10		5		5	ns
5	TrC	Clock Rise Time		20		15		10		5	ns
6	TdC(AUv)	Clock RE to Upper Address (A_{16}-A_{23}) Valid Delay		90		80		60		50	ns
7	ThC(AUv)	Clock RE to Upper Address Valid Hold Time	5		5		5		5		ns
8	TdC(ST)	Clock RE to R/$\overline{W}$ and B/$\overline{W}$ Valid Delay		110		90		60		50	ns
9	TdC(A)	Clock RE to Lower Address (A_0-A_{15}) Valid Delay		90		90		60		50	ns
10	TdC(Az)	Clock RE to Lower Address (A_0-A_{15}) Float Delay		60		60		50		40	ns
11	TdC(ALr)	Clock RE to ALE RE Delay		70		60		50		30	ns
12	TdC(AL)	Clock FE to ALE FE Delay		70		60		50		40	ns
13	TdC(DS)	Clock RE to $\overline{DS}$ (Read) FE Delay		60		60		50		40	ns
14	TdC(DSf)	Clock FE to $\overline{DS}$ (Write) FE Delay		60		60		50		45	ns
15	TdC(DSr)	Clock FE to $\overline{DS}$ RE Delay		60		60		50		40	ns
16	TdC(DO)	Clock RE to Data Out Valid Delay		90		90		65		60	ns
17	TsDI(C)	Data in to Clock FE Set-up Time	20		15		10		10		ns
18	TdA(AL)	Address Valid to ALE FE Delay	50		35		20		20		ns
19	ThAL(A)	ALE FE to Lower Address Valid Hold Time	60		40		30		30		ns
20	TwAL	ALE Width (HIGH)	80		60		45		40		ns
21	TdAz(DS)	Lower Address Float to $\overline{DS}$ LOW Delay	0		0		0		0		ns
22	TdAL(DS)	ALE FE to $\overline{DS}$ (Read) FE Delay	75		35		35		35		ns
23	TdAL(DI)	ALE FE to Data in Required Valid Delay		300		215		190		150	ns
24	TdA(DI)	Address Valid to Data in Required Valid Delay		410		305		240		190	ns
25	TdDS(A)	$\overline{DS}$ RE to Address Active Delay	80		45		30		20		ns
26	TdDS(AI)	$\overline{DS}$ RE 10 ALE RL Delay	75		40		40		35		ns
27	TdA(DS)	Address Valid to $\overline{DS}$ (Read) FE Delay	160		110		90		70		ns
28	TdDO(DSr)	Data Out Valid to $\overline{DS}$ RE Delay	230		150		125		80		ns
29	TdDO(DSf)	Data Out Valid to $\overline{DS}$ FE Delay	55		35		20		15		ns
30	ThDS(DO)	$\overline{DS}$ RE to Data Out Valid Hold Time	85		45		40		25		ns
31	TdDS(DI)	$\overline{DS}$ (Read) FE to Data in Required Valid Delay		205		155		125		100	ns
33	ThDI(DS)	$\overline{DS}$ RE to Data in Hold Time	0		0		0		0		ns
34	TwDSmw	$\overline{DS}$ (Write) Width (LOW)	185		110		105		80		ns
35	TwDSmr	$\overline{DS}$ (Read) Width (LOW)	275		220		160		130		ns
36	TdC(RBr)	Clock FE to $\overline{RBEN}$ RE Delay*		70		65		50		30	ns
37	ThDS(ST)	$\overline{DS}$ RE to B/$\overline{W}$, N/$\overline{S}$, R/$\overline{W}$ and M/$\overline{IO}$ Valid Hold Time	70		45		40		25		ns
38	TdC(TRf)	Clock RE to $\overline{TBEN}$ or $\overline{RBEN}$ FE Delay		60		60		50		35	ns
39	TdC(TRr)	Clock RE to $\overline{TBEN}$ RE Delay		60		60		45		45	ns
40	TdC(ST)	Clock RE to M/$\overline{IO}$ and N/$\overline{S}$ Valid Delay		90		75		65		50	ns
41	TdS(AL)	R/$\overline{W}$, M/$\overline{IO}$, B/$\overline{W}$ and N/$\overline{S}$ Valid to ALE FE Delay	60		35		20		20		ns
42	TsWT(C)	$\overline{WAIT}$ to Clock FF Set-up Time	20		20		10		10		ns
43	ThWT(C)	$\overline{WAIT}$ to Clock FE Hold Time	20		20		35		35		ns
44	TwDRQ	$\overline{DREQ}$ Pulse Width (Single Transfer Mode)	20		20		20		20		ns
45	TsDRQ(C)	$\overline{DREQ}$ Valid to Clock RE Set-up Time	60		50		30		20		ns
46	ThDRQ(C)	Clock RE to $\overline{DREQ}$ Valid Hold Time	20		20		20		20		ns
47	TdC(INTf)	Clock FE to $\overline{INT}$ FE Delay		150		150		105		105	ns

*These must not occur simultaneously.

Note: RE = rising edge
FE = falling edge

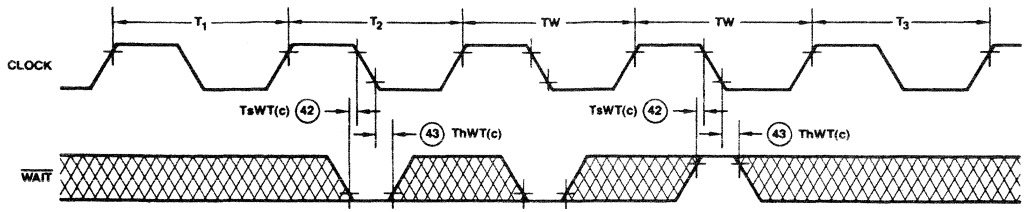
Am9516A CLOCK-CYCLE-TIME-DEPENDENT CHARACTERISTICS

The parameters listed below are also shown in the Switching specification. However, they are dependent on the actual values of the clock periods. The equations below define that dependence so that the exact limit for these parameters may be determined for any given system in relation to its specific clock characteristics.

Number	Parameters	Derivation
18	TdA (AL)	0.5TcC – #9 + (#12 – tr)
19	ThAL (A)	0.5TcC – #12 (ALE FE @ 0.8V) + #10
21	TdAz (DS)	#13 – #10
22	TdAL (DS)	0.5TcC – #12 + #13
23	TdAL (DI)	2TcC – #12 – #17
24	TdA (DI)	2.5TcC – #9 – #17
25	TdDS (A)	0.5TcC – #15 + #9
26	TsDS (AL)	0.5TcC – #15 + #11 (ALE RE)
27	TdA (DS)	TcC – #9 + #13
28	TdDO (DSr)	1.5TcC – #16 + #15
29	TdDO (DSf)	0.5TcC – #16 + #14
30	ThDS (DO)	0.5TcC – #15 + #32
31	TdDS (DI)	1.5TcC – #13 – #17
34	TwDSmw	TcC – #14 + #15
35	TwDSmr	1.5TcC – #13 + #15
37	ThDS (ST)	0.5TcC – #15 + (#40 – tr)
41	TdS (AL)	0.5TcC – #40 + (#12 – tr)

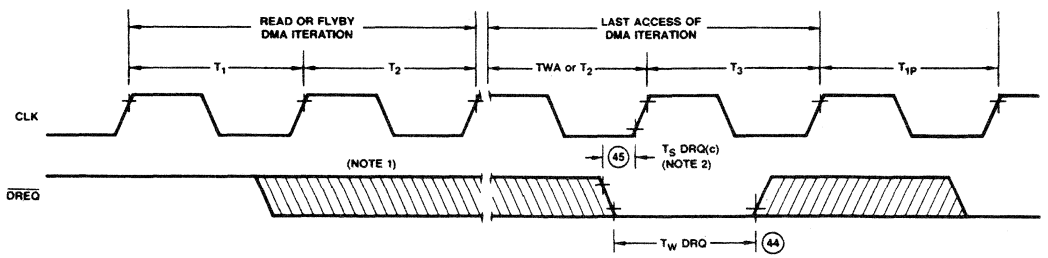
NOTE: tr (nominal) = 10ns
#32 CLCK RE to Data Out Not Valid Delay = 20ns (4 and 6 MHz)

$\overline{\text{WAIT}}$ Timing



WF007680

Sampling $\overline{\text{DREQ}}$ During Single Transfer DMA Operations



WF007670

Notes: 1. HIGH-to-LOW $\overline{\text{DREQ}}$ transitions will only be recognized after the HIGH-to-LOW transition of the clock during T_1 of a read or flyby DMA iteration.

2. A HIGH-to-LOW $\overline{\text{DREQ}}$ transition must meet the conditions in Note 1 and must occur TsDRQ(c) before state T_3 of the last access of the DMA iteration if the channel is to retain bus control and immediately start the next iteration. $\overline{\text{DREQ}}$ may go HIGH before TsDRQ(c) if it has met the TwDRQ parameter.

3. Flyby and Search transactions have only a single access; parameter TsDRQ(c) should be referenced to the start of T_3 of the access. All other operations will always have two or three accesses per iteration.

SWITCHING CHARACTERISTICS
UDC AS BUS SLAVE BUS EXCHANGE

			Preliminary								
			4MHz		6MHz		8MHz		10MHz		
Number	Parameters	Description	Min	Max	Min	Max	Min	Max	Min	Max	Units
61	TdIN(DO)	$\overline{INTACK}$ FE to Data Output Valid Delay		135		135		120		110	ns
62	TdIN(DOz)	$\overline{INTACK}$ RE to Data Output Float Delay		80		80		45		35	ns
63	TdDS(DO)	$\overline{DS}$ FE (IOR) to Data Output Driven Delay		135*		135		120		110	ns
64	TdDS(DOz)	$\overline{DS}$ RE (IOR) to Data Output Float Delay		80		80		45		40	ns
65	TsDI(DS)	Data Valid to $\overline{DS}$ RE (IOW) Set-up Time	40		40		40		35		ns
66	ThDS(DI)	$\overline{DS}$ RE (IOW) to Data Valid Hold Time	40		30		0		0		ns
67	TwDS	$\overline{DS}$ Low Width	150*		150*		125		100		ns
68	TwIN	$\overline{INTACK}$ Low Width	150		150		125		100		ns
69	ThDS(CS)	$\overline{DS}$ RE to $\overline{CS}$ Valid Hold Time	20		20		15		10		ns
70	ThDS(PD)	$\overline{DS}$ RE to P/$\overline{D}$ Valid Hold Time	20		20		15		10		ns
71	TsPD(DS)	P/$\overline{D}$ Valid to $\overline{DS}$ FE Set-up Time (IOR)	10		10		10		10		ns
		P/$\overline{D}$ Valid to $\overline{DS}$ FE Set-up Time (IOW)	50		50		40		30		
72	TsCS(DS)	$\overline{CS}$ Valid to $\overline{DS}$ FE Set-up Time	30		30		20		10		ns
73	TrDS	$\overline{DS}$ RE to $\overline{DS}$ FE Recovery Time (for Commands Only)	4TcC		4TcC		4Tcc		4Tcc		ns
74	TwRST	$\overline{RESET}$ Low Width	3TcC		3TcC		3Tcc		3Tcc		ns
75	TdC(BRQf)	Clock RE to BREQ RE Delay		165		150		125		100	ns
76	TdC(BRQr)	Clock FE to BREQ FE Delay		150		150		125		100	ns
77	TdBRQ(CTRz)	BREQ FE to Control Bus Float Delay		140		140		100		60	ns
78	TdBRQ(ADz)	BREQ FE to AD Bus Float Delay		140		140		100		60	ns
79	TdBRQ(BAK)	BREQ RE to BACK RE Required Delay	0		0		0		0		ns
80	TsBAK(C)	BACK Valid to Clock RE Set-up Time	50		45		30		20		ns
81	TdRES(ADz)	(Reset) FE to A and AD Buses Float Delay		135		135		125		100	ns
82	TdRES(CTRz)	(Reset) FE to Control Bus Float Delay		100		100		100		75	ns
83	TdRES(DSz)	(Reset) FE to $\overline{DS}$ Float Delay		90		90		80		60	ns
84	TsRW(DS)	R/$\overline{W}$ Valid to $\overline{DS}$ FE Set-up Time (IOW)	2		2		2		2		ns
85	ThDS(RW)	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time (IOW)	−10		−10		−10		−10		ns
86	TsRW(DS)	R/$\overline{W}$ Valid to $\overline{DS}$ FE Set-up Time (IOR)	20		20		15		15		ns
87	ThDS(RW)	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time (IOR)	20		20		15		15		ns

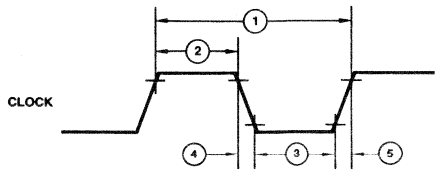
*2000ns for slow readable registers (worst case)
Note: RE = rising edge
FE = falling edge

SWITCHING CHARACTERISTICS
UDC-PERIPHERAL INTERFACE

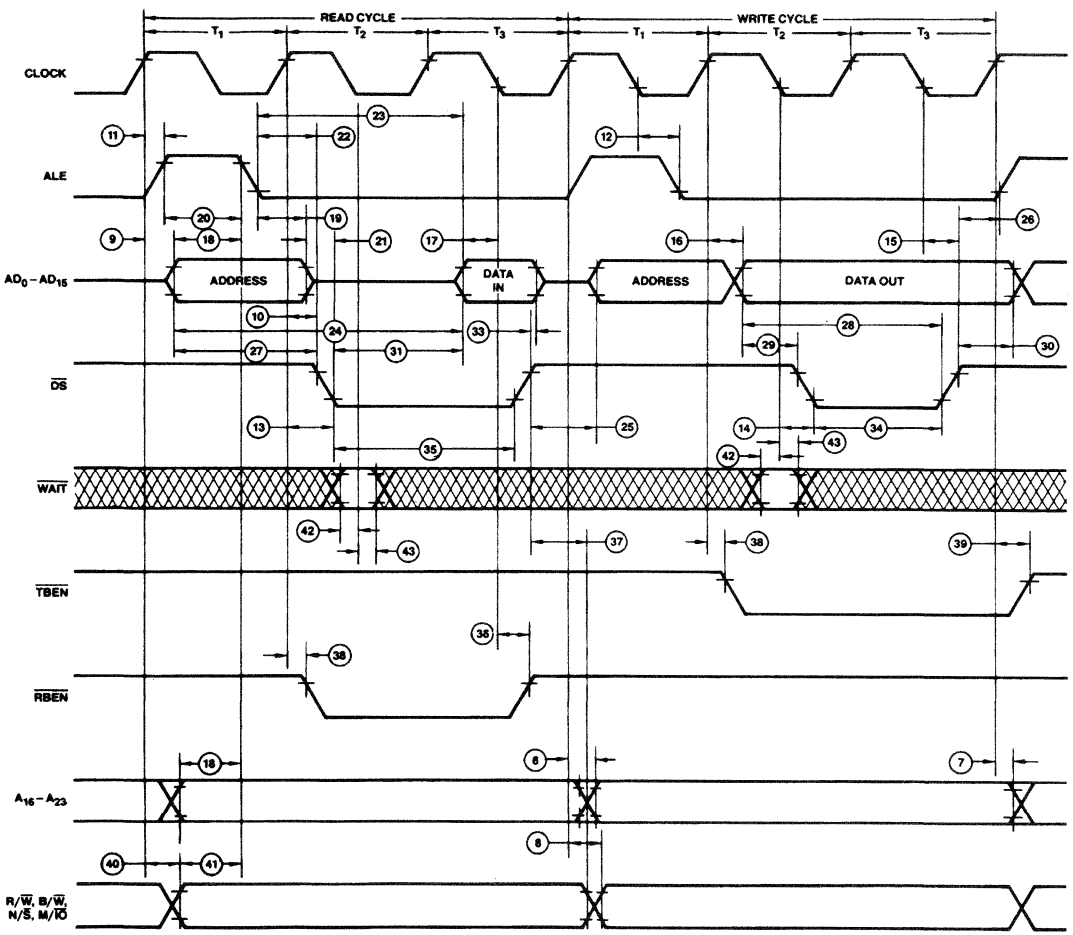
			Preliminary								
			4MHz		6MHz		8MHz		10MHz		
Number	Parameters	Description	Min	Max	Min	Max	Min	Max	Min	Max	Units
90	TCHDL	Clock RE to Pulsed $\overline{DACK}$ FE Delay (Flyby Transactions Only)		100		85		50		40	ns
91	TCHDH	Clock RE to Pulsed $\overline{DACK}$ RE Delay (To Flyby Transactions Only)		100		85		50		40	ns
92	TDSK	$\overline{DS}$ RE to Pulsed $\overline{DACK}$ RE Delay (FROM Flyby Transactions Only)	10		10		10		10		ns
93	TDAD	Clock RE to Level $\overline{DACK}$ Valid Delay		100		85		60		50	ns
94	TDAH	Clock FE to Level $\overline{DACK}$ Valid Hold Time		100		85		60		50	ns
95	TEIDL	Clock FE to Internal $\overline{EOP}$ LOW Delay		110		90		80		70	ns
96	TEIDH	Clock FE to Internal $\overline{EOP}$ RE Delay		110		90		80		70	ns
97	TES	External $\overline{EOP}$ Valid to Clock RE Set-up Time During Operation	10		10		10		10		ns
98	TEW	External $\overline{EOP}$ Pulse Width Required During Operation	20		20		20		20		ns
99	TES(BH)	External $\overline{EOP}$ Valid to Clock RE Set-up Time During Bus Hold	10		10		10		10		ns
100	TEW(BH)	External $\overline{EOP}$ Pulse Width Required During Bus Hold	20		20		20		20		ns

Note: RE = rising edge
FE = falling edge

AC Timing when UDC is a Bus Master

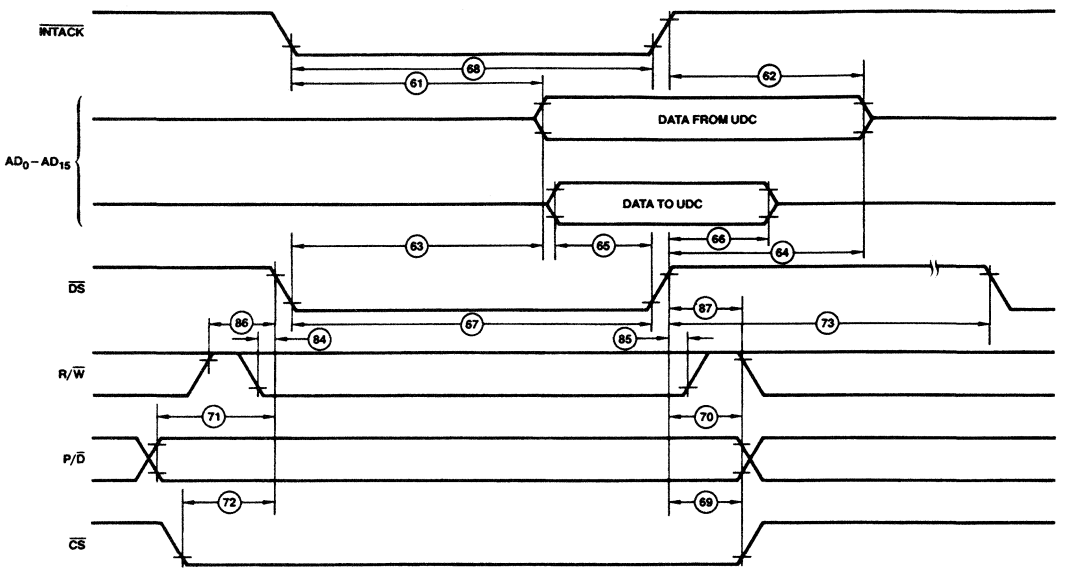


WF007630



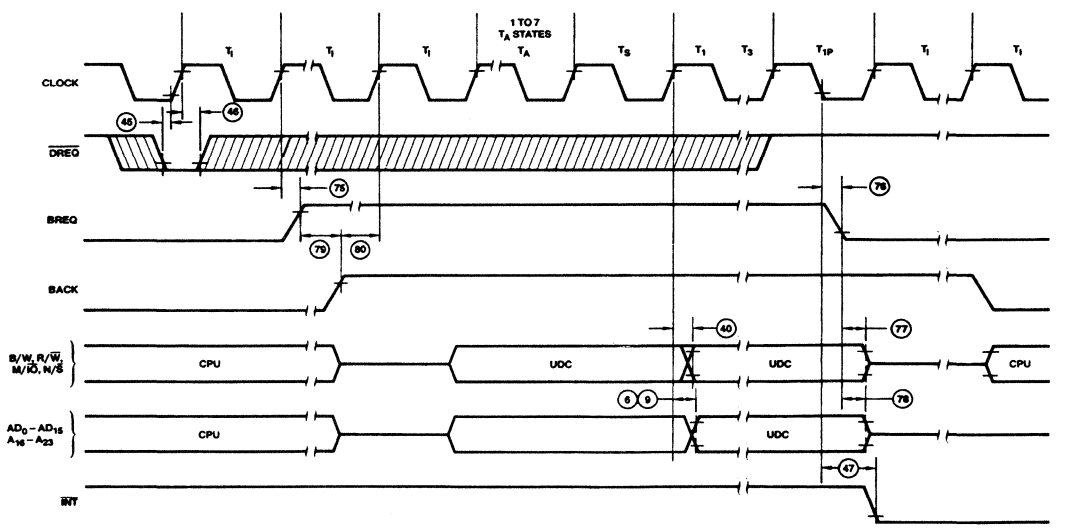
WF007711

AC Timing when UDC is a Bus Slave



WF007720

Bus Exchange Timing



WF007730

Am9517A/8237A*

Multimode DMA Controller

DISTINCTIVE CHARACTERISTICS

- Four independent DMA channels, each with separate registers for Mode Control, Current Address, Base Address, Current Word Count and Base Word Count
- Transfer modes: Block, Demand, Single Word, Cascade
- Independent Autoinitialization of all channels
- Memory-to-memory transfers
- Memory block initialization
- Address increment or decrement
- Master system disable
- Enable/disable control of individual DMA requests
- 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
- Compressed timing option speeds transfers – up to 2.5M bytes/second
- +5 volt power supply
- Advanced N-channel silicon gate MOS technology
- 40-pin Hermetic DIP package, 44-pin PLCC package
- New 9517A-5 5MHz version for higher speed CPU compatability

GENERAL DESCRIPTION

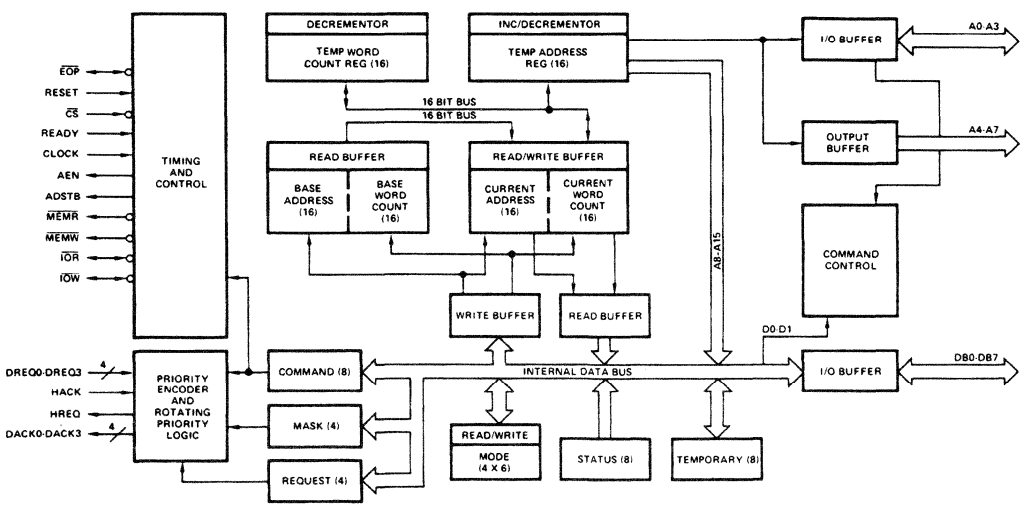
The Am9517A/8237A Multimode Direct Memory Access (DMA) Controller is a peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information to or from the system memory. Memory-to-memory transfer capability is also provided. The Am9517A/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 Am9517A/8237A is designed to be used in conjunction with an external 8-bit address register such as the Am74LS373. 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. An external $\overline{EOP}$ signal can terminate a DMA or memory-to-memory transfer. This is useful for block search or compare operations using external comparators or for intelligent peripherals to abort erroneous services.

BLOCK DIAGRAM



BD003250

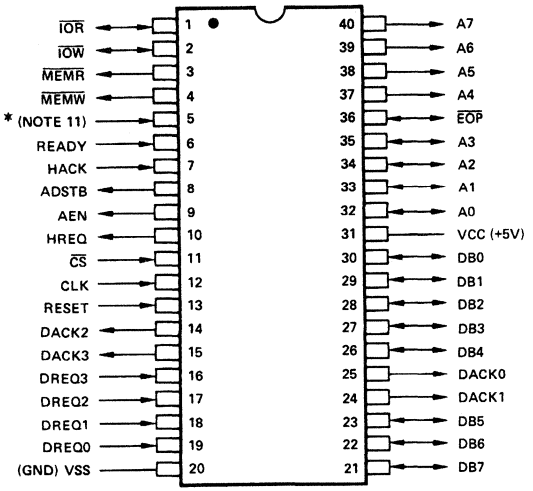
*The 8237A is an AMD-invented device more commonly referred to as the Am9517A.

Publication #	Rev.	Amendment
03040	C	/0

Issue Date: April 1987

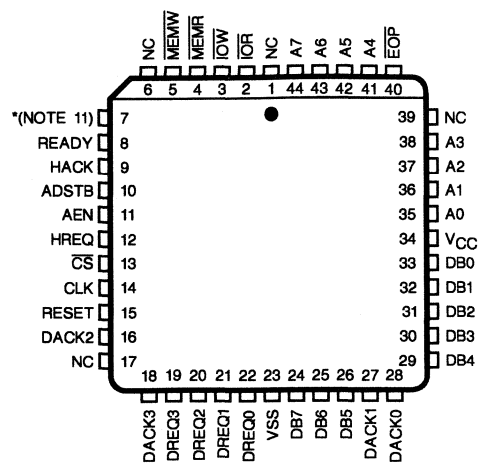
CONNECTION DIAGRAMS
Top View

DIPs



CD005072

PLCC



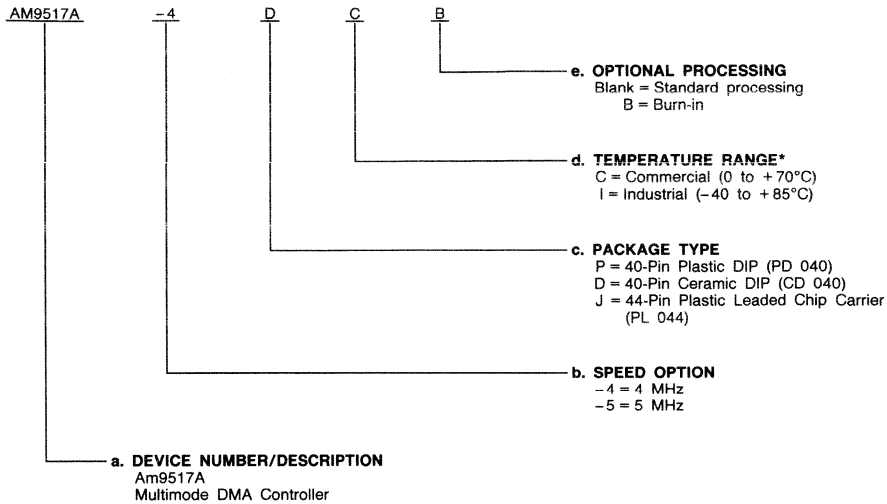
CD009911

Note: Pin 1 is marked for orientation.
*See Note 11 under DC Characteristics table.

ORDERING INFORMATION

Am9517A

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM9517A-4	PC, DC, DCB, DIB, JC
AM9517A-5	

Valid Combinations

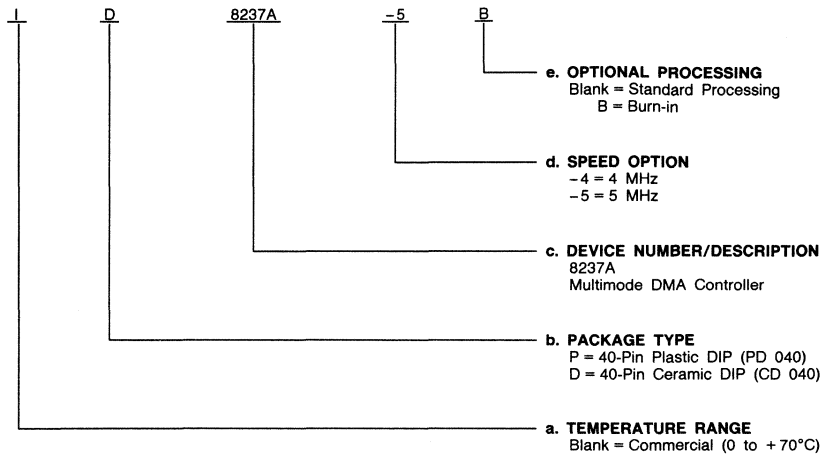
Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessor and Peripheral Military Handbook (Order #09275A/0) for electrical performance characteristics.

ORDERING INFORMATION (Cont'd.)

8237A

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D	8237A-4
	8237A-5
D	8237A-4B
	8237A-5B

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Pin No.*	Name	I/O	Description
31	V_{CC}		Power: +5 Volt supply.
20	V_{SS}		Ground.
12	CLK	I	Clock Input: Clock Input controls the internal operations of the Am9517A/8237A and its rate of data transfers. The input may be driven at up to 3MHz for the standard Am9517A/8237A and up to 5MHz for the Am9517A-5/8237A-5.
11	$\overline{CS}$	I	Chip Select: Chip Select is an active low input used to select the Am9517A/8237A as an I/O device during the Idle cycle.This allows CPU communication on the data bus.
13	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.
6	READY	I	Ready: Ready is an input used to extend the memory read and write pulses from the Am9517A/8237A to accommodate slow memories or I/O peripheral devices. Ready must not make transitions during its specified setup/hold time.
7	HACK	I	Hold Acknowledge: The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system buses.
19-16	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.
30-26, 23-21	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 Am9517A/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 Am9517A/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.
1	$\overline{IOR}$	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 Am9517A/8237A to access data from a peripheral during a DMA Write transfer.
2	$\overline{IOW}$	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 Am9517A/8237A. In the Active cycle, it is an output control signal used by the Am9517A/8237A to load data to the peripheral during a DMA Read transfer.
36	$\overline{EOP}$	I/O	End of Process: End of Process is an active low bidirectional open-drain signal. Information concerning the completion of DMA service is available at the bidirectional $\overline{EOP}$ pin. The Am9517A/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 Am9517A/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 Am9517A/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 prvent erroneous end of process inputs.
32-35	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 registers to be load or read. In the Active cycle, they are outputs and provide the lower 4 bits of the output address.
37-40	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 DMA service.
10	HREQ	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 the Am9517A/8237A to issue the HRQ. After HRQ goes active, at least one clock cycle (TCY) must occur before HLDA goes active.
25, 24 14, 15	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.
9	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 in other system bus drivers during DMA transfers. AEN is active-high.
8	ADSTB	O	Address Strobe. The active-high Address Strobe is used to strobe the upper address byte into an external latch.
3	$\overline{MEMR}$	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.
4	$\overline{MEMW}$	O	Memory Write: The Memory Write signal is an active low three-state output used to write data to the selected memory location during a DMA Write or a a memory-to-memory transfer.

*Applies to DIPs only.

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

Am9517/8237A Internal Registers.

DETAILED DESCRIPTION

The Am9517A/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 Am9517A/8237A contains 344 bits of internal memory in the form of registers. The table shown above 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.

The Am9517A/8237A contains three basic blocks of control logic. The Timing Control block generates internal timing and external control signals for the Am9517A/8237A. The Program Command Control block decodes the various commands given to the Am9517A/8237A by the microprocessor prior to servicing a DMA Request. It also decodes each channel's Mode Control word. The Priority Encoder block resolves priority contention among DMA channels requesting service simultaneously.

The Timing Control block derives internal timing from the clock input. In Am9080A systems this input will usually be the $\phi 2$ TTL clock from an Am8224. However, any appropriate system clock will suffice.

DMA Operation

The Am9517A/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 Am9517A/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 Am9517A/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 0 (S0) is the first state of a DMA service. The Am9517A/8237A has requested a hold but the processor has not yet returned an acknowledge. An acknowledge from the CPU will signal that 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 before S4 by the use of the Ready line on the Am9517A/8237A.

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 each complete 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 and S24) for the write-to-memory half of the transfer. The Temporary Data register is used for intermediate storage of the memory byte.

Idle Cycle

When no channel is requesting service, the Am9517A/8237A will enter the Idle cycle and perform "SI" states. In this cycle the Am9517A/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 Am9517A/8237A. When $\overline{CS}$ is LOW and HACK is LOW, the Am9517A/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 Am9517A/8237A in the Program Condition. These commands are decoded as sets of addresses when both $\overline{CS}$ and $\overline{IOW}$ are active and do not make use of the data bus. Functions include Clear First/Last Flip/Flop and Master Clear.

Active Cycle

When the Am9517A/8237A is in the idle cycle and a channel requests a DMA service, the device will output a HREQ 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 Am9517A/8237A will make a one-byte transfer during each HREQ/HACK handshake. When DREQ goes active, HREQ will go active. After the CPU responds by driving HACK active, a one-byte transfer will take place. Following the transfer, HREQ will go inactive, the word count will be decremented and the address will be either incremented or decremented. When the word count goes to zero, a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so.

To perform a single transfer, DREQ must be held active only until the corresponding DACK goes active. If DREQ is held continuously active, HREQ will go inactive following each transfer and then will go active again and a new one-byte transfer will be made following each rising edge of HACK. In 8080A/Am9080A systems, this will ensure one full machine cycle of execution between DMA transfers. Details of timing between the Am9517A/8237A and other bus control protocols will depend upon the characteristics of the microprocessor involved.

Block Transfer Mode: In Block Transfer mode, the Am9517A/8237A will continue making transfers until a TC (caused by the word count going to zero) or an external End of Process ($\overline{EOP}$) is encountered. DREQ need be held active only until DACK becomes active. An Autoinitialize 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 will continue making transfers until a TC or external $\overline{EOP}$ is encountered or until DREQ goes inactive. Thus, the device requesting service may discontinue transfers by bringing DREQ inactive. Service may be resumed by asserting an active DREQ once again. During the time between services when the microprocessor is allowed to operate, the intermediate values of address and word count may be read from the Am9517A/8237A Current Address and Current Word Count

registers. Autoinitialization will only occur following a TC or $\overline{EOP}$ at the end of service. Following Autoinitialization, an active-going DREQ edge is required to initiate a new DMA service.

Cascade Mode: This mode is used to cascade more than one Am9517A/8237A together for simple system expansion. The HREQ and HACK signals from the additional Am9517A/8237A are connected to the DREQ and DACK signals of a channel of the initial Am9517A/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 in the initial device is used only for prioritizing the additional device, it does not output any address or control signals of its own. These would conflict with the outputs of the active channel in the added device. The Am9517A/8237A will respond to DREQ with DACK but all other outputs except HREQ will be disabled.

Figure 1 shows two additional devices cascaded into an initial device using two of the previous channels. This forms a two level DMA system. More Am9517A/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 devices forming a third level.

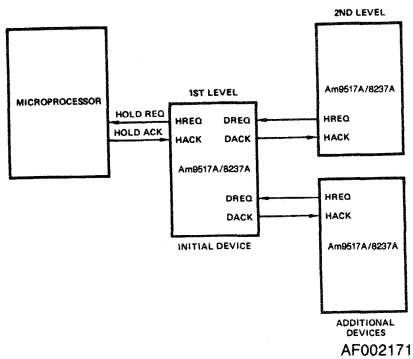


Figure 1. Cascaded Am9517A/8237As

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 $\overline{IOR}$ and $\overline{MEMW}$. Read transfers move data from memory to an I/O device by activating $\overline{MEMR}$ and $\overline{IOW}$. Verify transfers are pseudo transfers; the Am9517A/8237A operates as in Read or Write transfers generating addresses, responding to $\overline{EOP}$, etc. However, the memory and I/O control lines remain inactive.

Memory-to-Memory: The Am9517A/8237A includes a block move capability that allows blocks of data to be moved from one memory address space to another. When Bit C0 in the Command register is set to a logical 1, channels 0 and 1 will operate as memory-to-memory transfer channels. Channel 0 forms the source address and channel 1 forms the destination address. The channel 1 word count is used. A memory-to-memory transfer is initiated by setting a software DMA request for channel 0. Block Transfer Mode should be used for memory-to-memory. When channel 0 is programmed for a fixed source address, a single source word may be written into a block of memory.

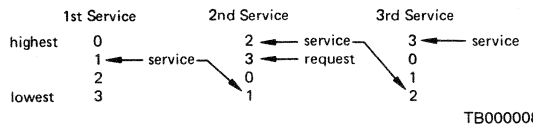
When setting up the Am9517A/8237A for memory-to-memory operation, it is suggested that both channels 0 and 1 be masked out. Further, the channel 0 word count should be initialized to the same value used in channel 1. No DACK outputs will be active during memory-to-memory transfers.

The Am9517A/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 may be found in Timing Diagram 2.

Autoinitialize: By programming a bit in the Mode register, a channel may be set up for an Autoinitialize operation. During Autoinitialization, 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 set by $\overline{EOP}$ when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to repeat its service without CPU intervention.

Priority: The Am9517A/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.

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.



The priority encoder selects the highest priority channel requesting service on each active-going HACK edge. Once a channel is started, its operation will not be suspended if a request is received by a higher priority channel. The high priority channel will only gain control after the lower priority channel releases HREQ. When control is passed from one channel to another, the CPU will always gain bus control. This ensures generation of rising HACK edge to be used to initiate selection of the new highest-priority requesting channel.

Compressed Timing: To achieve even greater throughput where system characteristics permit, the Am9517A/8237A can compress the transfer time to two clock cycles. From Timing Diagram 3 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 Timing Diagram 4.

Extended Write: For Flyby Transactions late write is normally used, as this allows sufficient time for the $\overline{IOR}$ signal to get data from the peripheral onto the bus before $\overline{MEMW}$ is activated. In some systems, performance can be improved by starting the write cycle earlier. This is especially true for memory-to-memory transactions.

Address Generation: To reduce pin count, the Am9517A/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 3-state enable. The lower order address bits are output by the Am9517A/8237A directly. Lines A0-A7 should be connected to the address bus. Timing Diagram 1 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 Am9517A/8237A executes S1 states only when updating of A8-A15 in the latch is necessary. This means for long services that S1 states may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

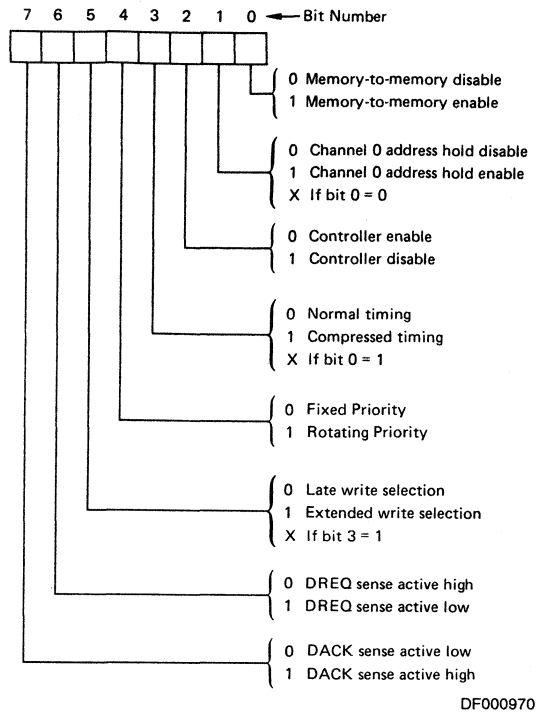
Register Description

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. Autoinitialization takes place only after an $\overline{EOP}$.

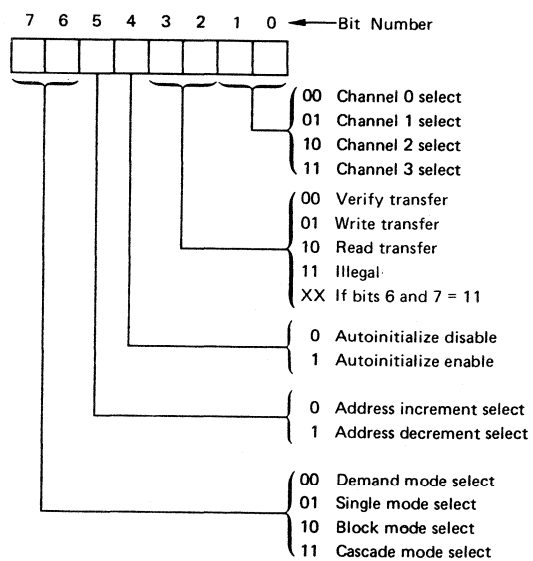
Current Word Count Register: Each channel has a 16-bit Current Word Count register. This register should be programmed with, and will return on a CPU read, a value one less than the number of words to be transferred. The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes to zero, 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 Autoinitialize back to its original value. Autoinitialize can occur only when an $\overline{EOP}$ occurs. Note that the contents of the Word Count register will be FFFF (hex) following on internally generated $\overline{EOP}$.

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 values 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 during DMA, programming by the microprocessor. Accordingly, writing to these registers when intermediate values are in the Current registers will overwrite the intermediate values. The Base registers cannot be read by the microprocessor.

Command Register: This 8-bit register controls the operation of the Am9517A/8237A. It is programmed by the microprocessor in the Program Condition and is cleared by Reset. The following table lists the function of the command bits. See Figure 2 for address coding.

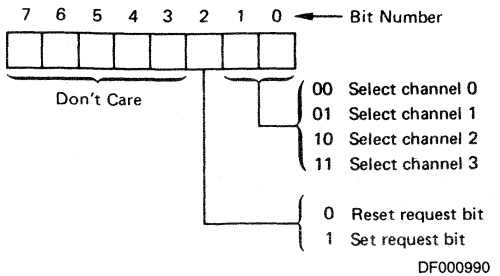


Mode Register: Each channel has a 6-bit Mode register associated with it. When the register is being written to by the microprocessor in the Program Condition, bits 0 and 1 determine which channel Mode register is to be written to.



Request Register: The Am9517A/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 nonmaskable and subject to prioritization by the Priority Encoder network.

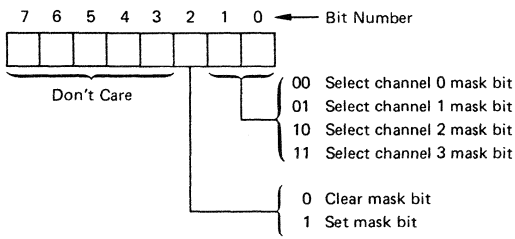
Each register bit is set or reset separately under software control or is cleared upon generation of a TC or external EOP. The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word. See Figure 2 for address coding.



DF000990

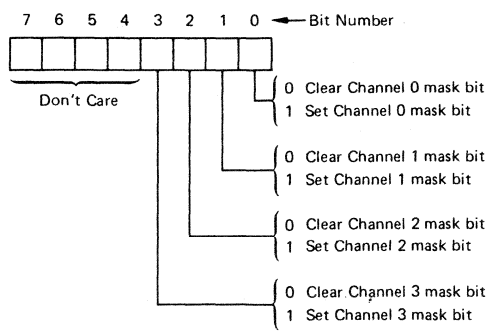
Software requests will be serviced only if the channel is in Block mode. When initiating a memory-to-memory transfer, the software request for channel 0 should be set.

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 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 2 for instruction addressing.



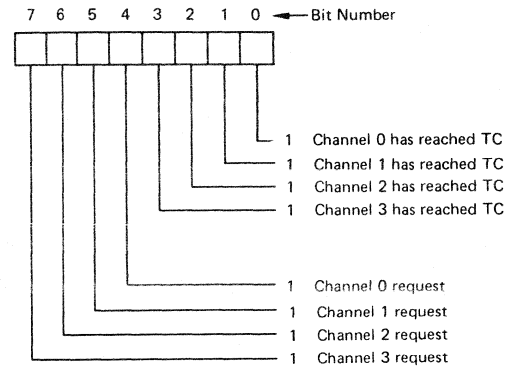
DF001000

All four bits of the Mask Register may also be written with a single command.



DF001010

Status Register: The Status registers may be read out of the Am9517A/8237A by the microprocessor. It indicates which channels have reached a terminal count and which channels have pending DMA requests. Bits 0-3 are set each time a TC is reached by that channel, including after each Autoinitialization. These bits are cleared by Reset and each Status Read. Bits 4-7 are set whenever their corresponding channel is requesting service.



DF001020

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: There are three 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 may be issued prior to writing or reading Am9517A/8237A address or word count information. This initializes the Flip/Flop to a known state so that subsequent accesses to register contents by the microprocessor will address lower and upper bytes in the correct sequence. When the Flip/Flop is cleared it addresses the lower byte and when set it addresses the upper byte.

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 Am9517A/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.

Interface 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 2. Register and Function Addressing

Channel	Register	Operation	Signals							Internal Flip/Flop	Data Bus DB0 – DB7
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base & Current Address	Write	0 0	1 1	0 0	0 0	0 0	0 0	0 0	0 1	A0 – A7 A8 – A15
	Current Address	Read	0 0	0 0	1 1	0 0	0 0	0 0	0 0	0 1	A0 – A7 A8 – A15
	Base & Current Word Count	Write	0 0	1 1	0 0	0 0	0 0	0 0	1 1	0 1	W0 – W7 W8 – W15
	Current Word Count	Read	0 0	0 0	1 1	0 0	0 0	0 0	1 1	0 1	W0 – W7 W8 – W15
1	Base & Current Address	Write	0 0	1 1	0 0	0 0	0 0	1 1	0 0	0 1	A0 – A7 A8 – A15
	Current Address	Read	0 0	0 0	1 1	0 0	0 0	1 1	0 0	0 1	A0 – A7 A8 – A15
	Base & Current Word Count	Write	0 0	1 1	0 0	0 0	0 0	1 1	1 1	0 1	W0 – W7 W8 – W15
	Current Word Count	Read	0 0	0 0	1 1	0 0	0 0	1 1	1 1	0 1	W0 – W7 W8 – W15
2	Base & Current Address	Write	0 0	1 1	0 0	0 0	1 1	0 0	0 0	0 1	A0 – A7 A8 – A15
	Current Address	Read	0 0	0 0	1 1	0 0	1 1	0 0	0 0	0 1	A0 – A7 A8 – A15
	Base & Current Word Count	Write	0 0	1 1	0 0	0 0	1 1	0 0	1 1	0 1	W0 – W7 W8 – W15
	Current Word Count	Read	0 0	0 0	1 1	0 0	1 1	0 0	1 1	0 1	W0 – W7 W8 – W15
3	Base & Current Address	Write	0 0	1 1	0 0	0 0	1 1	1 1	0 0	0 1	A0 – A7 A8 – A15
	Current Address	Read	0 0	0 0	1 1	0 0	1 1	1 1	0 0	0 1	A0 – A7 A8 – A15
	Base & Current Word Count	Write	0 0	1 1	0 0	0 0	1 1	1 1	1 1	0 1	W0 – W7 W8 – W15
	Current Word Count	Read	0 0	0 0	1 1	0 0	1 1	1 1	1 1	0 1	W0 – W7 W8 – W15

Figure 3. Word Count and Address Register Command Codes

APPLICATIONS INFORMATION

Figure 4 shows a convenient method for configuring a DMA system with the Am9517A/8237A Controller and a microprocessor system. The Multimode DMA Controller issues a Hold Request to the processor whenever there is at least one valid DMA Request from a peripheral device. When the processor replies with a Hold Acknowledge signal, the Am9517A/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 eight bits on the eight Address outputs and the most significant eight bits on the Data Bus. The contents of the Data Bus are then latched into the Am74LS373 register to complete the full 16 bits of the Address Bus. The Am74LS373 is a high-speed, low power, 8-bit, 3-state register in a 20-pin package. After the initial transfer takes place, the register is updated only after a carry or borrow is generated in the least significant address byte. Four DMA channels are provided when one Am9517A/8237A is used.

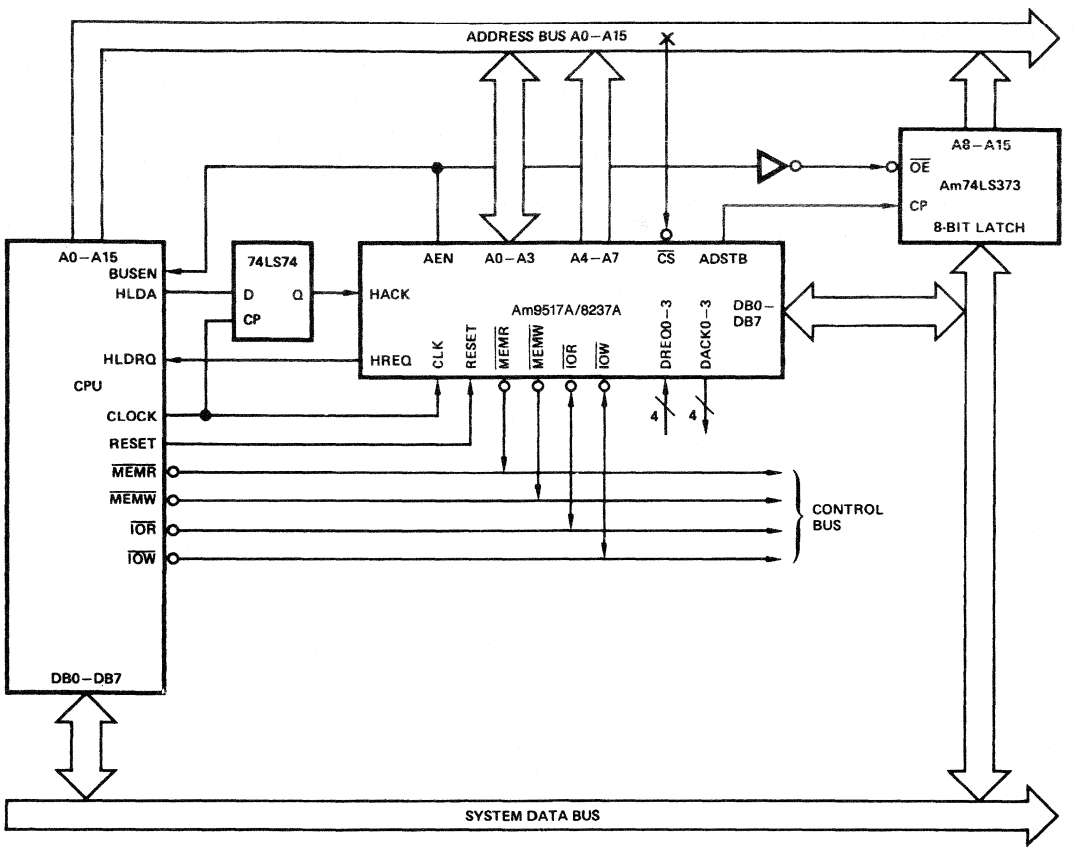


Figure 4. Basic DMA Configuration

ABSOLUTE MAXIMUM RATINGS

Storage temperature ..... −65 to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to +7.0V
All Signal Voltages with Respect to V_{SS} .. −0.5V to +7.0V
Power Dissipation (Package Limitation) 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to 70°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ±10%

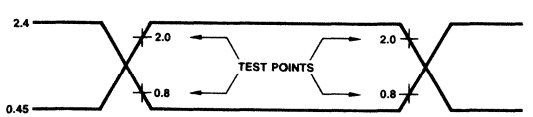
Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified. (Note 1)

Parameters	Description	Test Conditions	Min	Typ	Max	Units
VOH	Output HIGH Voltage	IOH = −200 μA	2.4			Volts
		IOH = −100 μA, (HREQ Only)	3.3			
VOL	Output LOW Voltage	IOL = 3.2 mA			0.40 V	Volts
VIH	Input HIGH Voltage		2.0		VCC + 0.5	Volts
VIL	Input LOW Voltage		−0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VI ≤ VCC	−10		+10	μA
IOZ	Output Leakage Current	VCC ≤ VO ≤ VSS + 0.40	−10		+10	μA
ICC	VCC Supply Current	T_A = +25°C		65	130	mA
		T_A = 0°C		75	150	
CO	Output Capacitance (Note 14)	fc = 1.0 MHz, Inputs = 0 V		4	8	pF
CI	Input Capacitance			8	15	pF
CIO	I/O Capacitance			10	18	pF
COHREQ	Output Capicitance (HREQ)	fc = 1.0 MHz, Inputs = 0 V		18	20	pF

Notes:
1. Typical values are for T_A = 25°C, nominal supply voltage and nominal processing parameters.
2. Input timing parameters assume transition times of 20 ns or less. Waveform measurement points for both input and output signals are 2.0 V for HIGH and 0.8 V for LOW, unless otherwise noted.
3. Output loading is 1 Standard TTL gate plus 50 pF capacitance unless noted otherwise.
4. The new $\overline{IOW}$ 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.
5. TDQ is specified for two different output HIGH levels. TDQ1 is measured at 2.0 V. TDQ2 is measured at 3.3 V. The value for TDQ2 assumes an external 3.3 kΩ pull-up resistor connected from HREQ to VCC.
6. DREQ should be held active until DACK is returned.
7. DREQ and DACK signals may be active high or active low.
   Timing diagrams assume the active high mode.
8. Output loading on the data bus is 1 Standard TTL gate plus 15 pF for the minimum value and 1 Standard TTL gate plus 100 pF for the maximum value.
9. Successive read and/or write operations by the external processor to program or examine the controller must be timed to allow at least 450 ns for the Am9517A-4/8237A-4, and 400 ns for the Am9517A-5/8237A-5 as recovery time between active read or write pulses.
10. Parameters are listed in alphabetical order.
11. Pin 5 is an input that should always be at a logic high level.
    An internal pull-up resistor will establish a logic high when the pin is left floating. Alternatively, pin 5 may be tied to VCC.
12. Signals $\overline{READ}$ and $\overline{WRITE}$ refer to $\overline{IOR}$ and $\overline{MEMW}$ respectively for peripheral-to-memory DMA operations and to $\overline{MEMR}$ and $\overline{IOW}$ respectively for memory-to-peripheral DMA operations.
13. If N wait states are added during the write-to-memory half of a memory-to-memory transfer, this parameter will increase by N (TCY).
14. All output pins except HREQ.
15. Because $\overline{EOP}$ high from clock high is load dependent, users wishing to test these parameters should use a 2k pull-up resistor and a tester with 50 pF or less load capacitance. Time constant RC = 120 ns is added to the specified number in the data sheet for testing.

SWITCHING TEST INPUT WAVEFORM



WF003310

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified
ACTIVE CYCLE (Notes 2, 3, 10, 11, and 12)

Parameters	Description	Am9517A-4/8237A-4		Am9517A-5/8237A-5		Units
		Min	Max	Min	Max	
TAEL	AEN HIGH from CLK LOW (S1) Delay Time		225		200	ns
TAET	AEN LOW from CLK HIGH (S1) Delay Time		150		130	ns
TAFAB	ADR Active to Float Delay from CLK HIGH		120		90	ns
TAFC	$\overline{READ}$ or $\overline{WRITE}$ Float from CLK HIGH		120		120	ns
TAFDB	DB Active to Float Delay from CLK HIGH		190		170	ns
TAHR	ADR from $\overline{READ}$ HIGH Hold Time	TCY-100		TCY-100		ns
TAHS	DB from ADSTB LOW Hold Time	40		30		ns
TAHW	ADR from WRITE HIGH Hold Time	TCY-50		TCY-50		ns
TAK	DACK Valid from CLK LOW Delay Time		220		170	ns
	$\overline{EOP}$ HIGH from CLK HIGH Delay Time		190		170	ns
	$\overline{EOP}$ LOW to CLK HIGH Delay Time		190		170	ns
TASM	ADR Stable from CLK HIGH		190		170	ns
TASS	DB to ADSTB LOW Set-up Time	100		100		ns
TCH	Clock High Time (Transitions $\leqslant$ 10ns)	100		80		ns
TCL	Clock Low Time (Transitions $\leqslant$ 10ns)	110		68		ns
TCY	CLK Cycle Time	250		200		ns
TDCL	CLK HIGH to $\overline{READ}$ or $\overline{WRITE}$ LOW Delay (Note 4)		200		190	ns
TDCTR	$\overline{Read}$ HIGH from CLK HIGH (S4) Delay Time (Note 4)		210		190	ns
TDCTW	$\overline{WRITE}$ HIGH from CLK HIGH (S4) Delay Time (Note 4)		150		130	ns
TDQ1	HREQ Valid from CLK HIGH Delay Time (Note 5)		120		120	ns
TDQ2			190		120	ns
TEPS	$\overline{EOP}$ LOW from CLK LOW Set-up Time	45		40		ns
TEPW	$\overline{EOP}$ Pulse Width	225		220		ns
TFAAB	ADR Float to Active Delay from CLK HIGH		190		170	ns
TFAC	$\overline{READ}$ or $\overline{WRITE}$ Active from CLK HIGH		150		150	ns
TFADB	DB Float to Active Delay from CLK HIGH		225		200	ns
THS	HACK Valid to CLK HIGH Set-up Time	75		75		ns
TIDH	Input Data from $\overline{MEMR}$ HIGH Hold Time	0		0		ns
TIDS	Input Data to $\overline{MEMR}$ HIGH Set-up Time	190		170		ns
TODH	Output Data from $\overline{MEMW}$ HIGH Hold Time	20		10		ns
TODV	Output Data Valid to $\overline{MEMW}$ HIGH (Note 13)	125		125		ns
TQS	DREQ to CLK LOW (S1, S4) Set-up Time	0		0		ns
TRH	CLK to READY LOW Hold Time	20		20		ns
TRS	READY to CLK LOW Set-up Time	60		60		ns
TSTL	ADSTB HIGH from CLK HIGH Delay Time		150		130	ns
TSTT	ADSTB LOW from CLK HIGH Delay Time		110		90	ns

Notes: See notes under DC Characteristics table.

SWITCHING CHARACTERISTICS over operating ranges (Cont'd.)

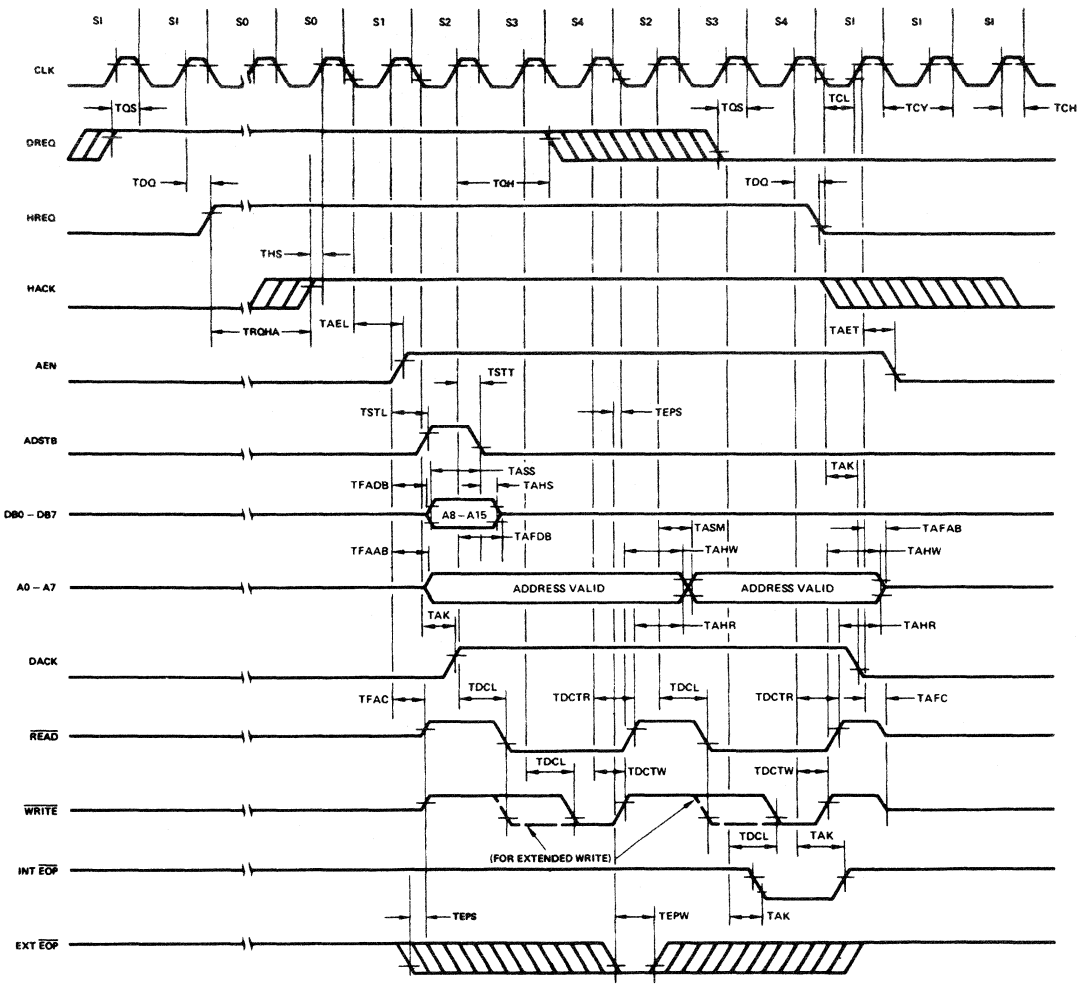
PROGRAM CONDITION (Idle Cycle) (Notes 2, 3, 10, and 11)

Parameters	Description	Am9517A-4/8237A-4		Am9517A-5/8237A-5		Units
		Min	Max	Min	Max	
TAR	ADR Valid or $\overline{CS}$ LOW to $\overline{READ}$ LOW	50		50		ns
TAW	ADR Valid to $\overline{WRITE}$ HIGH Set-up Time	150		130		ns
TCW	CS LOW to $\overline{WRITE}$ HIGH Set-up Time	150		130		ns
TDW	Data Valid to $\overline{WRITE}$ HIGH Set-up Time	150		130		ns
TRA	ADR or CS Hold from $\overline{READ}$ HIGH	0		0		ns
TRDE	Data Access from $\overline{READ}$ LOW (Note 8)		200		140	ns
TRDF	DB Float Delay from $\overline{READ}$ HIGH	20	100	0	70	ns
TRSTD	Power Supply HIGH to RESET LOW Set-up Time	500		500		μs
TRSTS	RESET to First $\overline{IOWR}$	2TCY		2TCY		ns
TRSTW	RESET Pulse Width	300		300		ns
TRW	$\overline{READ}$ Width	250		200		ns
TWA	ADR from $\overline{WRITE}$ HIGH Hold Time	20		20		ns
TWC	CS HIGH from $\overline{WRITE}$ HIGH Hold Time	20		20		ns
TWD	Data from $\overline{WRITE}$ HIGH Hold Time	30		30		ns
TWWS	Write Width	200		160		ns

Notes: See notes under DC Characteristics table.

SWITCHING WAVEFORMS

Timing Diagram 1. Active Cycle Timing Diagram

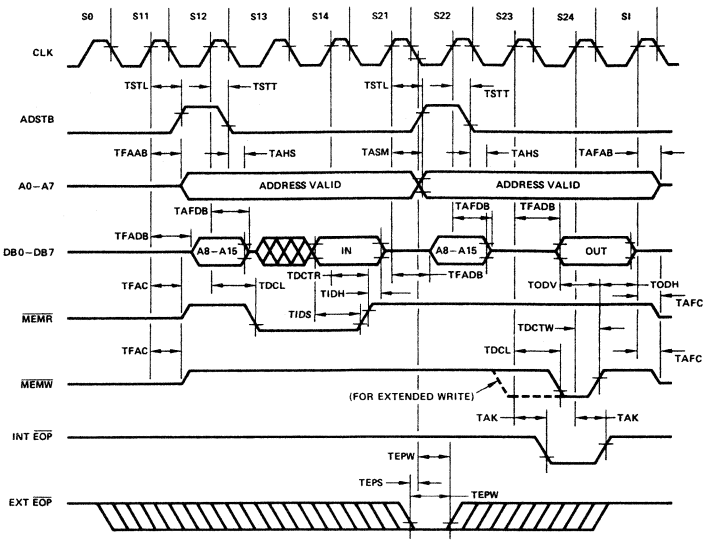


WF003300

Note: $\overline{EOP}$ must precede AEN in single transfer mode.

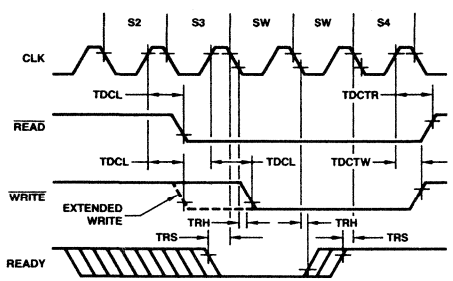
SWITCHING WAVEFORMS (Cont'd.)

Timing Diagram 2. Memory-to-Memory



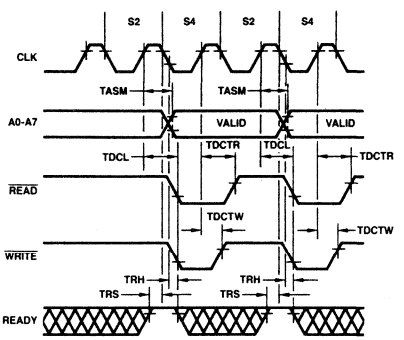
WF003320

Timing Diagram 3. Ready Timing



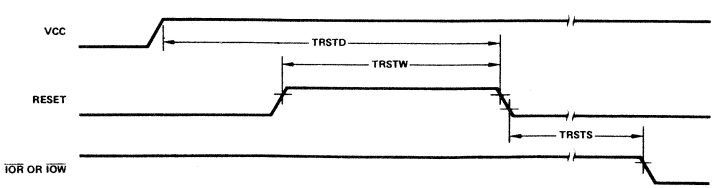
WF003330

Timing Diagram 4. Compressed Timing



WF003340

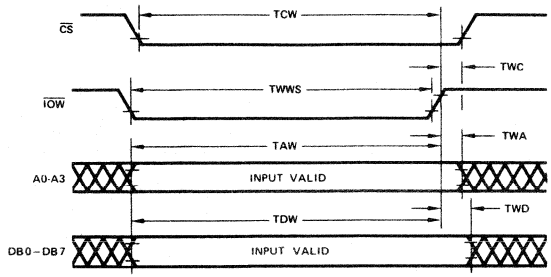
Timing Diagram 5. Reset Timing



WF003350

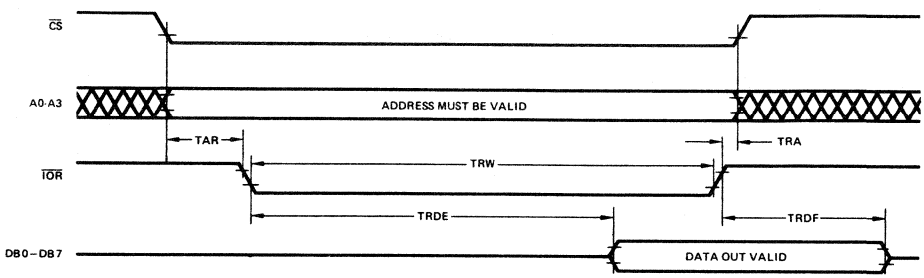
SWITCHING WAVEFORMS (Cont'd.)

Timing Diagram 6. Program Condition Write Timing



WF003360

Timing Diagram 7. Program Condition Read Cycle



WF003370

Am9518/AmZ8068

Data Ciphering Processor

DISTINCTIVE CHARACTERISTICS

- **Encrypts and decrypts data**
  Implements National Bureau of Standards standard data encryption algorithm.
- **High-Speed Operation**
  Am9518 and AmZ8068 throughput over 1.3 and 1.7M bytes per second, respectively. Operates at data rates fast enough for disk controllers, high-speed DMA, telecommunication channels.
- **Supports three ciphering options**
  Electronic Code Book for disk applications. Chain Block Cipher for high-speed telecommunications. Cipher Feedback for low-to-medium speed, byte-oriented communications.
- **Three separate key registers on-chip**
  Separate registers for encryption key, decryption key and master key improve system security and throughput by eliminating need to reload keys frequently.
- **Three separate data ports provide flexible interface, improved security**
  The DCP utilizes a master port, slave port and key port. Functions of the three ports can be programmed by the user to provide for simple interface to AmZ8000 and Am2900 systems and to provide total hardware separation of encrypted data, clear data and keys.

GENERAL DESCRIPTION

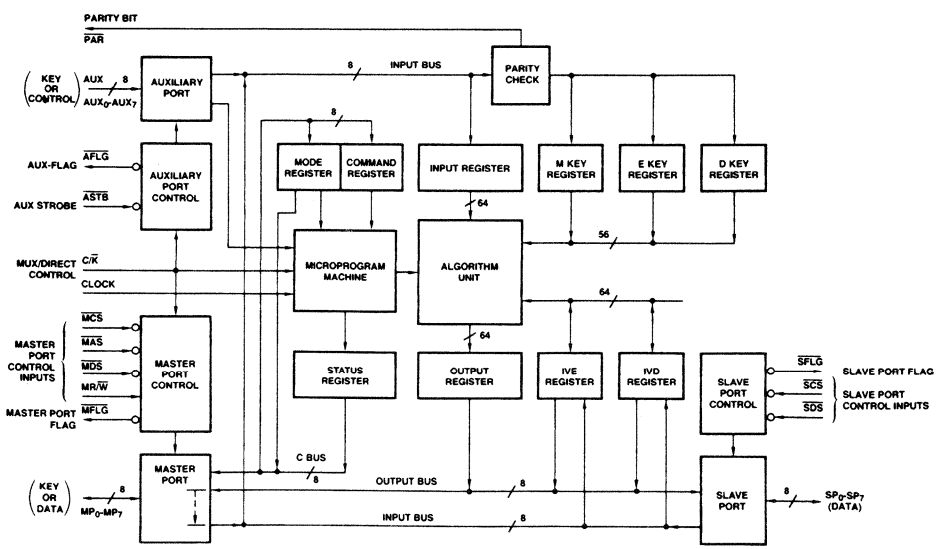
The Am9518/AmZ8068 Data Ciphering Processor is an N-channel silicon gate LSI product containing the circuitry necessary to encrypt and decrypt data using the National Bureau of Standards encryption algorithm. It is designed to be used in a variety of environments, including dedicated controllers, communication concentrators, terminals and peripheral task processors in general processor systems.

The DCP provides a high throughput rate using Cipher Feedback, Electronic Code Book or Cipher Block Chain operating modes. Separate ports for key input, clear data and enciphered data enhance security.

The system communicates with the DCP using commands entered in the master port and through auxiliary control lines. Once set up, data can flow through the DCP at high speeds because input, output and ciphering activities are all performed concurrently. External DMA control can easily be used to enhance throughput in some system configurations.

This device is designed to interface directly to the AmZ8000 CPU bus and, with a minimum of external logic, to the 2900, MC68000, 8086, 8085, and 8051 families of processors.

BLOCK DIAGRAM



BD003290

Publication #	Rev.	Amendment
00618	B	/0

Issue Date: April 1985

Am9519A

Universal Interrupt Controller

DISTINCTIVE CHARACTERISTICS

- Eight individually maskable interrupt inputs reduce CPU overhead
- Unlimited interrupt channel expansion with no extra hardware
- Programmable 1-byte to 4-byte response provides vector address and message protocol for 8-bit CPUs
- Rotating and fixed priority resolution logic
- Software interrupt request capability
- Common vector and polled mode options
- Automatic hardware clear of in-service interrupts reduces software overhead
- Polarity control of interrupt inputs and outputs
- Reset minimizes software initialization by automatically generating CALL to location zero

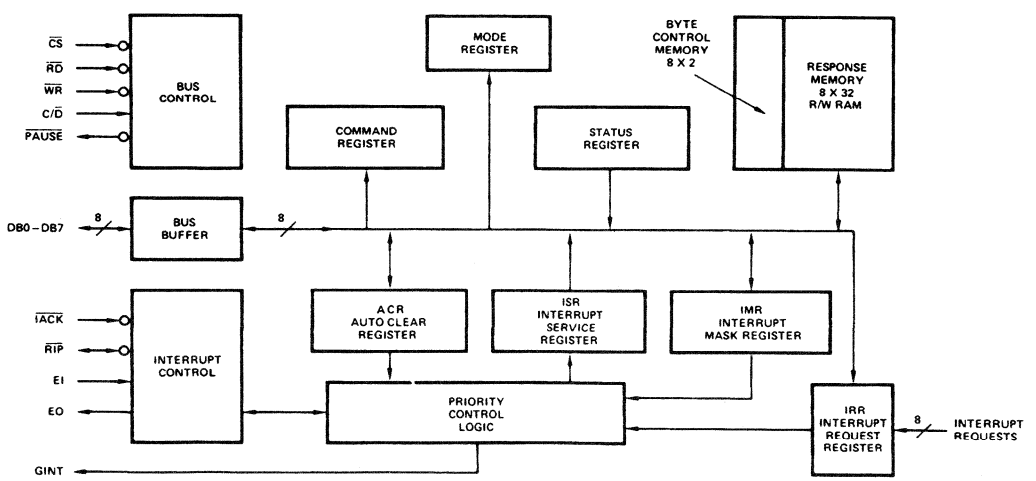
GENERAL DESCRIPTION

The Am9519A Universal Interrupt Controller is a processor support circuit that provides a powerful interrupt structure to increase the efficiency and versatility of microcomputer-based systems. A single Am9519A manages up to eight maskable interrupt request inputs, resolves priorities and supplies up to four bytes of fully programmable response for each interrupt. It uses a simple expansion structure that allows many units to be cascaded for control of large numbers of interrupts. Several programmable control features are provided to enhance system flexibility and optimization.

The Universal Interrupt Controller is designed with a general purpose interface to facilitate its use with a wide range of digital systems, including most popular 8-bit microprocessors. Since the response bytes are fully programmable, any instruction or vectoning protocol appropriate for the host processor may be used.

When the Am9519A controller receives an unmasked interrupt request, it issues a Group Interrupt output to the CPU. When the interrupt is acknowledged, the controller outputs the one-to-four byte response associated with the highest priority unmasked interrupt request. The ability of the CPU to set interrupt requests under software control permits hardware prioritization of software tasks and aids system diagnostic and maintenance procedures.

BLOCK DIAGRAM



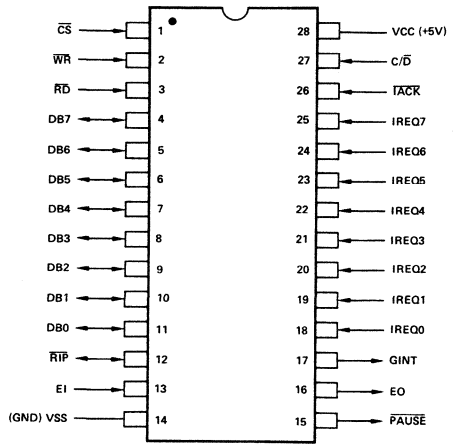
BD003280

Publication #	Rev.	Amendment
00147	C	/0

Issue Date: **May 1987**

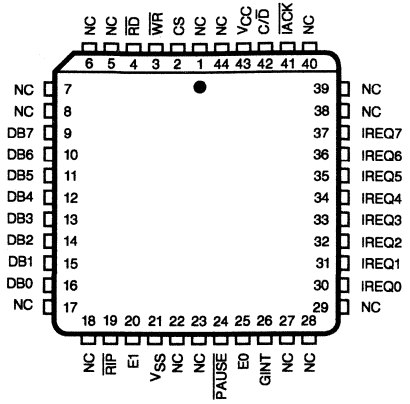
CONNECTION DIAGRAMS
Top View

DIPs



CD005101

PLCC



CD010650

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing

AM9519A-1DCB

e. OPTIONAL PROCESSING
Blank = Standard processing
B = Burn-in

d. TEMPERATURE RANGE*
C = Commercial (0 to +70°C)
I = Industrial (−40 to +85°C)

c. PACKAGE TYPE
P = 28-Pin Plastic DIP (PD 028)
D = 28-Pin Ceramic DIP (CD 028)
J = 28-Pin Plastic Leaded Chip Carrier (PL 028)

b. SPEED OPTION
−1 = Faster Response Time (see Switching Characteristics)

a. DEVICE NUMBER/DESCRIPTION
Am9519A
Universal Interrupt Controller

Valid Combinations	
AM9519A	PC, DC, DCB, DIB, JC
AM9519A-1	PC, DC, DCB, JC

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order # 09275A/0) for electrical performance characteristics.

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
28	V_{CC}		+5 Volt Power Supply.
14	V_{SS}		Ground.
11-4	DB0-DB7	I/O	(Data Bus). The eight bidirectional data bus signals are used to transfer information between the Am9519A and the system data bus. The direction of transfer is controlled by the $\overline{IACK}$, $\overline{WR}$ and $\overline{RD}$ input signals. Programming and control information are written into the device; status and response data are output by it.
1	$\overline{CS}$	I	(Chip Select). The active low Chip Select input enables read and write operations on the data bus. Interrupt acknowledge responses are not conditioned by $\overline{CS}$.
3	$\overline{RD}$	I	(Read). The active low Read signal is conditioned by $\overline{CS}$ and indicates that information is to be transferred from the Am9519A to the data bus.
2	$\overline{WR}$	I	(Write). The active low Write signal is conditioned by $\overline{CS}$ and indicates that data bus information is to be transferred from the data bus to a location within the Am9519A.
27	C/$\overline{D}$	I	(Control/Data). The C/$\overline{D}$ control signal selects source and destination locations for data bus read and write operations. Data read or write transfers are made to or from preselected internal registers or memory locations. Control write operations load the command register and control read operations output the status register.
18-25	IREQ0-IREQ7	I	(Interrupt Request). The Interrupt Request signals are used by external devices to indicate that service by the host CPU is desired. IREQ inputs are accepted asynchronously and they may be programmed for either a HIGH-to-LOW or LOW-to-HIGH edge transition. Active inputs are latched internally in the Interrupt Request Register. After the IRR bit is cleared, an IREQ transition of the programmed polarity must occur to initiate another request.
12	$\overline{RIP}$	I/O	(Response In Process). Response In Process is a bidirectional signal used when two or more Am9519A circuits are cascaded. It permits multibyte response transfers to be completed without interference from higher priority interrupts. An Am9519A that is responding to an acknowledged interrupt will treat $\overline{RIP}$ as an output and hold it LOW until the acknowledge response is finished. An Am9519A without an acknowledged interrupt will treat $\overline{RIP}$ as an input and will ignore $\overline{IACK}$ pulses as long as $\overline{RIP}$ is LOW. The $\overline{RIP}$ output is open drain and requires an external pull-up resistor to VCC.
26	$\overline{IACK}$	I	(Interrupt Acknowledge). The active-low Interrupt Acknowledge line indicates that the external system is asking for interrupt response information. Depending on the programmed state of the Am9519A, it will accept 1, 2, 3 or 4 $\overline{IACK}$ pulses; one response byte is transferred per pulse. The first $\overline{IACK}$ pulse causes selection of the highest priority unmasked pending interrupt request and generates a $\overline{RIP}$ output signal.
15	$\overline{PAUSE}$	O	(Pause). The active-low Pause signal is used to coordinate interrupt responses with data bus and control timing. Pause goes LOW when the first $\overline{IACK}$ is received and remains LOW until $\overline{RIP}$ goes LOW. The external system can use Pause to stretch the acknowledge cycle and allow the control timing to automatically adjust to the actual priority resolution delays in the interrupt system. Second, third and fourth response bytes do not cause Pause to go LOW. Pause is an open drain output and requires an external pull-up resistor to VCC.
16	EO	O	(Enable Out). The active-high EO signal is used to implement daisy-chained cascading of several Am9519A circuits. EO is connected to the EI input of the next lower priority chip. On receipt of an interrupt acknowledge, each EO will go inactive until it has been determined that no valid interrupt request is pending on that chip. If an active request is present, EO remains LOW. EO is also held LOW when the master mask bit is active, thus disabling all lower priority chips.
13	EI	I	(Enable in). The active-high EI signal is used to implement daisy-chained cascading of several Am9519A circuits. EI is connected to EO of the next higher priority chip. It may also be used as a hardware disable input for the interrupt system. When EI is LOW, $\overline{IACK}$ inputs will not affect ISR; however, $\overline{PAUSE}$ will go LOW until $\overline{RIP}$ goes LOW. EI is internally pulled up to VCC so that no external pull-up is needed when EI is not used.
17	GINT	O	(Group Interrupt). The Group Interrupt output signal indicates that at least one unmasked interrupt request is pending. It may be programmed for active-high or active-low polarity. When active-low, the output is open drain and requires an external pull-up resistor to VCC. Since a glitch on GINT occurs approximately 100nsec after the last IACK pulse, this pin should not be connected to edge sensitive devices.

PRODUCT OVERVIEW

Register Description

Interrupt Request Register (IRR): The 8-bit IRR is used to store pending interrupt requests. A bit in the IRR is set whenever the corresponding IREQ input goes active. Bits may also be set under program control from the CPU, thus permitting software generated interrupts. IRR bits may be cleared under program control. An IRR bit is automatically cleared when its interrupt is acknowledged. All IRR bits are cleared by a reset function.

Interrupt Service Register (ISR): The 8-bit ISR contains one bit for each IREQ input. It is used to indicate that a pending interrupt has been acknowledged and to mask all lower priority interrupts. When a bit is set by the acknowledge logic in the ISR, the corresponding IRR bit is cleared. If an acknowledged interrupt is not programmed to be automatically cleared, its ISR bit must be cleared by the CPU under program control when it is desired to permit interrupts from lower priority devices. When the interrupt is programmed for automatic clearing, the ISR bit is automatically reset during the acknowledge sequence. All ISR bits are cleared by a reset function.

Interrupt Mask Register (IMR): The 8-bit IMR is used to enable or disable the individual interrupt inputs. The IMR bits correspond to the IREQ inputs, and all eight may be loaded, set or cleared in parallel under program control. In addition, individual IMR bits may be set or cleared by the CPU. Care must be taken therefore when disabling a specific channel by setting its IMR bit. If that bit is causing the GINT pin to be active, a lock-up condition can occur if the CPU recognizes the interrupt and then the Am9519A removes the request. During the $\overline{IACK}$ cycle, $\overline{PAUSE}$ will go LOW and stay LOW. The solution is to disable CPU interrupts prior to writing to the IMR and then re-enable them. A reset function will set all eight

mask bits, disabling all requests. A mask bit that is set does not disable the IRR, and an IREQ that arrives while a corresponding mask bit is set will cause an interrupt later when the mask bit is cleared. Only unmasked interrupt inputs can generate a Group Interrupt output.

Response Memory: An 8 x 32 read/write response memory is included in the Am9519A. It is used to store up to four bytes of response information for each of the eight interrupt request inputs. All bits in the memory are programmable, allowing any desired vector, opcode, instruction or other data to be entered. The Am9519A transfers the interrupt response information for the highest priority unmasked interrupt from the memory to the data bus when the $\overline{IACK}$ input is active.

Auto Clear Register: The 8-bit Auto Clear register contains one bit for each IREQ input and specifies the operating mode for each of the ISR bits. When an auto clear bit is off, the corresponding ISR bit is set when that interrupt is acknowledged and is cleared by software command. When an auto clear bit is on, the corresponding ISR bit is cleared by the hardware by the rising edge of the last acknowledge pulse. A reset function clears all auto clear bits.

Status Register: The 8-bit Status register contains information concerning the internal state of the chip. It is especially useful when operating in the polled mode to identify interrupting devices. Figure 1 shows the status register bit assignments. The polarity of the GINT bit 7 is not affected by the GINT polarity control. Bits S0-S2 are set asynchronously to a status register read operation. It is recommended to read the register twice and to compare the binary vectors for equality prior to proceeding with the device service in polled mode. The polarity of the GINT bit 7 is not affected by the GINT polarity control (Mode bit 3). The Status register is read by executing a read operation ($\overline{CS}$ = 0, RS = 0) with the control location selected (C/$\overline{D}$ = 1).

Mode Register: The 8-bit Mode register controls the operating options of the Am9519A. Figure 2 shows the bit assignments for the Mode register. The five low order mode bits (0 through 4) are loaded in parallel by command. Bits 5, 6 and 7 are controlled by separate commands. (See Figure 4.) The Mode register cannot be read out directly to the data bus, but Mode bits 0, 2 and 7 are available as part of the Status register.

Command Register: The 8-bit Command register stores the last command entered. Depending upon the command opcode, it may initiate internal actions or precondition the part for subsequent data bus transfers. The Command register is loaded by executing a write operation ($\overline{WR}$ = 0) with the control location selected (C/$\overline{D}$ = 1), as shown in Figure 3.

Byte Count Register: The length in bytes of the response associated with each interrupt is independently programmed so that different interrupts may have different length responses. The byte count for each response is stored in eight 2-bit Byte Count registers. For a given interrupt, the Am9519A will expect to receive a number of $\overline{IACK}$ pulses that equal the corresponding byte count and will hold $\overline{RIP}$ LOW until the count is satisfied.

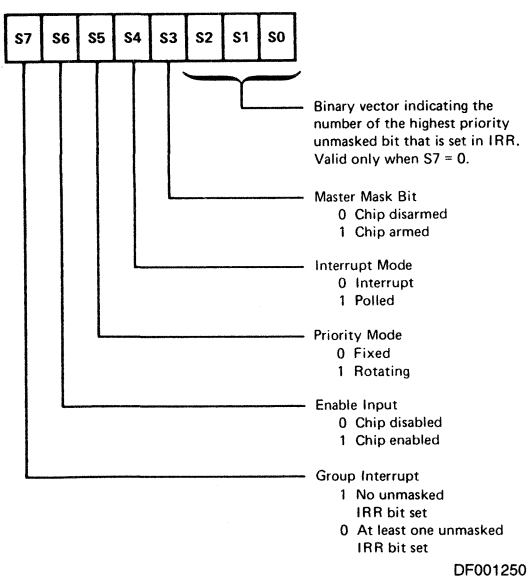


Figure 1. Status Register Bit Assignments

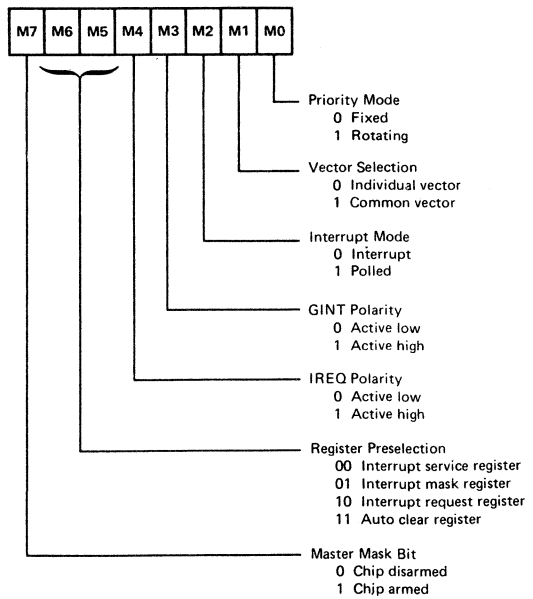


Figure 2. Mode Register Bit Assignments

DETAILED DESCRIPTION

Interrupts are used to improve system throughput and response time by eliminating heavy dependence on software polling procedures. Interrupts allow external devices to asynchronously modify the instruction sequence of a program being executed. In systems with multiple interrupts, vectoring can further improve performance by allowing direct identification of the interrupting device and its associated service routine. The Am9519A Universal Interrupt Controller contains, on one chip, all of the circuitry necessary to detect, prioritize and manage eight vectored interrupts. It includes many

options and operating modes that permit the design of sophisticated interrupt systems.

Reset

The reset function is accomplished by software command or automatically during power-up. The reset command may be issued by the CPU at any time. Internal power-up circuitry is triggered when VCC reaches a predetermined threshold, causing a brief internal reset pulse. In both cases, the resulting internal state of the machine is that all registers are cleared except the Mask register which is set. Thus, no Group Interrupt will be generated, and no interrupt requests will be recognized. The response memory and Byte Count registers are not affected by reset. Their contents after power-up are unpredictable and must be established by the host CPU during initialization.

Operating Sequence

A brief description of a typical sequence of events in an operating interrupt system will illustrate the general interactions among the host CPU, the interrupt controller and the interrupting peripheral.

1. The Am9519A controller is initialized by the CPU to customize its configuration and operation for the application at hand. Both the controller and the CPU are then enabled to accept interrupts.
2. One (or more) of the interrupt request inputs to the controller becomes active indicating that peripheral equipment is asking for service. The controller asynchronously accepts and latches the request(s).
3. If the request is masked, no further action takes place. If the request is not masked, a Group Interrupt output is generated by the controller.
4. The GINT signal is recognized by the CPU which normally will complete the execution of the current instruction, insert an interrupt acknowledge sequence into its instruction execution stream, and disable its internal interrupt structure. The controller expects to receive one or more $\overline{IACK}$ signals from the CPU during the acknowledge sequence.
5. When the controller receives the $\overline{IACK}$ signal, it brings $\overline{PAUSE}$ low and selects the highest priority unmasked pending request. When selection is complete, the $\overline{RIP}$ output is brought low and the first byte in the response memory associated with the selected request is output on the data bus. $\overline{PAUSE}$ stays low until RIP goes low. $\overline{RIP}$ stays low until the last byte of the response has been transferred.
6. During the acknowledge sequence, the IRR bit corresponding to the selected request is automatically cleared, and the corresponding ISR bit is set by the falling edge of $\overline{IACK}$. When the ISR bit is set, the Group Interrupt output is disabled until a higher priority request arrives or the ISR bit is cleared. The ISR bit will be cleared by either hardware or software.
7. If a higher priority request arrives while the current request is being serviced, GINT will be output by the controller, but will be recognized and acknowledged only if the CPU has its interrupt input enabled. If acknowledged, the corresponding higher priority ISR bit will be set and the requests nested.

Information Transfers

Figure 3 shows the control signal configurations for all information transfer operations between the Am9519A and the data bus. The following conventions are assumed: $\overline{RD}$ and $\overline{WR}$ active are mutually exclusive; $\overline{RD}$, $\overline{WR}$ and C/$\overline{D}$ have no meaning unless $\overline{CS}$ is LOW; active $\overline{IACK}$ pulses occur only when $\overline{CS}$ is HIGH.

For reading, the Status register is selected directly by the C/$\overline{D}$ control input. Other internal registers are read by preselecting the desired register with mode bits 5 and 6, and then executing a data read. The response memory can be read only with $\overline{IACK}$ pulses. For writing, the Command register is selected directly by the C/$\overline{D}$ control input. The Mask and Auto Clear registers are loaded following specific commands to that effect. To load each level of the response memory, the response preselect command is issued to select the desired level. An appropriate number of data write operations are then executed to load that level.

CONTROL INPUT					DATA BUS OPERATION
$\overline{CS}$	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{IACK}$	
0	0	0	1	1	Transfer contents of preselected data register to data bus
0	0	1	0	1	Transfer contents of data bus to preselected data register
0	1	0	1	1	Transfer contents of status register to data bus
0	1	1	0	1	Transfer contents of data bus to command register
1	X	X	X	0	Transfer contents of selected response memory location to data bus
1	X	X	X	1	No information transferred

Figure 3. Summary of Data Bus Transfers

The Pause output may be used by the host CPU to ensure that propertiming relationships are maintained with the Am9519A when $\overline{IACK}$ is active. The $\overline{IACK}$ pulse width required depends on several variables, including: operating temperature, internal logic delays, number of interrupt controllers chained together, and the priority level of the interrupt being acknowledged. When delays in these variables combine to delay selection of a request following the falling edge of the first $\overline{IACK}$, the Pause output may be used to extend the $\overline{IACK}$ pulse, if necessary. Pause will remain LOW until a request has been selected, as indicated by the falling edge of $\overline{RIP}$. Typically, the internal interrupt selection process is quite fast, especially for systems with a single Am9519A, and Pause will consequently remain LOW for only a very brief interval and will not cause extension of the $\overline{IACK}$ timing.

Operating Options

The Mode register specifies the various combinations of operating options that may be selected by the CPU. It is cleared by power-up or by a reset command. Mode bit 0 specifies the rotating/fixed priority mode (see Figure 2). In the fixed mode, priority is assigned to the request inputs based upon their physical location at the chip interface, with IREQ0 the highest and IREQ7 the lowest. In the rotating mode, relative priority is the same as for the fixed mode and the most recently serviced request is assigned the lowest priority. In the fixed mode, a lower priority request might never receive service if enough higher priority requests are active. In the rotating mode, any request will receive service within a maximum of seven other service cycles no matter what pattern the request inputs follow.

Mode bit 1 selects the individual/common vector option. Individual vectoring provides a unique location in the response memory for each interrupt request. The common vector option always supplies the response associated with IREQ0 no matter which request is being acknowledged.

Mode bit 2 specifies interrupt or polled operation. In the polled mode, the Group Interrupt output is disabled. The CPU may

read the Status register to determine if a request is pending. Since $\overline{IACK}$ pulses are not normally supplied in polled mode, the IRR bit is not automatically cleared, but may be cleared by command. With no $\overline{IACK}$ input, the ISR and the response memory are not used. An Am9519A in the polled mode has EI connected to EO so that in multichip interrupt systems the polled chip is functionally removed from the priority hierarchy.

Mode bit 3 specifies the sense of the GINT output. When active high polarity is selected, the output is a two-state configuration. For active low polarity, the output is open drain and requires an external pull-up resistor to provide the high logic level. The open drain output allows wired-or configurations with other similar output signals.

Mode bit 4 specifies the sense of the IREQ inputs. When active low polarity is selected, the IRR responds to falling edges on the request inputs. When active high is selected, the IRR responds to rising edges.

Mode bits 5 and 6 specify the register that will read on subsequent data read operations (C/$\overline{D}$ = 0, $\overline{RD}$ = 0). This preselection remains valid until changed by a reset or a command.

Mode bit 7 is the master mask bit that disables all request inputs. It is used to disable all interrupts without modifying the IMR so that the previous IMR contents are valid when interrupts are re-enabled. When the master mask bit is LOW, it causes the EO line to remain disabled (LOW). Thus, for multiple-chip interrupt systems, one master mask bit can disable the whole interrupt structure. Alternatively, portions of the structure may be disabled. The state of the master mask bit is available as bit S3 of the Status register.

Programming

After reset, the Am9519A must be initialized by the CPU to perform useful work. At a minimum, the master mask bit and at least one of the IMR bits should be enabled. If vectoring is to be used, the response memory must be loaded; if not, the mode must be changed to a non-vectored configuration. Normally, the first step will be to modify the Mode register and the Auto clear register to establish the configuration desired for the application. Then the response memory and byte count will be loaded for those request levels that will be in use. The response memory for every channel must be written even if the channel is not used. Every byte need not be written, only those specified by the byte count. Finally, the master mask bit and at least portions of the IMR will be enabled to allow interrupt processing to proceed.

Commands

The host CPU configures, changes and inspects the internal condition of the Am9519A using the set of commands shown in Figure 4. An "X" entry in the table indicates a "don't care" state. All commands are entered by directly loading the Command register as shown in Figure 3 (C/$\overline{D}$ = 1, $\overline{WR}$ = 0). Figure 5 shows the coding assignments for the Byte Count registers. (A detailed description of each command is contained in the Am9519A Application Note AMPUB-071.)

BY1	BY0	COUNT
0	0	1
0	1	2
1	0	3
1	1	4

Figure 5. Byte Count Coding

COMMAND CODE								COMMAND DESCRIPTION
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	Reset
0	0	0	0	1	X	X	X	Clear all IRR and all IMR bits
0	0	0	1	1	B2	B1	B0	Clear IRR and IMR bit specified by B2, B1, B0
0	0	1	0	0	X	X	X	Clear all IMR bits
0	0	1	0	1	B2	B1	B0	Clear IMR bit specified by B2, B1, B0
0	0	1	1	0	X	X	X	Set all IMR bits
0	0	1	1	1	B2	B1	B0	Set IMR bit specified by B2, B1, B0
0	1	0	0	0	X	X	X	Clear all IRR bits
0	1	0	0	1	B2	B1	B0	Clear IRR bit specified by B2, B1, B0
0	1	0	1	0	X	X	X	Set all IRR bits
0	1	0	1	1	B2	B1	B0	Set IRR bit specified by B2, B1, B0
0	1	1	0	X	X	X	X	Clear highest priority ISR bit
0	1	1	1	0	X	X	X	Clear all ISR bits
0	1	1	1	1	B2	B1	B0	Clear ISR bit specified by B2, B1, B0
1	0	0	M4	M3	M2	M1	M0	Load Mode register bits 0 – 4 with specified pattern
1	0	1	0	M6	M5	0	0	Load Mode register bits 5, 6 with specified pattern
1	0	1	0	M6	M5	0	1	Load Mode register bits 5, 6 and set mode bit 7
1	0	1	0	M6	M5	1	0	Load Mode register bits 5, 6 and clear mode bit 7
1	0	1	1	X	X	X	X	Preselected IMR for subsequent loading from data bus
1	1	0	0	X	X	X	X	Preselected Auto Clear register for subsequent loading from data bus
1	1	1	BY1	BY0	L2	L1	L0	Load BY1, BY0 into byte count register and preselect response memory level specified by L2, L1, L0 for subsequent loading from data bus

Figure 4. Am9519A Command Summary

APPLICATIONS

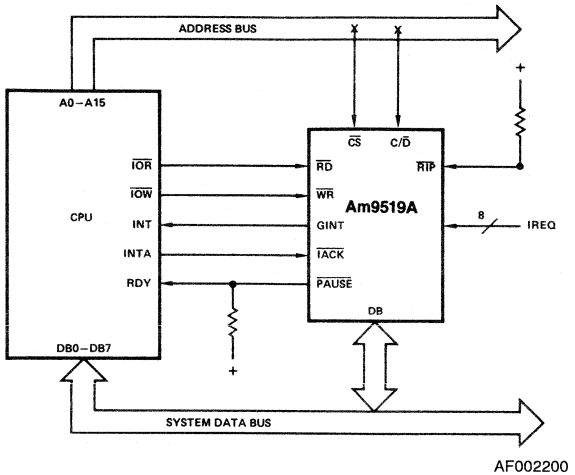


Figure 6. Base Interrupt System Configuration

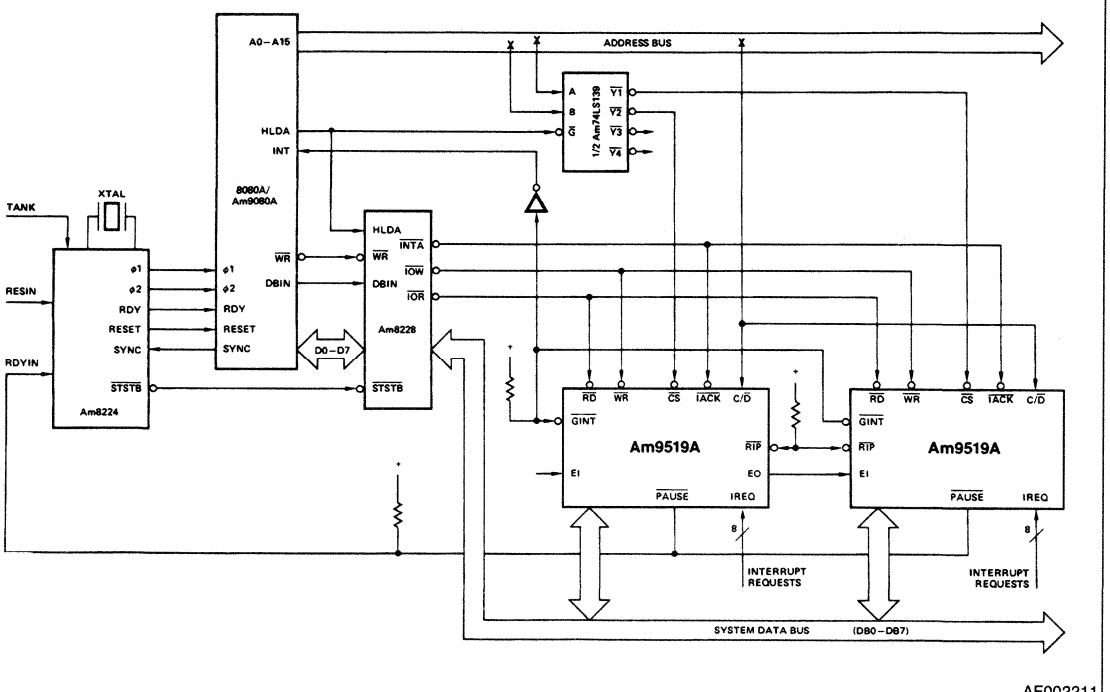


Figure 7. Expanded Interrupt System Configuration

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
VCC with Respect to VSS ..... −0.5V to +7.0V
All Signal Voltages
with Respect to VSS ..... −0.5V to +7.0V
Power Dissipitation (Package Limitation) 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ± 5%

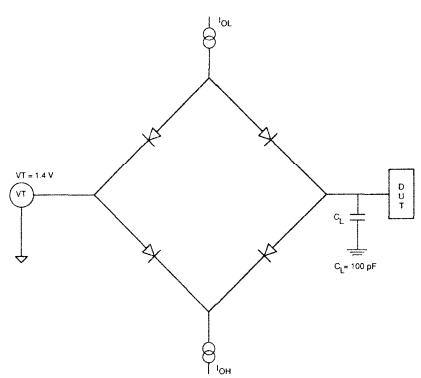
Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ± 10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range.

Parameters	Description	Test Conditions		Min	Max	Units
VOH	Output High Voltage (Note 8)	IOH = −200μA		2.4		Volts
		IOH = −100μA (EO only)		2.4		
VOL	Output Low Voltage	IOL = 3.2mA			0.4	Volts
		IOL = 1.0mA (EO only)			0.4	
VIH	Input High Voltage			2.0	VCC	Volts
VIL	Input Low Voltage			−0.5	0.8	Volts
IIX	Input Load Current	VSS ≤ VIN ≤ VCC	EI Input	−60	10	μA
			Other Inputs	−10	10	
IOZ	Output Leakage Current	VSS ≤ VOUT ≤ VCC, Output Off		−10	10	μA
ICC	VCC Supply Current	Commercial			125	mA
		Industrial			185	
CO	Output Capacitance	fc = 1.0MHz			15	pF
CI	Input Capacitance	T_A = 25°C			10	
CIO	I/O Capacitance	All pins at 0V			20	

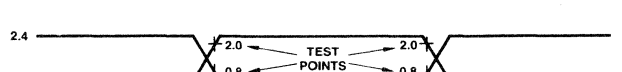
SWITCHING TEST CIRCUIT



TC004200

This test circuit is the dynamic load of a Teradyne J941.

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF007820

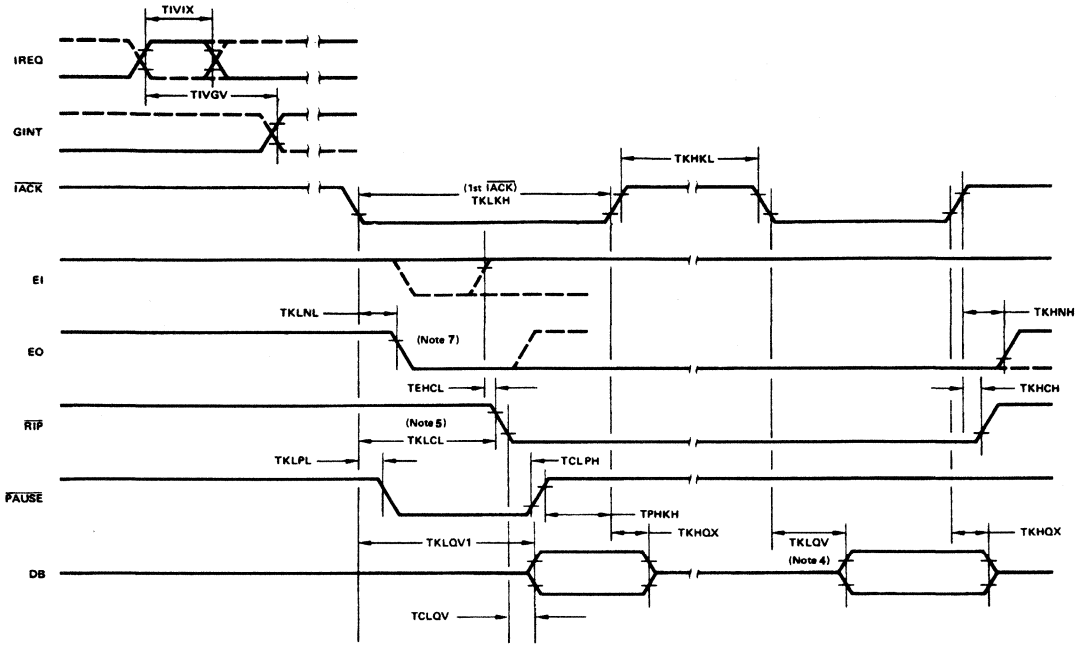
See section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating ranges (Notes 1, 2)

Parameters	Description	Am9519A		Am9519A-1		Units
		Min	Max	Min	Max	
TAVRL	C/$\overline{D}$ Valid and $\overline{CS}$ LOW to Read LOW	0		0		ns
TAVWL	C/$\overline{D}$ Valid and $\overline{CS}$ LOW to Write LOW	0		0		ns
TCLPH	$\overline{RIP}$ LOW to $\overline{PAUSE}$ HIGH (Note 3)	75	375	75	375	ns
TCLQV	$\overline{RIP}$ LOW to Data Out Valid (Note 4)		50		40	ns
TDVWH	Data in Valid to Write HIGH	250		200		ns
TEHCL	Enable in HIGH to $\overline{RIP}$ LOW (Note 5)	30	300	30	300	ns
TIVGV	Interrupt Request Valid to Group Interrupt Valid	100	800		650	ns
TIVIX	Interrupt Request Valid to Interrupt Request Don't Care (IREQ Pulse Duration)	250		250		ns
TKHCH	$\overline{IACK}$ HIGH to $\overline{RIP}$ HIGH (Note 5)		450		350	ns
TKHKL	$\overline{IACK}$ HIGH to $\overline{IACK}$ LOW ($\overline{IACK}$ Recovery)	140		120		ns
TKHNH	$\overline{IACK}$ HIGH to EO HIGH (Notes 6, 7)		975		750	ns
TKHQX	$\overline{IACK}$ HIGH to Data Out Invalid	20	200	20	100	ns
TKLCL	$\overline{IACK}$ LOW to $\overline{RIP}$ LOW (Notes 5, 9)	75	600	75	450	ns
TKLKH	$\overline{IACK}$ LOW to $\overline{IACK}$ HIGH (1st $\overline{IACK}$) (Note 9)	975		800		ns
TKLNL	$\overline{IACK}$ LOW to EO LOW (Notes 6, 7, 9)		125		100	ns
TKLPL	$\overline{IACK}$ LOW to $\overline{PAUSE}$ LOW (Note 9)	25	175	25	125	ns
TKLQV	$\overline{IACK}$ LOW to Data Out Valid (Notes 4, 9)	25	300	25	200	ns
TKLQV1	1st $\overline{IACK}$ LOW to Data Out Valid (Note 9)	75	650	75	490	ns
TPHKH	$\overline{PAUSE}$ HIGH to $\overline{IACK}$ HIGH	0		0		ns
TRHAX	Read HIGH to C/$\overline{D}$ and $\overline{CS}$ Don't Care	0		0		ns
TRHQX	Read HIGH to Data Out Invalid	20	200	20	100	ns
TRLQV	Read LOW to Data Out Valid		300		200	ns
TRLQX	Read LOW to Data Out Unknown	35		35		ns
TRLRH	Read LOW to Read HIGH ($\overline{RD}$ Pulse Duration)	300		250		ns
TWHAX	Write HIGH to C/$\overline{D}$ and $\overline{CS}$ Don't Care	25		25		ns
TWHDX	Write HIGH to Data in Don't Care	25		25		ns
TWHRW	Write HIGH to Read or Write LOW (Write Recovery)	600		400		ns
TWLWH	Write LOW to Write HIGH ($\overline{WR}$ Pulse Duration)	300		250		ns
TKHIH	$\overline{IACK}$ HIGH to GINT inactive		1000		800	ns

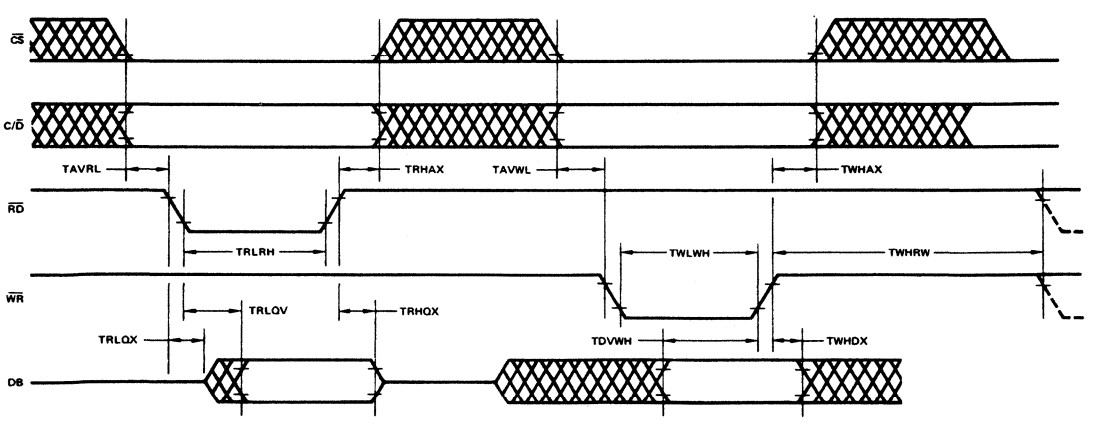
Notes:
1. Transition abbreviations used for the switching parameter symbols include: H = HIGH, L = LOW, V = Valid, X = unknown or don't care, Z = high-impedance.
2. Signal abbreviations used for the switching parameter symbols include: R = Read, W = Write, Q = Data Out, D = Data In, A = Address ($\overline{CS}$ and C/$\overline{D}$), K = Interrupt Acknowledge, N = Enable Out, E = Enable In, P = Pause, C = $\overline{RIP}$.
3. During the first $\overline{IACK}$ pulse, $\overline{PAUSE}$ will be LOW long enough to allow for priority resolution and will not go HIGH until after RIP goes LOW (TCLPH).
4. TKLQV applies only to second, third and fourth $\overline{IACK}$ pulses while $\overline{RIP}$ is LOW. During the first $\overline{IACK}$ pulse, Data Out will be valid following the falling edge of $\overline{RIP}$ (TCLQV).
5. $\overline{RIP}$ is pulled LOW to indicate that an interrupt request has been selected. $\overline{RIP}$ cannot be pulled LOW until EI is HIGH following an internal delay. TKLCL will govern the falling edge of $\overline{RIP}$ when EI is always HIGH or is HIGH early in the acknowledge cycle. The TEHCL will govern when EI goes HIGH later in the cycle. The rising edge of EI will be determined by the length of the preceding priority resolution chain.
   $\overline{RIP}$ remains LOW until after the rising edge of the $\overline{IACK}$ pulse that transfers the last response byte for the selected IREQ.
6. Test conditions for the EO line assume an output loading of IOL = 1.0 mA and IOH = −100 μA. Since EO normally only drives EI of another Am9519A, higher speed operations can be specified with this more realistic test condition.
7. The arrival of $\overline{IACK}$ will cause EO to go LOW, disabling additional circuits that may be connected to EO. If no valid interrupt is pending, EO will return HIGH when EI is HIGH. If a pending request is selected, EO will stay LOW until after the last $\overline{IACK}$ pulse for that interrupt is complete and $\overline{RIP}$ goes HIGH.
8. VOH specifications do not apply to $\overline{RIP}$, $\overline{PAUSE}$, or to $\overline{GINT}$ when active-low. These outputs are open drain, and VOH levels will be determined by external circuitry.
9. $\overline{CS}$ must be HIGH for at least 100ns prior to $\overline{IACK}$ going LOW.

SWITCHING WAVEFORMS



WF003551

Interrupt Operations



WF003560

Data Bus Transfers

Am9520/Am9521/AmZ8065

Burst Error Processor

DISTINCTIVE CHARACTERISTICS

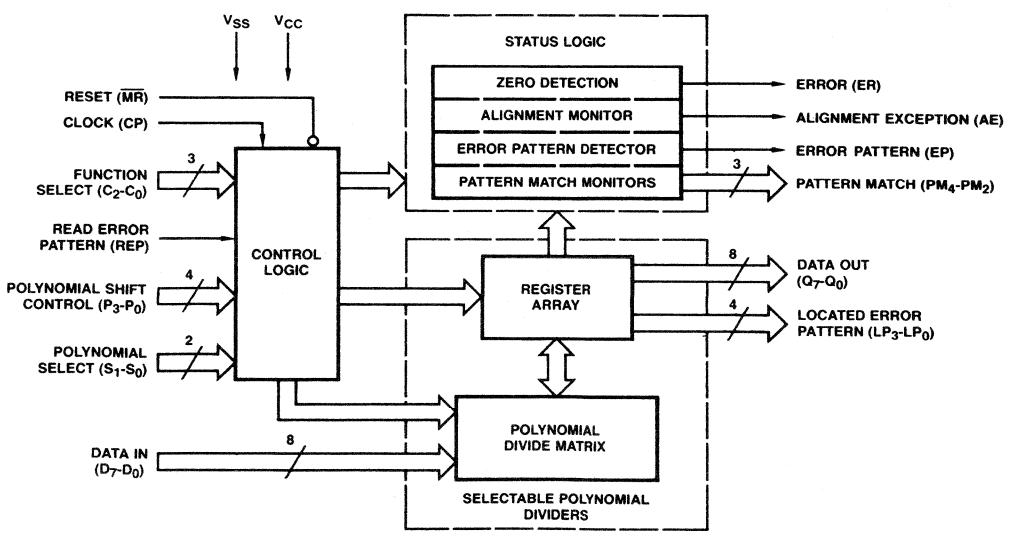
- **Provides for detection and correction of burst errors**
  Detects errors in serial data up to 585K bits long. Allows correction of error bursts of up to 12 bits.
- **High-Speed Operation**
  Effective data rates up to 20 Mbits/second for Am9520/Am9521/AmZ8065 and 30 Mbits/second for -1 versions. Fast enough for high-performance hard and soft disk systems.
- **Selectable Industry-Standard Polynomials**
  35-bit and 32-bit polynomials on Am9521. Am9520/AmZ8065 additionally has popular IBM 56-bit and 48-bit versions.
- **Three correction algorithms provide flexibility**
  Full-period clock-around method for conforming to current practices. Chinese remainder theorem reduces correction time by orders of magnitude. Reciprocal polynomial makes correction possible with 48-bit code.
- **Designed for use in both microprogrammed and microprocessor disk controller systems**
  Device complements both AmZ8000 and Am2900 microprocessor families and can also be used with other microprocessors.

GENERAL DESCRIPTION

The Burst Error Processor (BEP) provides for error detection and correction for high-performance disk systems and other systems in which high-speed serial data transfer takes place. As data density and transfer rates increase in both hard and floppy disks and other storage media, error detection and correction become increasingly important. The BEP is an LSI circuit that facilitates the most common error detection and correction schemes accommodating data streams of up to 585K bits at up to 20M bits/second effective data rate.

The BEP provides a choice of four standard polynomials, including the popular 56-bit and 48-bit versions, to satisfy a broad range of applications. The device divides the data stream by the selected polynomial using the rules of algebra in polynomial fields. The resulting remainder is the check word, which is then appended to the data for writing on the disk as a record. When the record is read back, the BEP computes the syndrome for data validation. If an error is detected, the location and pattern of this burst in the data stream is determined for corrections.

BLOCK DIAGRAM



BD003300

Publication # 00617 | Rev. B | Amendment /0
Issue Date: April 1985

Am9568

Data Ciphering Processor
(DCP)

DISTINCTIVE CHARACTERISTICS

- **Encrypts and decrypts data**
  Implements National Bureau of Standards Data Encryption Standard (DES) algorithm
- **Throughput over 1.5M bytes per second**
  Operates at data rates fast enough for disk controllers, high-speed DMA, telecommunication channels
- **Supports three ciphering options**
  Electronic Code Book for disk applications, Cipher Block Chain for high-speed telecommunications, and Cipher Feedback for low-to-medium speed, byte-oriented communications
- **Three separate key registers on one chip**
  Separate registers for encryption key, decryption key and master key improve system security and throughput by eliminating need to reload keys frequently.
- **Three separate data ports provide flexible interface, improved security**
  The DCP utilizes a Master Port, Slave Port and Key Port. Functions of the three ports can be programmed by the user to provide for simple interface to iAPX86 and Am2900 systems and to provide total hardware separation of encrypted data, clear data and keys.

GENERAL DESCRIPTION

The Am9568 Data Ciphering Processor is an N-channel silicon gate LSI product containing the circuitry necessary to encrypt and decrypt data using the National Bureau of Standards Encryption Algorithm. It is designed to be used in a variety of environments, including dedicated controllers, communication concentrators, terminals and peripheral task processors in general processor systems.

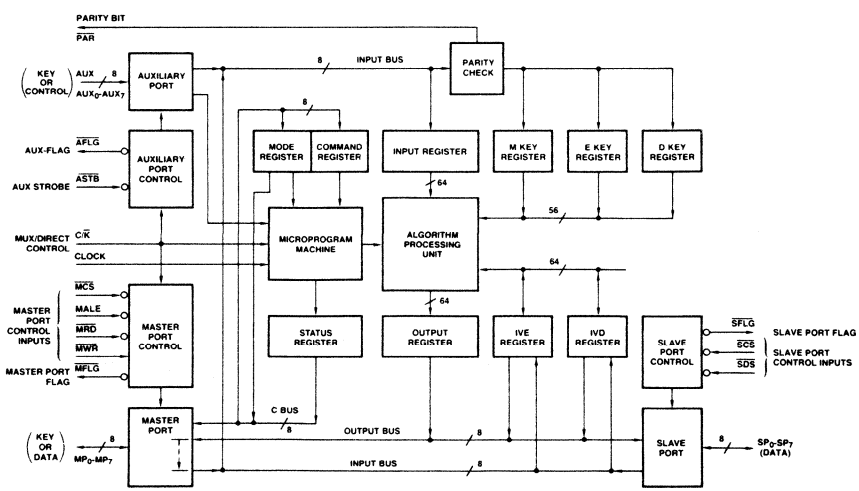
The DCP provides a high throughput rate using Cipher Feedback, Electronic Code Book or Cipher Block Chain operating modes. Separate ports for key input, clear data and enciphered data enhance security.

The system communicates with the DCP using commands entered in the Master Port and through auxiliary control lines. Once set up, data can flow through the DCP at high speeds because input, output and ciphering activities are all performed concurrently. External DMA control can easily be used to enhance throughput in some system configurations.

This device is designed to interface directly to the iAPX86, 88 CPU bus and, with a minimum of external logic, to the 2900 and 8051 families of processors.

BLOCK DIAGRAM

NTExport of this device from the United States is subject to control by the U.S. Department of State.



BD003370

Publication # 05178 | Rev. B | Amendment /0
Issue Date: April 1985

Am9580A

Hard Disk Controller (HDC)

PRELIMINARY

DISTINCTIVE CHARACTERISTICS

- Supports ST506/412 and IBM double density-floppy format
- Controls up to four drives, any mix of hard and floppy disk drives
- Two on-chip 512-byte sector buffers support zero-sector interleaving
- Error correct algorithms supported:
  - CRC/CCITT
  - Single-Burst Reed-Solomon
  - Double-Burst Reed-Solomon
  - External ECC (user-definable Error Correcting Code)
- Linked-list command and data structure
- On-chip DMA controller supports 32-bit addressing and 8/16-bit data

GENERAL DESCRIPTION

The Am9580A Hard Disk Controller (HDC) is a single-chip solution to the problems encountered in designing Data Formatters and Disk System Controllers. Together with its companion part, the Am9582 Disk Data Separator (DDS), the Am9580A provides all the functions which, until now, have been found only on sophisticated board-level products.

The Am9580A is flexible enough to cope with the differing requirements in today's broad marketplace while using the advanced technology and innovative features that tomorrow's market will demand.

The Am9580A supports both rigid and flexible disk drives and their respective data formats. The Am9580A can control up to four drives, allowing any mix of rigid and flexible drives. The characteristics of each drive are independently user-programmable.

A sophisticated on-chip DMA Controller fetches commands, writes status information, fetches data to be written on disk and writes data that has been read from disk. The DMA operation is programmable to adjust the bus occupancy, data bus width (8-bit or 16-bit), and Wait State insertion. Two sector buffers allow zero sector interleaving to access data on physically adjacent sectors, improving both file access time and system throughput. Sector sizes of 128, 256 and 512 bytes are programmable.

The Am9580A insures data integrity by selecting one of two methods: either by selecting an error detecting code (CRC-CCITT), or one of two error correcting codes (Single- or Double-Burst Reed-Solomon). Additionally, the HDC provides handshake signals to control external ECC circuitry to implement any user-definable ECC algorithm.

The Am9580A provides signals which are necessary to control external Encode/Decode and Address Mark circuitry (e.g., the Am9582). By partitioning the disk control system this way, future developments in the field of data encoding (e.g., RLL codes) will be able to take advantage of the HDC's advanced data formatting and control capabilities.

The flexible, user-programmable disk interface can be configured to control ST506/412 or standard double-density floppy-disk interfaces. With additional circuitry, it can easily be adapted to other interface standards.

The Am9580A provides a comprehensive, high-level command set for multi-sector disk I/O, marginal data recovery, diagnostics, and error recovery. Commands may be linked together to be executed sequentially by the Am9580 without any host intervention. This linked-list command structure also simplifies command insertion, deletion, or rearrangement.

Publication #	Rev.	Amendment
05308	B	/0

Issue Date: **August 1986**

Am9582

Disk Data Separator (DDS)

PRELIMINARY

DISTINCTIVE CHARACTERISTICS

- Complete Single-Chip Disk Data Separator for Floppy Disk and Hard Disk Drives (minimal external components)
- Complete on-chip Phase-Locked-Loop (PLL), frequency may be dynamically changed
- Supports:
  - 4 to 16 Mbit/sec MFM data rate for hard disks
  - 125 to 500 kbit/sec FM data rate for single-density floppy disks
  - 250 to 1000 kbit/sec MFM data rate for double-density floppy disks
- On-chip Write Pre-Compensation Logic (frequency proportional)
- On-chip Address Mark Generator/Detector
- One Am9582 can support both floppy- and hard-disk drives. The on- chip analog section can be dynamically switched between the two modes. No external component needs to be switched.

GENERAL DESCRIPTION

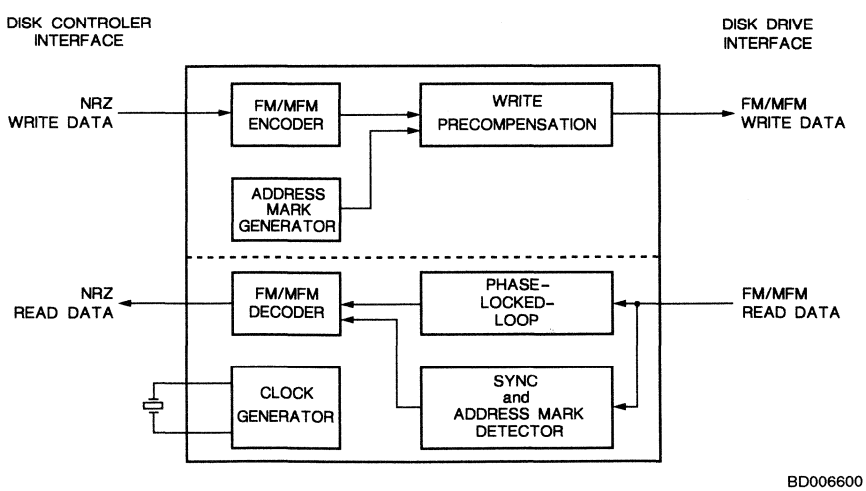
The Am9582 Disk Data Separator (DDS) is a single-chip solution to several functions associated with reading and writing data to systems having floppy- or hard-disk drives. The Am9582 is divided into two basic sections: the Read Section and the Write Section.

The Read Section contains an on-chip Phase-Locked-Loop (PLL) to provide a read clock signal that tracks the FM (Frequency Modulation) or MFM (Modified Frequency Modulation) serial data from the disk. The MFM or FM data is then fed into the MFM/FM decoder to be converted into NRZ (Non Return to Zero) data. A built-in Address Mark detector recognizes the standard address marks for both floppy and hard disks.

The Write Section contains an encoder which encodes the incoming NRZ data and the write (reference) clock into a single stream of either FM or MFM encoded data. The Write Section also contains an Address Mark generator and the Write Pre-Compensation Logic. This Address Mark generator can generate the standard Address Marks for both floppy- and hard-disk data formats. Write Pre-Compensation compensates bit-shifting caused by the characteristics (pulse superpositioning) of the magnetic media.

The Am9582 is the companion device for the Am9580A Hard Disk Controller or the Am9590 ESDI Hard Disk Controller. These chip sets provide a complete disk-controller solution to interface systems with ST506- or ST412-type hard-disk drives and floppy-disk drives. Also, the Am9582 can be integrated on disk drives offering a NRZ data interface (e.g., ESDI or SMD) to implement the on-drive data separator functions (MFM) or the PLL functions (RLL).

BLOCK DIAGRAM



Publication # 06104 | Rev. A | Amendment /0
Issue Date: August 1986

Am95C85

Content Addressable Data Manager

PRELIMINARY

DISTINCTIVE CHARACTERISTICS

- High-performance sorting, searching, and updating
- 1K byte software-reconfigurable memory array
- Programmable record size
- Cascadable up to 256 devices
- Content-addressable operation, independent of record size
- Intelligent peripheral with sixteen powerful instructions
- Stack mode allows inserting of data without resorting
- Up to 16-MHz operation
- CMOS technology

GENERAL DESCRIPTION

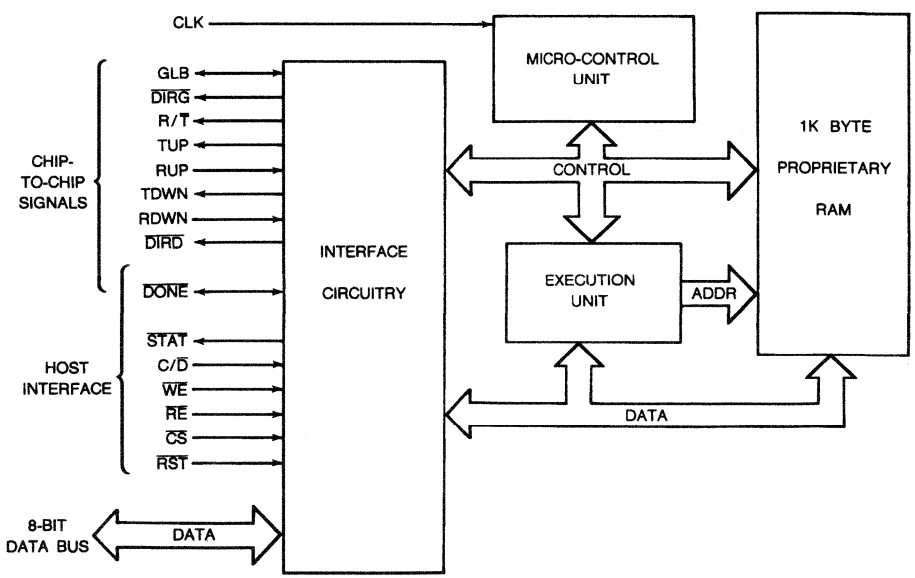
The Am95C85 Content Addressable Data Manager (CADM) is an intelligent CMOS peripheral device designed to enhance the performance of applications involving sorting, searching, and insertion or deletion. Orders of magnitude performance improvement can be seen when compared to the implementation using software algorithms.

The CADM uses an on-chip proprietary 1K byte memory for data manipulation. This specially designed memory can be easily reconfigured to meet different application requirements. The data stored in the CADM are collated into records that consist of a key field and a pointer field. The length of these two fields are software programmable. The sorting and searching of records are based on the values of the key fields. A mask register is also provided to selectively mask out unwanted bits in the key field for comparison. For applications that require large storage area for data manipulation, the CADM can be easily cascaded up to 256 devices.

Content-addressable operation allows the host to retrieve data without having to do extensive searching. Address generation for memory access is done internally, relieving the host from the burden of physical address calculation. Stack-mode operation allows the user to delete records simply by popping the records out of memory, and to insert records by pushing the records into the memory.

By providing content-addressable searching, automatic sorting, programmable record length, and address-independent operation, the CADM allows the host to off-load repetitive, time-consuming data manipulation. For applications that require substantial sorting, searching, and updating operations, the CADM offers significant improvement in overall performance.

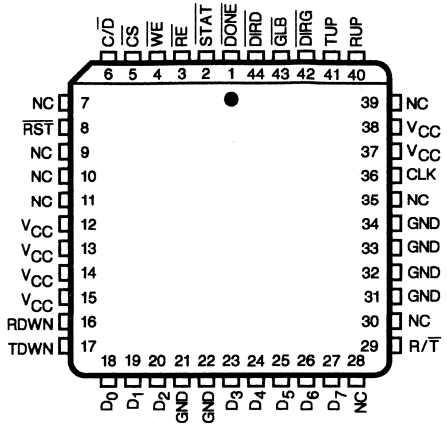
BLOCK DIAGRAM



BD005352

Publication # 06956 | Rev. B | Amendment /0
Issue Date: March 1987

CONNECTION DIAGRAM
Top View

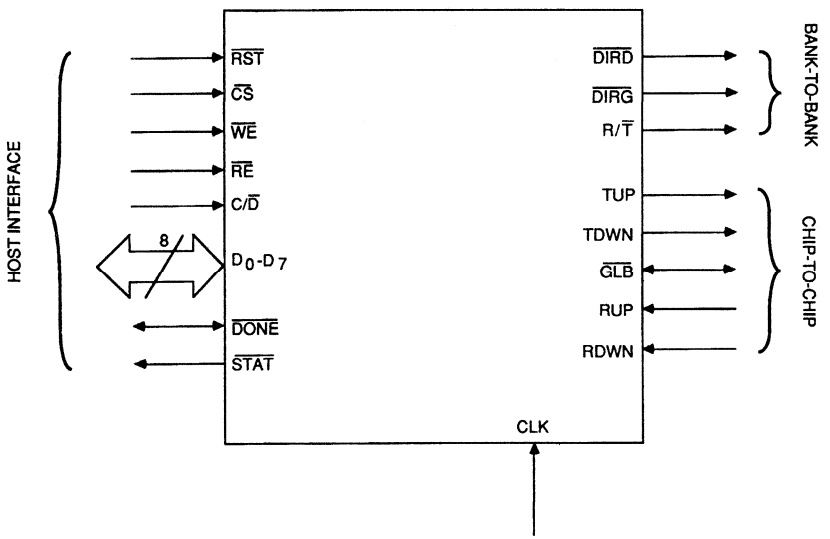


NC = No Connection

CD010442

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL

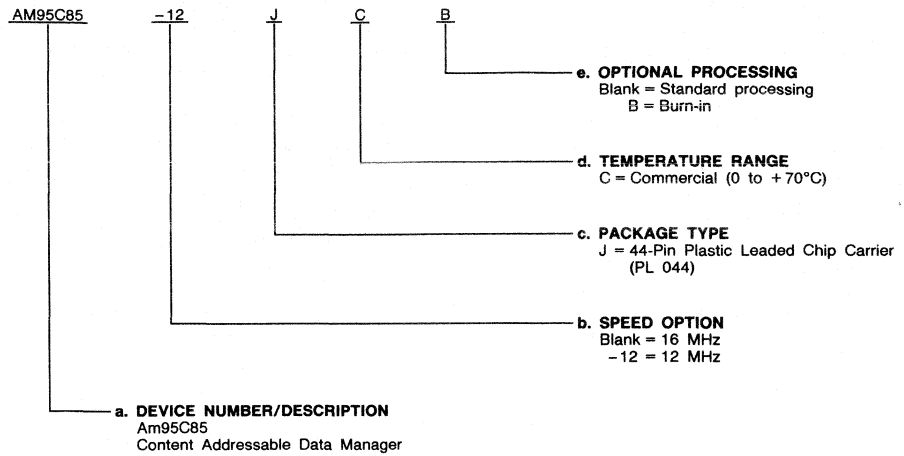


LS002891

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM95C85	JC, JCB
AM95C85-12	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Host Interface

C/$\overline{D}$ Command/$\overline{Data}$ (Input)
A HIGH on this input allows the command register to be loaded with the information on the data bus. A LOW on this input allows the data to be read from, or written into, the internal RAM.

$\overline{CS}$ $\overline{Chip\ Select}$ (Input; Active LOW)
The $\overline{CS}$ input enables the host CPU to perform read or write operations with the Am95C85 devices. The read and write inputs are ignored when $\overline{CS}$ is HIGH.

$D_0 - D_7$ Data Bus (Input/Output; Three State)
The eight bidirectional data pins are used for information exchanges between the Am95C85 (CADM) and the host processor, and between CADM parts themselves. A HIGH on a data line corresponds to a Logic "1," and a LOW corresponds to a Logic "0." These lines act as inputs when $\overline{WE}$ and $\overline{CS}$ are active, and as outputs when $\overline{RE}$ and $\overline{CS}$ are active. D_0 is the least significant bit and D_7 the most significant bit.

$\overline{DONE}$ $\overline{Done}$ (Input/Output; Active LOW, Three State)
This signal indicates the termination of an operation, and is precharged to HIGH at the beginning of a new command, data writes, or data reads. A LOW on this output indicates the device is ready for the next command or data transfer.

$\overline{RE}$ $\overline{Read\ Enable}$ (Input; Active LOW)
The $\overline{RE}$ input, together with $\overline{CS}$ and C/$\overline{D}$ inputs, are used to control data transfer from the Am95C85 to the host. The Am95C85 will put the data onto the data bus when $\overline{RE}$, $\overline{CS}$, and C/$\overline{D}$ inputs are LOW.

$\overline{RST}$ $\overline{Reset}$ (Input; Active LOW)
A LOW on this input will reset the Am95C85. Any command under execution is terminated.

$\overline{STAT}$ $\overline{Status}$ (Output; Active LOW, Three State)
When LOW, the $\overline{STAT}$ output indicates that an exception condition has occurred following the execution of an instruction or data transfer. This pin is precharged to HIGH at the beginning of a new command, or when a write or read is initiated.

$\overline{WE}$ $\overline{Write\ Enable}$ (Input; Active LOW)
The simultaneous occurrence of $\overline{WE}$ and $\overline{CS}$ indicates that information from the data bus is to be transferred to the Am95C85. The C/$\overline{D}$ input determines whether the data will be loaded into the command register or internal RAM.

Chip-to-Chip Communication

These pins are used in chip-to-chip communications in multiple Am95C85 memory configurations. They do not affect the system interface.

$\overline{GLB}$ $\overline{Global}$ (Input/Output; Active LOW, Three State)
This signal is used for part-to-part synchronization during instruction execution. All CADM devices in the same bank should have this pin connected together and pulled up through a resistor to the power supply. This pin is precharged to HIGH at the beginning of a new command, or when a Write or Read is initiated.

RDWN Receive from Downward (Input; Active HIGH)
This pin should be connected to TUP of the next lower order CADM in cascade. The last chip in the daisy chain should have this pin pulled up to the power supply through a resistor.

RUP Receive from Upward (Input; Active HIGH)
This pin should be connected to TDWN of the next higher order CADM in cascade. The first chip in the daisy chain should have this pin pulled up to the power supply through a resistor.

TDWN Transmit Downward (Output; Active HIGH)
This signal is issued by the higher order CADM to the next lower order CADM, in cascade, to synchronize the chip-to-chip data transfer. It should be connected to RUP of the next lower order CADM.

TUP Transmit Upward (Output; Active HIGH)
This handshaking signal is issued by the lower order CADM to the next higher order CADM, in cascade, during chip-to-chip data transfer. It should be connected to RDWN of the next higher order CADM.

Bank-to-Bank Control

Bank-to-bank communication is needed when multiple banks of Am95C85 devices are used in a system. The CADM array can be grouped into multiple banks and separated by buffers. The following signals are used to control the direction of buffer signals that separate the banks. They can be left unconnected if only one bank is used.

$\overline{DIRD}$ $\overline{Direction\ of\ Done\ Signal}$ (Output; Active LOW, Three State)
This signal is used to control the direction of the circuit that is buffering the $\overline{DONE}$ signal. All CADMs in the same bank should have this pin connected together and pulled up through a resistor to the power supply. When driven LOW, this signal indicates that the CADM is driving the $\overline{DONE}$ signal. This pin is precharged to HIGH at the beginning of a new command, or when a Write or Read is initiated.

$\overline{DIRG}$ $\overline{Direction\ of\ Global\ Signal}$ (Output; Active LOW, Three State)
This signal is used to control the direction of the circuit that is buffering the $\overline{GLB}$ signal. All CADMs in the same bank should have this pin connected together and pulled up through a resistor to the power supply. When driven LOW, this signal indicates that the CADM is driving the $\overline{GLB}$ signal. This pin is precharged to HIGH at the beginning of a new command, or when a Write or Read is initiated.

R/$\overline{T}$ Receive/$\overline{Transmit}$ (Output)
This output is driven LOW when the Am95C85 is driving the data bus. It should be used to control the direction of buffers which isolate the data bus from specific Am95C85 banks.

FUNCTIONAL DESCRIPTION

Introduction

The Am95C85 Content Addressable Data Manager (CADM) is an intelligent peripheral device intended to relieve the host CPU of many of the time-consuming tasks associated with data-list manipulation. Sorting and finding data are tasks implemented by both applications software and operating systems. By providing these functions in hardware, which were previously the responsibility of software, execution time is reduced. This performance improvement can be 100 to 500 times, depending upon the application.

The 44-pin Am95C85 contains 1K byte of RAM whose organization is programmable. It contains a micro-engine, registers, pointers, and an instruction decoder. Most of these functions are transparent to the user.

The Programmer's View

Hardware

The Am95C85 CADM interacts with the host system through the use of a command port, data port, and two status pins called $\overline{STAT}$ and $\overline{DONE}$. Both the command and data ports are accessed through the single 8-bit data bus. The two ports are differentiated by the use of a Command/$\overline{Data}$ pin (C/$\overline{D}$). The familiar signals, $\overline{RE}$, $\overline{WE}$, and $\overline{CS}$ are used to write and read data or commands.

Data Array

The CADM contains 1K byte of internal RAM. It consists of a mask area, a record area, an unused area, and an input buffer area as shown in Figure 1. The mask area exists only if an SMB command is issued. The length of the mask area is the same length as the key, as specified by the KPL command. The data stored in this area is used to select the desired bits in the key field for comparison during the sorting and searching process. Those mask bits with "0" will cause the associated bits in the key field to be ignored during the comparison. The record space stores data as records in the CADM. This area starts from address "K" if the masking option is chosen, or "0" if the masking option is not chosen. It ends at the last address as programmed by the KPL command. The length of this area should be a multiple of (K + P) bytes. The last (K + P) bytes are designated as input buffer area. They are reserved to temporarily store the incoming record. The remaining area between the record area and the input buffer is unused area and is not accessible by the user. This area should be kept as small as possible to optimize the performance of the CADM.

The internal RAM structure, a patented AMD design, is unique in that the record width is controlled by the CPU, using the KPL command. Each record is comprised of two fields, referred to as a key field (K) and a pointer field (P). The KPL command sets the width of these two fields, then partitions the entire array into records, each with a length of K + P bytes. Figure 2 shows the logical model of the CADM data array. The length of K may vary from 1 to 255 bytes, and P may be set between 0 and 255 bytes. The variable record width provides significant flexibility that is very useful for general-purpose data manipulation. It allows complex operations, such as sort and search, to be performed on virtually any type of data. For example, the Am95C85 devices can be used to search a file-allocation table for a particular file address. It may then be reprogrammed to manipulate a disk-directory table. The Am95C85 can sort a database index file and is versatile enough to handle each of the tasks described above, even though each has a differernt record width.

The maximum number of records stored in each CADM depends on the record width (K + P) and the value of Last Address (LA). To efficiently use the memory space of the CADM, the LA should be programmed with the following value:

if mask bytes are used,

$$LA = \{INT[(1024 - 2*K - P)/(K + P)]\} * (K + P) + K - 1$$

if mask bytes are not used,

$$LA = \{INT[(1024 - K - P)/(K + P)]\} * (K + P) - 1$$

The Am95C85 array can be easily expanded if the application requires more record storage; up to 256 CADMs can be cascaded to meet the application requirements. The addition of hardware is transparent to software. The programmer still sees one command port, one data port, one $\overline{STAT}$ pin and one $\overline{DONE}$ pin. The only difference is that there is more record space for data manipulation. The number of CADM devices in cascade can be easily determined by reading the data port after a hardware or software reset.

Addressing Flexibility

To take advantage of the flexibility of the unique memory array, the Am95C85 allows several different addressing modes:

1) Auto-Increment Access
2) Stack Access
3) Indirect Random Access
4) Content-Addressable Access

The programmer will first issue a command that either directly, or by implication, places the Am95C85 CADM in a particular addressing mode. For example, the command AIM allows the host to read or write the currently addressed location, while subsequent reads and writes will be to the next byte (i.e., the CADM auto-increments the address pointer after each data access). Alternatively, STK sets the Stack-Access mode, which means that any subsequent data access physically moves all data below the current location for a read or write. A data read pops the byte at the current location, and moves all the data below up. A data write pushes a byte on the array at the address pointer moving all the data below down. The Stack-Access mode allows for immediate insertion or deletion of records (in previously sorted data), without the need for re-sorting.

The pointer into the memory array, the address pointer, is maintained by the Am95C85, although the programmer can load the address pointer through the use of the LAL (Load Address Long) and LAS (Load Address Short) commands.

The Find (FND) instruction implies a Content-Addressable Access mode. The description of the FND command is "set the address pointer to the key whose value is equal to the following bytes. If not present, point to the next higher value key." Following this instruction, the Am95C85 may be read to acquire the key plus pointer that was found. Since the FND instruction relies on the Am95C85 CADM data being in sorted order, the next section describes how a sort can be accomplished.

If more records matching a particular key value are to be located, additional FND commands without a key following the command can be issued. In this case, the value of the key contained in the input buffer space from the previous FND is used. The Address Pointer is incremented and the key comparisons are performed. This continues with each subsequent FND. To terminate this mode of operation, for instance to allow a new record to be sought, a command other than FND or RRB should be issued. The CADMs will then expect a subsequent FND command to be followed by a new key for which to search.

Host-Independent Sorting

Sorting may be accomplished on data which is in the form of a relational database index file. The programmer sets the length of the key and pointer fields by the KPL command and sets up K bytes of mask if the masking option is used. The data list may be loaded into the Am95C85 devices via DMA or slower programmed I/O. Two methods of sorting are possible:

1) Load data by DMA or I/O and then issue a Sort-Off-Line (SOF) command. This method loads all the data first and then performs the sort. The CPU can be performing other functions during the SOF execution. DMA completion must be detected by software before the SOF command is issued. $\overline{\text{DONE}}$ must be detected after the SOF command to signal that the sort has been completed.

2) Sort-On-Line (SON) command, followed by I/O or DMA load, allows each record to be placed in sorted order as it is loaded. If DMA is utilized, the CPU is free to perform other tasks during the entire operation. DMA complete, followed by $\overline{\text{DONE}}$, defines the end of the sort.

The $\overline{\text{DONE}}$ pin signals the acceptance of each byte of data and indicates the device is ready for the next byte. It also signals the completion of the active sort for the SOF command. In the case of SON, after the last byte of each record is received, $\overline{\text{DONE}}$ is asserted after the record is merged with existing records. After the last record is sent to the CADM array, the final $\overline{\text{DONE}}$ signal represents the end of Sort On Line.

The Hardware Designer's View

Reset

The CADM will go into the reset cycle after the hardware reset is asserted or a software-reset command is issued. Each device in an array will number itself and determine its chip address. The first device with RUP tied to HIGH assumes it has a chip address of 0, the next chip assumes an address of 1, and so on, until the last device with RDWN tied to HIGH numbers itself. Completion of reset is signaled by $\overline{\text{DONE}}$ going LOW. After reset, the address pointer is set to the first byte location in the last chip. The key length, K, is set to 1; the pointer length, P, is set to 0, and the last address is set to 1023. Masking is disabled. A hardware reset is required after power-up to bring the internal logic into a known state.

System Interface

All system interface signals are designed to be standard TTL compatible.

The system-control signals, $\overline{\text{RE}}$, $\overline{\text{WE}}$, $\overline{\text{CS}}$, and C/$\overline{\text{D}}$ are used to control the interface between the host and the CADM array. The command port access, with $\overline{\text{CS}}$ = "LOW" and C/$\overline{\text{D}}$ = "HIGH," is used to send commands to the device and is write-only. The data port access, with $\overline{\text{CS}}$ = "LOW" and C/$\overline{\text{D}}$ = "LOW," is used to transfer data between the host and the CADM array when reading and writing. These control signals should be connected to all the CADM devices in cascade.

The CADM data bus is used for host interface and chip-to-chip data transfer. Because of this, the CADM should not be directly connected to the host data bus. A transceiver is required to isolate the CADM data bus from the host data bus to avoid possible contention.

Two pins indicate the status of the Am95C85. $\overline{\text{DONE}}$ is used to indicate the completion of a command execution or data transfer. $\overline{\text{STAT}}$ going active indicates an exception condition following the execution of command or data transfer. The host should not drive the CADM data bus when $\overline{\text{DONE}}$ is inactive; otherwise, an unexpected outcome may occur. $\overline{\text{DONE}}$ may stay inactive forever if an invalid command sequence is issued. In this case, a reset is required to bring $\overline{\text{DONE}}$ back to LOW. If there is more than one CADM in cascade, the $\overline{\text{DONE}}$ pin from each CADM should be connected together and pulled up through a resistor to the power supply. Similarly, the $\overline{\text{STAT}}$ pin from each CADM should be connected together and also tied to the power supply through a pull-up resistor.

The CLK signal should reside between 1 MHz and its maximum rating.

Chip-To-Chip Communications

During the execution of some commands, it may be necessary to transfer data from one chip to another. These signals, TUP, RUP, TDWN, RDWN, and $\overline{\text{GLB}}$ are used to perform handshaking between the devices involved in the transfer. RDWN should be connected to the TUP of the next lower order chip, and TDWN should be connected to the RUP of the next lower order chip. The first device should have RUP pulled HIGH through a resistor to the power supply as should RDWN of the last device. Figure 3 shows the signal connections for cascading multiple devices.

Bank-To-Bank Control

As the number of CADMs used in the system increases, the capacitive load seen by each CADM device will increase. Depending on the system environment, up to 16 devices may be cascaded. If the effective load exceeds the specified test load, the designer will have two choices:

1) Reduce the clock frequency to the CADM array.

2) Insert a buffer circuit between banks of CADMs to increase driving capabilities.

If Option 1 is chosen, $\overline{\text{DIRD}}$, $\overline{\text{DIRG}}$ and $\overline{\text{T}}$/R are not used and can be left unconnected. If Option 2 is chosen, $\overline{\text{DIRD}}$, $\overline{\text{DIRG}}$ and $\overline{\text{T}}$/R signals are used to control the direction of buffering circuitry between banks. Even if the designer chooses Option 2, the clock frequency still has to be slowed down from its maximum rating because of buffer delay. The designer must decide which option is best suited to the system.

Command Summary

There are 16 commands to control the operation of the CADM. These commands are used to initialize the CADM, to control the internal pointers, to load the data, and perform sorting and searching. A command is loaded into the command register by writing an operation code into the command port. The command port is used to load the operation code only. For commands that require parameters following the command operation code, the parameters should be loaded through the data port. Commands requiring literal data are: LAS, FND, KPL, SMB, SON, LUD, and LAL. Table 1 summarizes the operation code, mnemonic, and functional description for each command.

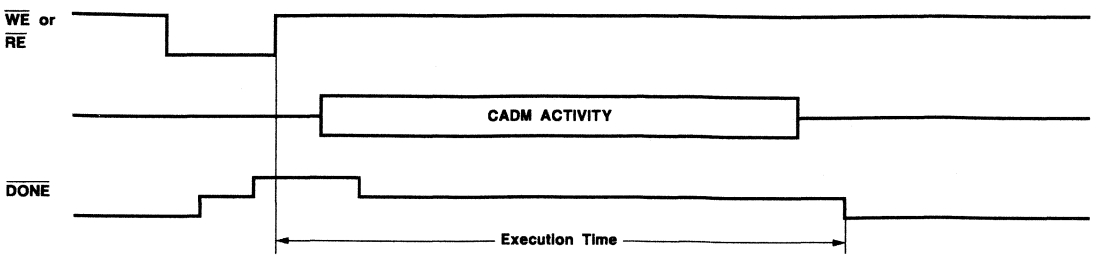
TABLE 1. COMMAND DESCRIPTIONS

OpCode	Mnemonic	Operands	Functional Description
00H	RST		Software reset command.
01H	LAS	Addr (LSB), Addr (MSB)	Load Address Short. Load the following two bytes of data at the address pointer of the currently active device.
02H	DEC		Decrement the address pointer by one.
03H	FND	Key (MSB), ... , Key (LSB) (Note 1)	Find the key specified following this command. Asserted $\overline{STAT}$ LOW if the key is not found.
04H	NXT		Set the address pointer to the first byte of the next record.
05H	RRB		Restore the address pointer to the first byte of the current record.
06H	AIM		Set Auto-Increment Mode. Address pointer is incremented by one after each data read/write.
07H	STK		Set Stack Mode. In Stack-Access mode, a read will pop data out of the data array at the address pointer and a write will push data into data array at the address pointer. The address pointer remains unchanged.
08H	KPL	K, P, LA (LSB), LA (MSB)	Load length of key and pointer fields and set the last address pointer.
09H	SMB	Mask (MSB), ... , Mask (LSB) (Note 1)	Set Mask Byte. The following K bytes of data will be used as mask during sorting and searching.
0AH	SON	Data (MSB), ... , Data (LSB) (Note 2)	Sort On Line. The CADM will insert the record into the data array in sorted order after the last byte of the record is loaded.
0BH	LUD	Data (MSB), ... , Data (LSB), ... (Note 3)	Load Unsorted Data. Data loaded following this command will be placed in the locations after existing meaningful data, if there is any.
0CH	SOF		Sort Off Line. Sort the existing data in the CADM in ascending order.
0DH	LAL	Addr (LSB), Addr (MSB), Chip Addr	Load Address Long. Load the following two bytes at the address pointer of the chip whose number is specified by the third byte.
0EH	PRE		Set the address pointer to the first byte of the previous record.
0FH	GSF		Get Status Full. Asserted $\overline{STAT}$ LOW if the CADM record space is full.

Notes: 1. Requires K Bytes following Opcode.
2. Requires integer multiples of (K + P) Bytes. Execution begins after each (K + P) Bytes are written.
3. Requires integer multiples of (K + P) Bytes.

Command Execution Time

The execution time of each command is expressed in clock cycles per byte of transfer. The execution times are measured from $\overline{WE}$ or $\overline{RE}$ to $\overline{DONE}$ as shown below.



WF024170

Command	Clock Cycles per Byte	Conditions
LUD	6	Command
	6	Per byte within a chip
	7	If crosses chip boundary
SMB	8	Command: first occurrence
	7	Command: all other occurrences
	6	Per mask byte for first k-1 bytes
	8	For last byte
SON	8	Command
	6	For first K + P – 1 bytes
	16 + S	For last byte (where S = binary search time; see FND performance equation)
AIM (Read/Write in AIM mode)	6	Command
	6	If on same chip
	9	If crosses chip boundary
KPL	9	Command; (+ 1 if only one chip in system)
	6	K: (+ 1 if user erroneously sets K = 0)
	7 + 2 * (9 – x)	P: where x = number of lower order zeros in K + P (binary)
	5	LA: (lsb)
	7	LA: (msb)
SOF	(See Sort-Off-Line performance equation)	
RST	8 + 4 * N	Where N = number of chips in system (Note: this applies to hardware and software RESETs).
LAL	4	Command
	5	lsb
	4	msb
	7	chip
LAS	4	Command
	5	lsb
	7	msb
DEC	8	If on same chip
	10	If crosses chip boundary
PRE	10	If on same chip
	13	If crosses chip boundary
RRB	7	Command
GSF	6	Command
FND	(See FIND performance equation)	
NXT	11	If on same chip
	14	If crosses chip boundary
STK	6	Command
–Push– (data write in Stack mode)	8	If on same chip. Add one clock cycle for every chip boundary crossing.
–Pop– (data read in Stack mode)	14	If on same chip. Add one clock cycle for every chip boundary crossing.

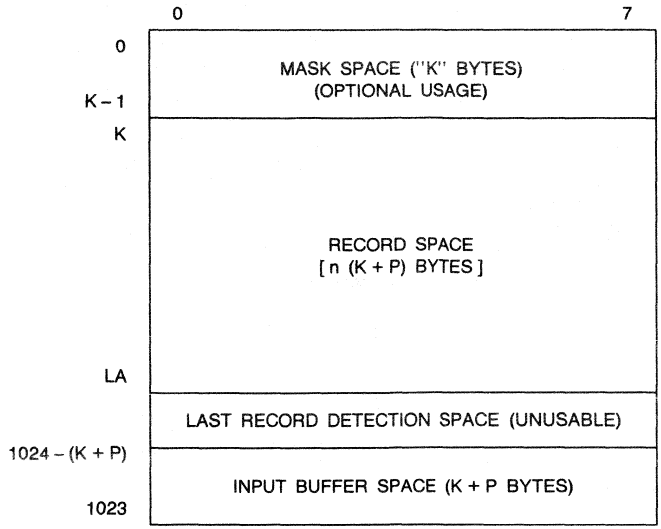


Figure 1. Am95C85 CADM Physical Model

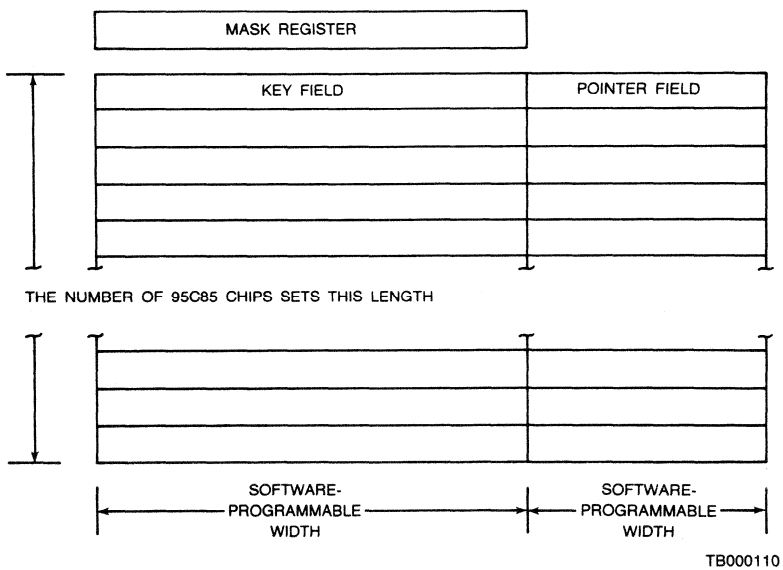


Figure 2. Am95C85 CADM Programmer's Model

Performance

"FND (Find)" Performance Equation

The "Find" performance equation assumes that the non-matching keys are different from the search key (in the most significant byte), and that the first match is found at the end of the longest possible binary search. The Find command's binary search is executed in parallel by all the CADM devices in an array. The device that finds the first occurrence terminates the operation by pulling $\overline{\text{DONE}}$ LOW. This explains why, for multi-chip arrays, the "Find" performance is independent of the total number of records. This equation includes the time required to load the search key.

"SOF (Sort-Off-Line)" Performance Equations

The Am95C85 CADM sorting performance is data-dependent. Best-case performance, quickest sort, is achieved from previously sorted data with no matching most significant bytes. The data which takes the longest time to sort is already sorted in reverse or descending order, and contains matching most significant bytes, where only the least significant bytes differ. The following two equations establish performance bounds for these two extremes.

"FIND" PERFORMANCE EQUATION

$$T_F = \frac{39 + 5K + (5.5 + 3K)\ (\lfloor \log_2 (n) \rfloor + 1)}{F}$$

"SORT OFF-LINE" PERFORMANCE EQUATIONS

Best-Case Performance:

$$T_{SB} = \frac{9 + N\left[\ 20 + 6\ (K + P) + 8.5\ (\lfloor \log_2\ (n + 1) \rfloor)\right]}{F}$$

Worst-Case Performance:

$$T_{SW} = \frac{9 + N\left[21 + \left(9 + \left\lceil \frac{N}{n} \right\rceil\right)(K + P) + (\lfloor \log_2 (n) \rfloor + 1)\ (5.5 + 3K)\right]}{F}$$

Where:

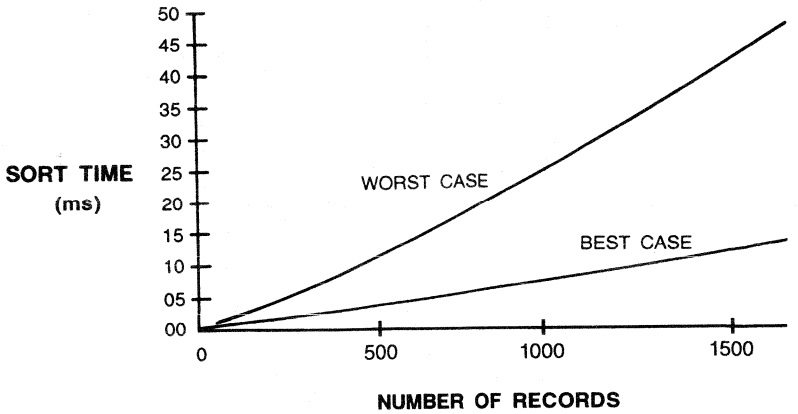
- N = Total no. of records
- n = No. of records in each chip
- K = No. of bytes/key
- P = No. of bytes/pointer
- F = Frequency of Am95C85 clock
- $\lfloor\ \rfloor$ = Truncate notation
- $\lceil\ \rceil$ = Round up notation
- T_{SB} = Time for sort (best case)
- T_{SW} = Time for sort (worst case)
- T_F = Time for find

$$n = \left\lfloor \frac{1024 - K - P - M}{K + P} \right\rfloor$$

M = K (if masking is used)
0 (if masking not used)

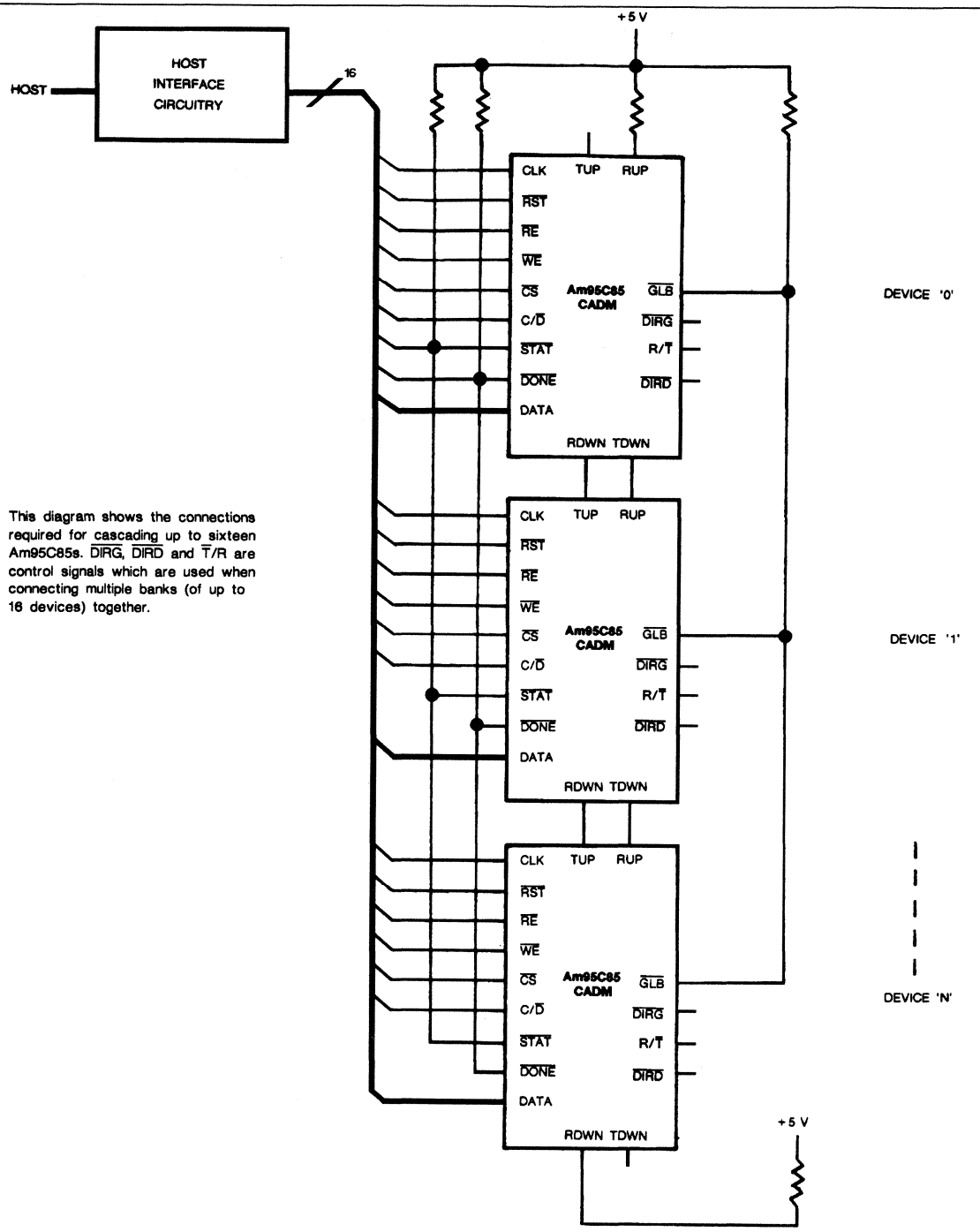
Am95C85 CADM SORT PERFORMANCE

(K = 8, P = 2)



OP001821

For This Case:
T_F = 6.4 μs
CLK = 16 MHz



This diagram shows the connections required for cascading up to sixteen Am95C85s. $\overline{DIRG}$, $\overline{DIRD}$ and $\overline{T}/R$ are control signals which are used when connecting multiple banks (of up to 16 devices) together.

Figure 3. Am95C85 Cascade Circuit

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin
with Respect to GND ..... −0.5 to +7

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.5 to +5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	16 MHz		Units
			Min.	Max.	
V_{OH}	Output HIGH Voltage	$I_{OH} = -16$ mA ($I_{OH} = -8$ mA for TUP, TDWN)	2.4	V_{CC}	V
V_{OL}	Output LOW Voltage	$I_{OL} = 16$ mA ($I_{OH} = 8$ mA for TUP, TDWN)	0	0.45	V
V_{IH}	Input HIGH Voltage		2.0	$V_{CC} + 0.5$	V
V_{IL}	Input LOW Voltage		−0.5	0.8	V
I_{IL}	Input Leakage Current	$0 V < V_{IN} < V_{CC}$		±10	μA
I_{LO}	Output Leakage Current	$0 V < V_{OUT} < V_{CC}$		±10	μA
I_{CC}	Maximum Average Power Supply Current	$V_{CC} = 5.5$ V, 16 MHz, Outputs loaded, worst-case data shifts during Push		200	mA
I_{CCS}	Maximum Average Power Supply Standby Current	$V_{CC} = 5.5$ V, 16 MHz, Outputs unloaded, No-ops		125	mA
V_{CC}	Power Supply Voltage		4.5	5.5	V

CAPACITANCE*

Parameter Symbol	Parameter Description	Test Conditions	16 MHz		Units
			Min.	Max.	
C_I	Input Capacitance	$f_C = 1$ MHz, Unmeasured Pins at GND		10	pF
C_O	Output Capacitance			13	
C_{IO}	I/O Capacitance			17	

*The capacitance values are guaranteed by design and are not tested.

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

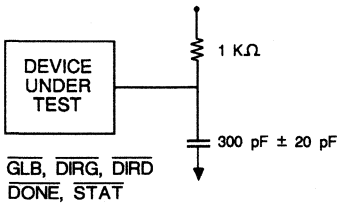
No.	Parameter Symbol	Parameter Description	16 MHz		12 MHz		Units
			Min.	Max.	Min.	Max.	
SYSTEM-TO-CADM TIMINGS							
1	t_{CC}	CLK Period	62	1000	82	1000	ns
2	t_{CH}	CLK HIGH Time	26		36		ns
3	t_{CL}	CLK LOW Time	26		36		ns
4	t_{DS}	Data Setup Before CLK LOW for Write	5		5		ns
5	t_{DH}	Data Hold After CLK LOW for Write	24		24		ns
6	t_{WS}	$\overline{WE}$ Setup Before CLK LOW	10		10		ns
7	t_{WW}	$\overline{WE}$ Pulse Width	86		106		ns
8	t_{WH}	$\overline{WE}$ Hold After CLK LOW	14		14		ns
9	t_{RS}	$\overline{RE}$ Setup Before CLK LOW	10		10		ns
10	t_{RR}	$\overline{RE}$ Pulse Width	86		106		ns
11	t_{RH}	$\overline{RE}$ Hold After CLK LOW	14		14		ns
12	t_{CSS}	$\overline{CS}$ Setup Before CLK LOW	10		10		ns
13	t_{CSW}	$\overline{CS}$ Pulse Width	86		106		ns
14	t_{CSH}	$\overline{CS}$ Hold After CLK LOW	14		14		ns
15	t_{CDS}	C/$\overline{D}$ Setup Before CLK LOW	10		10		ns
16	t_{CDW}	C/$\overline{D}$ Pulse Width	86		106		ns
17	t_{CDH}	C/$\overline{D}$ Hold After CLK LOW	14		14		ns
18	t_{SS}	$\overline{RST}$ Setup Before CLK LOW	10		10		ns
19	t_{SW}	$\overline{RST}$ Pulse Width	210		270		ns
20	t_{SH}	$\overline{RST}$ Hold After CLK LOW	14		14		ns
CADM-TO-SYSTEM TIMINGS							
21	t_{LDVR}	CLK LOW to Data Valid for Read		26		37	ns
22	t_{HDTR}	CLK HIGH to Data Three-State for Read		20		30	ns
23	t_{CHDH}	CLK HIGH to $\overline{DONE}$, $\overline{DIRD}$ HIGH		20		30	ns
24	t_{CLDT}	CLK LOW to $\overline{DONE}$, $\overline{DIRD}$ Three State		20		30	ns
25	t_{CHDL}	CLK HIGH to $\overline{DONE}$, $\overline{DIRD}$ LOW		20		30	ns
26	t_{CHSH}	CLK HIGH to $\overline{STAT}$ HIGH		20		30	ns
27	t_{CLST}	CLK LOW to $\overline{STAT}$ Three State		20		30	ns
28	t_{CHSL}	CLK HIGH to $\overline{STAT}$ LOW		20		30	ns
29	t_{CLRL}	CLK LOW to R/$\overline{T}$ LOW		26		37	ns
30	t_{CHRH}	CLK HIGH to R/$\overline{T}$ HIGH		20		30	ns

SWITCHING CHARACTERISTIC (Cont'd.)

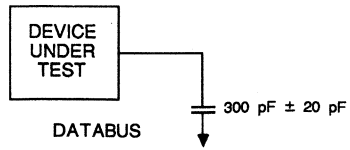
No.	Parameter Symbol	Parameter Description	16 MHz		12 MHz		Units
			Min.	Max.	Min.	Max.	
CADM-TO-CADM TIMINGS							
31	t_{CHRL}	CLK HIGH to R/$\overline{T}$ LOW for Interchip Data Move		20		30	ns
32	t_{HDVI}	CLK HIGH to Data Valid for Interchip Data Move	5	20	5	30	ns
33	t_{HDTI}	CLK HIGH to Data Three State for Interchip Data Move		20		30	ns
34	t_{CHGH}	CLK HIGH to $\overline{GLB}$ HIGH		20		30	ns
35	t_{CLGT}	CLK LOW to $\overline{GLB}$ Three State		20		30	ns
36	t_{CHGL}	CLK HIGH to $\overline{GLB}$, $\overline{DIRG}$ LOW		20		30	ns
37	t_{LDGH}	CLK LOW to $\overline{DIRG}$ HIGH		26		37	ns
38	t_{HDGT}	CLK HIGH to $\overline{DIRG}$ Three State		20		30	ns
39	t_{HTUH}	CLK HIGH to TUP HIGH		20		30	ns
40	t_{HTUL}	CLK HIGH to TUP LOW		20		30	ns
41	t_{LTUL}	CLK LOW to TUP LOW		15		20	ns
42	t_{RDSL}	RDWN Setup Before CLK LOW	6		6		ns
43	t_{RDHL}	RDWN Hold After CLK LOW	26		36		ns
44	t_{RUSL}	RUP Setup Before CLK LOW	6		6		ns
45	t_{RUHL}	RUP Hold After CLK LOW	26		36		ns
46	t_{HTDH}	CLK HIGH to TDWN HIGH		20		30	ns
47	t_{HTDL}	CLK HIGH to TDWN LOW		20		30	ns
48	t_{LTDH}	CLK LOW to TDWN HIGH		15		20	ns
49	t_{RUDT}	RUP HIGH to Data Three State	0	15	0	20	ns
50	t_{RURH}	RUP HIGH to R/$\overline{T}$ HIGH (for Pop)	0	15	0	25	ns
51	t_{RDDT}	RDWN LOW to Data Three State	0	15	0	20	ns
52	t_{RDRH}	RDWN LOW to R/$\overline{T}$ HIGH (for Push)	0	15	0	25	ns
53	t_{DHTD}	Data Hold After TDWN HIGH for Interchip Data Move (for Pop)	0		0		ns
54	t_{DHTU}	Data Hold After TUP LOW for Interchip Data Move (for Push)	0		0		ns
55	t_{DSL}	$\overline{DONE}$ Setup Before CLK LOW	6		6		ns
56	t_{DHL}	$\overline{DONE}$ Hold After CLK LOW	26		36		ns
57	t_{GSL}	$\overline{GLB}$ Setup Before CLK LOW	6		6		ns
58	t_{GHL}	$\overline{GLB}$ Hold After CLK LOW	26		36		ns

SWITCHING TEST CIRCUITS

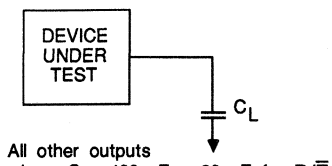
AC Loading



TC003981



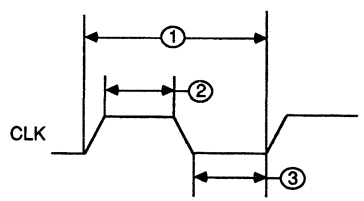
TC004001



All other outputs
where C_L = 100 pF ± 20 pF for R/$\overline{T}$
C_L = 20 pF for TUP, TDWN

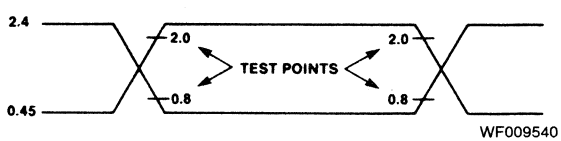
TC003991

SWITCHING TEST WAVEFORMS



WF023670

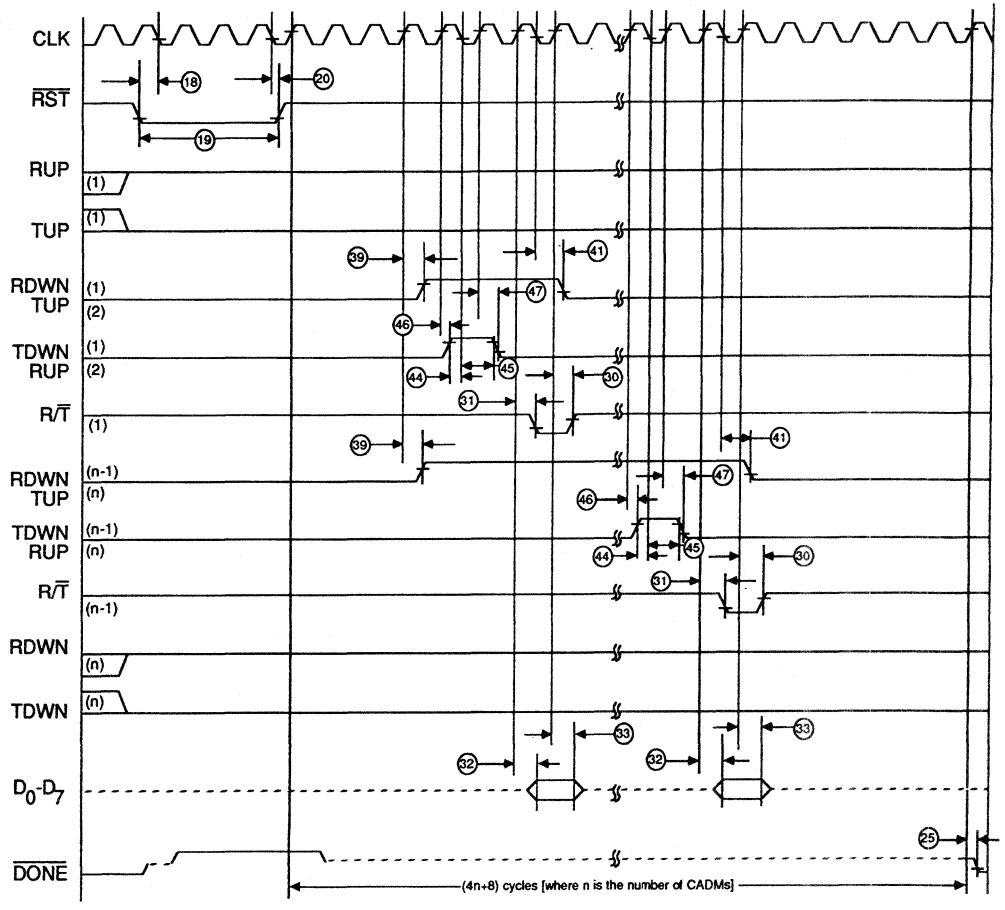
Clock



WF009540

Input/Output

SWITCHING WAVEFORMS

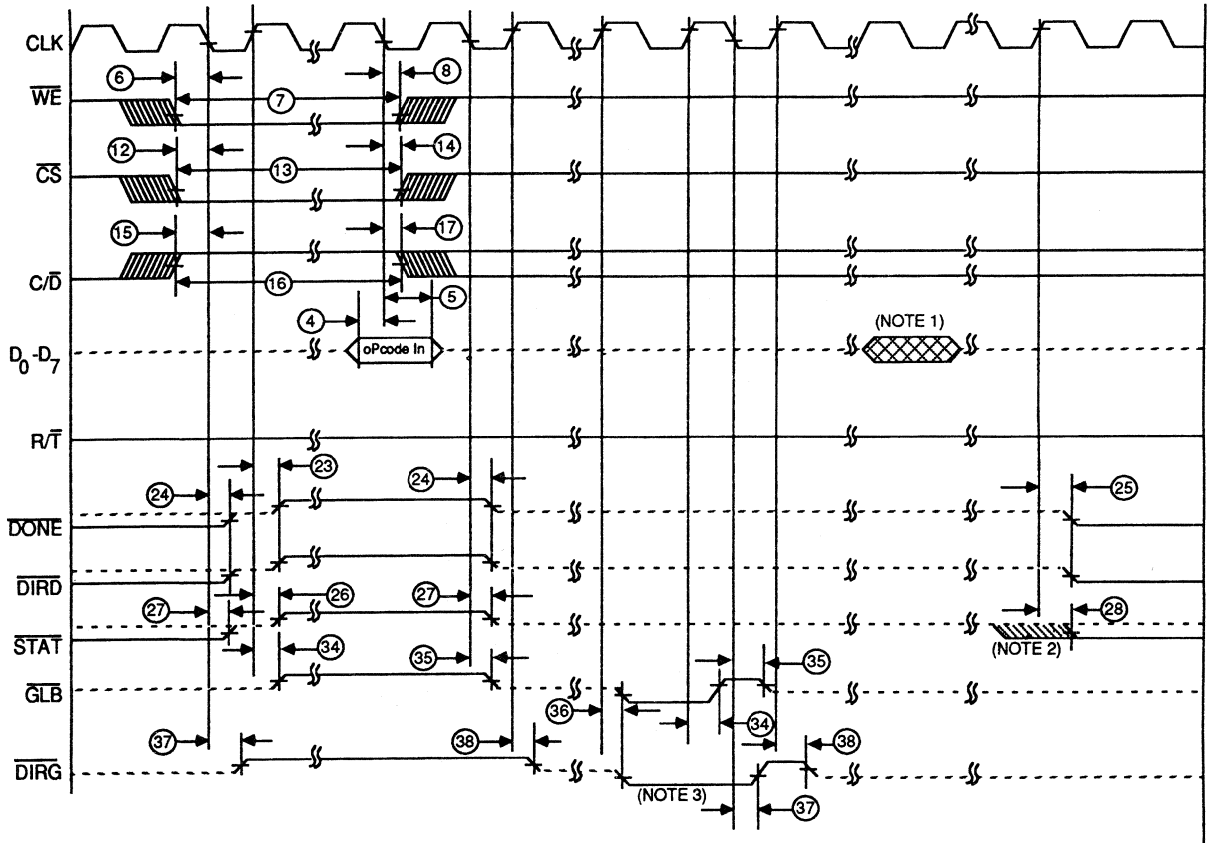


WF024040

Hardware Reset Timing

Note: TUP and TDWN are guaranteed to be LOW only when the CLK is LOW.

SWITCHING WAVEFORMS (Cont'd.)

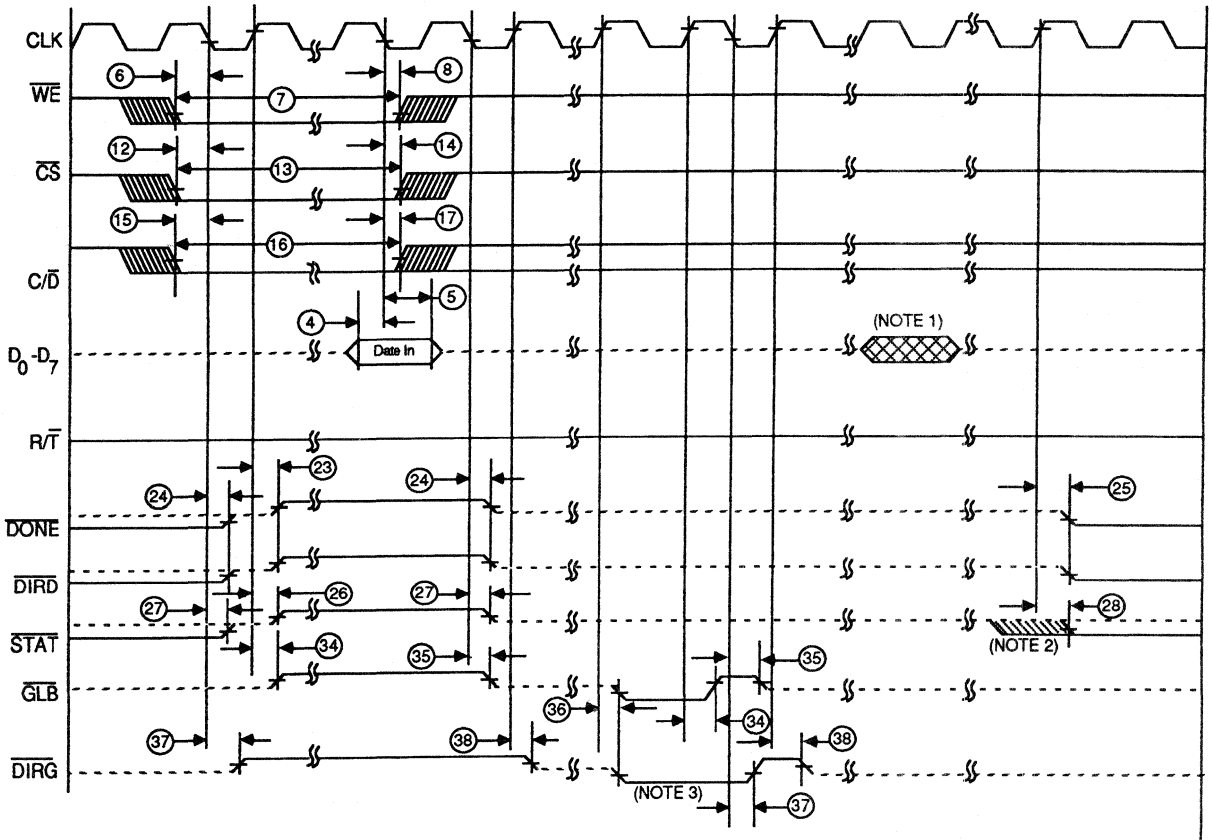


WF024050

Command Write Timing

Notes: 1. Any CADM may drive the data bus at any time during the command.
2. $\overline{STAT}$ is indeterminate one cycle before $\overline{DONE}$. It must be qualified with $\overline{DONE}$ being asserted.
3. $\overline{GLB}$ and $\overline{DIRG}$ may occur multiple times or not at all during the command.

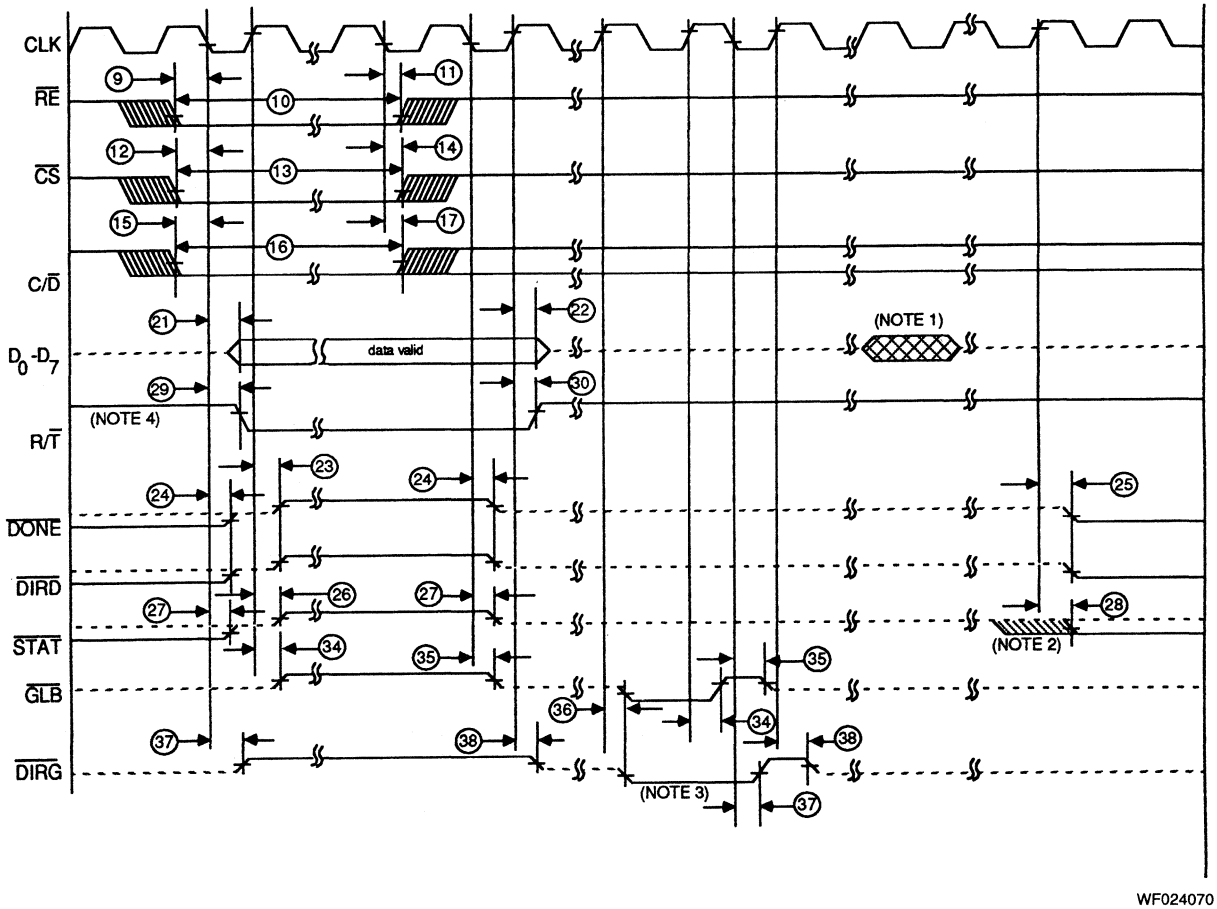
SWITCHING WAVEFORMS (Cont'd.)



Data Write Timing

Notes: 1. Any CADM may drive the data bus at any time during the write.
2. $\overline{STAT}$ is indeterminate one cycle before $\overline{DONE}$. It must be qualified with $\overline{DONE}$ being asserted.
3. $\overline{GLB}$ and $\overline{DIRG}$ may occur multiple times or not at all during the write.

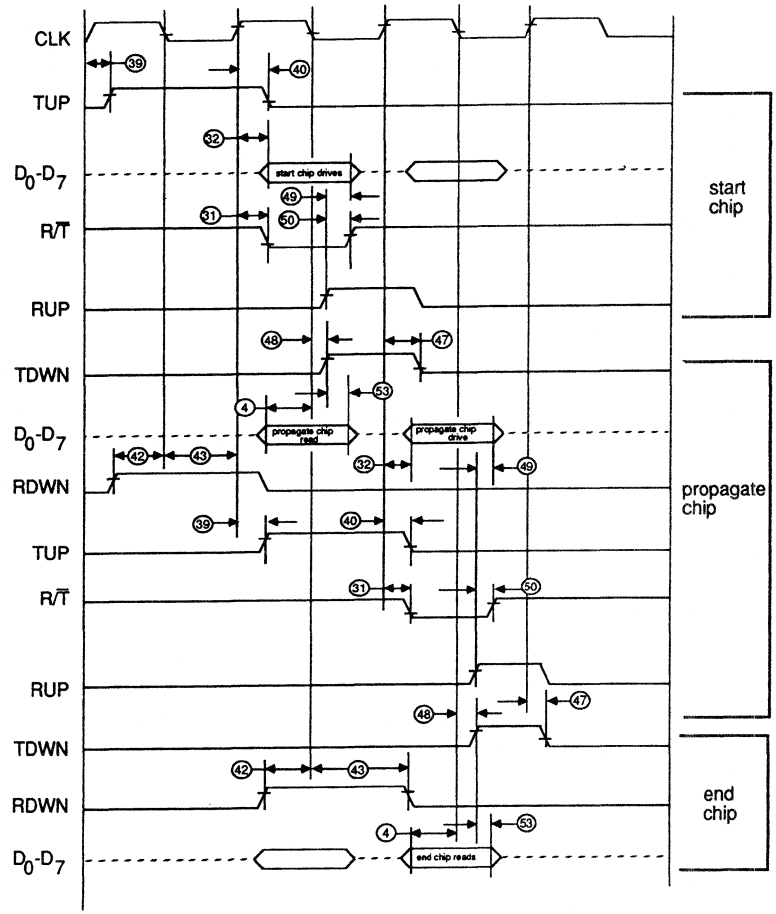
SWITCHING WAVEFORMS (Cont'd.)



Data Read Timing

Notes: 1. Any CADM may drive the data bus at any time during the read.
2. $\overline{STAT}$ is indeterminate one cycle before $\overline{DONE}$. It must be qualified with $\overline{DONE}$ being asserted.
3. $\overline{GLB}$ and $\overline{DIRG}$ may or may not occur during the read.
4. TUP and TDWN are not guaranteed to be LOW during the cycle when $\overline{T}$/R switches LOW during a read.

SWITCHING WAVEFORMS (Cont'd.)

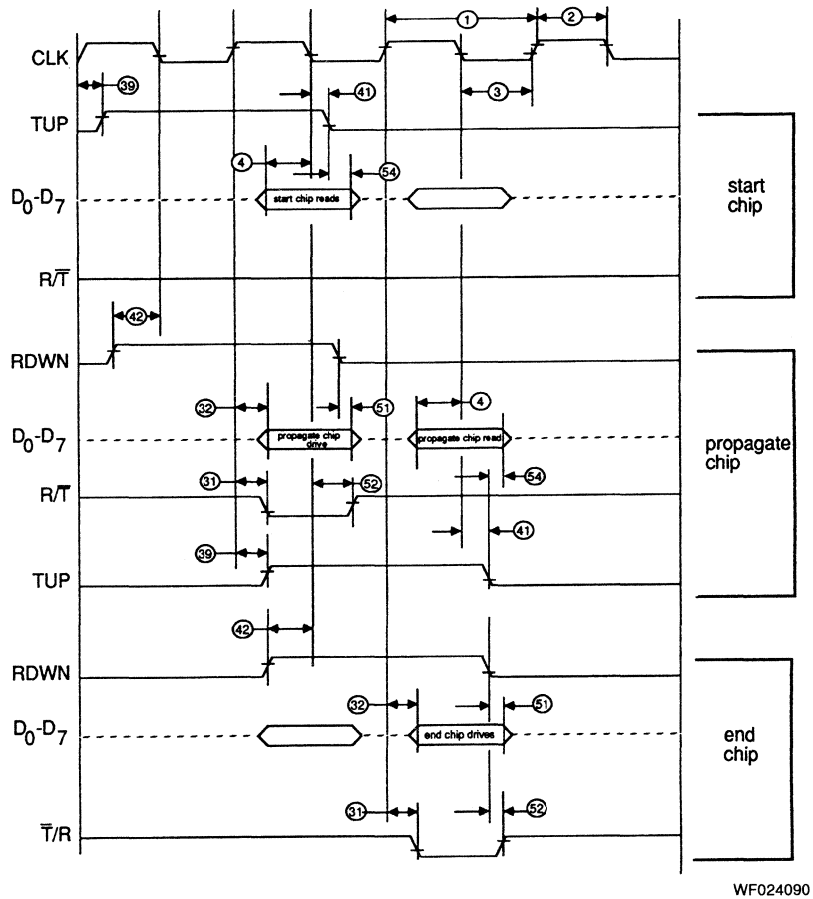


WF024080

Pop Timing

Note: TUP and TDWN are guaranteed to be LOW only when the CLK is LOW.

SWITCHING WAVEFORMS (Cont'd.)

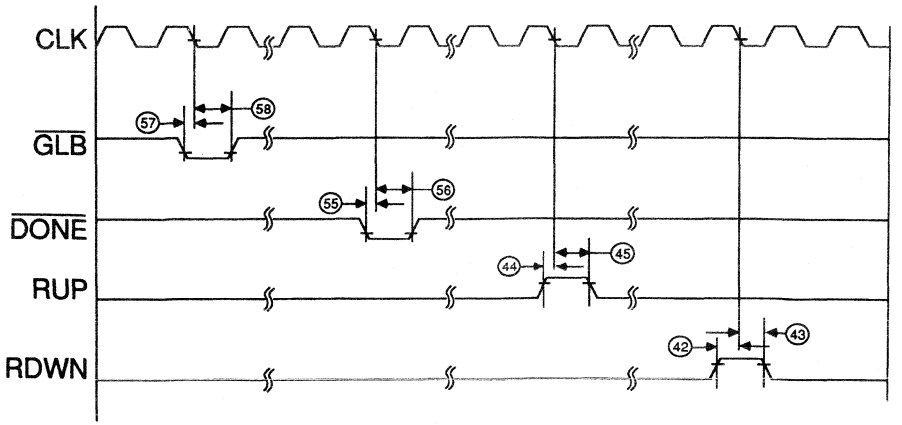


WF024090

Push Timing

Note: TUP and TDWN are guaranteed to be LOW only when the CLK is LOW.

SWITCHING WAVEFORMS (Cont'd.)

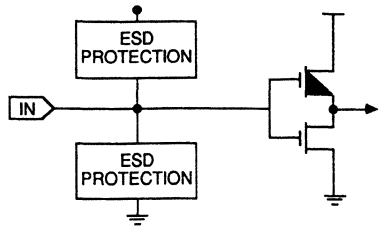


WF024100

Drive $\overline{\text{GLB}}$, $\overline{\text{DONE}}$, RUP, RDWN Timing

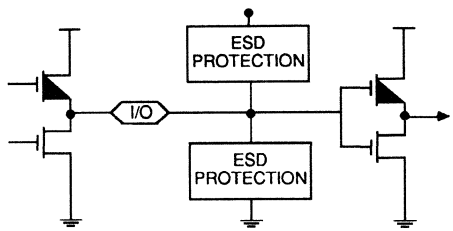
INPUT/OUTPUT CIRCUIT DIAGRAMS

Inputs: $\overline{RE}$, $\overline{WE}$, $\overline{CS}$, C/$\overline{D}$, $\overline{RST}$, RDWN, RUP



TC004110

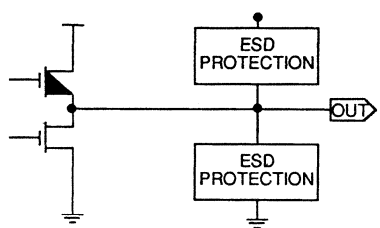
Bi-Directional: $\overline{DONE}$, D0-7, $\overline{GLB}$, TUP, TDWN*



TC004120

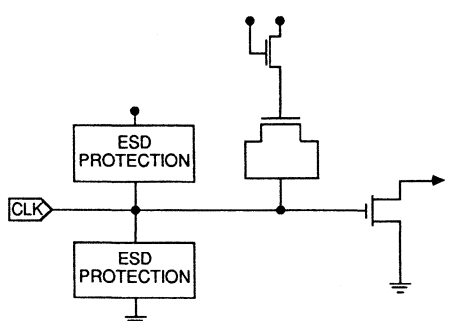
*TUP and TDWN are inputs for test mode only

Outputs: $\overline{STAT}$, $\overline{DIRG}$, $\overline{DIRD}$, R/$\overline{T}$



TC004130

Clocks: CLK



TC004140

iAPX86 FAMILY

AMD LITERATURE

To obtain literature in the U.S., write or call the AMD Literature Distribution Center, 901 Thompson Place, P.O. Box 3453 — M/S 82, Sunnyvale, CA 94088; (408) 732-2400, TOLL FREE (800) 538-8450. To obtain literature from international locations, contact the nearest AMD sales office or distributor (see listings in the back of this publication).

80186

High Integration 16-Bit Microprocessor
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

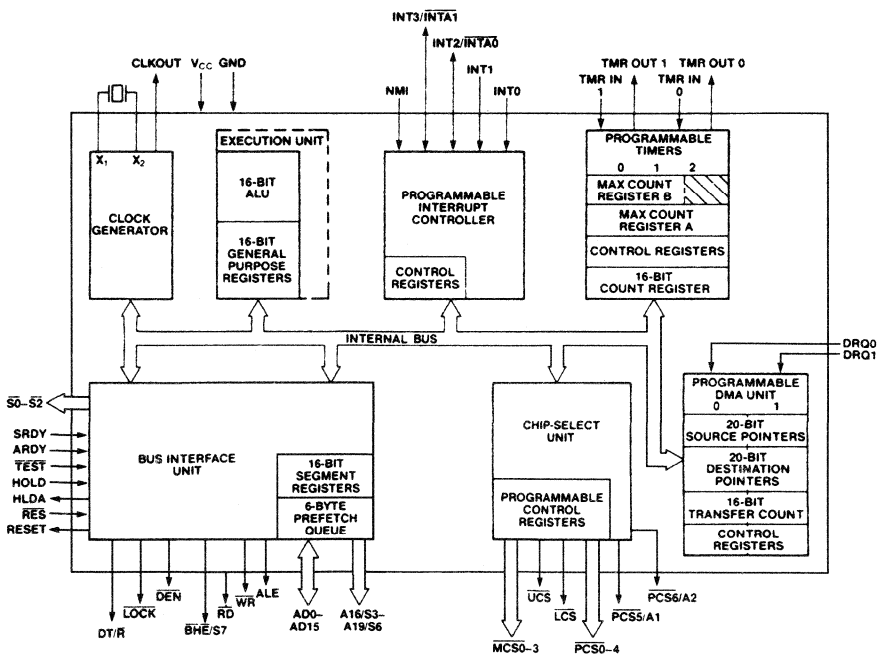
- Integrated feature set
  - Enhanced 10MHz 8086-1 CPU
  - Clock generator
  - Two independent, high-speed DMA channels
  - Programmable interrupt controller
  - Three programmable 16-bit timers
  - Programmable memory and peripheral chip-select logic
  - Programmable wait state generator
  - Local bus controller
- Available in 10MHz (80186-10), 8MHz (80186)
- High performance processor
  - Two times the performance of the standard 8086
  - 4M byte/sec bus bandwidth interface
- Direct addressing capability to 1M byte of memory
- Completely object code compatible with all existing iAPX 86, 88 software
  - Ten new instruction types
  - Compatible with 29843/45, 29833/63, 8284, and 8288 bus support components
- Optional numeric processor extension
- Available in 68-pin Plastic Leaded Chip Carrier (PLCC), Ceramic Leadless Chip Carrier (LCC), and Pin Grid Array (PGA) packages.

GENERAL DESCRIPTION

The 80186 is a highly integrated 16-bit microprocessor. It effectively combines 15-20 of the most common iAPX 86 system components onto one. The 80186 provides two times greater throughput than the standard 5MHz 8086. The 80186 is upward compatible with 8086 and 8088 software and adds 10 new instruction types to the existing set.

The 80186 comes in a 68-pin package and requires a single ±5V power supply.

BLOCK DIAGRAM



BD003560

Figure 1.

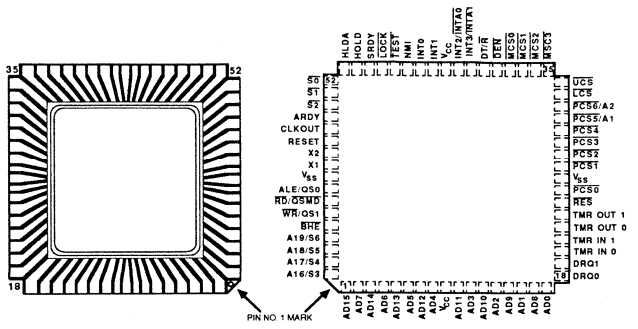
Publication #	Rev.	Amendment
03551	D	/0

Issue Date: **April 1987**

CONNECTION DIAGRAMS

Ceramic Leadless Chip Carrier (LCC*)

Top **Bottom**

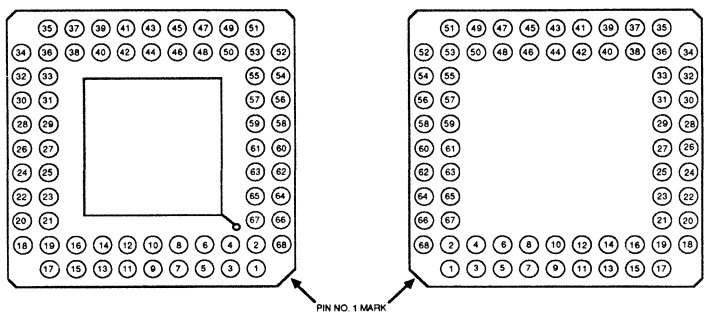


CD010500

*LCC package placed in socket top down.

Pin Grid Array (PGA)

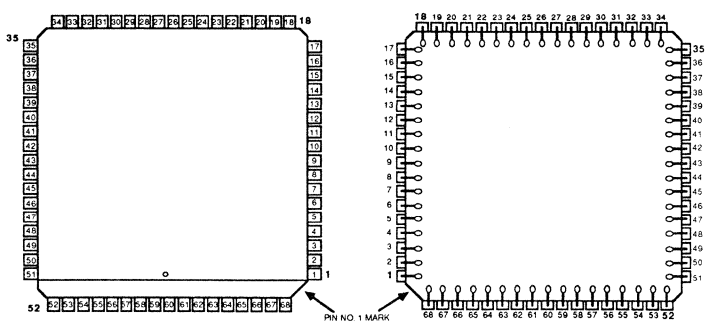
Pins Facing Up **Pins Facing Down**



CD010521

Plastic Leaded Chip Carrier (PLCC**)

Top **Bottom**



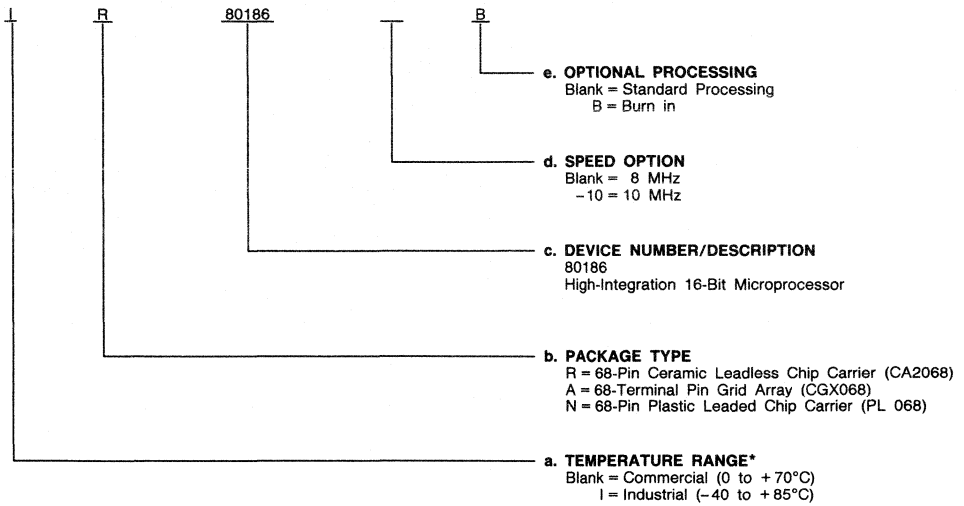
CD010530

**PLCC package placed in socket top up.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
A, R, N	80186
	80186-10
A, R, IA, IR	80186B

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
9, 43	V_{CC}, V_{CC}	I	System Power: +5 volt power supply.
26, 60	V_{SS}, V_{SS}	I	System Ground.
57	RESET	O	Reset Output indicates that the 80186 CPU is being reset; and can be used as a system reset. It is active HIGH, synchronized with the processor clock, and lasts an integer number of clock periods corresponding to the length of the $\overline{RES}$ signal.
59, 58	X1, X2	I	Crystal inputs, X1 and X2, provide an external connection for a fundamental mode parallel resonant crystal for the internal crystal oscillator. X1 can interface to an external clock instead of a crystal. The input or oscillator frequency is internally divided by two to generate the clock signal (CLKOUT).
56	CLKOUT	O	Clock Output provides the system with a 50% duty cycle waveform. All device pin timings are specified relative to CLKOUT. CLKOUT has sufficient MOS drive capabilities for a numeric processor extension.
24	$\overline{RES}$	I	System Reset causes the 80186 to immediately terminate its present activity, clear the internal logic, and enter a dormant state. This signal may be asynchronous to the 80186 clock. The 80186 begins fetching instructions approximately 7 clock cycles after $\overline{RES}$ is returned HIGH. $\overline{RES}$ is required to be LOW for greater than 4 clock cycles and is internally synchronized. For proper initialization, the LOW-to-HIGH transition of $\overline{RES}$ must occur no sooner than 50 microseconds after power up. This input is provided with a Schmitt-trigger to facilitate power-on $\overline{RES}$ generation via an RC network. When $\overline{RES}$ occurs, the 80186 will drive the status lines to an inactive level for one clock, and then tri-state them.
47	$\overline{TEST}$	I	$\overline{TEST}$ is examined by the WAIT instruction. If the $\overline{TEST}$ input is HIGH when "WAIT" execution begins, instruction execution will suspend. $\overline{TEST}$ will be resampled until it goes LOW, at which time execution will resume. If interrupts are enabled while the 80186 is waiting for $\overline{TEST}$, interrupts will be serviced. This input is synchronized internally.
20 21	TMR in 0, TMR IN1	I I	Timer inputs are used either as clock or control signals, depending upon the programmed timer mode. These inputs are active HIGH (or LOW-to-HIGH transitions are counted) and internally synchronized.
22 23	TMR OUT 0, TMR OUT 1	O O	Timer outputs are used to provide single pulse or continuous waveform generation, depending upon the timer mode selected.
18 19	DRQ0 DRQ1	I I	DMA Request is driven HIGH by an external device when it desires that a DMA channel (Channel 0 or 1) perform a transfer. These signals are active HIGH, level-triggered, and internally synchronized.
46	NMI	I	Non-Maskable Interrupt is an edge-triggered input which causes a type 2 interrupt. NMI is not maskable internally. A transition from a LOW to HIGH initiates the interrupt at the next instruction boundary. NMI is latched internally. An NMI duration of one clock or more will guarantee service. This input is internally synchronized.
45, 44 42 41	INT0, INT1 INT2/$\overline{INTA0}$ INT3/$\overline{INTA1}$	I I/O I/O	Maskable Interrupt Requests can be requested by strobing one of these pins. When configured as inputs, these pins are active HIGH. Interrupt Requests are synchronized internally. INT2 and INT3 may be configured via software to provide active-LOW interrupt-acknowledge output signals. All interrupt inputs may be configured via software to be either edge- or level-triggereed. To ensure recognition, all interrupt requests must remain active until the interrupt is acknowledged. When iRMX mode is selected, the function of these pins changes (see Interrupt Controller section of this data sheet).
65-68	A19/S6, A18/S5, A17/S4, A16/S3	O O O O	Address Bus Outputs (16-19) and Bus Cycle Status (3-6) reflect the four most significant address bits during T_1. These signals are active HIGH. During T_2, T_3, T_W, and T_4, status information is available on these lines as encoded below: S6: Low = Processor Cycle; High = DMA Cycle S3, S4, and S5 are defined as LOW during T_2-T_4.
10-17 1-8	AD15-AD0	I/O	Address/Data Bus (0-15) signals constitute the time multiplexed memory or I/O address (T_1) and data (T_2, T_3, T_W, and T_4) bus. The bus is active HIGH. A_0 is analogous to $\overline{BHE}$ for the lower byte of the data bus, pins D_7 through D_0. It is LOW during T_1 when a byte is to be transferred onto the lower portion of the bus in memory or I/O operations.
64	$\overline{BHE}$/S7	O	During T_1 the Bus High Enable signal should be used to determine if data is to be enabled onto the most significant half of the data bus, pins D_{15}-D_8. $\overline{BHE}$ is LOW during T_1 for read, write, an interrupt acknowledge cycles when a byte is to be transferred on the higher half of the bus. The S_7 status information is available during T_2, T_3, and T_4. S_7 is logically equivalent to $\overline{BHE}$. The signal is active LOW, and is tristated OFF during bus HOLD. **$\overline{BHE}$ and A0 Encodings** $\overline{BHE}$ Value / A0 Value / Function: 0 / 0 / Work Transfer 0 / 1 / Byte Transfer on upper half of data bus (D15-D8) 1 / 0 / Byte Transfer on lower half of data bus (D_7-D_0) 1 / 1 / Reserved
61	ALE/QS0	O	Address Latch Enable/Queue Status 0 is provided by the 80186 to latch the address into the 8282/8283 address latches. ALE is active HIGH. Addresses are guaranteed to be valid on the trailing edge of ALE. The ALE rising edge is generated off the rising edge of the CLKOUT immediately preceding T_1 of the associated bus cycle, effectively one-half clock cycle earlier than in the standard 8086. The trailing edge is generated off the CLKOUT rising edge in T_1 as in the 8086. Note that ALE is never floated.

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
63	$\overline{WR}$/QS1	O	Write Strobe/Queue Status 1 indicates that the data on the bus is to be written into a memory or an I/O device. $\overline{WR}$ is active for T_2, T_3, and T_W of any write cycle. It is active LOW, and floats during "HOLD." It is driven HIGH for one clock during Reset, and then floated. When the 80186 is in queue status mode, the ALE/QS0 and $\overline{WR}$/QS1 pins provide information about processor/instruction queue interaction.
62	$\overline{RD}$/$\overline{QSMD}$	O	Read Strobe indicates that the 80186 is performing a memory or I/O read cycle. $\overline{RD}$ is active LOW for T_2, T_3, and T_W of any read cycle. It is guaranteed not to go LOW in T_2 until after the Address Bus is floated. $\overline{RD}$ is active LOW, and floats during "HOLD." $\overline{RD}$ is driven HIGH for one clock during Reset, and then the output driver is floated. A weak internal pull-up mechanism on the $\overline{RD}$ line holds it HIGH when the line is not driven. During RESET the pin is sampled to determine whether the 80186 should provide ALE, $\overline{WR}$, and $\overline{RD}$, or if the Queue-Status should be provided. $\overline{RD}$ should be connected to GND to provide Queue-Status data.
55	ARDY	I	Asynchronous Ready informs the 80186 that the addressed memory space or I/O device will complete a data transfer. The ARDY input pin will accept an asynchronous input, and is active HIGH. Only the rising edge is internally synchronized by the 80186. This means that the falling edge of ARDY must be synchronized to the 80186 clock. If connected to V_{CC}, no WAIT states are inserted. Asynchronous ready (ARDY) or synchronous ready (SRDY) must be active to terminate a bus cycle. If line is unused, it may remain connected to V_{CC} or it may be connected to V_{SS} (in which case the programmer must initialize the part to inhibit the external pins).
49	SRDY	I	Synchronous Ready must be synchronized externally to the 80186. The use of SRDY provides a relaxed system-timing specification on the Ready input. This is accomplished by eliminating the one-half clock cycle which is required for internally resolving the signal level when using the ARDY input. This line is active HIGH. If this line is connected to V_{CC} no WAIT states are inserted. Asynchronous ready (ARDY) or synchronous ready (SRDY) must be active before a bus cycle is terminated. If line is unused, it may remain connected to V_{CC} or it may be connected to V_{SS} (in which case the programmer must initialize the part to inhibit the external pins).
48	$\overline{LOCK}$	O	$\overline{LOCK}$ output indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active LOW. The $\overline{LOCK}$ signal is requested by the LOCK prefix instruction and is activated at the beginning of the first data cycle associated with the instruction following the LOCK prefix. It remains active until the completion of the instruction following the LOCK prefix. No prefetches will occur while $\overline{LOCK}$ is asserted. When executing more than one LOCK instruction, always make sure there are six bytes of code between the end of the first LOCK instruction and the start of the second LOCK instruction. $\overline{LOCK}$ is active LOW, is driven HIGH for one clock during RESET, and then floated.
52-54	$\overline{S0}$, $\overline{S1}$, $\overline{S2}$	O	Bus cycle status $\overline{S0}$-$\overline{S2}$ are encoded to provide bus-transaction information:
50 51	HOLD (input) HLDA (output)	I O	HOLD indicates that another bus master is requesting the local bus. The HOLD input is active HIGH. HOLD may be asynchronous with respect to the 80186 clock. The 80186 will issue a HLDA in response to a HOLD request at the end of T_4 or T_1 Simultaneous with the issuance of HLDA, the 80186 will float the local bus and control lines. After HOLD is detected as being LOW, the 80186 will lower HLDA. When the 80186 needs to run another bus cycle, it will again drive the local bus and control lines.
34	$\overline{UCS}$	O	Upper Memory Chip Select is an active LOW output whenever a memory reference is made to the defined upper portion (1K-=256K block) of memory. This line is not floated during bus HOLD. The address range activating $\overline{UCS}$ is software programmable.
33	$\overline{LCS}$	O	Lower Memory Chip Select is active LOW whenever a memory reference is made to the defined lower portion (1K-256K) of memory. This line is not floated during bus HOLD. The address range activating $\overline{LCS}$ is software programmable.
38, 37, 36, 35	$\overline{MCS0}$-3	O	Mid-Range Memory Chip Select signals are active LOW when a memory reference is made to the defined mid-range portion of memory (8K-512K). These lines are not floated during bus HOLD. The address ranges activating $\overline{MCS0}$-3 are software programmable.
25 27-30	$\overline{PCS0}$ $\overline{PCS1}$-4	O O	Peripheral Chip Select signals 0-4 are active LOW when a reference is made to the defined peripheral area (65K byte I/O space). These lines are not floated during bus HOLD. The address ranges activating $\overline{PCS0}$-4 are software programmable.

QS1	QS0	Queue Operation
0	0	No queue operation
0	1	First opcode byte fetched from the queue
1	1	Subsequent byte fetched from the queue
1	0	Empty the queue

80186 Bus Cycle Status Information

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Bus Cycle Initiated
0	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

The status pins float during "HOLD."
$\overline{S2}$ may be used as a logical M/$\overline{IO}$ indicator, and $\overline{S1}$ as a DT/$\overline{R}$ indicator.
The status lines are driven HIGH for one clock during Reset, and then floated until a bus cycle begins.

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
31	$\overline{PCS5}$/A1	O	Peripheral Chip Select 5 or Latched A1 may be programmed to provide a sixth peripheral chip select, or to provide an internally latched A1 signal. The address range activating $\overline{PCS5}$ is software programmable. When programmed to provide latched A1, rather than $\overline{PCS5}$, this pin will retain the previously latched value of A1 during a bus HOLD. A1 is active HIGH.
32	$\overline{PCS6}$/A2	O	Peripheral Chip Select 6 or Latched A2 may be programmed to provide a seventh peripheral chip select, or to provide an internally latched A2 signal. The address range activating $\overline{PCS6}$ is software programmable. When programmed to provide latched A2, rather than $\overline{PCS6}$, this pin will retain the previously latched value of A2 during a bus HOLD. A2 is active HIGH.
40	DT/$\overline{R}$	O	Data Transmit/Receive controls the direction of data flow through the external 29833/29863 data bus transceiver. When LOW, data is transferred to the 80186. When HIGH the 80186 places write data on the data bus.
39	$\overline{DEN}$	O	Data Enable is provided as a 29833/29863 data bus transceiver output enable. $\overline{DEN}$ is active LOW during each memory and I/O access. $\overline{DEN}$ is HIGH whenever DT/$\overline{R}$ changes state.

DETAILED DESCRIPTION

Introduction

The following Functional Description describes the base architecture of the 80186. The architecture is common to the 8086, 8088, and 80286 microprocessor families as well. The 80186 is a very high integration 16-bit microprocessor. It combines 15-20 of the most common microprocessor system components onto one chip while providing twice the performance of the standard 8086. The 80186 is object code compatible with the 8086, 8088 microprocessors and adds 10 new instruction types to the existing 8086, 8088 instruction set.

80186 BASE ARCHITECTURE

The 8086, 8088, 80186, and 80286 family all contain the same basic set of registers, instructions, and addressing modes. The 80186 processor is upward compatible with the 8086, 8088, and 80286 CPUs.

Register Set

The 80186 base architecture has fourteen registers as shown in Figures 3a and 3b. These registers are grouped into the following categories.

General Registers

Eight 16-bit general purpose registers used to contain arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used as 16-bit registers or split into pairs of separate 8-bit registers.

Segment Registers

Four 16-bit special purpose registers select, at any given time, the segments of memory that are immediately addressable for code, stack, and data. (For usage, refer to Memory Organization.)

Base and Index Registers

Four of the general purpose registers may also be used to determine offset addresses of operands in memory. These registers may contain base addresses or indexes to particular locations within a segment. The addressing mode selects the specific registers for operand and address calculations.

Status and Control Registers

Two 16-bit special purpose registers record or alter certain aspects of the 80186 processor state. These are the Instruction Pointer Register, which contains the offset address of the next sequential instruction to be executed, and the Status Word Register, which contains status and control flag bits (see Figures 3a and 3b).

Status Word Description

The Status Word records specific characteristics of the result of logical and arithmetic instructions (bits 0, 2, 4, 6, 7, and 11) and controls the operation of the 80186 within a given operating mode (bits 8, 9, and 10). The Status Word Register is 16-bits wide. The function of the Status Word bits is shown in Table 2.

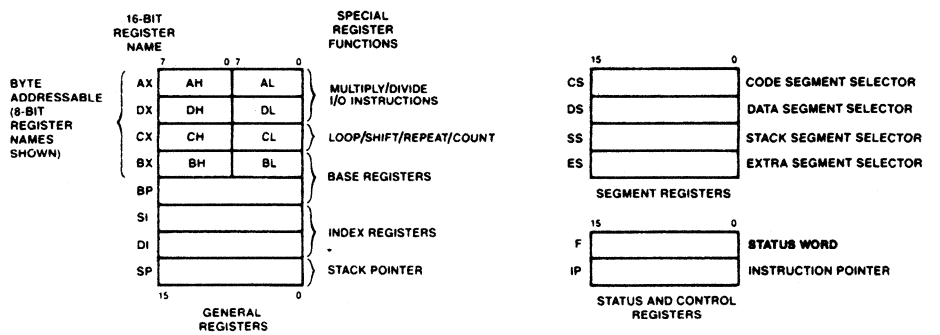


Figure 3a. 80186 General Purpose Register Set

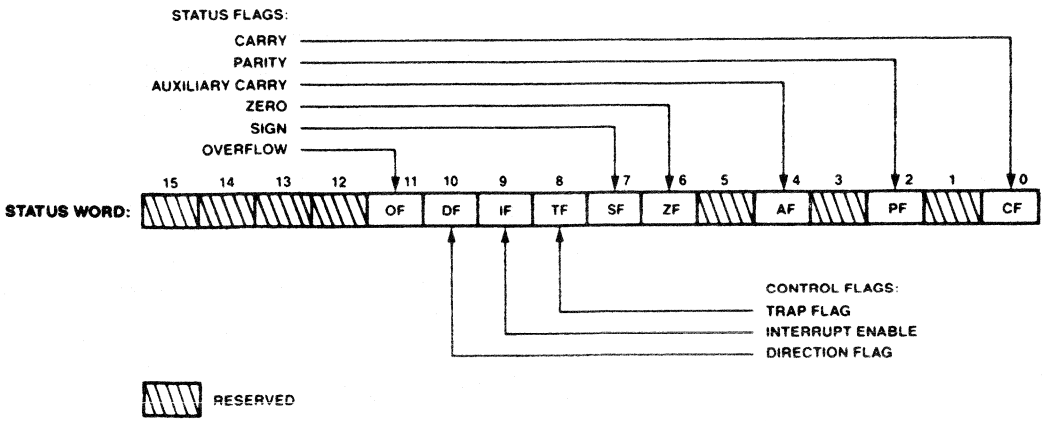


Figure 3b. Status Word Format

Table 2. Status Word Bit Function

Bit Position	Name	Function
0	CF	Carry Flag — Set on high-order bit carry or borrow; cleared otherwise.
2	PF	Parity Flag — Set if low-order 8 bits or result contain an even number of 1-bits; cleared otherwise.
4	AF	Set on carry from or borrow to the low order four bits of AL; cleared otherwise.
6	ZF	Zero Flag — Set if result is zero; cleared otherwise.
7	SF	Sign Flag — Set equal to high-order bit of result (0 if positive, 1 if negative).
8	TF	Single Step Flag — Once set, a single step interrupt occurs after the next instruction executes. TF is cleared by the single step interrupt.
9	IF	Interrupt-enable Flag — When set, maskable interrupts will cause the CPU to transfer control to an interrupt vector specified location.
10	DF	Direction Flag — Causes string instructions to auto decrement the appropriate index register when set. Clearing DF causes auto increment.
11	OF	Overflow Flag — Set if the signed result cannot be expressed within the number of bits in the destination operand; cleared otherwise.

Instruction Set

The instruction set is divided into seven categories: data transfer, arithmetic, shift/rotate/logical, string manipulation, control transfer, high-level instructions, and processor control. These categories are summarized in Figure 4.

An 80186 instruction can reference anywhere from zero to several operands. An operand can reside in a register, in the instruction itself, or in memory. Specific operand addressing modes are discussed later in this data sheet.

Memory Organization

Memory is organized in sets of segments. Each segment is a linear contiguous sequence of up to 64K (2^{16}) 8-bit bytes. Memory is addressed using a two-component address (a pointer) that consists of a 16-bit base segment and a 16-bit offset. The 16-bit base values are contained i one of four internal segment registers (code, data, stack, extra). The physical address is calculated by shifting the base value LEFT by four bits and adding the 16-bit offset value to yield a 20-bit physical address (see Figure 5). This allows for a 1 MByte physical address size.

All instructions that address operands in memory must specifiy the base segment and the 16-bit offset value. For speed and compact instruction encoding, the segment register used for physical address generation is implied by the addressing mode used (see Table 3). These rules follow the way programs are written (see Figure 6) as independent modules that require areas for code and data, a stack, and access to external data areas.

Special segment override instruction prefixes allow the implicit segment register selection rules to be overridden for special cases. The stack, data, and extra segments may coincide for simple programs.

GENERAL PURPOSE	
MOV	Move byte or word
PUSH	Push word onto stack
POP	Pop word off stack
PUSHA	Push all registers on stack
POPA	Pop all registers from stack
XCHG	Exchange byte or word
XLAT	Translate byte
INPUT/OUTPUT	
IN	Input byte or word
OUT	Output byte or word
ADDRESS OBJECT	
LEA	Load effective address
LDS	Load pointer using DS
LES	Load pointer using ES
FLAG TRANSFER	
LAHF	Load AH register from flags
SAHF	Store AH register in flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack

ADDITION	
ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
SUBTRACTION	
SUB	Subtract byte or word
SBB	Subtract byte or word with borrow
DEC	Decrement byte or word by 1
NEG	Negate byte or word
CMP	Compare byte or word
AAS	ASCII adjust for subtraction
DAS	Decimal adjust for subtraction
MULTIPLICATION	
MUL	Multiply byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII adjust for multiply
DIVISION	
DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
CBW	Convert byte to word
CWD	Convert word to doubleword
MOVS	Move byte or word string

INS	Input bytes or word string
OUTS	Output bytes or word string
CMPS	Compare byte or word string
SCAS	Scan byte or word string
LODS	Load byte or word string
STOS	Store byte or word string
REP	Repeat
REPE/ REPZ	Repeat while equal/zero
REPNE/ REPNZ	Repeat while not equal/not zero

LOGICALS	
NOT	"Not" byte or word
AND	"And" byte or word
OR	"Inclusive or" byte or word
XOR	"Exclusive or" byte or word
TEST	"Test" byte or word
SHIFTS	
SHL/SAL	Shift logical/arithmetic left byte or word
SHR	Shift logical right byte or word
SAR	Shift arithmetic right byte or word
ROTATES	
ROL	Rotate left byte or word
ROR	Rotate right byte or word
RCL	Rotate through carry left byte or word
RCR	Rotate through carry right byte or word

FLAG OPERATIONS	
STC	Set carry flag
CLC	Clear carry flag
CMC	Complement carry flag
STD	Set direction flag
CLD	Clear direction flag
STI	Set interrupt enable flag
CLI	Clear interrupt enable flag
EXTERNAL SYNCHRONIZATION	
HLT	Halt until interrupt or reset
WAIT	Wait for $\overline{\text{TEST}}$ pin active
ESC	Escape to extension processor
LOCK	Lock bus during next instruction
NO OPERATION	
NOP	No operation
HIGH LEVEL INSTRUCTIONS	
ENTER	Format stack for procedure entry
LEAVE	Restore stack for procedure exit
BOUND	Detects values outside prescribed range

Figure 4. 80186 Instruction Set

CONDITIONAL TRANSFERS		UNCONDITIONAL TRANSFERS	
JA/JNBE	Jump if above/not below nor equal	CALL	Call procedure
JAE/JNB	Jump if above or equal/not below	RET	Return from procedure
JB/JNAE	Jump if below/not above nor equal	JMP	Jump
JBE/JNA	Jump if below or equal/not above		
JC	Jump if carry	**ITERATION CONTROLS**	
JE/JZ	Jump if equal/zero		
JG/JNLE	Jump if greater/not less nor equal	LOOP	Loop
JGE/JNL	Jump if greater or equal/not less	LOOPE/LOOPZ	Loop if equal/zero
JL/JNGE	Jump if less/not greater nor equal	LOOPNE/LOOPNZ	Loop if not equal/not zero
JLE/JNG	Jump if less or equal/not greater	JCXZ	Jump if register CX = 0
JNC	Jump if not carry		
JNE/JNZ	Jump if not equal/not zero	**INTERRUPTS**	
JNO	Jump if not overflow		
JNP/JPO	Jump if not parity/parity odd	INT	Interrupt
JNS	Jump if not sign	INTO	Interrupt if overflow
JO	Jump if overflow	IRET	Interrupt return
JP/JPE	Jump if parity/parity even		
JS	Jump if sign		

Figure 4. 80186 Instruction Set (continued)

To access operands that do not reside in one of the four immediately available segments, a full 32-bit pointer can be used to reload both the base (segment) and offset values.

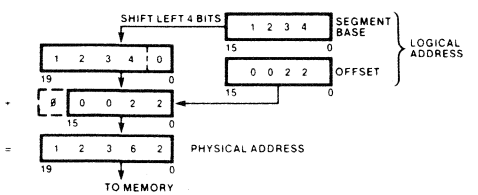


Figure 5. Two Component Address

Table 3. Segment Register Selection Rule

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Instructions	Code (CS)	Instruction prefetch and immediate data.
Stack	Stack (SS)	All stack pushes and pops; any memory references which use BP Register as a base register.
External Data (Global)	Extra (ES)	All string instruction references which use the DI register as an index.
Local Data	Data (DS)	All other data references.

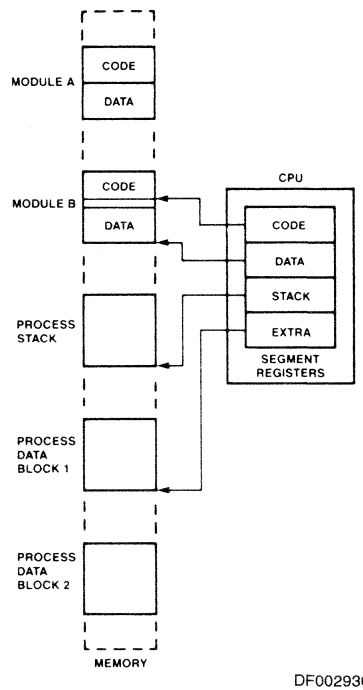


Figure 6. Segmented Memory Helps Structure Software

Addressing Modes

The 80186 provides eight categories of addressing modes to specify operands. Two addressing modes are provided for instructions that operate on register or immediate operands:

- *Register Operand Mode:* The operand is located in one of the 8- or 16-bit general registers.
- *Immediate Operand Mode:* The operand is included in the instruction.

Six modes are provided to specify the location of an operand in a memory segment. A memory operand address consists of two 16-bit components: a segment base and an offset. The segment base is supplied by a 16-bit segment register either implicity chosen by the addressing mode or explicitly chosen by a segment override prefix. The offset, also called the effective address, is calculated by summing any combination of the following three address elements:

- the *displacement* (an 8- or 16-bit immediate value contained in the instruction);
- the *base* (contents of either the BX or BP base registers); and
- the *index* (contents of either the SI or DI index registers)

Any carry out from the 16-bit addition is ignored. Eight-bit displacements are sign extended to 16-bit values.

Combinations of these three address elements define the six memory addressing modes, described below.

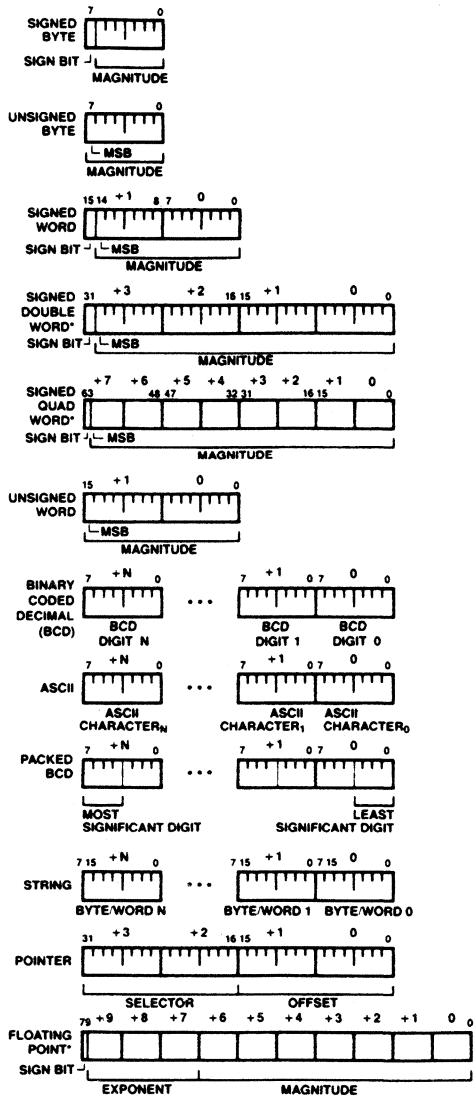
- *Direct Mode:* The operand's offset is contained in the instruction as an 8- or 16-bit displacement element.
- *Register Indirect Mode:*The operand's offset is in one of the registers SI, DI, BX, or BP.
- *Based Mode:* The operand's offset is the sum of an 8- or 16-bit displacement and the contents of a base register (BX or BP).
- *Indexed Mode:* The operand's offset is the sum of an 8- or 16-bit displacement and the contents of an index register (SI or DI).
- *Based Indexed Mode:* The operand's offset is the sum of the contents of a base register and an index register.
- *Based Indexed Mode with Displacement:* The operand's offset is the sum of a base register's contents, an index register's contents, and an 8- or 16-bit displacement.

Data Types

The 80186 directly supports the following data types:

- *Integer:* A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a 2's complement representation. Signed 32 and 64 bit integers are supported using a numeric data processor.
- *Ordinal:* An unsigned binary numeric value contained in an 8-bit byte or a 16-bit word.
- *Pointer:* A 16- or 32-bit quantity, composed of a 16-bit offset component or a 16-bit segment base component in addition to a 16-bit offset component.
- *String:* A contiguous sequence of bytes or words. A string may contain from 1K to 64K bytes.
- *ASCII:* A byte representation of alphanumeric and control characters using the ASCII standard of character representation.
- *BCD:* A byte (unpacked) representation of the decimal digits 0-9.
- *Packed BCD:* A byte (packed) representation of two decimal digits (0-9). One digit is stored in each nibble (4-bits) of the byte.
- *Floating Point:* A signed 32-, 64-, or 80-bit real number representation. (Floating point operands are supported using a numeric data processor configuration.)

In general, individual data elements must fit within defined segment limits. Figure 7 graphically represents the data types supported by the 80186.



DF002940

NOTE: *SUPPORTED BY 80186 WITH A NUMERIC DATA PROCESSOR

Figure 7. 80186 Supported Data Types

I/O Space

The I/O space consists of 64K 8-bit or 32K 16-bit ports. Separate instructions address the I/O space with either an 8-bit port address, specified in the instruction, or a 16-bit port address in the DX register, 8-bit port addresses are zero extended such that A_{15}-A_8 are LOW. I/O port addresses 00F8(H) through 00FF(H) are reserved.

Interrupts

An interrupt transfers execution to a new program location. The old program address (CS:IP) and machine state (Status Word) are saved on the stack to allow resumption of the interrupted program. Interrupts fall into three classes: hardware initiated, INT instructions, and instruction exceptions. Hardware initiated interrupts occur in response to an external input and are classified as non-maskable or maskable.

Programs may cause an interrupt with an INT instruction. Instruction exceptions occur when an unusual condition, which prevents further instruction processing, is detected while attempting to execute an instruction. If the exception was caused by executing an ESC instruction with the ESC trap bit set in the relocation register, the return instruction will point to the ESC instruction, or to the segment override prefix immediately preceding the ESC instruction if the prefix was present. In all other cases, the return address from an exception will point at the instruction immediately following the instruction causing the exception.

A table containing up to 256 pointers defines the proper interrupt service routine for each interrupt. Interrupts 0-31, some of which are used for instruction exceptions, are reserved. Table 4 shows the 80186 predefined types and default priority levels. For each interrupt, an 8-bit vector must be supplied to the 80186 which identifies the appropriate table entry. Exceptions supply the interrupt vector internally. In addition, internal peripherals and noncascaded external interrupts will generate their own vectors through the internal interrupt controller. INT instructions contain or imply the vector and allow access to all 256 interrupts. Maskable hardware initiated interrupts supply the 8-bit vector to the CPU during an interrupt acknowledge bus sequence. Non-maskable hardware interrupts use a predefined internally supplied vector.

Interrupt Sources

The 80186 can service interrupts generated by software or hardware. The software interrupts are generated by specific instructions (INT, ESC, unused OP, etc.) or the results of conditions specified by instructions (array bounds check, INTO, DIV, IDIV, etc.) All interrupt sources are serviced by an indirect call through an element of a vector table. This vector table is indexed by using the interrupt vector type (Table 4), multiplied by four. All hardware-generated interrupts are sampled at the end of each instruction. Thus, the software interrupts will begin service first. Once the service routine is entered and interupts are enabled, any hardware source of sufficient priority can interrupt the service routine in progress.

The software generated 80186 interrupts are described below.

DIVIDE ERROR EXCEPTION (TYPE 0)

Generated when a DIV or IDIV instuction quotient cannot be expressed in the number of bits in the destination.

Table 4. 80186 Interrupt Vectors

Interrupt Name	Vector Type	Default Priority	Related Instructions
Divide Error Exception	0	*1	DIV, IDIV
Single Step Interrupt	1	12**2	All
NMI	2	1	All
Breakpoint Interrupt	3	*1	INT
INT0 Detected Overflow Exception	4	*1	INT0
Array Bounds Exception	5	*1	BOUND
Unused-Opcode Exception	6	*1	Undefined Opcodes
ESC Opcode Exception	7	*1***	ESC Opcodes
Timer 0 Interrupt	8	2A****	
Timer 1 Interrupt	16	2B****	
Timer 2 Interrupt	17	2C****	
Reserved	9	3	
DMA 0 Interrupt	10	4	
DMA 1 Interrupt	11	5	
INT0 Interrupt	12	6	
INT1 Interrupt	13	7	
INT2 Interrupt	14	8	
INT3 Interrupt	15	9	

NOTES:

*1. These are generated as the result of an instruction execution.

**2. This is handled as in the 8086.

****3. All three timers constitute one source of request to the interrupt controller. The Timer interrupts all have the same default priority level with respect to all other interrupt sources. However, they have a defined priority ordering amongst themselves. (Priority 2A is higher priority than 2B.) Each Timer interrupt has a separate vector type number.

4. Default priorities for the interrupt sources are used only if the user does not program each source into a unique priority level.

***5. An escape opcode will cause a trap only if the proper bit is set in the peripheral control block relocation register.

SINGLE-STEP INTERRUPT (TYPE 1)

Generated after most instructions if the TF flag is set. Interrupts will not be generated after prefix instructions (e.g., REP), instructions which modify segment registers (e.g., POP DS), or the WAIT instruction.

NON-MASKABLE INTERRUPT-NMI (TYPE 2)

An external interrupt source which cannot be masked.

BREAKPOINT INTERRUPT (TYPE 3)

A one-byte version of the INT instruction. It uses 12 as an index into the service routine address table (because it is a type 3 interrupt).

INTO DETECTED OVERFLOW EXCEPTION (TYPE 4)

Generated during an INTO instruction if the 0F bit is set.

ARRAY BOUNDS EXCEPTION (TYPE 5)

Generated during a BOUND instruction if the array index is outside the array bounds. The array bounds are located in memory at a location indicated by one of the instruction operands. The other operand indicates the value of the index to be checked.

UNUSED OPCODE EXCEPTION (TYPE 6)

Generated if execution is attempted on undefined opcodes.

ESCAPE OPCODE EXCEPTION (TYPE 7)

Generated if execution is attempted of ESC opcodes (D8H-DFH). This exception will only be generated if a bit in the relocation register is set. The return address of this exception will point to the ESC instruction causing the exception. If a segment override prefix preceded the ESC instruction, the return address will point to the segment override prefix.

Hardware-generated interrupts are divided into two groups: maskable interupts and non-maskable interrupts. The 80186 provides maskable hardware interrupt request pins INTO-INT3. In addition, maskable interrupts may be generated by the 80186 integrated DMA controller and the integrated timer unit. The vector types for these interrupts is shown in Table 4. Software enables these inputs by setting the interrupt flag bit (IF) in the Status Word. The interrupt controller is discussed in the peripheral section of this data sheet.

Further maskable interrupts are disabled while servicing an interrupt because the IF bit is reset as part of the response to an interrupt or exception. The saved Status Word will reflect the enable status of the processor prior to the interrupt. The interrupt flag will remain zero unless specifically set. The interrupt return instruction restores the Status Word, thereby restoring the original status of IF bit. If the interrupt return re-enables interrupts, and another interrupt is pending, the 80186 will immediately service the highest-priority interrupt pending, i.e., no instructions of the main line program will be executed.

Non-Maskable Interrupt Request (NMI)

A non-maskable interrupt (NMI) is also provided. This interrupt is serviced regardless of the state of the IF bit. A typical use of NMI would be to activate a power failure routine. The activation of this input causes an interrupt with an internally supplied vector value of 2. No external interrupt acknowledge sequence is performed. The IF bit is cleared at the beginning of an NMI interrupt to prevent maskable interrupts from being serviced.

Single-Step Interrupt

The 80186 has an internal interrupt that allows programs to execute one instruction at a time. It is called the single-step interrupt and is controlled by the single-step flag bit (TF) in the Status Word. Once this bit is set, an internal single-step interrupt will occur after the next instruction has been executed. The interrupt clears the TF bit and uses an internally supplied vector of 1. The IRET instruction is used to set the TF bit and transfer control to the next instruction to be single-stepped.

Initialization and Processor Reset

Processor initialization or startup is accomplished by driving the $\overline{RES}$ input pin LOW. $\overline{RES}$ forces the 80186 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as $\overline{RES}$ is active. After $\overline{RES}$ becomes inactive and an internal processing interval elapses, the 80186 begins execution with the instruction at physical location FFFF0(H). $\overline{RES}$ also sets some registers to predefined values as shown in Table 5.

Table 5. 80186 Initial Register State after RESET

Status Word	F002(H)
Instruction Pointer	0000(H)
Code Segment	FFFF(H)
Data Segment	0000(H)
Extra Segment	0000(H)
Stack Segment	0000(H)
Relocation Register	20FF(H)
UMCS	FFFB(H)

80186 CLOCK GENERATOR

The 80186 provides an on-chip clock generator for both internal and external clock generation. The clock generator features a crystal oscillator, a divide-by-two counter, synchronous and asynchronous ready inputs, and reset circuitry.

Oscillator

The oscillator circuit of the 80186 is designed to be used with a parallel resonant fundamental mode crystal. This is used as the time base for the 80186. The crystal frequency selected will be double the CPU clock frequency. Use of an LC or RC circuit is not recommended with this oscillator. If an external oscialltor is used, it can be connected directly to input pin X1 in lieu of a crystal. The output of the oscillator is not directly available outside the 80186. The recommended crystal configuration is shown in Figure 8.

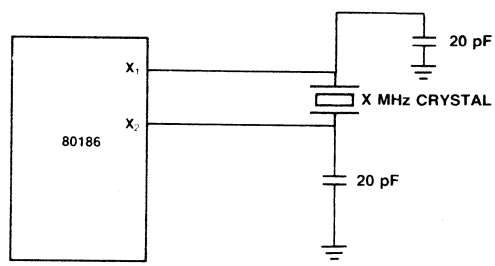


Figure 8. Recommended 80186 Crystal Configuration

Clock Generator

The 80186 clock generator provides the 50% duty cycle processor clock for the 80186. It does this by dividing the oscillator output by 2 forming the symmetrical clock. If an external oscillator is used, the state of the clock generator will change on the falling edge of the oscillator signal. The CLKOUT pin provides the processor clock signal for use outside the 80186. This may be used to drive other system components. All timings are referenced to the output clock.

READY Synchronization

The 80186 provides both synchronous and asynchronous ready inputs. Asynchronous ready synchronization is accomplished by circuitry which samples ARDY in the middle of T_2, T_3 and again in the middle of each T_W until ARDY is sampled HIGH. One-half CLKOUT cycle of resolution time is used. Full synchronization is performed only on the rising edge of ARDY, i.e., the falling edge of ARDY must be synchronized to the CLKOUT signal if it will occur during T_2 or T_W. HIGH-to-LOW transitions of ARDY must be performed synchronously to the CPU clock.

A second ready input (SRDY) is provided to interface with externally synchronized ready signals. This input is sampled at the end of T_2 and again at the end of each T_W until it is sampled HIGH. By using this input rather than the asynchronous ready input, the half-clock cycle resolution time penalty is eliminated.

This input must satisfy set-up and hold times to guarantee proper operation of the circuit.

In addition, the 80186, as part of the integrated chip-select logic, has the capability to program WAIT states for memory and peripheral blocks. This is discussed in the Chip Select/Ready Logic description.

RESET Logic

The 80186 provides both a $\overline{RES}$ input pin and a synchronized RESET pin for use with other system components. The $\overline{RES}$ input pin on the 80186 is provided with hysteresis in order to facilitate power-on Reset generation via an RC network. RESET is guaranteed to remain active for at least five clocks given a $\overline{RES}$ input of at least six clocks. RESET may be delayed up to two and one-half clocks behind $\overline{RES}$.

Multiple 80186 processors may be synchronized through the $\overline{RES}$ input pin, since this input resets both the processor and divide-by-two internal counter in the clock generator. In order to insure that the divide-by-two counters all begin counting at the same time, the active going edge of $\overline{RES}$ must satisfy a 25 ns setup time before the falling edge of the 80186 clock input. In addition, in order to insure that all CPUs begin executing in the same clock cycle, the reset must satisfy a 25 ns setup time before the rising edge of the CLKOUT signal of all the processors.

LOCAL BUS CONTROLLER

The 80186 provides a local bus controller to generate the local bus control signals. In addition, it employs a HOLD/HLDA protocol for relinquishing the local bus to other bus masters. It also provides control lines that can be used to enable external buffers and to direct the flow of data on and off the local bus.

Memory/Peripheral Control

The 80186 provides ALE, $\overline{RD}$, and $\overline{WR}$ bus control signals. The $\overline{RD}$ and $\overline{WR}$ signals are used to strobe data from memory to the 80186 or to strobe data from the 80186 to memory. The ALE line provides a strobe to address latches for the multiplexed address/data bus. The 80186 local bus controller does not provide a memory/$\overline{I/O}$ signal. If this is required, the user will have to use the $\overline{S2}$ signal (which will require external latching), make the memory and I/O spaces nonoverlapping, or use only the integrated chip-select circuitry.

Transceiver Control

The 80186 generates two control signals to be connected to 29833/29863 transceiver chips. This capability allows the addition of transceivers for extra buffering without adding external logic. These control lines, DT/$\overline{R}$ and $\overline{DEN}$, are generated to control the flow of data through the transceivers. The operation of these signals is shown in Table 6.

Table 6. Transceiver Control Signals Description

Pin Name	Function
$\overline{DEN}$ (Data Enable)	Enables the output drivers of the transceivers. It is active LOW during memory, I/O, or INTA cycles.
DT/$\overline{R}$ (Data Transmit/Receive)	Determines the direction of travel through the transceivers. A HIGH level directs data away from the processor during write operations, while a LOW level directs data toward the processor during a read operation.

Local Bus Arbitration

The 80186 uses a HOLD/HLDA system of local bus exchange. This provides an asynchronous bus exchange mechanism. This means multiple masters utilizing the same bus can operate at separate clock frequencies. The 80186 provides a single HOLD/HLDA pari through which all other bus masters may gain cotnrol of the local bus. This requires external circuitry to arbitrate whcih external device will gain control of the bus from the 80186 when there is more than one alternate local bus master. When the 80186 relinquishes control of the local bus, it floats $\overline{DEN}$, $\overline{RD}$, $\overline{WR}$, $\overline{S0}$-$\overline{S2}$, $\overline{LOCK}$, AD0-AD15, A16-A19, $\overline{BHE}$, and DT/$\overline{R}$ to allow another master to drive these lines directly.

The 80186 HOLD latency time, i.e., the time between HOLD request and HOLD acknowledge, is a function of the activity occurring in the processor when the HOLD request is received. A HOLD request is the highest-priority activity request which the processor may receive: higher than instruction fetching or internal DMA cycles. However, if a DMA cycle is in progress, the 80186 will complete the transfer before relinquishing the bus. This implies that if a HOLD request is received just as a DMA transfer begins, the HOLD latency time can be as great as 4 bus cycles. This will occur if a DMA word transfer operation is taking place from an odd address to an odd address. This is a total of 16 clocks or more, if WAIT states are required. In addition, if locked transfers are performed, the HOLD latency time will be increased by the length of the locked transfer.

Local Bus Controller and Reset

Upon receipt of a RESET pulse from the $\overline{RES}$ input, the local bus controller will perform the following actions:

- Drive $\overline{DEN}$, $\overline{RD}$, and $\overline{WR}$ HIGH for one clock cycle, then float.

  NOTE: $\overline{RE}$ is also provided with an internal pull-up device to prevent the processor from inadvertently entering Queue Status mode during reset.
- Drive $\overline{S0}$-$\overline{S2}$ to the passive state (all HIGH) and then float.
- Drive $\overline{LOCK}$ HIGH and then float.
- Tristate AD0-15, A16-19, $\overline{BHE}$, DT/$\overline{R}$.
- Drive ALE LOW (ALE is never floated).
- Drive HLDA LOW.

INTERNAL PERIPHERAL INTERFACE

All the 80186 integrated peripherals are controlled via 16-bit registers contained within an internal 256-byte control block. This control block may be mapped into either memory or I/O space. Internal logic will recognize the address and respond to

the bus cycle. During bus cycles to internal registers, the bus controller will signal the operation externally (i.e., the $\overline{RD}$, $\overline{WR}$, status, address, data, etc., lines will be driven as in a normal bus cycle), but D_{15-0}, SRDY, and ARDY will be ignored. The base address of the control block must be on an even 256-byte boundary (i.e., the lower 8 bits of the base address are all zeros). All of the defined registers within this control block may be read or written by the 80186 CPU at any time. The location of any register contained within the 256-byte control block is determined by the current base address of the control block.

The control block base address is programmed via a 16-bit relocation register contained within the control block at offset FEH from the base address of the control block (see Figure 9). It provides the upper 12 bits of the base address of the control block. Note that mapping the control register block into an address range corresponding to a chip-select range is not recommended (the chip select circuitry is discussed later in this date sheet). In addition, bit 12 of this register determines whether the control block will be mapped into I/O or memory space. If this bit is 1, the control block will be located in memory space, whereas if the bit is 0, the control block will be located in I/O space. If the control register block is mapped into I/O space, the upper 4 bits of the base address must be programmed as 0 (since I/O addresses are only 16 bits wide).

In addition to providing relocation information for the control block, the relocation register contains bits which place the interrupt controller into iRMX mode, and cause the CPU to interrupt upon encountering ESC instructions. At RESET, the relocation register is set to 20FFH. This causes the control block to start at FF00H in I/O space. An offset map of the 256-byte control register block is shown in Figure 10.

The integrated 80186 peripherals operate semiautonomously from the CPU. Access to them for the most part is via software read/write of the control and data locations in the control block. Most of these registers can be both read and written. A few dedicated lines, such as interrupts and DMA request provide real-time communication between the CPU and peripherals as in a more conventional system utilizing discrete peripheral blocks. The overall interaction and function of the peripheral blocks has not substantially changed.

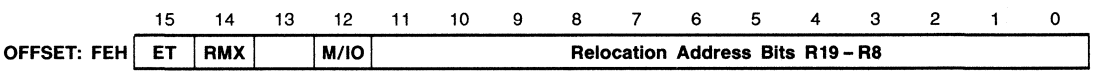
CHIP-SELECT/READY GENERATION LOGIC

The 80186 contains logic which provides programmable chip-select generation for both memories and peripherals. In addition, it can be programmed to provide READY (or WAIT state) generation. It can also provide latched address bits A1 and A2. The chip-select lines are active for all memory and I/O cycles in their programmed areas, whether they be generated by the CPU or by the integrated DMA unit.

Memory Chip Selects

The 80186 provides 6 memory chip select outputs for 3 address areas: upper memory, lower memory, and midrange memory. One each is provided for upper memory and lower memory, while four are provided for midrange memory.

The range for each chip select is user-programmable and can be set to 2K, 4K, 8K, 16K, 32K, 64K, 128K (plus 1K and 256K for upper and lower chip selects). In addition, the beginning or base address of the midrange memory chip select may also be selected. Only one chip select may be programmed to be active for any memory location at a time. All chip select sizes are in bytes, whereas 80186 memory is arranged in words. This means that if, for example, 16 64K x 1 memories are used, the memory block size will be 128K, not 64K.



ET = ESC Trap/No ESC Trap (1/0)
M/IO = Register block located in Memory/ I/O Space (1/0)
RMX = Normal Interrupt Controller mode/ IRMX compatible Interrupt Controller mode (0/1)

Figure 9. Relocation Register

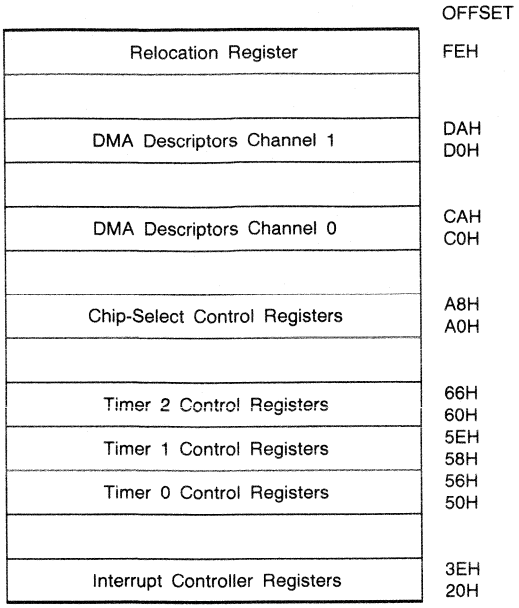


Figure 10. Internal Register Map

Upper Memory $\overline{CS}$

The 80186 provides a chip select, called $\overline{UCS}$, for the top of memory. The top of memory is usually used as the system memory because after reset the 80186 begins executing at memory location FFFF0H.

The upper limit of memory defined by this chip select is always FFFFFH, while the lower limit is programmable. By programming the lower limit, the size of the select block is also defined. Table 7 shows the relationship between the base address selected and the size of the memory block obtained.

Table 7. UMCS Programming Values

Starting Address (Base Address)	Memory Block Size	UMCS Value (Assuming R0 = R1 = R2 = 0)
FFC00	1K	FFF8H
FF800	2K	FFB8H
FF000	4K	FF38H
FE000	8K	FE38H
FC000	16K	FC38H
F8000	32K	F838H
F0000	64K	F038H
E0000	128K	E038H
C0000	256K	C038H

The lower limit of this memory block is defined in the UMCS register (see Figure 11). This register is at offset A0H in the internal control block. The legal values for bits 6-13 and the resulting starting address and memory block sizes are given in Table 7. Any combination of bits 6-13 not shown in Table 7 will result in undefined operation. After reset, the UMCS register is programmed for a 1K area. It must be reprogrammed if a larger upper memory area is desired.

Any internally generally 20-bit address whose upper 16-bits are greater than or equal to UMCS (with bits 0-5 "0") will cause UCS to be activated. UMCS bits R2-R0 are used to specify READY mode for the area of memory defined by this chip-select register, as explained below.

Lower Memory $\overline{CS}$

The 80186 provides a chip select for low memory called LCS. The bottom of memory contains the interrupt vector table, starting at location 00000H.

The lower limit of memory defined by this chip select is always 0H, while the upper limit is programmable. By programming the upper limit, the size of the memory block is also defined. Table 8 shows the relationship between the upper address selected and the size of the memory block obtained.

Table 8. LMCS Programming Values

Upper Address	Memory Block Size	LMCS Value (Assuming R0 = R1 = R2 = 0)
003FFH	1K	0038H
007FFH	2K	0078H
00FFFH	4K	00F8H
01FFFH	8K	01F8H
03FFFH	16K	03F8H
07FFFH	32K	07F8H
0FFFFH	64K	0FF8H
1FFFFH	128K	1FF8H
3FFFFH	256K	3FF8H

The upper limit of this memory block is defined in the LMCS register (see Figure 12). This register is at offset A2H in the internal control block. The legal values for bits 6-15 and the resulting upper address and memory block sizes are given in Table 8. Any combination of bits 6-15 not shown in Table 8 will result in undefined operation. After reset, the LMCS register value is undefined. However, the $\overline{LCS}$ chip-select line will not become active until the LMCS register is accessed.

Any internally generated 20-bit address whose upper 16 bits are less than or equal to LMCS (with bits 0-5 "1") will cause $\overline{LCS}$ to be active. LMCS register bits R2-R0 are used to specify the READY mode for the area of memory defined by this chip-select register.

Mid-Range Memory $\overline{CS}$

The 80186 provides four $\overline{MCS}$ lines which are active within a user-locatable memory block. This block can be located anywhere within the 80186 1M byte memory address space exclusive of the areas defined by $\overline{UCS}$ and $\overline{LCS}$. Both the base address and size of this memory block are programmable.

The size of the memory block defined by the midrange select lines, as shown in Table 9, is determined by bits 8-14 of the MPCS register (see Figure 13). This register is at location A8H in the internal control block. One and only one of bits 8-14 must be set at a time. Unpredictable operation of the $\overline{MCS}$ lines will otherwise occur. Each of the four chip-select lines is active for one of the four equal contiguous divisions of the mid-range block. Thus, if the total block size is 23K, each chip select is active for 8K of memory with $\overline{MCS0}$ being active for the first range and $\overline{MCS3}$ being active for the last range.

The EX and MS in MPCS relate to peripheral functionally as described in a later section.

Table 9. MMCS Programming Values

Total Block Size	Individual Select Size	MMCS Bits 14 – 8
8K	2K	0000001B
16K	4K	0000010B
32K	8K	0000100B
64K	16K	0001000B
128K	32K	0010000B
256K	64K	0100000B
512K	128K	1000000B

The base address of the mid-range memory block is defined by bits 15-9 of the MMCS register (see Figure 14). This register is at offset A6H in the internal control block. These bits correspond to bits A19-A13 of the 20-bit memory address. Bits A12-A0 of the base address area always 0. The base address may be set at any integer multiple of the size of the total memory block selected. For example, if the midrange block size is 32K (or the size of the block for which each $\overline{MCS}$ line is active is 8K), the block could be located at 10000H or 18000H, but not at 14000H, since the first few integer multiples of a 32K memory block are 0H, 8000H, 10000H, 18000H, etc. After reset, the contents of both of these registers is undefined. However, none of the $\overline{MCS}$ lines will be active until both the MMCS and MPCS registers are accessed.

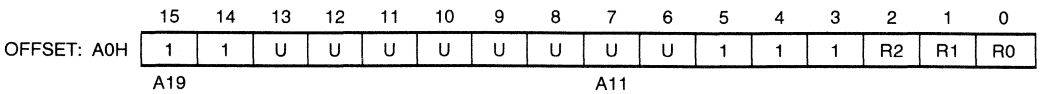


Figure 11. UMCS Register

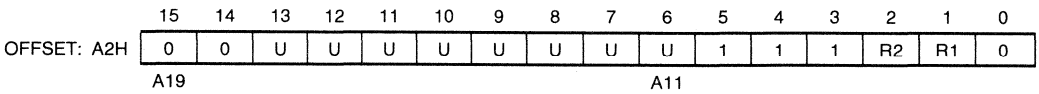


Figure 12. LMCS Register

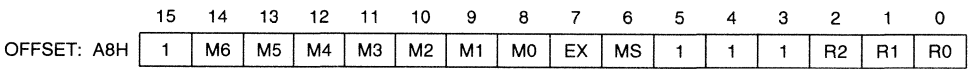


Figure 13. MPCS Register

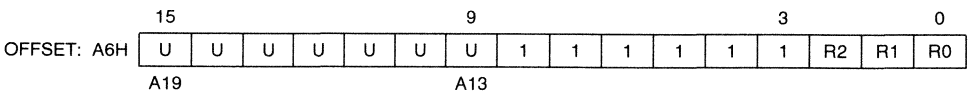


Figure 14. MMCS Register

MMCS bits R2-R0 specify READY mode of operation for all mid-range chip selects. All devices in mid-range memory must use the same number of WAIT states.

The 512K block size for the mid-range memory chip selects is a special case. When using 512K, the base address would have to be at either locations 00000H or 80000H. If it were to be programmed at 00000H when the $\overline{LCS}$ line was programmed, there would be an internal conflict between the $\overline{LCS}$ ready generation logic and the $\overline{MCS}$ ready generation logic. Likewise, if the base address were programmed at 80000H, there would be a conflict with the $\overline{UCS}$ ready generation logic. Since the $\overline{LCS}$ chip-select line does not become active until programmed, while the $\overline{UCS}$ line is active at reset, the memory base can be set only at 00000H. If this base address is selected, however, the $\overline{LCS}$ range must not be programmed.

Peripheral Chip Selects

The 80186 can generate chip selects for up to seven peripheral devices. These chip selects are active for seven contiguous blocks of 128 bytes above a programmable base address. This base address may be located in either memory or I/O space.

Seven $\overline{CS}$ lines called $\overline{PCS0}$-6 are generated by the 80186. The base address is user-programmable; however it can only be a multiple of 1K bytes, i.e., the least significant 10 bits of the starting address are always 0.

$\overline{PCS5}$ and $\overline{PCS6}$ can also be programmed to provide latched address bits A1, A2. If so programmed, they cannot be used as peripheral selects. These outputs can be connected directly to the A0, A1 pins used for selecting internal registers of 8-bit peripheral chips. This scheme simplifies the hardware interface because the 8-bit registers of peripherals are simply

treated as 16-bit registers located on even boundaries in I/O space or memory space where only the lower 8-bits of the register are significant: the upper 8-bits are "don't cares."

The starting address of the peripheral chip-select block is defined by the PACS register (see Figure 15). This register is located at offset A4H in the internal control block. Bits 15-6 of this register correspond to bits 19-10 of the 20-bit Programmable Base Address (PBA) of the peripheral chip-select block. Bits 9-0 of the PBA of the peripheral chip-select block are all zeros. If the chip-select block is located in I/O space, bits 12-15 must be programmed zero, since the I/O address is only 16 bits wide. Table 10 shows the address range of each peripheral chip select with respect to the PBA contained in PACS register.

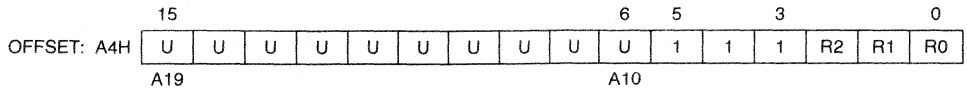


Figure 15. PACS Register

The user should program bits 15-6 to correspond to the desired peripheral base location. PACS bits 0-2 are used to specify READY mode for $\overline{PCS0}$-$\overline{PCS3}$.

Table 10. PCS Address Ranges

PCS Line	Active between Locations
PCS0	PBA — PBA + 127
PCS1	PBA + 128 — PBA + 255
PCS2	PBA + 256 — PBA + 383
PCS3	PBA + 384 — PBA + 511
PCS4	PBA + 512 — PBA + 639
PCS5	PBA + 640 — PBA + 767
PCS6	PBA + 768 — PBA + 895

The mode of operation of the peripheral chip selects is defined by the MPCS register (which is also used to set the size of the mid-range memory chip-select block, see Figure 16). This register is located at offset A8H in the internal control block. Bit 7 is used to select the function of $\overline{PCS5}$ and $\overline{PCS6}$, while bit 6 is used to select whether the peripheral chip selects are mapped into memory or I/O space. Table 11 describes the programming of these bits. After reset, the contents of both the MPCS and the PACS registers are undefined, however none of the PCS lines will be active until both of the MPCS and PACS registers are accessed.

Table 11. MS, EX Programming Values

Bit	Description
MS	1 = Peripherals mapped into memory space. 0 = Peripherals mapped into I/O space.
EX	0 = 5 $\overline{PCS}$ lines. A1, A2 provided. 1 = 7 $\overline{PCS}$ lines. A1, A2 are not provided.

MPCS bits 0-2 are used to specify READY mode for $\overline{PCS4}$-$\overline{PCS6}$ as outlined below.

READY Generation Logic

The 80186 can generate a "READY" signal internally for each of the memory or peripheral $\overline{CS}$ lines. The number of WAIT states to be inserted for each peripheral or memory is programmable to provide 0-3 wait states for all accesses to the area for which the chip select is active. In addition, the 80186 may be programmed to either ignore external READY for each chip- select range individually or to factor external READY with the integrated ready generator.

READY control consists of 3 bits for each $\overline{CS}$ line or group of lines generated by the 80186. The interpretation of the ready bits is shown in Table 12.

Table 12. READY Bits Programming

R2	R1	R0	Number of WAIT States Generated
0	0	0	0 wait states, external RDY also used.
0	0	1	1 wait state inserted, external RDY also used.
0	1	0	2 wait states inserted, external RDY also used.
0	1	1	3 wait states inserted, external RDY also used.
1	0	0	0 wait states, external RDY ignored.
1	0	1	1 wait state inserted, external RDY ignored.
1	1	0	2 wait states inserted, external RDY ignored.
1	1	1	3 wait states inserted, external RDY ignored.

The internal ready generator operates in parallel with external READY, not in series if the external READY is used (R2 = 0). This means, for example, if the internal generator is set to insert two wait states, but activity on the external READY lines will insert four wait states, the processor will only insert four wait states, not six. This is because the two wait states generated by the internal generator overlapped the first two wait states generated by the external ready signal. Note that the external ARDY and SRDY lines are always ignored during cycles accessing internal peripherals.

R2-R0 of each control word specifies the READY mode for the corresponding block, with the exception of the peripheral chip selects: R2-R0 of PACS set the $\overline{PCS0}$-3 READY mode, R2-R0 of MPCS set the $\overline{PCS4}$-6 READY mode.

Chip Select/Ready Logic and Reset

Upon reset, the Chip-Select/Ready Logic will perform the following actions:

- All chip-select outputs will be driven HIGH.
- Upon leaving RESET, the UCS line will be programmed to provide chip selects to a 1K block with the accompanying READY control bits set at 011 to allow the maximum number of internal wait states in conjunction with external Ready consideration (i.e. UMCS resets to FFFBH).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET: A8H	1	M6	M5	M4	M3	M2	M1	M0	EX	MS	1	1	1	R2	R1	R0

Figure 16. MPCS Register

- No other chip select or READY control registers have any predefined values after RESET. They will not become active until the CPU accesses their control registers. Both the PACS and MPCS registers must be accessed before the $\overline{PCS}$ lines will become active.

DMA CHANNELS

The 80186 DMA controller provides two independent high-speed DMA channels. Data transfers can occur between memory and I/O spaces (e.g., Memory to I/O) or within the same space (e.g., Memory to Memory or I/O to I/O). Data can be transferred either in bytes (8 bits) or in words (16 bits) to or from even or odd addresses. Each DMA channel maintains both a 20-bit source and destination pointer which can be optionally incremented or decremented after each data transfer (by one or two depending on byte or word transfers). Each data transfer consumes 2 bus cycles (a minimum of 8 clocks), one cycle to fetch data and the other to store data. This provides a maximum data transfer rate of one Mword/sec or 2 MBytes/sec.

DMA Operation

Each channel has six registers in the control block which define each channel's specific operation. The control registers consist of a 20-bit Source pointer (2 words), a 20-bit Destination pointer (2 words), a 16-bit Transfer Counter, and a 16-bit Control Word. The format of the DMA Control Blocks is shown in Table 13. The Transfer Count Register (TC) specifies the number of DMA transfers to be performed. Up to 64K byte or word transfers can be performed with automatic termination. The Control Word defines the channel's operation (see Figure 18). All registers may be modified or altered during any DMA activity. Anyl changes made to these registers will be reflected immediately in DMA operation.

Table 13. DMA Control Block Format

Register Name	Register Address	
	Ch. 0	Ch. 1
Control Word	CAH	DAH
Transfer Count	C8H	D8H
Destination Pointer (upper 4 bits)	C6H	D6H
Destination Pointer	C4H	D4H
Source Pointer (upper 4 bits)	C2H	D2H
Source Pointer	C0H	D0H

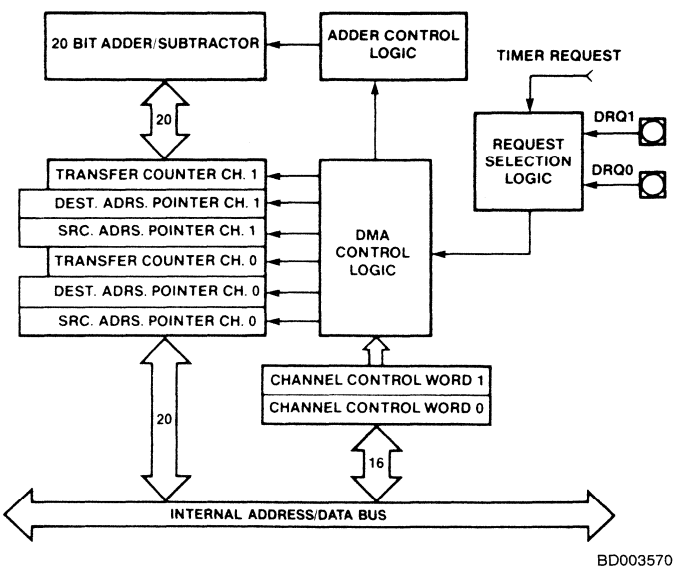


Figure 17. DMA Unit Block Diagram

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M/$\overline{IO}$	DESTINATION DEC	INC	M/$\overline{IO}$	SOURCE DEC	INC	TC	INT	SYN		P	TDRQ	X	CHG/$\overline{NOCHG}$	ST/$\overline{STOP}$	$\overline{B}$/W

X = DON'T CARE

Figure 18. DMA Control Register

DMA Channel Control Word Register

Each DMA Channel Control Word determines the mode of operation for the particular 80186 DMA channel. This register specifies:

- the mode of synchronization;
- whether bytes or words will be transferred;
- whether interrupts will be generated after the last transfer;
- whether DMA activity will cease after a programmed number of DMA cycles;
- the relative priority of the DMA channel with respect to the other DMA channel;
- whether the source pointer will be incremented, decremented, or maintained constant after each transfer;
- whether the source pointer addresses memory or I/O space;
- whether the destination pointer will be incremented, decremented, or maintained constant after each transfer; and
- whether the destination pointer will address memory or I/O space.

The DMA channel control registers may be changed while the channel is operating. However, any changes made during operation will affect the current DMA transfer.

DMA Control Word Bit Descriptions

$\overline{B}$/W: Byte/Word (0/1) Transfers.

ST/$\overline{STOP}$: Start/stop (1/0) Channel.

CHG/$\overline{NOCHG}$: Change/Do not change (1/0) ST/$\overline{STOP}$ bit. If this bit is set when writing to the control word, the ST/$\overline{STOP}$ bit will be programmed by the write to the control word. If this bit is cleared when writing the control word, the ST/$\overline{STOP}$ bit will not be altered. This bit is not stored; it will always be a 0 on read.

INT: Enable Interrupts to CPU on byte count termination.

TC: If set, DMA will terminate when the contents of the Transfer Count register reach zero. The ST/$\overline{STOP}$ bit will also be reset at this point if TC is set. If this bit is cleared, the DMA unit will decrement the transfer count register for each DMA cycle, but the DMA transfer will not stop when the contents of the TC register reach zero.

SYN: (2 bits)
- 00 No synchronization
  NOTE: The ST bit will be cleared automatically when the contents of the TC register reach zero regardless of the state of the bit.
- 01 Source synchronization.
- 10 Destination synchronization.
- 11 Unused.

SOURCE: INC Increment source pointer by 1 or 2 (depends on $\overline{B}$/W) after each transfer.

M/$\overline{IO}$ Source pointer is in M/IO space (1/0).

DEC Decrement source pointer by 1 or 2 (depends on $\overline{B}$/W) after each transfer.

DEST: INC Increment destination pointer by 1 or 2 ($\overline{B}$/W) after each transfer.

M/$\overline{IO}$ Destination pointer is in M/IO space (1/0).

DEC Decrement destination pointer by 1 or 2 (depending on $\overline{B}$/W) after each transfer.

P Channel priority-relative to other channel.
0 low priority.
1 high priority.
Channels will alternate cycles if both set at same priority level.

TDRQ
0: Disable DMA requests from timer 2.
1: Enable DMA requests from timer 2.

Bit 3 Bit 3 is not used.

If both INC and DEC are specified for the same pointer, the pointer will remain constant after each cycle.

DMA Destination and Source Pointer Registers

Each DMA channel maintains a 20-bit source and a 20-bit destination pointer. Each of these pointers takes up two full 16-bit registers in the peripheral control block. The lower four bits of the upper register contain the upper four bits of the 20-bit physical address (see Figure 18a). These pointers may be individually incremented or decremented after each transfer. If word transfers are performed the pointer is incremented or decremented by two. Each pointer may point into either memory or I/O space. Since the DMA channels can perform transfers to or from odd addresses, there is no restriction on values for the pointer registers. Higher transfer rates can be obtained if all word transfers are performed to even ad-

dresses, since this will allow data to be accessed in a single memory access.

DMA Transfer Count Register

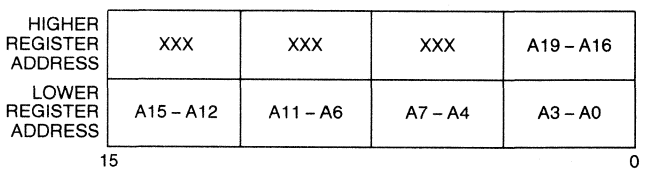
Each DMA channel maintains a 16-bit transfer count register (TC). This register is decremented after every DMA cycle, regardless of the state of the TC bit in the DMA Control Register. If the TC bit in the DMA control word is set, however, DMA activity will terminate when the transfer count register reaches zero.

DMA Requests

Data transfers may be either source or destination synchronized, that is either the source of the data or the destination of the data may request the data transfer. In addition, DMA transfers may be unsynchronized; that is, the transfer will take place continually until the correct number of transfers has occurred. When source or unsynchronized transfers are performed, the DMA channel may begin another transfer immediately after the end of a previous DMA transfer. This allows complete transfer to take place every 2 bus cycles or eight clock cycles (assuming no wait states). No prefetching occurs when destination synchronization is performed, however. Data will not be fetched from the source address until the destination device signatls that it is ready to receive it. When destination synchronized transfers are requested, the DMA controller will relinquish control of the bus after every transfer. If no other bus activity is initiated, another DMA cycle will begin after two processor clocks. This is done to allow the destination device time to remove its request if another transfer is not desired. Since the DMA controller will relinquish the bus, the CPU can initiate a bus cycle. As a result, a complete bus cycle will often be inserted between destination synchronized transfers. These lead to the maximum DMA transfer rates shown in Table 14.

Table 14. Maximum DMA Transfer Rates

Type of Synchronization Selected	CPU Running	CPU Halted
Unsynchronized	2MBytes/sec	2MBytes/sec
Source Synch	2MBytes/sec	2MBytes/sec
Destination Synch	1.3MBytes/sec	1.5MBytes/sec



XXX = DON'T CARE

Figure 18a. DMA Memory Pointer Register Format

DMA Acknowledge

No explicit DMA acknowledge pulse is provided. Since both source and destination pointers are maintained, a read from a requesting source, or a write to a requesting destination, should be used as the DMA acknowledge signal. Since the chip-select lines can be programmed to be active for a given block of memory or I/O space, and the DMA pointers can be programmed to point to the same given block, a chip-select line could be used to indicate a DMA acknowledge.

DMA Priority

The DMA channels may be programmed such that one channel is always given priority over the other, or they may be programmed such as to alternate cycles when both have DMA requests pending. DMA cycles always have priority over internal CPU cycles except between locked memory accesses or word accesses the odd memory locations; however, an external bus hold takes priority over an internal DMA cycle. Because an interrupt request cannot suspend a DMA operation and the CPU cannot access memory during a DMA cycle, interrupt latency time will suffer during sequences of continuous DMA cycles. An NMI request, however, will cause all internal DMA activity to halt. This allows the CPU to quickly respond to the NMI request.

DMA Programming

DMA cycles will occur whenever the ST/STOP bit of the Control Register is set. If synchronized transfers are programmed, a DRQ must also have been generated. Therefore, the source and destination transfer pointers, and the transfer count register (if used) must be programmed before this bit is set.

Each DMA register may be modified while the channel is operating. If the CHG/NOCHG bit is cleared when the control register is written, the ST/STOP bit of the control register will not be modified by the write. If multiple channel registers are modified, it is recommended that a LOCKED string transfer be used to prevent a DMA transfer from occurring between updates to the channel registers.

DMA Channels and Reset

Upon RESET, the DMA channels will perform the following actions:

- The Start/Stop bit for each channel will be rest to STOP.
- Any transfer in progress is aborted.

TIMERS

The 80186 provides three internal 16-bit programmable timers (see Figure 19). Two of these are highly flexible and are connected to four external pins (2 per timer). They can be used to count external events, time external events, generate nonrepetitive waveforms, etc. The third timer is not connected to any external pins, and is useful for real-time coding and time delay applications. In addition, this third timer can be used as a prescaler to the other two, or as a DMA request source.

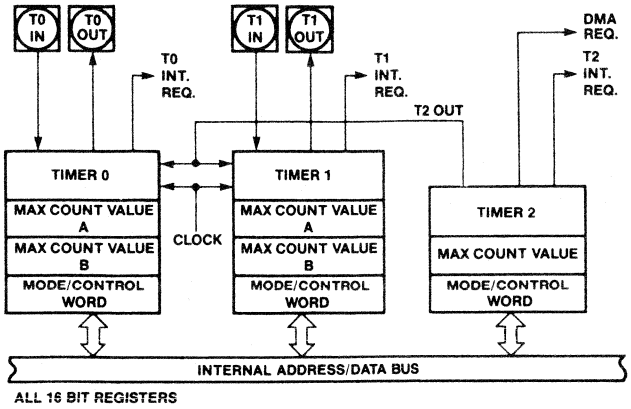


Figure 19. Timer Block Diagram

Timer Operation

The timers are controlled by 11 16-bit registers in the internal peripheral control block. The configuration of these registers is shown in Table 15. The count register contains the current value of the timer. It can be read or written at any time independent of whether the timer is running or not. The value of this register will be incremented for each timer event. Each of the timers is equipped with a MAX COUNT register, which defines the maximum count the timer will reach. After reaching the MAX COUNT register value, the timer count value will reset to zero during that same clock, i.e., the maximum count value is never stored in the count register itself. Timers 0 and 1 are, in addition, equipped with a second MAX COUNT register, which enables the timers to alternate their count between two different MAX COUNT values programmed by the user. If a single MAX COUNT register is used, the timer output pin will switch LOW for a single clock, 2 clocks after the maximum count value has been reached. In the dual MAX COUNT register mode, the output pin will indicate which MAX COUNT register is currently in use, thus allowing nearly complete freedom in selecting waveform duty cycles. For the timers with two MAX COUNT registers, the RIU bit in the control register determines which is used for the comparison.

Each timer gets serviced every fourth CPU-clock cycle, and thus can operate at speeds up to one-quarter the internal clock frequency (one-eighth the crystal rate). External clocking of the timers may be done at up to a rate of one-quarter of the internal CPU-clock rate (2 MHz for an 8 MHz CPU clock). Due to internal synchronization and pipelining of the timer circuitry, a timer output may take up to 6 clocks to respond to any individual clock or gate input.

Since the count registers and the maximum count registers are all 16 bits wide, 16 bits of resolution are provided. Any Read or Write access to the timers will add one wait state to the minimum four-clock bus cycle. However, this is needed to synchronize and coordinate the internal data flows between the internal timers and the internal bus.

The timers have several programmable options.

- All three timers can be set to halt or continue on a terminal count.
- Timers 0 and 1 can select between internal and external clocks, alternate between MAX COUNT registers and be set to retrigger on external events.
- The timers may be programmed to cause an interrupt on terminal count.

These options are selectable via the timer mode/control word.

Timer Mode/Control Register

The mode/control register (see Figure 20) allows the user to program the specific mode of operation or check the current programmed status for any of the three integrated timers.

Table 15. Timer Control Block format

Register Name	Register Offset		
	Tmr. 0	Tmr. 1	Tmr. 2
Mode/Control Word	56H	5EH	66H
Max Count B	54H	5CH	not present
Max Count A	52H	5AH	62H
Count Register	50H	58H	60H

15	14	13	12	11		5	4	3	2	1	0
EN	$\overline{INH}$	INT	RIU	0	....	MC	RTG	P	EXT	ALT	CONT

Figure 20. Timer Mode/Control Register

ALT:

The ALT bit determines which of two MAX COUNT registers is used for count comparison. If ALT = 0, register A for that timer is always used, while if ALT = 1, the comparison will alternate between register A and register B when each maximum count is reached. This alternation allows the user to change one MAX COUNT register while the other is being used, and thus provides a method of generating nonrepetitive waveforms. Square waves and pulse outputs of any duty cycle are a subset of available signals obtained by not changing the final count registers. The ALT bit also determines the function of the timer output pin. If ALT is zero, the output pin will go LOW for one clock, the clock after the maximum count is reached, If ALT is one, the output pin will reflect the current MAX COUNT register being used (0/1 for B/A).

CONT:

Setting the CONT bit causes the associated timer to run continuously, while resetting it causes the timer to halt upon maximum count. If CONT = 0 and ALT = 1, the timer will count to the MAX COUNT register A value, reset, count to the register B value, reset, and halt.

EXT:

The external bit selects between internal and external clocking for the timer. The external signal may be asynchronous with respect to the 80186 clock. If this bit is set, the timer will count LOW-to-HIGH transitions on the input pin. If cleared, it will count an internal clock while using the input pin for control. In this mode, the function of the external pin is defined by the RTG bit. The maximum input to output transition latency time may be as much as 6 clocks. However, clock inputs may be pipelined as closely together as every 4 clocks without losing clock pulses.

P:

The prescaler bit is ignored unless internal clocking has been selected (EXT = 0). If the P bit is a zero, the timer will count at one-fourth the internal CPU clock rate. If the P bit is a one, the output of timer 2 will be used as a clock for the timer. Note that the user must initialize and start timer 2 to obtain the prescaled clock.

RTG:

Retrigger bit is only active for internal clocking (EXT = 0). In this case it determines the control function provided by the input pin.

If RTG = 0, the input level gates the internal clock on and off. If the input pin is HIGH, the timer will count; if the input pin is LOW, the timer will hold its value. As indicated previously, the input signal may be asynchronous with respect to the 80186 clock.

When RTG = 1, the input pin detects LOW-to-HIGH transitions. The first such transition starts the timer running, clearing the timer value to zero on the first clock, and then incrementing thereafter. Further transitions on the input pin will again reset the timer to zero, from which it will start counting up again. If CONT = 0, when the timer has reached maximum count, the EN bit will be cleared, inhibiting further timer activity.

EN:

The enable bit provides programmer control over the timer's RUN/HALT status. When set, the timer is enabled to increment subject to the input pin constraints in the internal clock mode (discussed previously). When cleared, the timer will be inhibited from counting. All input pin transitions during the time EN is zero will be ignored. If CONT is zero, the EN bit is automatically cleared upon maximum count.

INH:

The inhibit bit allows for selective updating of the enable (EN) bit. If INH is a one during the write to the mode/control word, then the state of the EN bit will be modified by the write. If INH is a zero during the write, the EN bit will be unaffected by the operation. This bit is not stored; it will always be a 0 on a read.

INT:

When set, the INT bit enables interrupts from the timer, which will be generated on every terminal count. If the timer is configured in dual MAX COUNT register mode, an interrupt will be generated each time the value in MAX COUNT register A is reached, and each time the value in MAX COUNT register B is reached. If this enable bit is cleared after the interrupt request has been generated, but before a pending interrupt is serviced, the interrupt request will still be in force. (The request is latched in the Interrupt Controller.)

MC:

The Maximum Count bit is set whenever the timer reaches its final maximum count value. If the timer is configured in dual MAX COUNT register mode, this bit will be set each time the value in MAX COUNT register A is reached, and each time the value in MAX COUNT register B is reached. This bit is set regardless of the timer's interrupt-enable bit. The MC bit gives the user the ability to monitor timer status through software instead of through interrupts.

RIU:

The Register In Use bit indicates which MAX COUNT register is currently being used for comparison to the timer count value. A zero value indicates register A. The RIU bit cannot be written, i.e., its value is not affected when the control register is written. It is always cleared when the ALT bit is zero.

Not all mode bits are provided for timer 2. Certain bits are hardwired as indicated below:

ALT = 0, EXT = 0, P = 0, RTG = 0, RIU = 0

Count Registers

Each of the three timers has a 16-bit count register. The current contents of this register may be read or written by the processor at any time. If the register is written into while the timer is counting, the new value will take effect in the current count cycle.

Max Count Registers

Timers 0 and 1 have two MAX COUNT registers, while timer 2 has a single MAX COUNT register. These contain the number of events the timer will count. In timers 0 and 1, the MAX COUNT register used can alternate between the two max count values whenever the current maximum count is reached. The condition which causes a timer to reset is equivalent between the current count value and the max count being used. This means that if the count is changed to be above the max count value, or if the max count value is changed to be below the current value, the timer will not reset to zero, but rather will count to its maximum value, "wrap around" to zero, then count until the max count is reached.

Timers and Reset

Upon RESET, the Timers will perform the following actions:

- All EN (Enable) bits are reset preventing timer counting.
- All SEL (Select) bits are reset to zero. This selects MAX COUNT register A, resulting in the Timer Out pins going HIGH upon RESET.

INTERRUPT CONTROLLER

The 80186 can receive interrupts from a number of sources, both internal and external. The internal interrupt controller serves to merge these requests on a priority basis, for individual service by the CPU. Internal interrupt sources (Timers and DMA channels) can be disabled by their own control registers or by mask bits within the interrupt controller. The 80186 interrupt controller has its own control registers that set the mode of operation for the controller.

The interrupt controller will resolve priority among requests that are pending simultaneously. Nesting is provided so interrupt service routines for lower priority interrupts may themselves be interrupted by higher priority interrupts. A block diagram of the interrupt controller is shown in Figure 21.

The interrupt controller has a special iRMX 86 compatibility mode that allows the use of the 80186 within the iRMX 86 operating system interrupt structure. The controller is set in this mode by setting bit 14 in the peripheral control block relocation register (see iRMX 86 Compatibility Mode section). In this mode, the internal 80186 interrupt controller functions as a "slave" controller to an external "master" controller. Special initialization software must be included to properly set up the 80186 interrupt controller in iRMX 86 mode.

MASTER (NON-iRMX) MODE OPERATION

Interrupt Controller External Interface

For external interrupt sources, five dedicated pins are provided. One of these pins is dedicated to NMI, non-maskable interrupt. This is typically used for power-fail interrupts, etc. The other four pins may function either as four interrupt input lines with internally generated interrupt vectors, as an interrupt line and an interrupt acknowledge line (called the "cascade mode") along with two other input lines with internally generated interrupt vectors, or as two interrupt input lines and two dedicated interrupt acknowledge ouput lines. When the interrupt lines are configured in cascade mode, the 80186 interrupt controller will not generate internal interrupt vectors.

External sources in the cascade mode use externally generated interrupt vectors. When an interrupt is acknowledged, two $\overline{\text{INTA}}$ cycles are initiated and the vector is read into the 80186 on the second cycle. The capability to interface to external 8259A programmable interrupt controllers is thus provided when the inputs are configured in cascade mode.

Interrupt Controller Modes of Operation

The basic modes of operation of the interrupt controller in non-iRMX mode are similar to the 8259A. The interrupt controller responds identically to internal interrupts in all three modes: the difference is only in the interpretation of function of the four external interrupt pins. The interrupt controller is set into one of these three modes by programming the correct bits in the INT0 and INT1 control registers. The modes of interrupt controller operation are as follows:

Fully Nested Mode

When in the fully nested mode four pins are used as direct interrupt requests. The vectors for these four inputs are generated internally. An in-service bit is provided for every interrupt source. If a lower-priority device requests an interrupt while the in-service bit (IS) is set, no interrupt will be generated by the interrupt controller. In addition, if another interrupt request occurs from the same interrupt source while the inservice bit is set, no interrupt will be generated by the interrupt controller. This allows interrupt service routines to operate with interrupts enabled without being themselves interrupted by lower-priority interrupts. Since interrupts are enabled, higher-priority interrupts will be serviced.

When a service routine is completed, the proper IS bit must be reset by writing the proper pattern to the EOI register. This is required to allow subsequent interrupts from this interrupt source and to allow servicing of lower-priority interrupts. An EOI command is issued at the end of the service routine just before the issuance of the return from interrupt instruction. If the fully nested structure has been upheld, the next highest-priority source with its IS bit set is then serviced.

Cascade Mode

The 80186 has four interrupt pins and two of them have dual functions. In the fully nested mode the four pins are used as direct interrupt inputs and the corresponding vectors are generated internally. In the cascade mode, the four pins are configured into interrupt input-dedicated acknowledge signal pairs. The interconnection is shown in Figure 22. INT0 is an interrupt input interfaced to an 8259A, while INT2/$\overline{\text{INTA0}}$ serves as the dedicated interrupt acknowledge signal to that peripheral. The same is true for INT1 and INT3/$\overline{\text{INTA1}}$. Each pair can selectively be placed in the cascade or non-cascade mode by programming the proper value into INT0 and INT1 control registers. The use of the dedicated acknowledge signals eliminates the need for the use of external logic to generate $\overline{\text{INTA}}$ and device select signals.

The primary cascade mode allows the capability to serve up to 128 external interrupt sources through the use of external master and slave 8259As. Three levels of priority are created, requiring priority resolution in the 80186 interrupt controller, the master 8259As, and the slave 8259As. If an external interrupt is serviced, one IS bit is set at each of these levels. When the interrupt service routine is completed, up to three end-of-interrupt commands must be issued by the programmer.

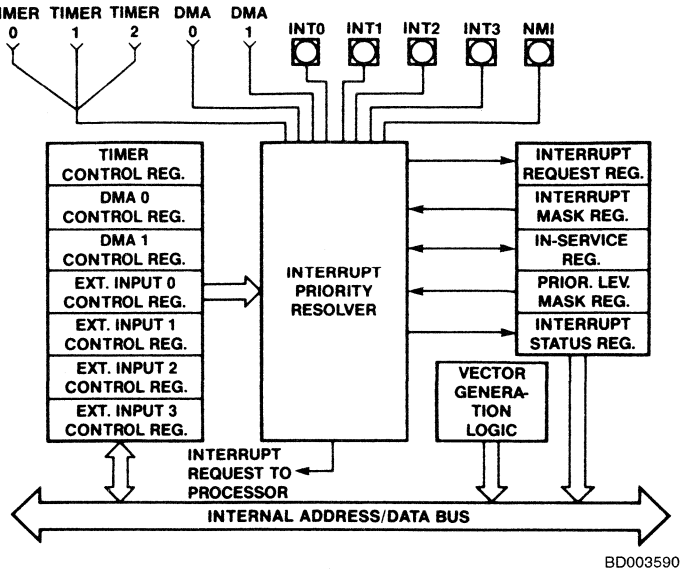


Figure 21. Interrupt Controller Block Diagram

Special Fully Nested Mode

This mode is entered by setting the SFNM bit in INT0 or INT1 control register. It enables complete nestability with external 8259A masters. Normally, an interrupt request from an interrupt source will not be recognized unless the in-service bit for that source is reset. If more than one interrupt source is connected to an external interrupt controller, all of the interrupts will be funneled through the same 80186 interrupt request pin. As a result, if the external interrupt controller receives a higher-priority interrupt, its interrupt will not be recognized by the 80186 controller until the 80186 in-service bit is reset. In special fully nested mode, the 80186 interrupt controller will allow interrupts from an external pin regardless of the state of the in-service bit for an interrupt source in order to allow multiple interrupts from a single pin. An in-service bit will continue to be set, however, to inhibit interrupts from other lower-priority 80186 interrupt sources.

Special procedures should be followed when resetting IS bits at the end of interrupt service routines. Software polling of the external master's IS register is required to determine if there is more than one bit set. If so, the IS bit in the 80186 remains active and the next interrupt service routine is entered.

Operation in a Polled Environment

The controller may be used in a polled mode if interrupts are undesirable. When polling, the processor disables interrupts and then polls the interrupt controller whenever it is convenient. Polling the interrupt controller is accomplished by reading the Poll Word (Figure 9). bit 15 in the poll word indicates to the processor that an interrupt of high enough priority is requesting service. Reading the Poll Word causes the In-Service bit of the highest-priority source to be set.

It is desirable to be able to read the Poll Word information without guaranteeing service of any pending interrupt, i.e., not set the indicated in-service bit. The 80186 provides a Poll Status Word in addition to the conventional Poll Word to allow this to be done. Poll Word information is duplicated in the Poll Status Word, but reading the Poll Status Word does not set the associated in-service bit. These words are located in two adjacent memory locations in the register file.

Master Mode Features

Programmable Priority

The user can program the interrupt sources into any of eight different priority levels. The programming is done by placing a 3-bit priority level (0–7) in the control register of each interrupt source. (A source with a priority level of 4 has higher priority over all priority levels from 5 to 7. Priority registers containing values lower than 4 have greater priority.) All interrupt sources have preprogrammed default priority levels (see Table 4).

If two requests with the same programmed priority level are pending at once, the priority ordering scheme shown in Table 4 is used. If the serviced interrupt routine reenables interrupts, it allows other requests to be serviced.

End-of-Interrupt Command

The end-of-interrupt (EOI) command is used by the programmer to reset the In-Service (IS) bit when an interrupt service routine is completed. The EOI command is issued by writing the proper pattern to the EOI register. There are two types of EOI commands, specific and nonspecific. The nonspecific command does not specify which IS bit is reset. When issued, the interrupt controller automatically resets the IS bit of the highest priority source with an active service routine. A specific EOI command requires that the programmer send the interrupt vector type to the interrupt controller indicating which source's IS bit is to be reset. This command is used when the fully nested structure has been disturbed or the highest priority IS bit that was set does not belong to the service routine in progress.

Trigger Mode

The four external interrupt pins can be programmed in either edge- or level-trigger mode. The control register for each external source has a level-trigger mode (LTM) bit. All interrupt inputs are active HIGH. In the edge sense mode or the level-trigger mode the interrupt request must remain active (HIGH) until the interrupt request is acknowledged by the 80186 CPU. In the edge-sense mode, if the level remains high after the interrupt is acknowledged, the input is disabled and no further requests will be generated. The input level must go LOW for at least one clock cycle to reenable the input. In the level-trigger mode, no such provision is made: holding the interrupt input HIGH will cause continuous interrupt requests.

Interrupt Vectoring

The 80186 Interrupt Controller will generate interrupt vectors for the integrated DMA channels and the integrated Timers. In addition, the Interrupt Controller will generate interrupt vectors for the external interrupt lines if they are not configured in Cascade or Special Fully Nested Mode. The interrupt vectors generated are fixed and cannot be changed (see Table 4).

Interrupt Controller Registers

The Interrupt Controller register model is shown in Figure 23. It contains 15 registers. All registers can both be read or written unless specified otherwise.

In-Service Register

This register can be read from or written into. The format is shown in Figure 24. It contains the In-Service bit for each of the interrupt sources. The In-Service bit for each of the interrupt sources. The In-Service bit is set to indicate that a source's service routine is in progress. When an In-Service bit is set, the interrupt controller will not generate interrupts to the CPU when it receives interrupt requests from devices with a lower programmed priority level. The TMR bit is the In-Service bit for all three timers; the D0 and D1 bits are the In-Service bits for the two DMA channels; the 10–13 are the In-Service bits for the external interrupt pins. The IS bit is set when the processor acknowledges an interrupt request either by an interrupt acknowledge or by reading the poll register. The IS bit is reset at the end of the interrupt service routine by an end-of-interrupt command issued by the CPU.

Interrupt Request Register

The internal interrupt sources have interrupt request bits inside the interrupt controller. The format of this register is shown in Figure 24. A read from this register yields the status of these bits. The TMR bit is the logical OR of all timer interrupt requests. D0 and D1 are the interrupt request bits for the DMA channels.

The state of the external interrupt input pins is also indicated. The state of the external interrupt pins is not a stored condition inside the interrupt controller, therefore the external interrupt bits cannot be written. The external interrupt request bits show exactly when an interrupt request is given to the interrupt controller, so if edge-triggered mode is selected, the bit in the register will be HIGH only after an inactive-to-active transition. For internal interrupt sources, the register bits are set when a request arrives and are reset when the processor acknowledges the requests.

Mask Register

This is a 16-bit register that contains a mask bit for each interrupt source. The format for this register is shown in Figure 24. A one in a bit position corresponding to a particular source serves to mask the source from generating interrupts. These mask bits are the exact same bits which are used in the individual control registers; programming a mask bit using the mask register will also change this bit in the individual control registers, and vice versa.

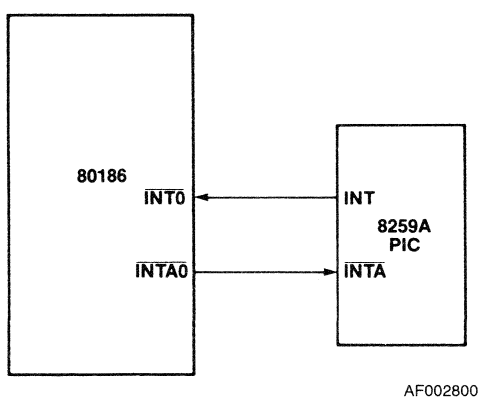


Figure 22. Cascade Mode Interrupt Connection

	OFFSET
INT3 CONTROL REGISTER	3EH
INT2 CONTROL REGISTER	3CH
INT1 CONTROL REGISTER	3AH
INT0 CONTROL REGISTER	38H
DMA 1 CONTROL REGISTER	36H
DMA 0 CONTROL REGISTER	34H
TIMER CONTROL REGISTER	32H
INTERRUPT CONTROLLER STATUS REGISTER	30H
INTERRUPT REQUEST REGISTER	2EH
IN-SERVICE REGISTER	2CH
PRIORITY MASK REGISTER	2AH
MASK REGISTER	28H
POLL STATUS REGISTER	26H
POLL REGISTER	24H
EOI REGISTER	22H

Figure 23. Interrupt Controller Registers (Non-IRMX 86 Mode)

Priority Mask Register

This register is used to mask all interrupts below particular interrupt priority levels. The format of this register is shown in Figure 25. The code in the lower three bits of this register inhibits interrupts of priority lower (a higher priority number) than the code specified. For example, 100 written into this register masks interrupts of level five (101), six (110), and seven (111). The register is reset to seven (111) upon RESET so all interrupts are unmasked.

Interrupt Status Register

This register contains general interrupt controller status information. The format of this register is shown in Figure 26. The bits in the status register have the following functions:

DHLT: DMA Halt Transfer; setting this bit halts all DMA transfers. It is automatically set whenever a non-maskable interrupt occurs, and it is reset when an IRET instruction is executed. The purpose of this bit is to allow prompt service of all non-maskable interrupts. This bit may also be set by the CPU.

IRTx: These three bits represent the individual timer interrupt request bits. These bits are used to differentiate the timer interrupts, since the timer IR bit in the interrupt request register is the "OR" function of all timer interrupt requests. Note that setting any one of these three bits initiates an interrupt request to the interrupt controller.

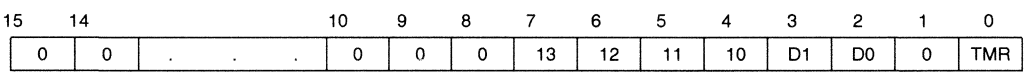


Figure 24. In-Service, Interrupt Request, and Mask Register Formats

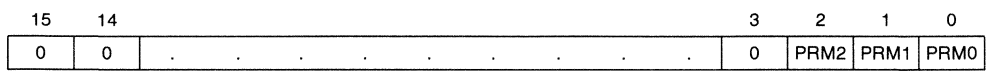


Figure 25. Priority Mask Register Format

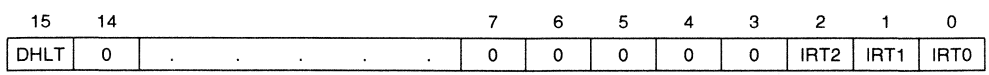


Figure 26. Interrupt Status Register Format

Timer, DMA 0, 1; Control Registers

These registers are the control words for all the internal interrupt sources. The format for these registers is shown in Figure 27. The three bit positions PR0, PR1, and PR2 represent the programmable priority level of the interrupt source. The MSK bit inhibits interrupt requests from the interrupt source. The MSK bits in the individual control registers are the exact same bits as are in the Mask Register; modifying them in the individual control registers will also modify them in the Mask Register, and vice versa.

INT0–INT3 Control Registers

These registers are the control words for the four external input pins. Figure 28 shows the format of the INT0 and INT1 Control registers; Figure 29 shows the format of the INT2 and INT3 Control registers. In cascade mode or special fully nested mode, the control words for INT2 and INT3 are not used.

The bits in the various control registers are encoded as follows:

PRO-2: Priority programming information. Highest priority = 000, lowest priority = 111.

LTM: Level-trigger mode bit. 1 = level-triggered; 0 = edge-triggered. Interrupt Input levels are active high. In level-triggered mode, an interrupt is generated whenever the external line is high. In edge-triggered mode, an interrupt will be generated only when this level is preceded by an inactive-to-active transition on the line. In both cases, the level must remain active until the interrupt is acknowledged.

MSK: Mask bit, 1 = mask; 0 = nonmask.

C: Cascade mode bit, 1 = cascade; 0 = direct

SFNM: Special fully nested mode bit, 1 = SFNM; 0 = normal nested mode.

EOI Register

The end of the interrupt register is a command register which can only be written into. The format of this register is shown in Figure 30. It initiates an EOI command when written to by the 80186 CPU.

The bits in the EOI register are encoded as follows:

S_X: Encoded information that specifies an interrupt source vector type as shown in Table 4. For example, to reset the In-Service bit for DMA channel 0, these bits should be set to 01010, since the vector type for DMA channel 0 is 10. Note that to reset the single In-Service bit for any of the three timers, the vector type for timer 0(8) should be written in this register.

15	14	. . . . .	4	3	2	1	0
0	0	. . . . .	0	MSK	PR2	PR1	PR0

Figure 27. Timer/DMA Control Register Formats

15	14	. . . . .	7	6	5	4	3	2	1	0
0	0	. . . . .	0	SFNM	C	LTM	MSK	PR2	PR1	PR0

Figure 28. INT0/INT1 Control Register Formats

15	14	. . . . .	5	4	3	2	1	0
0	0	. . . . .	0	LTM	MSK	PR2	PR1	PR0

Figure 29. INT2/INT3 Control Register Formats

NSPEC/: A bit that determines the type of EOI command.

SPEC Nonspecific = 1, Specific = 0.

Poll and Poll Status Registers

These registers contain polling information. The format of these registers is shown in Figure 31. They can only be read. Reading the Poll register constitutes a software poll. This will set the IS bit of the highest priority pending interrupt. Reading the poll status register will not set the IS bit of the highest priority pending interrupt; only the status of pending interrupts will be provided.

Encoding of the Poll and Poll Status register bits are as follows:

S_X: Encoded information that indicates the vector type of the highest priority interrupting source. Valid only when INTREQ = 1.

INTREQ: This bit determines if an interrupt request is present. Interrupt Request = 1; no Interrupt Request = 0.

iRMX 86 COMPATIBILITY MODE

This mode allows iRMX 86–80186 compatibility. The interrupt model of iRMX 86 requires one master and multiple slave 8259As in cascaded fashion. When iRMX mode is used, the internal 80186 interrupt controller will be used as a slave controller to an external master interrupt controller. The internal 80186 resources will be monitored through the internal interrupt controller, while the external controller functions as the system master interrupt controller.

Upon reset, the 80186 interrupt controller will be in the non-iRMX 86 mode of operation. To set the controller in the iRMX 86 mode, bit 14 of the Relocation Register should be set.

Because of pin limitations caused by the need to interface to an external 8259A master, the internal interrupt controller will no longer accept external inputs. There are however, enough 80186 interrupt controller inputs (internally) to dedicate one to each timer. In this mode, each timer interrupt source has its own mask bit, IS bit, and control word.

The iRMX 86 operating system requires peripherals to be assigned fixed priority levels. This is incompatible with the normal operation of the 80186 interrupt controller. Therefore, the initialization software must program the proper priority levels for each source. The required priority levels for the internal interrupt sources in iRMX mode are shown in Table 16.

Table 16. Internal Source Priority Level

Priority Level	Interrupt Source
0	Timer 0
1	(reserved)
2	DMA 0
3	DMA 1
4	Timer 1
5	Timer 2

These level assignments must remain fixed in the iRMX 86 mode of operation.

iRMX 86 Mode External Interface

The configuration of the 80186 with respect to an external 8259A master is shown in Figure 32. The INT0 input is used as the 80186 CPU interrupt input. INT3 functions as an output to send the 80186 slave-interrupt-request to one of the 8 master-PIC-inputs.

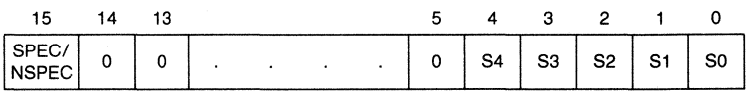


Figure 30. EOI Register Format

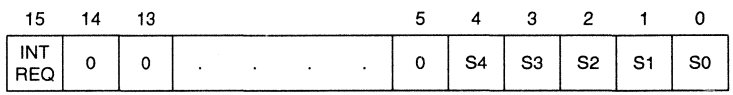


Figure 31. Poll Register Format

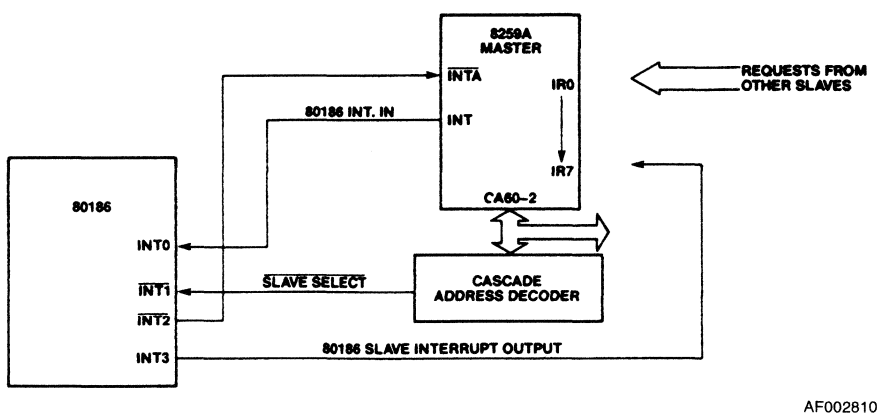


Figure 32. iRMX 86 Interrupt Controller Interconnection

Correct master-slave interface requires decoding of the slave addresses (CAS0–2). Slave 8259As do this internally. Because of pin limitations, the 80186 salve address will have to be decoded externally. $\overline{INT1}$ is used as a slave-select input. Note that the slave vector address is transferred internally, but the READY input must be supplied externally.

$\overline{INT2}$ is used as an acknowledge output, suitable to drive the $\overline{INTA}$ input of an 8259A.

Interrupt Nesting

iRMX 86 mode operation allows nesting of interrupt requests. When an interrupt is acknowledged, the priority logic masks off all priority levels except those with equal or higher priority.

Vector Generation in the iRMX 86 MODE

Vector generation in iRMX mode is exactly like that of an 8259A slave. The interrupt controller generates an 8-bit vector which the CPU multiplies by four and uses as an address into a vector table. The significant five bits of the vector are user-programmable while the lower three bits are generated by the priority logic. These bits represent the encoding of the priority level requesting service. The significant five bits of the vector are programmed by writing to the Interrupt Vector register at offset 20H.

Specific End-of-Interrupt

In iRMX mode the specific EOI command operates to reset an in-service bit of a specific priority. The user supplies a 3-bit priority-level value that points to an in-service bit to be reset. The command is executed by writing the correct value in the Specific EOI register at offset 22H.

Interrupt Controller Registers in the iRMX 86 Mode

All control and command registers are located inside the internal peripheral control block. Figure 33 shows the offsets of these registers.

End-of-Interrupt Register

The end-of-interrupt register is a command register which can only be written. The format of this register is shown in Figure 34. It initiates an EOI command when written by the 80186 CPU.

The bits in the EOI register are encoded as follows:

L_X: Encoded value indicating the priority of the IS bit to be reset.

In-Service Register

This register can be read from or written into. It contains the in-service bit for each of the internal sources. The format for this register is shown in Figure 35. Bit positions 2 and 3 correspond to the DMA channels; positions 0, 4, and 5 correspond to the integral timers. The source's IS bit is set when the processor acknowledges its interrupt request.

Interrupt Request Register

This register indicates which internal peripherals have interrupt requests pending. The format of this register is shown in Figure 35. The interrupt request bits are set when a request arrives from an internal source, and are rest when the processor acknowledges the request.

Mask Register

This register contains a mask bit for each interrupt source. The format for this register is shown in Figure 35. If the bit in this register corresponding to a particular interrupt source is set, any interrupts from that source will be masked. These mask bits are exactly the same bits which are used in the individual control registers, i.e., changing the state of a mask bit in this register will also change the state of the mask bit in the individual interrupt control register corresponding to the bit.

Control Registers

These registers are the control words for all the internal interrupt sources. The format of these registers is shown in Figure 36. Each of the timers and both of the DMA channels have their own Control Register.

The bits of the Control Registers are encoded as follows:

pr_X: 3-bit encoded field indicating a priority level for the source; note that each source must be programmed at specified levels.

msk: mask bit for the priority level indicated by pr_X bits.

	OFFSET
LEVEL 5 CONTROL REGISTER (TIMER 2)	3AH
LEVEL 4 CONTROL REGISTER (TIMER 1)	38H
LEVEL 3 CONTROL REGISTER (DMA 1)	36H
LEVEL 2 CONTROL REGISTER (DMA 0)	34H
LEVEL 0 CONTROL REGISTER (TIMER 0)	32H
INTERRUPT-REQUEST REGISTER	2EH
IN-SERVICE REGISTER	2CH
PRIORITY-LEVEL MASK REGISTER	2AH
MASK REGISTER	28H
SPECIFIC EOI REGISTER	22H
INTERRUPT VECTOR REGISTER	20H

Figure 33. Interrupt Controller Registers (iRMX86 Mode)

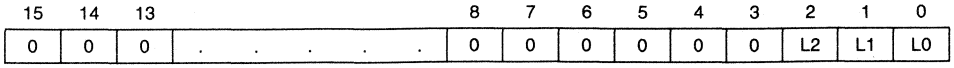


Figure 34. Specific EQI Register Format

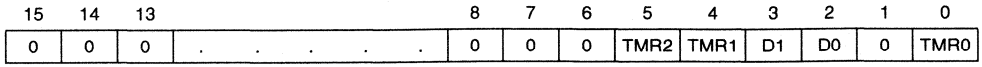


Figure 35. In-Service, Interrupt Request, and Mask Register Format

Interrupt Vector Register

This register provides the upper five bits of the interrupt vector address. The format of this register is shown in Figure 37. The interrupt controller itself provides the lower three bits of the interrupt vector as determined by the priority level of the interrupt request.

The format of the bits in this register is:

t_x: 5-bit field indicating the upper five bits of the vector address.

Priority-Level Mask Register

This register indicates the lowest priority-level interrupt which will be serviced.

The encoding of the bits in this register is:

m_x: 3-bit encoded field indication priority-level value. All levels of lower priority will be masked.

Interrupt Controller and Reset

Upon RESET, the interrupt controller will perform the following actions:

- All SFNM bits reset to 0, implying Fully Nested Mode.
- All PR bits in the various control registers set to 1. Tis places all sources at lowest priority (level 111).
- All LTM bits reset to 0, resulting in edge-sense mode.
- All Interrupt Service bits reset to 0.
- All Interrupt Request bits reset to 0.
- All MSK (Interrupt Mask) bits set to 1 (mask).
- All C (Cascade) bits reset to 0 (non-cascade).
- All PRM (Priority Mask) bits set to 1, implying no levels masked.
- Initialized to non-iRMX 86 mode.

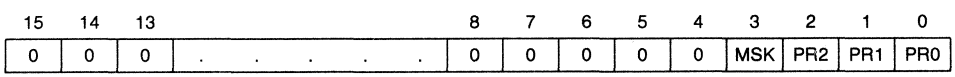


Figure 36. Control Word Format

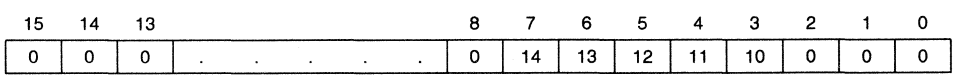


Figure 37. Interrupt Vector Register Format

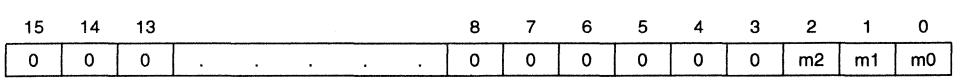


Figure 38. Priority Level Mask Register

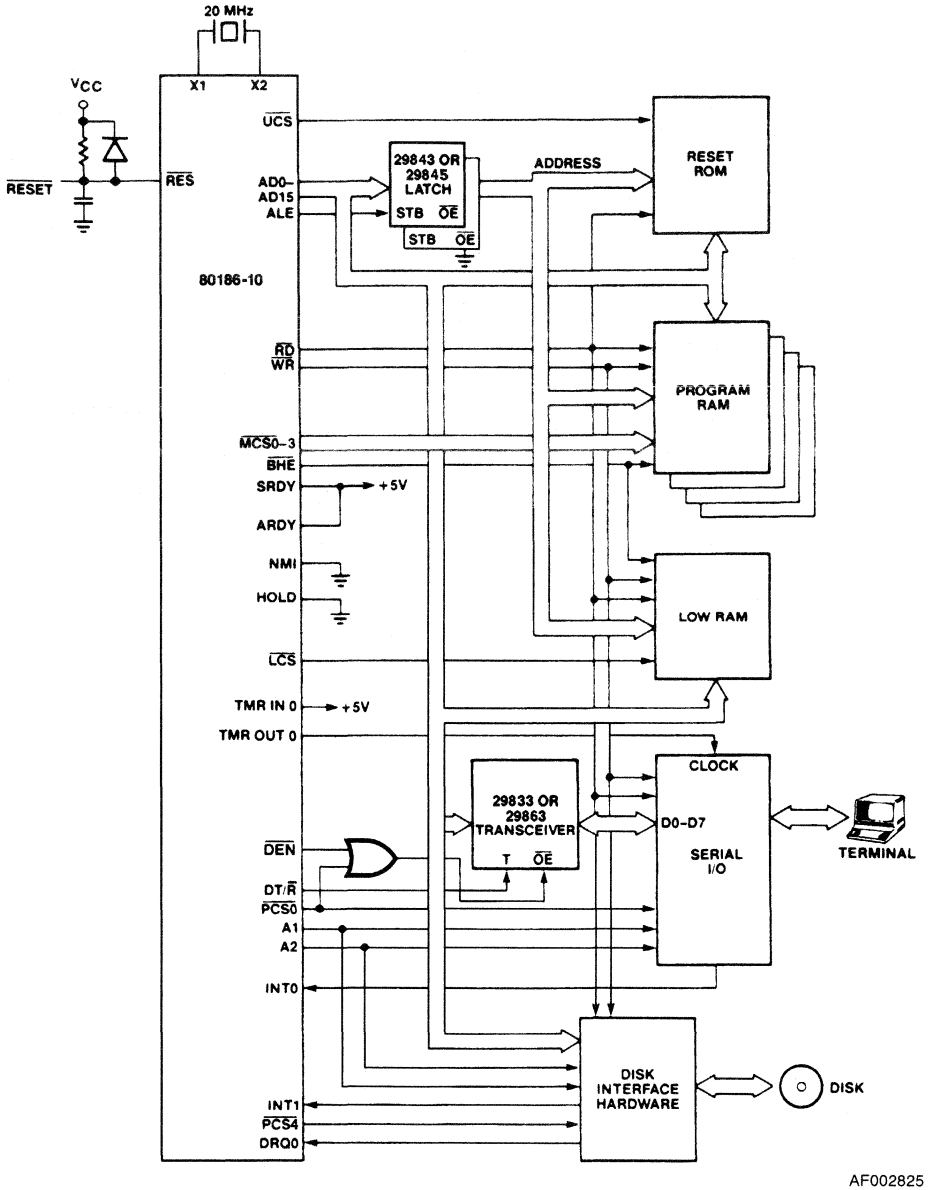


Figure 39. Typical 80186 Computer

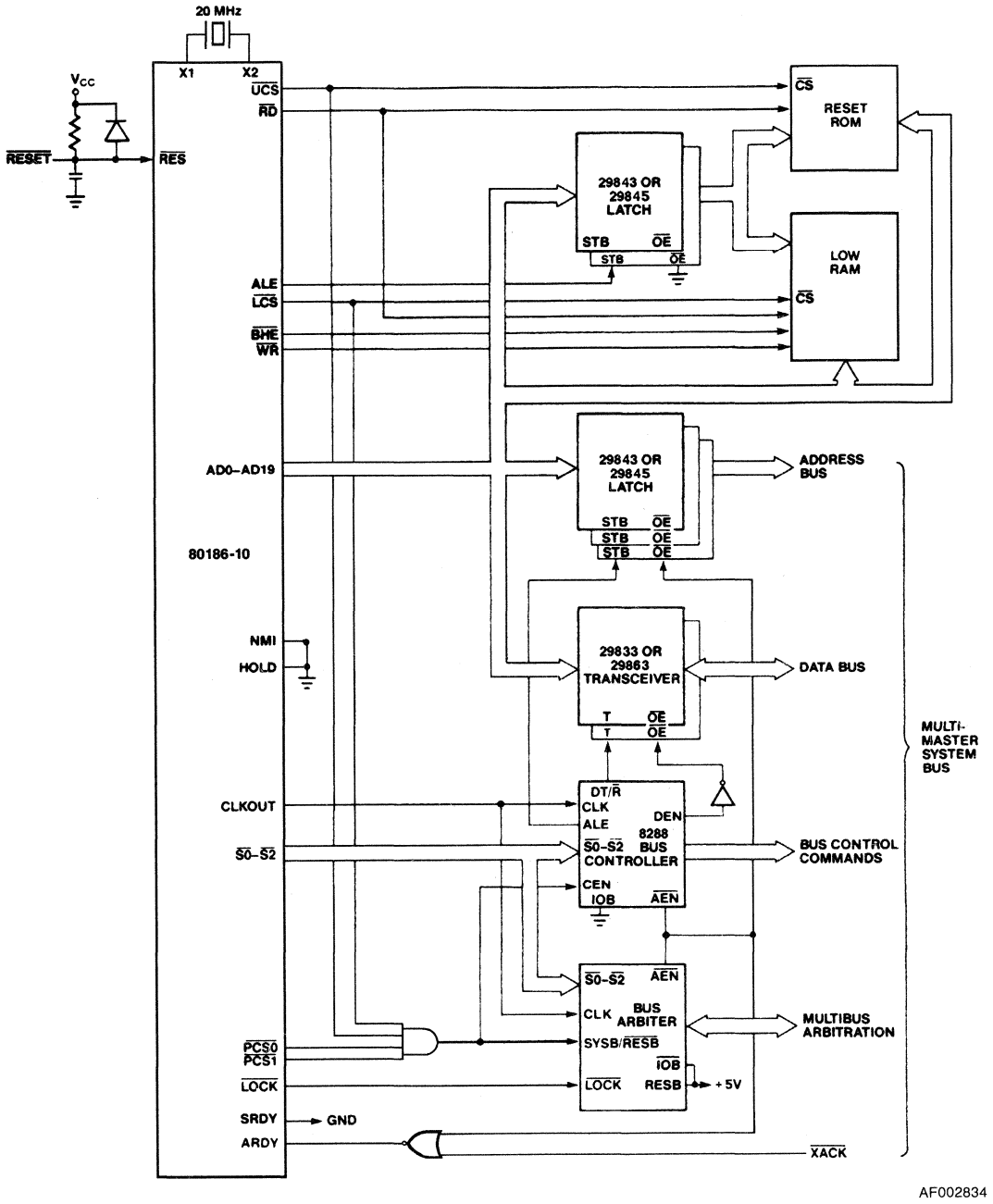


Figure 40. Typical 80186 Multi-Master Bus Interface

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground..... −1.0V to +7V
Power Dissipation (Steady State 70°C) ..... 2 Watt

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A)..... 0 to +70°C
(T_C).....0 to +110°C
Supply Voltage (V_{CC}) .....5 V ±10%

Industrial (I) Devices
Temperature (T_A).....−40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	+0.8	Volts
V_{IH}	Input High Voltage (All except X1 and $\overline{RES}$)		2.0	V_{CC} + 0.5	Volts
V_{IH1}	Input High Voltage ($\overline{RES}$)		3.0	V_{CC} + 0.5	Volts
V_{OL}	Output Low Voltage	I_A = 2.5mA for $\overline{S0}$ − $\overline{S2}$ I_A = 2.0mA for all other outputs		0.45	Volts
V_{OH}	Output High Voltage	I_{OA} = −400μA	2.4		Volts
I_{CC}	Power Supply Current	T_A = −40°C		600**	mA
		T_A = 0°C		500	
		T_A = 70°C		375	
I_{LI}	Input Leakage Current	0V < V_{IN} < V_{CC}		±10	μA
I_{LO}	Output Leakage Current	0.45V < V_{OUT} < V_{CC}		±10	μA
V_{CLO}	Clock Output Low	I_A = 4.0mA		0.6	Volts
V_{CHO}	Clock Output High	I_{OA} = −200μA	4.0		Volts
V_{CLI}	Clock Input Low Voltage		−0.5	0.6	Volts
V_{CHI}	Clock Input High Voltage		3.9	V_{CC} + 1.0	Volts
C_{IN}	Input Capacitance			10	pF
C_{IO}	I/O Capacitance			20	pF

SWITCHING CHARACTERISTICS

PIN TIMING

80186 Timing Requirements All Timings Measured At 1.5 Volts Unless Otherwise Noted

			80186-10 (10MHz)		80186 (8MHz)		
Parameters	**Description**	**Test Conditions**	**Min**	**Max**	**Min**	**Max**	**Units**
TDVCL	Data in Setup (A/D)		15		20		ns
TCLDX	Data in Hold (A/D)		8		10		ns
TARYHCH	Asynchronous Ready (AREADY) Active Setup Time*		15		20		ns
TARYLCL	AREADY Inactive Setup Time		25		35		ns
TCHARYX	AREADY Hold Time		15		15		ns
TARYCHL	AREADY Inactive Hold Time		15		15		ns
TSRYCL	Synchronous Ready (SREADY) Transition Setup Time		20		20		ns
TCLSRY	SREADY Transition Hold Time		15		15		ns
THVCL	HOLD Setup*		20		25		ns
TINVCH	INTR, NMI, $\overline{TEST}$, TIMERIN, Setup*		25		25		ns
TINVCL	DRQ0, DRQ1, Setup*		20		25		ns

*To guarantee recognition at next clock.
**For Industrial Grade Parts only.

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS (Cont'd.)

80186 Master Interface Timing Responses

Parameters	Description	Test Conditions	80186-10 (10MHz)		80186 (8MHz)		Units
			Min	Max	Min	Max	
TCLAV	Address Valid Delay	C_L = 20 – 200 pF all outputs except TCLTMV	5	44	5	55	ns
TCLAX	Address Hold		10		10		ns
TCLAZ	Address Float Delay		TCLAX	30	TCLAX	35	ns
TCHCZ	Command Lines Float Delay			40		45	ns
TCHCV	Command Lines Valid Delay (after float)			45		55	ns
TLHLL	ALE Width		TCLCL-30		TCLCL-35		ns
TCHLH	ALE Active Delay			30		35	ns
TCHLL	ALE Inactive Delay			30		35	ns
TLLAX	Address Hold to ALE Inactive		TCHCL-20		TCHCL-25		ns
TCLDV	Data Valid Delay		10	40	10	44	ns
TCLDOX	Data Hold Time		10		10		ns
TWHDX	Data Hold after WR		TCLCL-34		TCLCL-40		ns
TCVCTV	Control Active Delay1		5	56	10	70	ns
TCHCTV	Control Active Delay2		10	44	10	55	ns
TCVCTX	Control Inactive Delay		5	44	5	55	ns
TCVDEX	$\overline{DEN}$ Inactive Delay (Non-Write Cycle)		10	56	10	70	ns
TAZRL	Address Float to $\overline{RD}$ Active		0		0		ns
TCLRL	$\overline{RD}$ Active Delay		10	40	5	50	ns
TCLRH	$\overline{RD}$ Inactive Delay		10	44	10	55	ns
TRHAV	$\overline{RD}$ Inactive to Address Active		TCLCL-40		TCLCL-40		ns
TCLHAV	HLDA Valid Delay		5	40	5	50	ns
TRLRH	$\overline{RD}$ Width		2TCLCL-46		2TCLCL-50		ns
TWLWH	$\overline{WR}$ Width		2TCLCL-34		2TCLCL-40		ns
TAVAL	Address Valid to ALE Low		TCLCH-19		TCLCH-25		ns
TCHSV	Status Active Delay		10	45	10	55	ns
TCLSH	Status Inactive Delay		10	50	10	65	ns
TCLTMV	Timer Output Delay	100 pF Max.		48		60	ns
TCLRO	Reset Delay			48		60	ns
TCHQSV	Queue Status Delay			28		35	ns
TCHDX	Status Hold Time		10		10		ns
TAVCH	Address Valid to Clock High		10		10		ns
TCLLV	$\overline{LOCK}$ Valid/Invalid Deloay		5	60	5	65	ns

80186 Chip-Select Timing Responses

Parameters	Description	Test Conditions	80186-10 (10MHz)		80186 (8MHz)		Units
			Min	Max	Min	Max	
TCLCSV	Chip-Select Active Delay			45		66	ns
TCXCSX	Chip-Select Hold from Command Inactive		35		35		ns
TCHCSX	Chip-Select Inactive Delay		5	32	5	35	ns

SWITCHING CHARACTERISTICS (Cont'd.)

80186 CLKIN Requirements

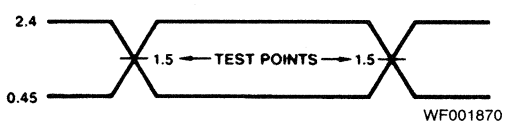
Parameters	Description	Test Conditions	80186-10 (10MHz)		80186 (8MHz)		Units
			Min	Max	Min	Max	
TCKIN	CLKIN Period		50	250	62.5	250	ns
TCKHL	CLKIN Fall Time	3.5 to 1.0 volts		10		10	ns
TCKLH	CLKIN Rise Time	1.0 to 3.5 volts		10		10	ns
TCLCK	CLKIN Low Time	1.5 volts	20		25		ns
TCHCK	CLKIN High Time	1.5 volts	20		25		ns

80186 CLKOUT Timing (200 pF load)

Parameters	Description	Test Conditions	Min	Max	Min	Max	Units
TCICO	CLKIN to CLKOUT Skew			25		50	ns
TCLCL	CLKOUT Period		100	500	125	500	ns
TCLCH	CLKOUT Low Time	1.5 volts	½TCLCL-6.0		½TCLCL-7.5		ns
TCHCL	CLKOUT High Time	1.5 volts	½TCLCL-6.0		½TCLCL-7.5		ns
TCH1CH2	CLKOUT Rise Time	1.0 to 3.5 volts		12		15	ns
TCL2CL1	CLKOUT Fall Time	3.5 to 1 volts		12		15	ns

All timings measured at 1.5 volts unless otherwise noted.

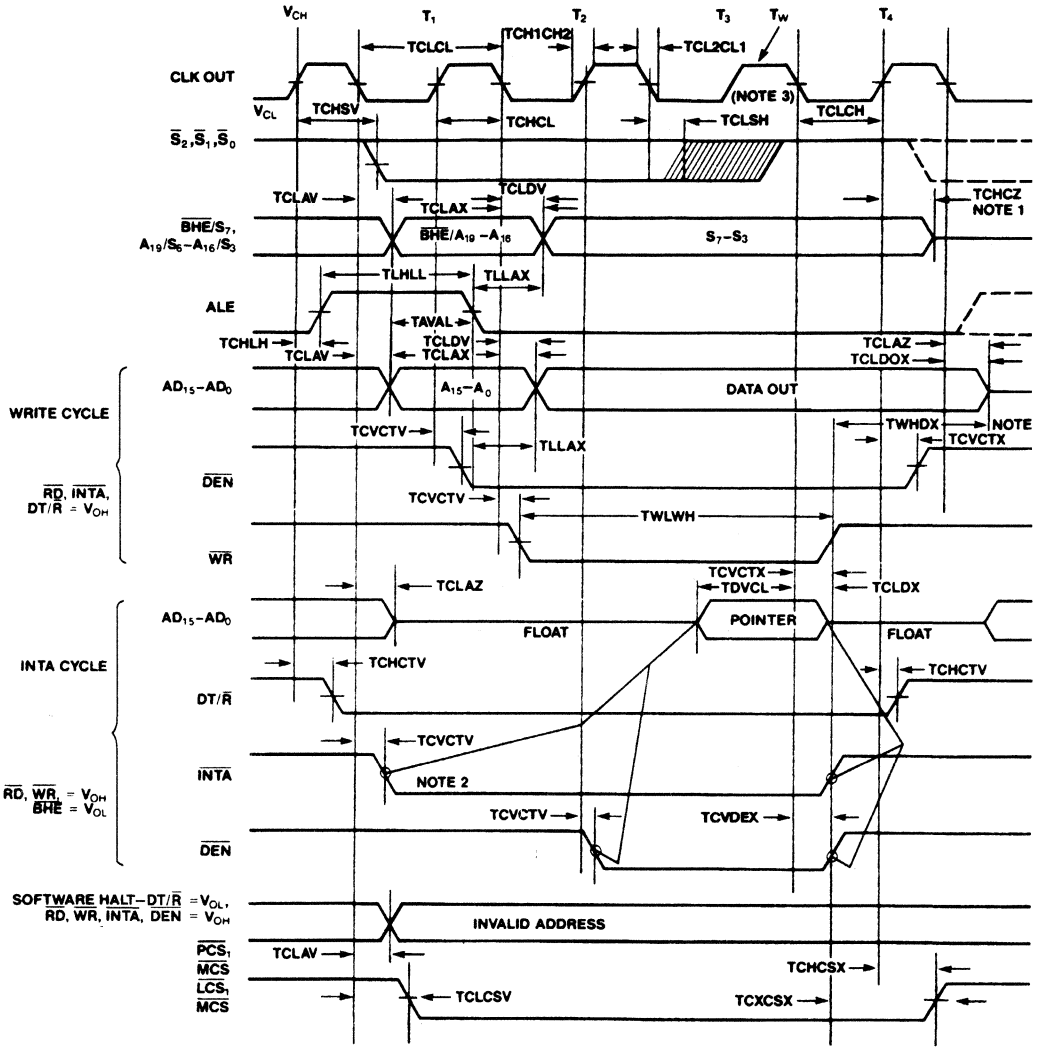
SWITCHING TEST INPUT/OUTPUT WAVEFORM



AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0." The clock is driven at 4.3V and 0.25V. Timing measurements are made at 1.5V for both a logic "1" and "0."

SWITCHING WAVEFORMS

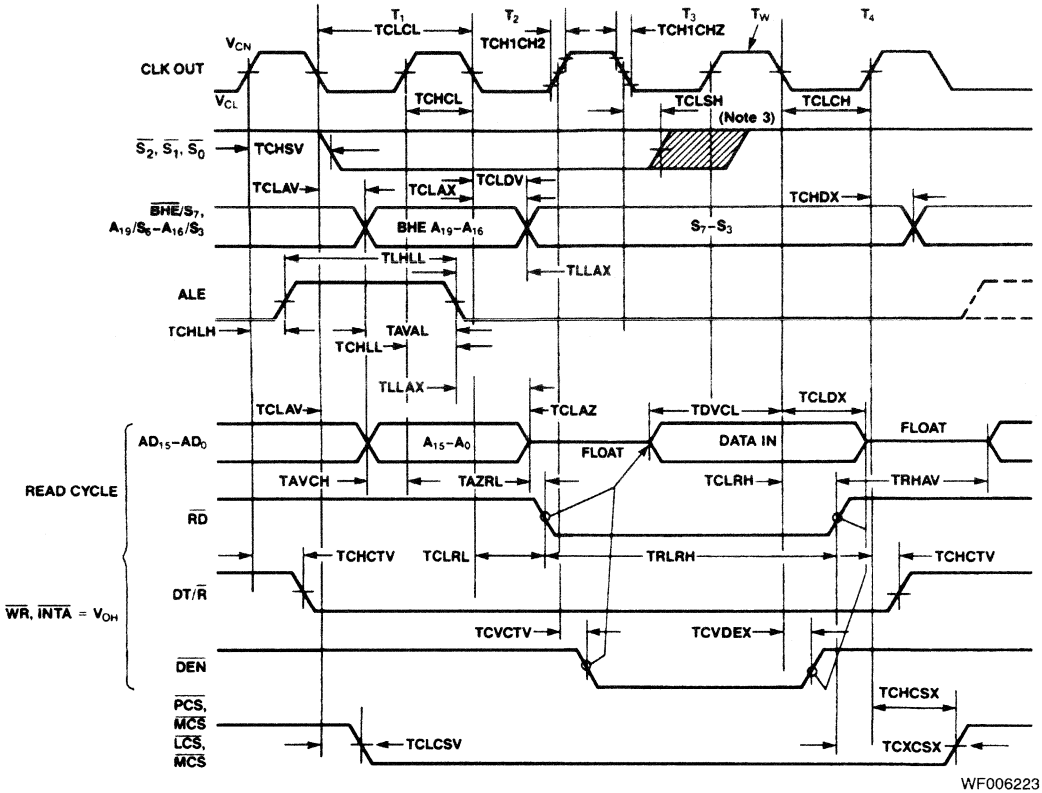
MAJOR CYCLE TIMING



WF006212

SWITCHING WAVEFORMS (Cont'd.)

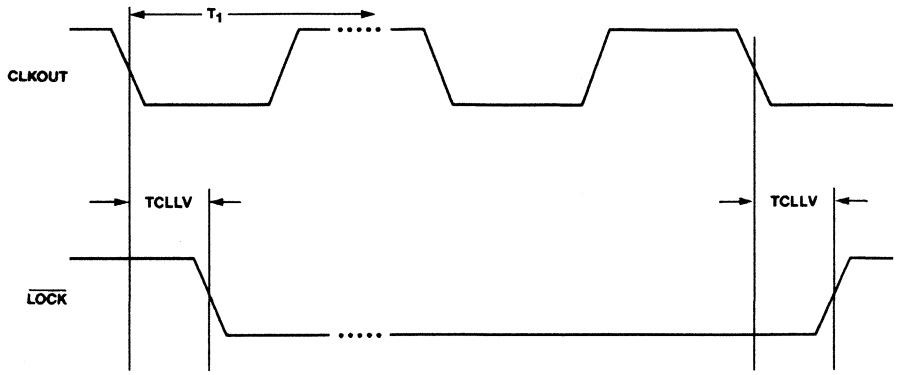
MAJOR CYCLE TIMING (Cont'd.)



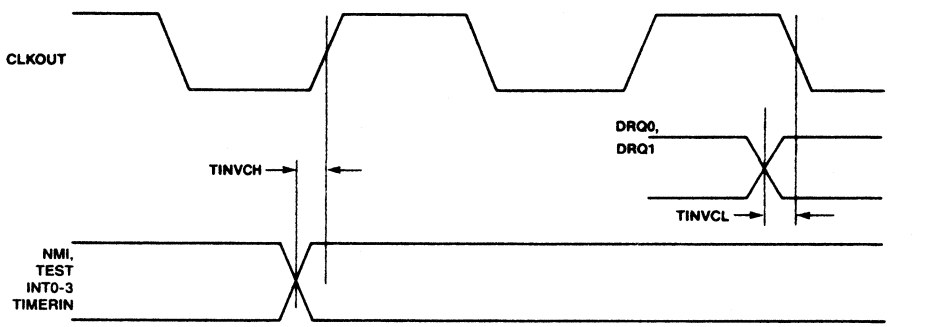
WF006223

Notes:
1. Following a Write cycle, the Local Bus is floated by the 80186 only when the 80186 enters a "Hold Acknowledge" state.
2. INTA occurs one clock later in RMX-mode.
3. Status inactive just prior to T_4.

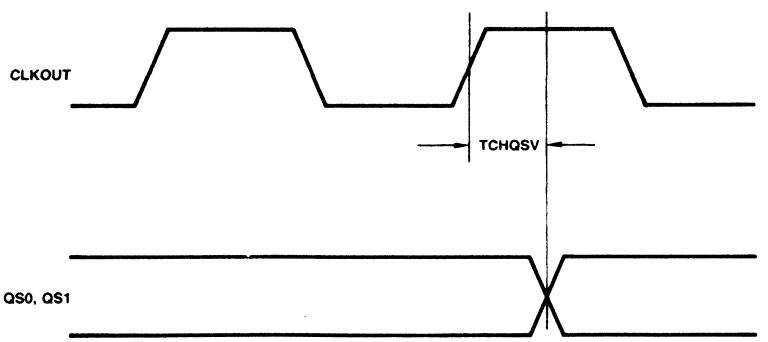
SWITCHING WAVEFORMS (Cont'd.)



WF006231

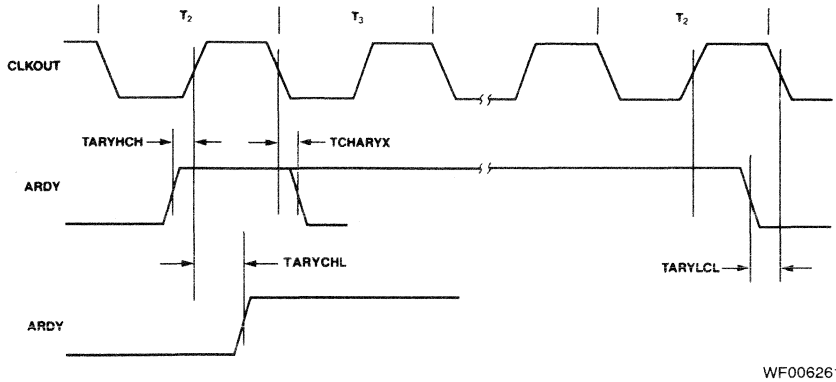


WF006240

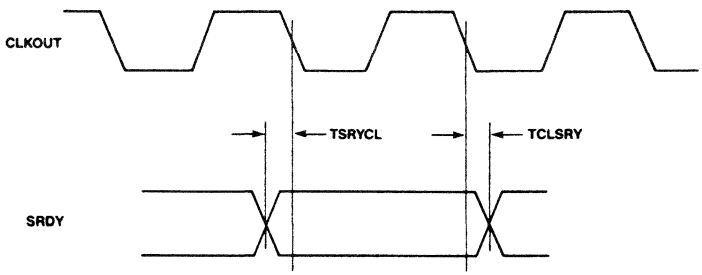


WF006250

SWITCHING WAVEFORMS (Cont'd.)

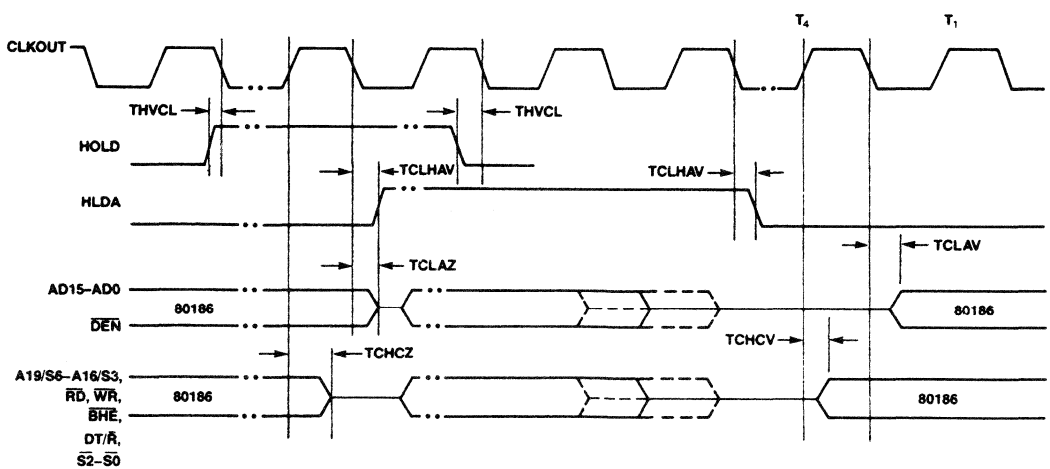


WF006261



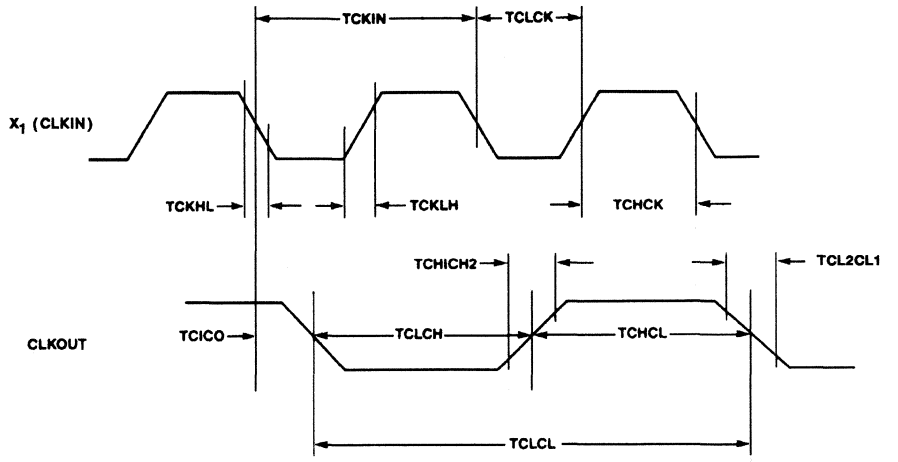
WF006270

HOLD-HLDA TIMING



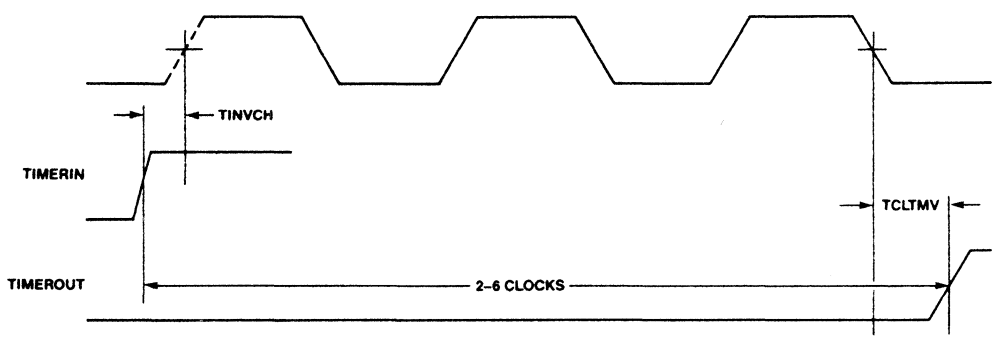
WF006280

SWITCHING WAVEFORMS (Cont'd.)



WF006292

TIMER ON 80186



WF006300

80186 INSTRUCTION TIMINGS

The following instruction timings represent the minimum execution time in clock cycles for each instruction. The timings given are based on the following assumptions:

- The opcode, along with any data or displacement required for execution of a particular instruction, has been pre-fetched and resides in the queue at the time it is needed.
- No wait states or bus HOLDS occur.
- All word-data is located on even-address boundaries.

All jumps and calls include the time required to fetch the opcode of the next instruction at the destination address.

All instructions which involve memory reference can require one (and in some cases, two) additional clocks above the minimum timings shown. This is due to the asynchronous nature of the handshake between the BIU and the Execution unit.

INSTRUCTION SET SUMMARY

FUNCTION	FORMAT	Clock Cycles	Comments
DATA TRANSFER			
MOV = Move:			
Register to Register/Memory	1 0 0 0 1 0 0 w | mod reg r/m	2/12	
Register/memory to register	1 0 0 0 1 0 1 w | mod reg r/m	2/9	
Immediate to register/memory	1 1 0 0 0 1 1 w | mod 0 0 0 r/m | data | data if w = 1	12 – 13	8/16-bit
Immediate to register	1 0 1 1 w reg | data | data if w = 1	3 – 4	8/16-bit
Memory to accumulator	1 0 1 0 0 0 0 w | addr-low | addr-high	9	
Accumulator to memory	1 0 1 0 0 0 1 w | addr-low | addr-high	8	
Register/memory to segment register	1 0 0 0 1 1 1 0 | mod 0 reg r/m	2/9	
Segment register to register/memory	1 0 0 0 1 1 0 0 | mod 0 reg r/m	2/11	
PUSH = Push:			
Memory	1 1 1 1 1 1 1 1 | mod 1 1 0 r/m	16	
Register	0 1 0 1 0 reg	10	
Segment register	0 0 0 reg 1 1 0	9	
Immediate	0 1 1 0 1 0 s 1 | data | data if s = 0	10	
PUSHA = Push All	0 1 1 0 0 0 0 0	36	
POP = Pop:			
Memory	1 0 0 0 1 1 1 1 | mod 0 0 0 r/m	20	
Register	0 1 0 1 1 reg	10	
Segment register	0 0 0 reg 1 1 1 (reg ≠ 01)	8	
POPA = Pop All	0 1 1 0 0 0 0 1	51	
XCHG = Exchange:			
Register/memory with register	1 0 0 0 0 1 1 w | mod reg r/m	4/17	
Register with accumulator	1 0 0 1 0 reg	3	
IN = Input from:			
Fixed port	1 1 1 0 0 1 0 w | port	10	
Variable port	1 1 1 0 1 1 0 w	8	
OUT = Output to:			
Fixed port	1 1 1 0 0 1 1 w | port	9	
Variable port	1 1 1 0 1 1 1 w	7	
XLAT = Translate byte to AL	1 1 0 1 0 1 1 1	11	
LEA = Load EA to register	1 0 0 0 1 1 0 1 | mod reg r/m	6	
LDS = Load pointer to DS	1 1 0 0 0 1 0 1 | mod reg r/m (mod ≠ 11)	18	
LES = Load pointer to ES	1 1 0 0 0 1 0 0 | mod reg r/m (mod ≠ 11)	18	
LAHF = Load AH with flags	1 0 0 1 1 1 1 1	2	
SAHF = Store AH into flags	1 0 0 1 1 1 1 0	3	
PUSHF = Push flags	1 0 0 1 1 1 0 0	9	
POPF = Pop flags	1 0 0 1 1 1 0 1	8	

Shaded areas indicate new 80186 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Cont'd.)

FUNCTION	FORMAT	Clock Cycles	Comments
ARITHMETIC			
ADD = Add:			
Reg/memory with register to either	0 0 0 0 0 0 d w | mod reg r/m	3/10	
Immediate to register / memory	1 0 0 0 0 0 s w | mod 0 0 0 r/m | data | data if s w = 01	4/16	
Immediate to accumulator	0 0 0 0 0 1 0 w | data | data if w = 1	3/4	8/16-bit
ADC = Add with carry:			
Reg/memory with register to either	0 0 0 1 0 0 d w | mod reg r/m	3/10	
Immediate to register/memory	1 0 0 0 0 0 s w | mod 0 1 0 r/m | data | data if s w = 01	4/16	
Immediate to accumulator	0 0 0 1 0 1 0 w | data | data if w = 1	3/4	8/16-bit
INC = Increment:			
Register/memory	1 1 1 1 1 1 1 w | mod 0 0 0 r/m	3/15	
Register	0 1 0 0 0 reg	3	
SUB = Subtract:			
Reg/memory and register to either	0 0 1 0 1 0 d w | mod reg r/m	3/10	
Immediate from register/memory	1 0 0 0 0 0 s w | mod 1 0 1 r/m | data | data if s w = 01	4/16	
Immediate from accumulator	0 0 1 0 1 1 0 w | data | data if w = 1	3/4	8/16-bit
SBB = Subtract with borrow:			
Reg/memory and register to either	0 0 0 1 1 0 d w | mod reg r/m	3/10	
Immediate from register/memory	1 0 0 0 0 0 s w | mod 0 1 1 r/m | data | data if s w = 01	4/16	
Immediate from accumulator	0 0 0 1 1 1 0 w | data | data if w = 1	3/4	8/16-bit
DEC = Decrement:			
Register/memory	1 1 1 1 1 1 1 w | mod 0 0 1 r/m	3/15	
Register	0 1 0 0 1 reg	3	
CMP = Compare:			
Register/memory with register	0 0 1 1 1 0 1 w | mod reg r/m	3/10	
Register with register/memory	0 0 1 1 1 0 0 w | mod reg r/m	3/10	
Immediate with register/memory	1 0 0 0 0 0 s w | mod 1 1 1 r/m | data | data if s w = 01	3/10	
Immediate with accumulator	0 0 1 1 1 1 0 w | data | data if w = 1	3/4	8/16-bit
NEG = Change sign	1 1 1 1 0 1 1 w | mod 0 1 1 r/m	3	
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1	8	
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1	4	
AAS = ASCII adjust for subtract	0 0 1 1 1 1 1 1	7	
DAS = Decimal adjust for subtract	0 0 1 0 1 1 1 1	4	
MUL = Mulitiply (unsigned)	1 1 1 1 0 1 1 w | mod 1 0 0 r/m		
Register-Byte		26 - 28	
Register-Word		35 - 37	
Memory-Byte		32 - 34	
Memory-Word		41 - 43	
IMUL = Integer multiply (signed):	1 1 1 1 0 1 1 w | mod 1 0 1 r/m		
Register-Byte		25 - 28	
Register-Word		34 - 37	
Memory-Byte		31 - 34	
Memory-Word		40 - 43	
IMUL = Integer immediate multiply (signed)	0 1 1 0 1 0 s 1 | mod reg r/m | data | data if s = 0	22 - 25/29 - 32	
DIV = Divide (unsigned):	1 1 1 1 0 1 1 w | mod 1 1 0 r/m		
Register-Byte		29	
Register-Word		38	
Memory-Byte		35	
Memory-Word		44	

Shaded areas indicate new 80186 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Cont'd.)

FUNCTION	FORMAT	Clock Cycles	Comments
ARITHMETIC (Continued):			
IDIV = Integer divide (signed):	1 1 1 1 0 1 1 w | mod 1 1 1 r/m		
Register-Byte		44 – 52	
Register-Word		53 – 61	
Memory-Byte		50 – 58	
Memory-Word		59 – 67	
AAM = ASCII adjust for multiply	1 1 0 1 0 1 0 0 | 0 0 0 0 1 0 1 0	19	
AAD = ASCII adjust for divide	1 1 0 1 0 1 0 1 | 0 0 0 0 1 0 1 0	15	
CBW = Convert byte to word	1 0 0 1 1 0 0 0	2	
CWD = Convert word to double word	1 0 0 1 1 0 0 1	4	
LOGIC **Shift/Rotate Instructions:**			
Register/Memory by 1	1 1 0 1 0 0 0 w | mod TTT r/m	2/15	
Register/Memory by CL	1 1 0 1 0 0 1 w | mod TTT r/m		
Register/Memory by Count	1 1 0 0 0 0 0 w | mod TTT r/m | count	5 + n/17 + n	

TTT	Instruction
0 0 0	ROL
0 0 1	ROR
0 1 0	RCL
0 1 1	RCR
1 0 0	SHL/SAL
1 0 1	SHR
1 1 1	SAR

AND = And:					
Reg/memory and register to either	0 0 1 0 0 0 d w	mod reg r/m		3/10	
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1	4/16	
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w = 1		3/4	8/16-bit
TEST = And function to flags, no result:					
Register/memory and register	1 0 0 0 0 1 0 w	mod reg r/m			3/10	
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1	4/10	
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w = 1		3/4	8/16-bit
OR = Or:					
Reg/memory and register to either	0 0 0 0 1 0 d w	mod reg r/m			3/10	
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w = 1	4/16	
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w = 1		3/4	8/16-bit
XOR = Exclusive or:					
Reg/memory and register to either	0 0 1 1 0 0 d w	mod reg r/m			3/10	
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1	4/16	
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w = 1		3/4	8/16-bit
NOT = Invert register/memory	1 1 1 1 0 1 1 w	mod 0 1 0 r/m			3	
STRING MANIPULATION:						
MOVS = Move byte/word	1 0 1 0 0 1 0 w				8 + 8n	
CMPS = Compare byte/word	1 0 1 0 0 1 1 w				5 + 22n	
SCAS = Scan byte/word	1 0 1 0 1 1 1 w				5 + 15n	
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w				6 + 11n	
STOS = Stor byte/wd from AL/A	1 0 1 0 1 0 1 w				6 + 9n	
INS = Input byte/wd from DX port	0 1 1 0 1 1 0 w				8	
OUTS = Output byte/wd to DX port	0 1 1 0 1 1 1 w				7	

Shaded areas indicate new 80186 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Cont'd.)

FUNCTION	FORMAT	Clock Cycles	Comments
STRING MANIPULATION (Continued):			
Repeated by count in CX			
MOVS = Move string	1 1 1 1 0 0 1 0 | 1 0 1 0 0 1 0 w	14	
CMPS = Compare string	1 1 1 1 0 0 1 z | 1 0 1 0 0 1 1 w	22	
SCAS = Scan string	1 1 1 1 0 0 1 z | 1 0 1 0 1 1 1 w	15	
LODS = Load string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 1 0 w	12	
STOS = Store string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 0 1 w	10	
INS = Input string	1 1 1 1 0 0 1 0 | 0 1 1 0 1 1 0 w	8 + 8n/14	Repeated/ Not Repeated
OUTS = Output string	1 1 1 1 0 0 1 0 | 0 1 1 0 1 1 1 w	8 + 8n/14	Repeated/ Not Repeated
CONTROL TRANSFER			
CALL = Call:			
Direct within segment	1 1 1 0 1 0 0 0 | disp-low | disp-high	14	
Register memory indirect within segment	1 1 1 1 1 1 1 1 | mod 0 1 0 r/m	13/19	
Direct intersegment	1 0 0 1 1 0 1 0 | segment offset segment selector	23	
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 0 1 1 r/m | (mod ≠ 11)	38	
JMP = Unconditional jump:			
Short/long	1 1 1 0 1 0 1 1 | disp-low	13	
Direct within segment	1 1 1 0 1 0 0 1 | disp-low | disp-high	13	
Register/memory indirect within segment	1 1 1 1 1 1 1 1 | mod 1 0 0 r/m	11/17	
Direct intersegment	1 1 1 0 1 0 1 0 | segment offset segment selector	13	
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 1 0 1 r/m | (mod ≠ 11)	26	
RET = Return from CALL:			
Within segment	1 1 0 0 0 0 1 1	16	
Within seg adding immed to SP	1 1 0 0 0 0 1 0 | data-low | data-high	18	
Intersegment	1 1 0 0 1 0 1 1	22	
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0 | data-low | data-high	25	

Shaded areas indicate new 80186 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Cont'd.)

FUNCTION	FORMAT	Clock Cycles	Comments
CONTROL TRANSFER (Continued):			
JE/JZ = Jump on equal zero	0 1 1 1 0 1 0 0 | disp	4/13	13 if JMP taken
JL/JNGE = Jump on less not greater or equal	0 1 1 1 1 1 0 0 | disp	4/13	4 if JMP not taken
JLE/JNG = Jump on less or equal not greater	0 1 1 1 1 1 1 0 | disp	4/13	
JB/JNAE = Jump on below not above or equal	0 1 1 1 0 0 1 0 | disp	4/13	
JBE/JNA = Jump on below or equal not above	0 1 1 1 0 1 1 0 | disp	4/13	
JP/JPE = Jump on parity parity even	0 1 1 1 1 0 1 0 | disp	4/13	
JO = Jump on overflow	0 1 1 1 0 0 0 0 | disp	4/13	
JS = Jump on sign	0 1 1 1 1 0 0 0 | disp	4/13	
JNE/JNZ = Jump on not equal not zero	0 1 1 1 0 1 0 1 | disp	4/13	
JNL/JGE = Jump on not less greater or equal	0 1 1 1 1 1 0 1 | disp	4/13	
JNLE/JG = Jump on not less or equal greater	0 1 1 1 1 1 1 1 | disp	4/13	
JNB/JAE = Jump on not below above or equal	0 1 1 1 0 0 1 1 | disp	4/13	
JNBE/JA = Jump on not below or equal above	0 1 1 1 0 1 1 1 | disp	4/13	
JNP/JPO = Jump on not par / par odd	0 1 1 1 1 0 1 1 | disp	4/13	
JNO = Jump on not overflow	0 1 1 1 0 0 0 1 | disp	4/13	
JNS = Jump on not sign	0 1 1 1 1 0 0 1 | disp	4/13	
LOOP = Loop CX times	1 1 1 0 0 0 1 0 | disp	5/15	
LOOPZ/LOOPE = Loop while zero equal	1 1 1 0 0 0 0 1 | disp	6/16	
LOOPNZ/LOOPNE = Loop while not zero equal	1 1 1 0 0 0 0 0 | disp	6/16	
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1 | disp	16 5	JMP taken/ JMP not taken
ENTER = Enter Procedure	1 1 0 0 1 0 0 0 | data-low | data-high | L		
L = 0		15	
L = 1		25	
L > 1		22 + 16(n - 1)	
LEAVE = Leave Procedure	1 1 0 0 1 0 0 1	8	
INT = Interrupt:			
Type specified	1 1 0 0 1 1 0 1 | type	47	
Type 3	1 1 0 0 1 1 0 0	45	if INT. taken/
INTO = Interrupt on overflow	1 1 0 0 1 1 1 0	48/4	if INT. not taken
IRET = Interrupt return	1 1 0 0 1 1 1 1	28	
BOUND = Detect value out of range	0 1 1 0 0 0 1 0 | mod reg r/m	33 - 35	

Shaded areas indicate new 80186 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Cont'd.)

FUNCTION	FORMAT	Clock Cycles	Comments
PROCESSOR CONTROL			
CLC = Clear carry	1 1 1 1 1 0 0 0	2	
CMC = Complement carry	1 1 1 1 0 1 0 1	2	
STC = Set carry	1 1 1 1 1 0 0 1	2	
CLD = Clear direction	1 1 1 1 1 1 0 0	2	
STD = Set direction	1 1 1 1 1 1 0 1	2	
CLI = Clear interrupt	1 1 1 1 1 0 1 0	2	
STI = Set interrupt	1 1 1 1 1 0 1 1	2	
HLT = Halt	1 1 1 1 0 1 0 0	2	
WAIT = Wait	1 0 0 1 1 0 1 1	6	if $\overline{\text{test}}$ = 0
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0	2	
ESC = Processor Extension Escape	1 0 0 1 1 T T T | mod LLL r/m (TTT LLL are opcode to processor extension)	6	

FOOTNOTES

The effective Address (EA) of the memory operand is computed according to the mod and r/m fields:

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

SEGMENT OVERRIDE PREFIX

0 0 1 reg 1 1 0

reg is assigned according to the following:

reg	Segment Register
00	ES
01	CS
10	SS
11	DS

REG is assigned according to the following table:

16-Bit (w = 1)	8-Bit (w = 0)
000 AX	000 AL
001 CX	001 CL
010 DX	010 DL
011 BX	011 BL
100 SP	100 AH
101 BP	101 CH
110 SI	110 DH
111 DI	111 BH

The physical addresses of all operands addressed by the BP register are computed using the SS segment register. The physical addresses of the destination operands of the string primitive operations (those addressed bythe DI register) are computed using the ES segment, which may not be overridden.

80188

High Integration 8-Bit Microprocessor
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

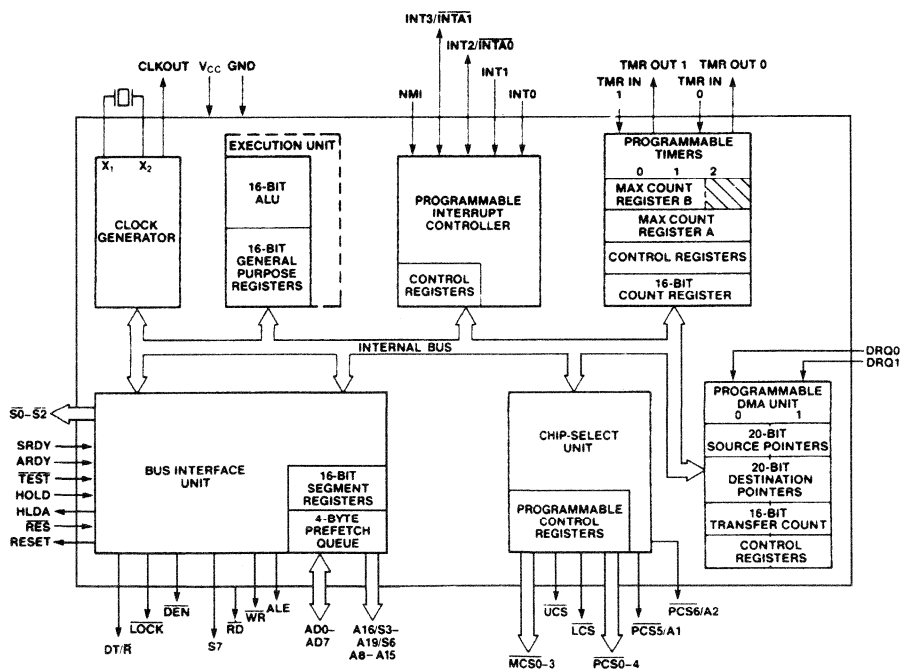
- Integrated feature set
  - Enhanced 10 MHz 8088-1 CPU
  - Clock generator
  - Two independent, high-speed DMA channels
  - Programmable interrupt controller
  - Three programmable 16-bit timers
  - Programmable memory and peripheral chip-select logic
  - Programmable wait state generator
  - Local bus controller
- Eight-bit data bus interface, 16-bit internal architecture
- Available in 10 MHz (80188-10), 8 MHz (80188)
- High-performance processor
  - Two times the performance of the standard 8088
  - 2.25 Mbyte/sec bus bandwidth interface
- Direct addressing capability to 1 Mbyte of memory
- Completely object code compatible with all existing iAPX 86, 88 software
  - Ten new instruction types
  - Compatible with 29843/45, 29833/63, 8284, and 8288 bus support components
- Optional numeric processor extension
- Available in 68-pin Plastic Leaded Chip Carrier (PLCC), Ceramic Leadless Chip Carrier (LCC), and Pin Grid Array (PGA) packages.

GENERAL DESCRIPTION

The 80188 is a highly-integrated microprocessor with an 8-bit data bus interface and a 16-bit internal architecture for high performance. It effectively combines 15-20 of the most common iAPX 86 system components onto one. The 8 MHz 80188 provides two times greater throughput than the standard 5 MHz 8088. The 80188 is upward compatible with 8086 and 8088 software and adds 10 new instruction types to the existing set.

The 80188 comes in a 68-pin package and requires a single +5 V power supply.

BLOCK DIAGRAM

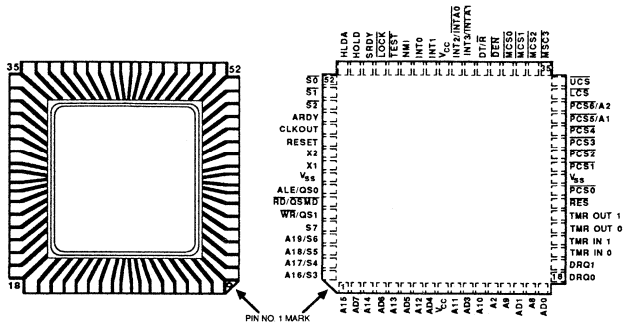


BD003561

Publication # 06878 | Rev. B | Amendment /0
Issue Date: April 1987

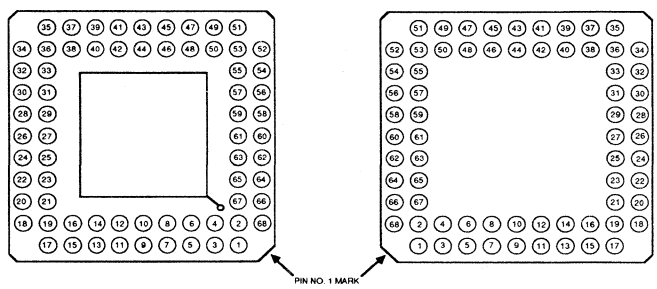
CONNECTION DIAGRAMS

Leadless Chip Carrier (LCC*)

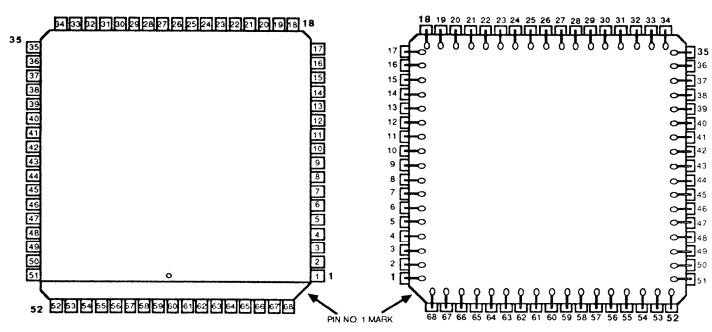


*LCC package placed in socket top down.

Pin Grid Array (PGA)

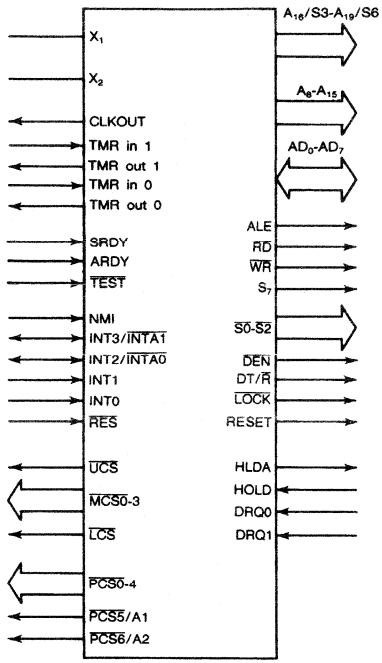


Plastic Leaded Chip Carrier (PLCC**)



**PLCC package placed in socket top up.

LOGIC SYMBOL

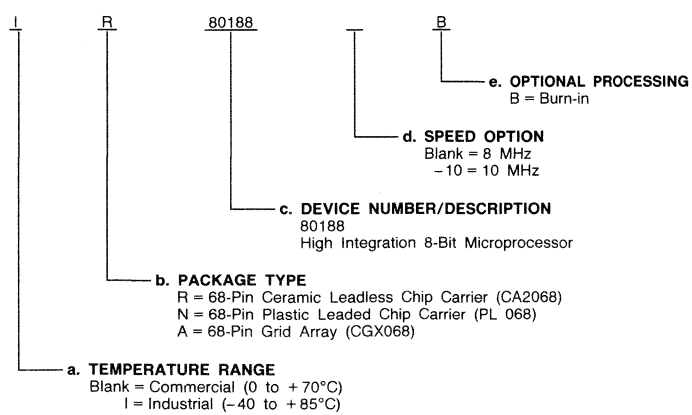


LS001970

ORDERING INFORMATION
Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Temperature Range
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
A, R, N	80188
	80188-10
A, R, IA, IR	80188B

Valid Combinations

Valid combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Active State	Name	I/O	Description
	V_{CC}, V_{CC}	I	System Power: +5 volt power supply.
	V_{SS}, V_{SS}	I	System Ground.
Active HIGH	RESET	O	Reset Output indicates that the 80188 CPU is being reset; and can be used as a system reset. It is active HIGH, synchronized with the processor clock, and lasts an integer number of clock periods corresponding to the length of the $\overline{RES}$ signal.
	X1, X2	I	Crystal inputs, X1 and X2, provide an external connection for a fundamental mode parallel resonant crystal for the internal crystal oscillator. X1 can interface to an external clock instead of a crystal. The input or oscillator frequency is internally divided by two to generate the clock signal (CLKOUT).
	CLKOUT	O	Clock Output provides the system with a 50% duty cycle waveform. All device pin timings are specified relative to CLKOUT. CLKOUT has sufficient MOS drive capabilities for a numeric processor extension.
Active LOW	$\overline{RES}$	I	System Reset causes the 80188 to immediately terminate its present activity, clear the internal logic, and enter a dormant state. This signal may be asynchronous to the 80188 clock. The 80188 begins fetching instructions approximately 7 clock cycles after $\overline{RES}$ is returned HIGH. $\overline{RES}$ is required to be LOW for greater than 4 clock cycles and is internally synchronized. For proper initialization, the LOW-to-HIGH transition of $\overline{RES}$ must occur no sooner than 50 microseconds after power up. This input is provided with a Schmitt-trigger to facilitate power-on $\overline{RES}$ generation via an RC network. When $\overline{RES}$ occurs, the 80188 will drive the status lines to an inactive level for one clock, and then tri-state them.
Active LOW	$\overline{TEST}$	I	$\overline{TEST}$ is examined by the WAIT instruction. If the $\overline{TEST}$ input is HIGH when "WAIT" execution begins, instruction execution will suspend. $\overline{TEST}$ will be resampled until it goes LOW, at which time execution will resume. If interrupts are enabled while the 80188 is waiting for $\overline{TEST}$, interrupts will be serviced. This input is synchronized internally.
Active HIGH	TMR in 0, TMR IN1	I I	Timer inputs are used either as clock or control signals, depending upon the programmed timer mode. These inputs are active HIGH (or LOW-to-HIGH transitions are counted) and internally synchronized.
Active HIGH	TMR OUT 0, TMR OUT 1	O O	Timer outputs are used to provide single pulse or continuous waveform generation, depending upon the timer mode selected.
Active HIGH	DRQ0 DRQ1	I I	DMA Request is driven HIGH by an external device when it desires that a DMA channel (Channel 0 or 1) perform a transfer. These signals are active HIGH, level-triggered, and internally synchronized.
Active HIGH	NMI	I	Non-Maskable Interrupt is an edge-triggered input which causes a type 2 interrupt. NMI is not maskable internally. A transition from a LOW to HIGH initiates the interrupt at the next instruction boundary. NMI is latched internally. An NMI duration of one clock or more will guarantee service. This input is internally synchronized.
Active HIGH or LOW	INT0, $\overline{INT1}$ INT2/$\overline{INTA0}$ INT3/$\overline{INTA1}$	I I/O I/O	Maskable Interrupt Requests can be requested by strobing one of these pins. When configured as inputs, these pins are active HIGH. Interrupt Requests are synchronized internally. INT2 and INT3 may be configured via software to provide active-LOW interrupt-acknowledge output signals. All interrupt inputs may be configured via software to be either edge- or level-triggereed. To ensure recognition, all interrupt requests must remain active until the interrupt is acknowledged. When iRMX mode is selected, the function of these pins changes (see Interrupt Controller section of this data sheet).
Active HIGH	A19/S6, A18/S5, A17/S4, A16/S3	O O O O	Address Bus Outputs (16-19) and Bus Cycle Status (3-6) reflect the four most significant address bits during T_1. These signals are active HIGH. During T_2, T_3, T_W, and T_4, status information is available on these lines as encoded below: S3, S4, and S5 are defined as LOW during T_2-T_4.
Active HIGH	AD7-AD0	I/O	Address/Data Bus (0-7) signals constitute the time multiplexed memory or I/O address (T_1) and data (T_2, T_3, T_W, and T_4) bus.
Active HIGH	A15-A8	O	Address-only Bus (8-15), containing valid address from T_1-T_4.
Active HIGH	S7	O	This signal is always HIGH to indicate that the 80188 has an 8-bit data bus and is tri-state OFF during bus hold.
Active HIGH	ALE/QS0	O	Address Latch Enable/Queue Status 0 is provided by the 80188 to latch the address into the 8282/8283 address latches. Addresses are guaranteed to be valid on the trailing edge of ALE. The ALE rising edge is generated off the rising edge of the CLKOUT immediately preceding T_1 of the associated bus cycle, effectively one-half clock cycle earlier than in the standard 8088. The trailing edge is generated off the CLKOUT rising edge in T_1 as in the 8088. Note that ALE is never floated.

	Low	High
S6	Processor Cycle	DMA Cycle

PIN DESCRIPTION (Cont.)

Active State	Name	I/O	Description
Active LOW	$\overline{WR}$/QS1	O	Write Strobe/Queue Status 1 indicates that the data on the bus is to be written into a memory or an I/O device. $\overline{WR}$ is active for T_2, T_3, and T_W of any write cycle. Floats during "HOLD." It is driven HIGH for one clock during Reset, and then floated. When the 80188 is in queue status mode, the ALE/QS0 and $\overline{WR}$/QS1 pins provide information about processor/instruction queue interaction.

QS1	QS0	Queue Operation
0	0	No queue operation
0	1	First opcode byte fetched from the queue
1	1	Subsequent byte fetched from the queue
1	0	Empty the queue

Active LOW	$\overline{RD}$/QSMD	O	Read Strobe indicates that the 80188 is performing a memory or I/O read cycle. $\overline{RD}$ is active LOW for T_2, T_3, and T_W of any read cycle. It is guaranteed not to go LOW in T_2 until after the Address Bus is floated. $\overline{RD}$ is active LOW, and floats during "HOLD." $\overline{RD}$ is driven HIGH for one clock during Reset, and then the output driver is floated. A weak internal pull-up mechanism on the $\overline{RD}$ line holds it HIGH when the line is not driven. During RESET the pin is sampled to determine whether the 80188 should provide ALE, $\overline{WR}$, and $\overline{RD}$, or if the Queue-Status should be provided. $\overline{RD}$ should be connected to GND to provide Queue-Status data.
Active HIGH	ARDY	I	Asynchronous Ready informs the 80188 that the addressed memory space or I/O device will complete a data transfer. The ARDY input pin will accept an asynchronous input, and is active HIGH. Only the rising edge is internally synchronized by the 80188. This means that the falling edge of ARDY must be synchronized to the 80188 clock. If connected to V_{CC}, no WAIT states are inserted. Asynchronous ready (ARDY) or synchronous ready (SRDY) /ust be active to terminate a bus cycle. If line is unused, it may remain connected to V_{CC} or it may be connected to V_{SS} (in which case the programmer must initialize the part to inhibit the external pins).
Active HIGH	SRDY	I	Synchronous Ready must be synchronized externally to the 80188. The use of SRDY provides a relaxed system-timing specification on the Ready input. This is accomplished by eliminating the one-half clock cycle which is required for internally resolving the signal level when using the ARDY input. This line is active HIGH. If this line is connected to V_{CC} no WAIT states are inserted. Asynchronous ready (ARDY) or synchronous ready (SRDY) must be active before a bus cycle is terminated. If line is unused, it may remain connected to V_{CC} or it may be connected to V_{SS} (in which case the programmer must initialize the part to inhibit the external pins).
Active LOW	$\overline{LOCK}$	O	$\overline{LOCK}$ output indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active LOW. The $\overline{LOCK}$ signal is requested by the LOCK prefix instruction and is activated at the beginning of the first data cycle associated with the instruction following the LOCK prefix. It remains active until the completion of the instruction following the LOCK prefix. No prefetches will occur while $\overline{LOCK}$ is asserted. When executing more than one LOCK instruction, there must be six bytes of code between the end of the first LOCK instruction and the start of the second LOCK instruction. $\overline{LOCK}$ is active LOW, is driven HIGH for one clock during RESET, and then floated.
	$\overline{S0}$, $\overline{S1}$, $\overline{S2}$	O	Bus cycle status $\overline{S0}$-$\overline{S2}$ are encoded to provide bus-transaction information:

80188 Bus Cycle Status Information			
$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Bus Cycle Initiated
0	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

The status pins float during "HOLD."
$\overline{S2}$ may be used as a logical M/$\overline{IO}$ indicator, and $\overline{S1}$ as a DT/$\overline{R}$ indicator.
The status lines are driven HIGH for one clock during Reset, and then floated until a bus cycle begins.

Active HIGH	HOLD (input) HLDA (output)	I O	HOLD indicates that another bus master is requesting the local bus. The HOLD input is active HIGH. HOLD may be asynchronous with respect to the 80188 clock. The 80188 will issue a HLDA in response to a HOLD request at the end of T_4 or T_1 Simultaneous with the issuance of HLDA, the 80188 will float the local bus and control lines. After HOLD is detected as being LOW, the 80188 will lower HLDA. When the 80188 needs to run another bus cycle, it will again drive the local bus and control lines.
Active LOW	$\overline{UCS}$	O	Upper Memory Chip Select is an active LOW output whenever a memory reference is made to the defined upper portion (1K-=256K block) of memory. This line is not floated during bus HOLD. The address range activating $\overline{UCS}$ is software programmable.
Active LOW	$\overline{LCS}$	O	Lower Memory Chip Select is active LOW whenever a memory reference is made to the defined lower portion (1K-256K) of memory. This line is not floated during bus HOLD. The address range activating $\overline{LCS}$ is software programmable.
Active LOW	$\overline{MCS0}$-3	O	Mid-Range Memory Chip Select signals are active LOW when a memory reference is made to the defined mid-range portion of memory (8K-512K). These lines are not floated during bus HOLD. The address ranges activating $\overline{MCS0}$-3 are software programmable.
Active LOW	$\overline{PCS0}$ $\overline{PCS1}$-4	O O	Peripheral Chip Select signals 0-4 are active LOW when a reference is made to the defined peripheral area (65K byte I/O space). These lines are not floated during bus HOLD. The address ranges activating $\overline{PCS0}$-4 are software programmable.

PIN DESCRIPTION (Cont.)

Active State	Name	I/O	Description
Active LOW or HIGH	$\overline{PCS5}$/A1	O	Peripheral Chip Select 5 or Latched A1 may be programmed to provide a sixth peripheral chip select, or to provide an internally latched A1 signal. The address range activating $\overline{PCS5}$ is software programmable. When programmed to provide latched A1, rather than $\overline{PCS5}$, this pin will retain the previously latched value of A1 during a bus HOLD.
Active LOW or HIGH	$\overline{PCS6}$/A2	O	Peripheral Chip Select 6 or Latched A2 may be programmed to provide a seventh peripheral chip select, or to provide an internally latched A2 signal. The address range activating $\overline{PCS6}$ is software programmable. When programmed to provide latched A2, rather than $\overline{PCS6}$, this pin will retain the previously latched value of A2 during a bus HOLD.
	DT/$\overline{R}$	O	Data Transmit/Receive controls the direction of data flow through the external 2946/47 data bus transceiver. When LOW, data is transferred to the 80188. When HIGH the 80188 places write data on the data bus.
Active LOW	$\overline{DEN}$	O	Data Enable is provided as a 2946/47 data bus transceiver output enable. $\overline{DEN}$ is active LOW during each memory and I/O access. $\overline{DEN}$ is HIGH whenever DT/$\overline{R}$ changes state.

FUNCTIONAL DESCRIPTION

Introduction

The following Functional Description describes the base architecture of the 80188. The architecture is common to the 8086, 8088, and 80286 microprocessor families as well. The 80188 is a very high integration 8-bit microprocessor. It combines 15-20 of the most common microprocessor system components onto one chip while providing twice the performance of the standard 8088. The 80188 is object code compatible with the 8086, 8088 microprocessors and adds 10 new instruction types to the existing 8086, 8088 instruction set.

80188 BASE ARCHITECTURE

The 8086, 8088, 80188, 80186 and 80286 family all contain the same basic set of registers, instructions, and addressing modes. The 80188 processor is upward compatible with the 8086, 8088, 80186 and 80286 CPUs.

Register Set

The 80188 base architecture has fourteen registers as shown in Figures 1 and 2. These registers are grouped into the following categories.

General Registers

Eight 16-bit general purpose registers used to contain arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used as 16-bit registers or split into pairs of separate 8-bit registers.

Segment Registers

Four 16-bit special purpose registers select, at any given time, the segments of memory that are immediately addressable for code, stack, and data. (For usage, refer to Memory Organization.)

Base and Index Registers

Four of the general purpose registers may also be used to determine offset addresses of operands in memory. These registers may contain base addresses or indexes to particular locations within a segment. The addressing mode selects the specific registers for operand and address calculations.

Status and Control Registers

Two 16-bit special purpose registers record or alter certain aspects of the 80188 processor state. These are the Instruction Pointer Register, which contains the offset address of the next sequential instruction to be executed, and the Status Word Register, which contains status and control flag bits (see Figures 1 and 2).

Status Word Description

The Status Word records specific characteristics of the result of logical and arithmetic instructions (bits 0, 2, 4, 6, 7, and 11) and controls the operation of the 80188 within a given operating mode (bits 8, 9, and 10). The Status Word Register is 16-bits wide. The function of the Status Word bits is shown in Table 2.

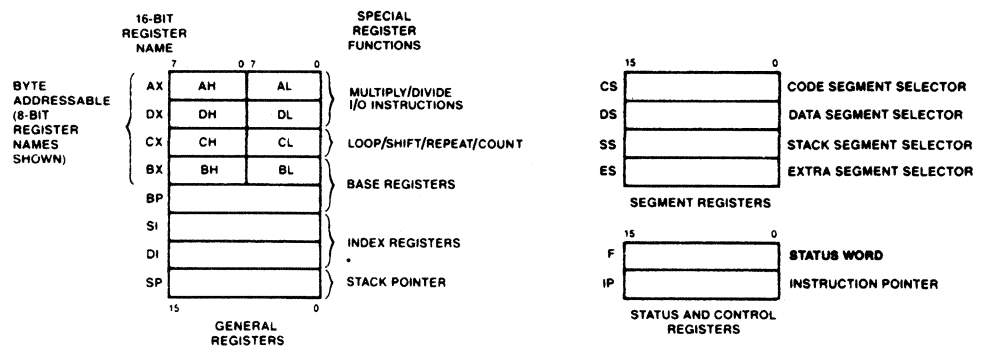


Figure 1. 80188 General Purpose Register Set

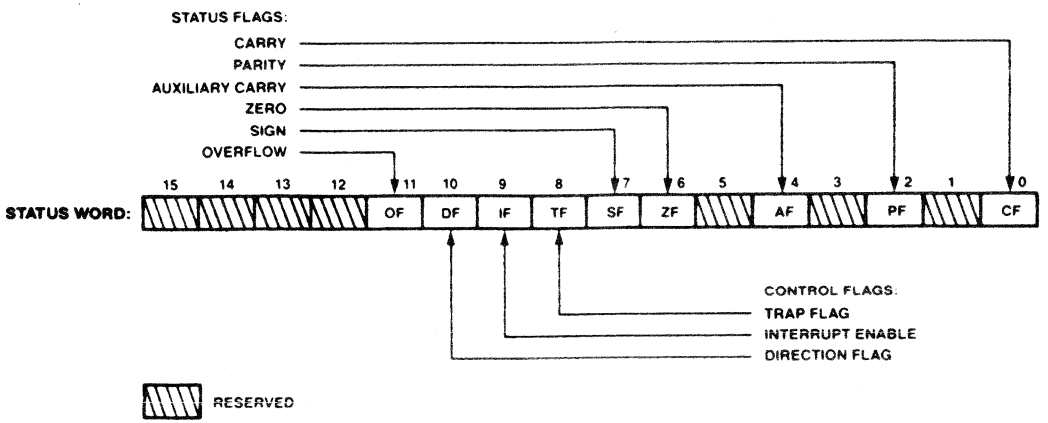


Figure 2. Status Word Format

Table 2. Status Word Bit Function

Bit Position	Name	Function
0	CF	Carry Flag — Set on high-order bit carry or borrow; cleared otherwise.
2	PF	Parity Flag — Set if low-order 8 bits or result contain an even number of 1-bits; cleared otherwise.
4	AF	Set on carry from or borrow to the low order four bits of AL; cleared otherwise.
6	ZF	Zero Flag — Set if result is zero; cleared otherwise.
7	SF	Sign Flag — Set equal to high-order bit of result (0 if positive, 1 if negative).
8	TF	Single Step Flag — Once set, a single step interrupt occurs after the next instruction executes. TF is cleared by the single step interrupt.
9	IF	Interrupt-enable Flag — When set, maskable interrupts will cause the CPU to transfer control to an interrupt vector specified location.
10	DF	Direction Flag — Causes string instructions to auto decrement the appropriate index register when set. Clearing DF causes auto increment.
11	OF	Overflow Flag — Set if the signed result cannot be expressed within the number of bits in the destination operand; cleared otherwise.

Instruction Set

The instruction set is divided into seven categories: data transfer, arithmetic, shift/rotate/logical, string manipulation, control transfer, high-level instructions, and processor control. These categories are summarized in Figure 3.

An 80188 instruction can reference anywhere from zero to several operands. An operand can reside in a register, in the instruction itself, or in memory. Specific operand addressing modes are discussed later in this data sheet.

Memory Organization

Memory is organized in sets of segments. Each segment is a linear contiguous sequence of up to 64K (2^{16}) 8-bit bytes. Memory is addressed using a two-component address (a pointer) that consists of a 16-bit base segment and a 16-bit offset. The 16-bit base values are contained i one of four internal segment registers (code, data, stack, extra). The physical address is calculated by shifting the base value LEFT by four bits and adding the 16-bit offset value to yield a 20-bit physical address (see Figure 4). This allows for a 1 MByte physical address size.

All instructions that address operands in memory must specifiy the base segment and the 16-bit offset value. For speed and compact instruction encoding, the segment register used for physical address generation is implied by the addressing mode used (see Table 3). These rules follow the way programs are written (see Figure 5) as independent modules that require areas for code and data, a stack, and access to external data areas.

Special segment override instruction prefixes allow the implicit segment register selection rules to be overridden for special cases. The stack, data, and extra segments may coincide for simple programs.

GENERAL PURPOSE	
MOV	Move byte or word
PUSH	Push word onto stack
POP	Pop word off stack
PUSHA	Push all registers on stack
POPA	Pop all registers from stack
XCHG	Exchange byte or word
XLAT	Translate byte
INPUT/OUTPUT	
IN	Input byte or word
OUT	Output byte or word
ADDRESS OBJECT	
LEA	Load effective address
LDS	Load pointer using DS
LES	Load pointer using ES
FLAG TRANSFER	
LAHF	Load AH register from flags
SAHF	Store AH register in flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack

ADDITION	
ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
SUBTRACTION	
SUB	Subtract byte or word
SBB	Subtract byte or word with borrow
DEC	Decrement byte or word by 1
NEG	Negate byte or word
CMP	Compare byte or word
AAS	ASCII adjust for subtraction
DAS	Decimal adjust for subtraction
MULTIPLICATION	
MUL	Multiply byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII adjust for multiply
DIVISION	
DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
CBW	Convert byte to word
CWD	Convert word to doubleword
MOVS	Move byte or word string

INS	Input bytes or word string
OUTS	Output bytes or word string
CMPS	Compare byte or word string
SCAS	Scan byte or word string
LODS	Load byte or word string
STOS	Store byte or word string
REP	Repeat
REPE/ REPZ	Repeat while equal/zero
REPNE/ REPNZ	Repeat while not equal/not zero

LOGICALS	
NOT	"Not" byte or word
AND	"And" byte or word
OR	"Inclusive or" byte or word
XOR	"Exclusive or" byte or word
TEST	"Test" byte or word
SHIFTS	
SHL/SAL	Shift logical/arithmetic left byte or word
SHR	Shift logical right byte or word
SAR	Shift arithmetic right byte or word
ROTATES	
ROL	Rotate left byte or word
ROR	Rotate right byte or word
RCL	Rotate through carry left byte or word
RCR	Rotate through carry right byte or word

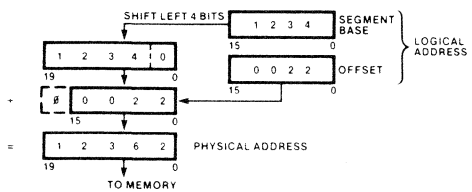
FLAG OPERATIONS	
STC	Set carry flag
CLC	Clear carry flag
CMC	Complement carry flag
STD	Set direction flag
CLD	Clear direction flag
STI	Set interrupt enable flag
CLI	Clear interrupt enable flag
EXTERNAL SYNCHRONIZATION	
HLT	Halt until interrupt or reset
WAIT	Wait for $\overline{\text{TEST}}$ pin active
ESC	Escape to extension processor
LOCK	Lock bus during next instruction
NO OPERATION	
NOP	No operation
HIGH LEVEL INSTRUCTIONS	
ENTER	Format stack for procedure entry
LEAVE	Restore stack for procedure exit
BOUND	Detects values outside prescribed range

Figure 3. 80188 Instruction Set

CONDITIONAL TRANSFERS		UNCONDITIONAL TRANSFERS	
JA/JNBE	Jump if above/not below nor equal	CALL	Call procedure
JAE/JNB	Jump if above or equal/not below	RET	Return from procedure
JB/JNAE	Jump if below/not above nor equal	JMP	Jump
JBE/JNA	Jump if below or equal/not above		
JC	Jump if carry	**ITERATION CONTROLS**	
JE/JZ	Jump if equal/zero		
JG/JNLE	Jump if greater/not less nor equal	LOOP	Loop
JGE/JNL	Jump if greater or equal/not less	LOOPE/LOOPZ	Loop if equal/zero
JL/JNGE	Jump if less/not greater nor equal	LOOPNE/LOOPNZ	Loop if not equal/not zero
JLE/JNG	Jump if less or equal/not greater	JCXZ	Jump if register CX = 0
JNC	Jump if not carry		
JNE/JNZ	Jump if not equal/not zero	**INTERRUPTS**	
JNO	Jump if not overflow		
JNP/JPO	Jump if not parity/parity odd	INT	Interrupt
JNS	Jump if not sign	INTO	Interrupt if overflow
JO	Jump if overflow	IRET	Interrupt return
JP/JPE	Jump if parity/parity even		
JS	Jump if sign		

Figure 3. 80188 Instruction Set (continued)

To access operands that do not reside in one of the four immediately available segments, a full 32-bit pointer can be used to reload both the base (segment) and offset values.

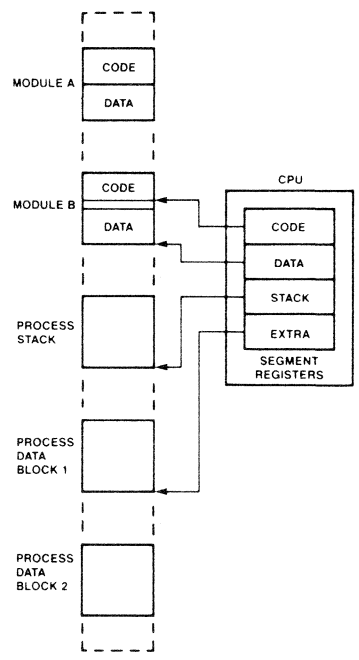


DF002920

Figure 4. Two Component Address

Table 3. Segment Register Selection Rule

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Instructions	Code (CS)	Instruction prefetch and immediate data.
Stack	Stack (SS)	All stack pushes and pops; any memory references which use BP Register as a base register.
External Data (Global)	Extra (ES)	All string instruction references which use the DI register as an index.
Local Data	Data (DS)	All other data references.



DF002930

Figure 5. Segmented Memory Helps Structure Software

Addressing Modes

The 80188 provides eight categories of addressing modes to specify operands. Two addressing modes are provided for instructions that operate on register or immediate operands:

- *Register Operand Mode:* The operand is located in one of the 8- or 16-bit general registers.
- *Immediate Operand Mode:* The operand is included in the instruction.

Six modes are provided to specify the location of an operand in a memory segment. A memory operand address consists of two 16-bit components: a segment base and an offset. The segment base is supplied by a 16-bit segment register either implicity chosen by the addressing mode or explicitly chosen by a segment override prefix. The offset, also called the effective address, is calculated by summing any combination of the following three address elements:

- the *displacement* (an 8- or 16-bit immediate value contained in the instruction);
- the *base* (contents of either the BX or BP base registers); and
- the *index* (contents of either the SI or DI index registers)

Any carry out from the 16-bit addition is ignored. Eight-bit displacements are sign extended to 16-bit values.

Combinations of these three address elements define the six memory addressing modes, described below.

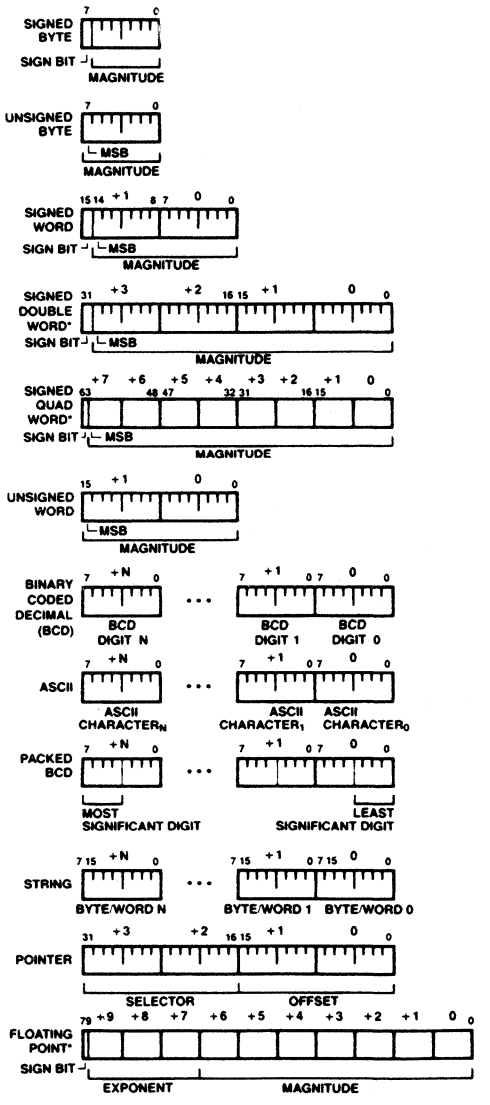
- *Direct Mode:* The operand's offset is contained in the instruction as an 8- or 16-bit displacement element.
- *Register Indirect Mode:*The operand's offset is in one of the registers SI, DI, BX, or BP.
- *Based Mode:* The operand's offset is the sum of an 8- or 16-bit displacement and the contents of a base register (BX or BP).
- *Indexed Mode:* The operand's offset is the sum of an 8- or 16-bit displacement and the contents of an index register (SI or DI).
- *Based Indexed Mode:* The operand's offset is the sum of the contents of a base register and an index register.
- *Based Indexed Mode with Displacement:* The operand's offset is the sum of a base register's contents, an index register's contents, and an 8- or 16-bit displacement.

Data Types

The 80188 directly supports the following data types:

- *Integer:* A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a 2's complement representation. Signed 32 and 64 bit integers are supported using a numeric data processor.
- *Ordinal:* An unsigned binary numeric value contained in an 8-bit byte or a 16-bit word.
- *Pointer:* A 16- or 32-bit quantity, composed of a 16-bit offset component or a 16-bit segment base component in addition to a 16-bit offset component.
- *String:* A contiguous sequence of bytes or words. A string may contain from 1K to 64K bytes.
- *ASCII:* A byte representation of alphanumeric and control characters using the ASCII standard of character representation.
- *BCD:* A byte (unpacked) representation of the decimal digits 0-9.
- *Packed BCD:* A byte (packed) representation of two decimal digits (0-9). One digit is stored in each nibble (4-bits) of the byte.
- *Floating Point:* A signed 32-, 64-, or 80-bit real number representation. (Floating point operands are supported using a numeric data processor configuration.)

In general, individual data elements must fit within defined segment limits. Figure 6 graphically represents the data types supported by the 80188.



DF002940

NOTE: *SUPPORTED BY 80188 WITH A NUMERIC DATA PROCESSOR

Figure 6. 80188 Supported Data Types

I/O Space

The I/O space consists of 64K 8-bit or 32K 16-bit ports. Separate instructions address the I/O space with either an 8-bit port address, specified in the instruction, or a 16-bit port address in the DX register, 8-bit port addresses are zero extended such that A_{15}-A_8 are LOW. I/O port addresses 00F8(H) through 00FF(H) are reserved.

Interrupts

An interrupt transfers execution to a new program location. The old program address (CS:IP) and machine state (Status Word) are saved on the stack to allow resumption of the interrupted program. Interrupts fall into three classes: hardware initiated, INT instructions, and instruction exceptions. Hardware initiated interrupts occur in response to an external input and are classified as non-maskable or maskable.

Programs may cause an interrupt with an INT instruction. Instruction exceptions occur when an unusual condition, which prevents further instruction processing, is detected while attempting to execute an instruction. If the exception was caused by executing an ESC instruction with the ESC trap bit set in the relocation register, the return instruction will point to the ESC instruction, or to the segment override prefix immediately preceding the ESC instruction if the prefix was present. In all other cases, the return address from an exception will point at the instruction immediately following the instruction causing the exception.

A table containing up to 256 pointers defines the proper interrupt service routine for each interrupt. Interrupts 0-31, some of which are used for instruction exceptions, are reserved. Table 4 shows the 80188 predefined types and default priority levels. For each interrupt, an 8-bit vector must be supplied to the 80188 which identifies the appropriate table entry. Exceptions supply the interrupt vector internally. In addition, internal peripherals and noncascaded external interrupts will generate their own vectors through the internal interrupt controller. INT instructions contain or imply the vector and allow access to all 256 interrupts. Maskable hardware initiated interrupts supply the 8-bit vector to the CPU during an interrupt acknowledge bus sequence. Non-maskable hardware interrupts use a predefined internally supplied vector.

Interrupt Sources

The 80188 can service interrupts generated by software or hardware. The software interrupts are generated by specific instructions (INT, ESC, unused OP, etc.) or the results of conditions specified by instructions (array bounds check, INTO, DIV, IDIV, etc.) All interrupt sources are serviced by an indirect call through an element of a vector table. This vector table is indexed by using the interrupt vector type (Table 4), multiplied by four. All hardware-generated interrupts are sampled at the end of each instruction. Thus, the software interrupts will begin service first. Once the service routine is entered and interupts are enabled, any hardware source of sufficient priority can interrupt the service routine in progress.

The software generated 80188 interrupts are described below.

DIVIDE ERROR EXCEPTION (TYPE 0)

Generated when a DIV or IDIV instuction quotient cannot be expressed in the number of bits in the destination.

Table 4. 80188 Interrupt Vectors

Interrupt Name	Vector Type	Default Priority	Related Instructions
Divide Error Exception	0	*1	DIV, IDIV
Single Step Interrupt	1	12**2	All
NMI	2	1	All
Breakpoint Interrupt	3	*1	INT
INTO Detected Overflow Exception	4	*1	INTO
Array Bounds Exception	5	*1	BOUND
Unused-Opcode Exception	6	*1	Undefined Opcodes
ESC Opcode Exception	7	*1***	ESC Opcodes
Timer 0 Interrupt	8	2A****	
Timer 1 Interrupt	16	2B****	
Timer 2 Interrupt	17	2C****	
Reserved	9	3	
DMA 0 Interrupt	10	4	
DMA 1 Interrupt	11	5	
INT0 Interrupt	12	6	
INT1 Interrupt	13	7	
INT2 Interrupt	14	8	
INT3 Interrupt	15	9	

NOTES:

*1. These are generated as the result of an instruction execution.

**2. This is handled as in the 8088.

****3. All three timers constitute one source of request to the interrupt controller. The Timer interrupts all have the same default priority level with respect to all other interrupt sources. However, they have a defined priority ordering amongst themselves. (Priority 2A is higher priority than 2B.) Each Timer interrupt has a separate vector type number.

4. Default priorities for the interrupt sources are used only if the user does not program each source into a unique priority level.

***5. An escape opcode will cause a trap only if the proper bit is set in the peripheral control block relocation register.

SINGLE-STEP INTERRUPT (TYPE 1)

Generated after most instructions if the TF flag is set. Interrupts will not be generated after prefix instructions (e.g., REP), instructions which modify segment registers (e.g., POP DS), or the WAIT instruction.

NON-MASKABLE INTERRUPT-NMI (TYPE 2)

An external interrupt source which cannot be masked.

BREAKPOINT INTERRUPT (TYPE 3)

A one-byte version of the INT instruction. It uses 12 as an index into the service routine address table (because it is a type 3 interrupt).

INTO DETECTED OVERFLOW EXCEPTION (TYPE 4)

Generated during an INTO instruction if the OF bit is set.

ARRAY BOUNDS EXCEPTION (TYPE 5)

Generated during a BOUND instruction if the array index is outside the array bounds. The array bounds are located in memory at a location indicated by one of the instruction operands. The other operand indicates the value of the index to be checked.

UNUSED OPCODE EXCEPTION (TYPE 6)

Generated if execution is attempted on undefined opcodes.

ESCAPE OPCODE EXCEPTION (TYPE 7)

Generated if execution is attempted of ESC opcodes (D8H-DFH). This exception will only be generated if a bit in the relocation register is set. The return address of this exception will point to the ESC instruction causing the exception. If a segment override prefix preceded the ESC instruction, the return address will point to the segment override prefix.

Hardware-generated interrupts are divided into two groups: maskable interupts and non-maskable interrupts. The 80188 provides maskable hardware interrupt request pins INT0-INT3. In addition, maskable interrupts may be generated by the 80188 integrated DMA controller and the integrated timer unit. The vector types for these interrupts is shown in Table 4. Software enables these inputs by setting the interrupt flag bit (IF) in the Status Word. The interrupt controller is discussed in the peripheral section of this data sheet.

Further maskable interrupts are disabled while servicing an interrupt because the IF bit is reset as part of the response to an interrupt or exception. The saved Status Word will reflect the enable status of the processor prior to the interrupt. The interrupt flag will remain zero unless specifically set. The interrupt return instruction restores the Status Word, thereby restoring the original status of IF bit. If the interrupt return re-enables interrupts, and another interrupt is pending, the 80188 will immediately service the highest-priority interrupt pending, i.e., no instructions of the main line program will be executed.

Non-Maskable Interrupt Request (NMI)

A non-maskable interrupt (NMI) is also provided. This interrupt is serviced regardless of the state of the IF bit. A typical use of NMI would be to activate a power failure routine. The activation of this input causes an interrupt with an internally supplied vector value of 2. No external interrupt acknowledge sequence is performed. The IF bit is cleared at the beginning of an NMI interrupt to prevent maskable interrupts from being serviced.

Single-Step Interrupt

The 80188 has an internal interrupt that allows programs to execute one instruction at a time. It is called the single-step interrupt and is controlled by the single-step flag bit (TF) in the Status Word. Once this bit is set, an internal single-step interrupt will occur after the next instruction has been executed. The interrupt clears the TF bit and uses an internally supplied vector of 1. The IRET instruction is used to set the TF bit and transfer control to the next instruction to be single-stepped.

Initialization and Processor Reset

Processor initialization or startup is accomplished by driving the $\overline{RES}$ input pin LOW. $\overline{RES}$ forces the 80188 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as $\overline{RES}$ is active. After $\overline{RES}$ becomes inactive and an internal processing interval elapses, the 80188 begins execution with the instruction at physical location FFFF0(H). $\overline{RES}$ also sets some registers to predefined values as shown in Table 5.

Table 5. 80188 Initial Register State after RESET

Status Word	F002(H)
Instruction Pointer	0000(H)
Code Segment	FFFF(H)
Data Segment	0000(H)
Extra Segment	0000(H)
Stack Segment	0000(H)
Relocation Register	20FF(H)
UMCS	FFFB(H)

THE 80188 COMPARED TO THE 80186

The 80188 CPU is an 8-bit processor designed around the 80186 internal structure. Most internal functions of the 80188 are identical to the equivalent 80186 functions. The 80188 handles the external bus the same way the 80186 does with the distinction of handling only 8 bits at a time. Sixteen bit operands are fetched or written in two consecutive bus cycles. Both processors will appear identical to the software engineer, with the exception of execution time. The internal register structure is identical and all instructions have the same end result. The differences between the 80188 and 80186 are outlined below. Internally, there are three differences between the 80188 and the 80186. All changes are related to the 8-bit bus interface.

- The queue length is 4 bytes in the 80188, whereas the 80186 queue contains 6 bytes, or three words. The queue was shortened to prevent overuse of the bus by the BIU when prefetching instructions. This was required because of the additional time necessary to fetch instructions 8 bits at a time.
- To further optimize the queue, the prefetching algorithm was changed. The 80188 BIU will fetch a new instruction to load into the queue each time there is a 1-byte hole (space available) in the queue. The 80186 waits until a 2-byte space is available.
- The internal execution time of the instruction is affected by the 8-bit interface. All 16-bit fetches and writes from/to memory take an additional four clock cycles. The CPU may also be limited by the speed of instruction fetches when a series of simple operations occurs. When the more sophisticated instructions of the 80188 are being used, the queue has time to fill and the execution proceeds as fast as the execution unit will allow.

The 80188 and 80186 are completely software compatible by virtue of their identical execution units. Software that is system dependent may not be completely transferable, but software that is not system dependent will operate equally well on an 80188 or an 80186.

The hardware interface of the 80188 contains the major differences between the two CPUs. The pin assignments are nearly identical, however, with the following functional changes.

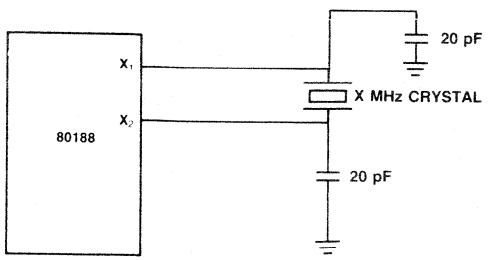
- A8-A15—These pins are only address outputs on the 80188. These address lines are latched internally and remain valid throughout a bus cycle in a manner similar to the 8085 upper address lines.
- $\overline{BHE}$ has no meaning on the 80188 and has been eliminated.

80188 CLOCK GENERATOR

The 80188 provides an on-chip clock generator for both internal and external clock generation. The clock generator features a crystal oscillator, a divide-by-two counter, synchronous and asynchronous ready inputs, and reset circuitry.

Oscillator

The oscillator circuit of the 80188 is designed to be used with a parallel resonant fundamental mode crystal. This is used as the time base for the 80188. The crystal frequency selected will be double the CPU clock frequency. Use of an LC or RC circuit is not recommended with this oscillator. If an external oscialltor is used, it can be connected directly to input pin X1 in lieu of a crystal. The output of the oscillator is not directly available outside the 80188. The recommended crystal configuration is shown in Figure 7.



X = 20 for 10 MHz (80188-1)
X = 16 for 8 MHz (80188-3)
X = 12 for 6 MHz (80188-6)

Figure 7. Recommended 80188 Crystal Configuration

Clock Generator

The 80188 clock generator provides the 50% duty cycle processor clock for the 80188. It does this by dividing the oscillator output by 2 forming the symmetrical clock. If an external oscillator is used, the state of the clock generator will change on the falling edge of the oscillator signal. The CLKOUT pin provides the processor clock signal for use outside the 80188. This may be used to drive other system components. All timings are referenced to the output clock.

READY Synchronization

The 80188 provides both synchronous and asynchronous ready inputs. Asynchronous ready synchronization is accomplished by circuitry which samples ARDY in the middle of T_2, T_3 and again in the middle of each T_W until ARDY is sampled HIGH. One-half CLKOUT cycle of resolution time is used. Full synchronization is performed only on the rising edge of ARDY, i.e., the falling edge of ARDY must be synchronized to the CLKOUT signal if it will occur during T_2 or T_W. HIGH-to-LOW transitions of ARDY must be performed synchronously to the CPU clock.

A second ready input (SRDY) is provided to interface with externally synchronized ready signals. This input is sampled at the end of T_2 and again at the end of each T_W until it is sampled HIGH. By using this input rather than the asynchronous ready input, the half-clock cycle resolution time penalty is eliminated.

This input must satisfy set-up and hold times to guarantee proper operation of the circuit.

In addition, the 80188, as part of the integrated chip-select logic, has the capability to program WAIT states for memory and peripheral blocks. This is discussed in the Chip Select/Ready Logic description.

RESET Logic

The 80188 provides both a $\overline{RES}$ input pin and a synchronized RESET pin for use with other system components. The $\overline{RES}$ input pin on the 80188 is provided with hysteresis in order to facilitate power-on Reset generation via an RC network. RESET is guaranteed to remain active for at least five clocks given a $\overline{RES}$ input of at least six clocks. RESET may be delayed up to two and one-half clocks behind $\overline{RES}$.

Multiple 80188 processors may be synchronized through the $\overline{RES}$ input pin, since this input resets both the processor and divide-by-two internal counter in the clock generator. In order to insure that the divide-by-two counters all begin counting at the same time, the active going edge of $\overline{RES}$ must satisfy a 25 ns setup time before the falling edge of the 80188 clock input. In addition, in order to insure that all CPUs begin executing in the same clock cycle, the reset must satisfy a 25 ns setup time before the rising edge of the CLKOUT signal of all the processors.

LOCAL BUS CONTROLLER

The 80188 provides a local bus controller to generate the local bus control signals. In addition, it employs a HOLD/HLDA protocol for relinquishing the local bus to other bus masters. It also provides control lines that can be used to enable external buffers and to direct the flow of data on and off the local bus.

Memory/Peripheral Control

The 80188 provides ALE, $\overline{RD}$, and $\overline{WR}$ bus control signals. The $\overline{RD}$ and $\overline{WR}$ signals are used to strobe data from memory to the 80188 or to strobe data from the 80188 to memory. The ALE line provides a strobe to address latches for the multiplexed address/data bus. The 80188 local bus controller does not provide a memory/$\overline{I/O}$ signal. If this is required, the user will have to use the $\overline{S2}$ signal (which will require external latching), make the memory and I/O spaces nonoverlapping, or use only the integrated chip-select circuitry.

Transceiver Control

The 80188 generates two control signals to be connected to 2946/2947 transceiver chips. This capability allows the addition of transceivers for extra buffering without adding external logic. These control lines, DT/$\overline{R}$ and $\overline{DEN}$, are generated to control the flow of data through the transceivers. The operation of these signals is shown in Table 6.

Table 6. Transceiver Control Signals Description

Pin Name	Function
$\overline{DEN}$ (Data Enable)	Enables the output drivers of the transceivers. It is active LOW during memory, I/O, or INTA cycles.
DT/$\overline{R}$ (Data Transmit/Receive)	Determines the direction of travel through the transceivers. A HIGH level directs data away from the processor during write operations, while a LOW level directs data toward the processor during a read operation.

Local Bus Arbitration

The 80188 uses a HOLD/HLDA system of local bus exchange. This provides an asynchronous bus exchange mechanism. This means multiple masters utilizing the same bus can operate at separate clock frequencies. The 80188 provides a single HOLD/HLDA pari through which all other bus masters may gain cotnrol of the local bus. This requires external circuitry to arbitrate whcih external device will gain control of the bus from the 80188 when there is more than one alternate local bus master. When the 80188 relinquishes control of the local bus, it floats $\overline{DEN}$, $\overline{RD}$, $\overline{WR}$, $\overline{S0}$-$\overline{S2}$, $\overline{LOCK}$, AD0-AD7, A8-A19, $\overline{S7}$ and DT/$\overline{R}$ to allow another master to drive these lines directly.

The 80188 HOLD latency time, i.e., the time between HOLD request and HOLD acknowledge, is a function of the activity occurring in the processor when the HOLD request is received. A HOLD request is the highest-priority activity request which the processor may receive: higher than instruction fetching or internal DMA cycles. However, if a DMA cycle is in progress, the 80188 will complete the transfer before relinquishing the bus. This implies that if a HOLD request is received just as a DMA transfer begins, the HOLD latency time can be as great as 4 bus cycles. This will occur if a DMA word transfer operation is taking place from an odd address to an odd address. This is a total of 16 clocks or more, if WAIT states are required. In addition, if locked transfers are performed, the HOLD latency time will be increased by the length of the locked transfer.

Local Bus Controller and Reset

Upon receipt of a RESET pulse from the $\overline{RES}$ input, the local bus controller will perform the following actions:

- Drive $\overline{DEN}$, $\overline{RD}$, and $\overline{WR}$ HIGH for one clock cycle, then float.

  NOTE: $\overline{RE}$ is also provided with an internal pull-up device to prevent the processor from inadvertently entering Queue Status mode during reset.
- Drive $\overline{S0}$-$\overline{S2}$ to the passive state (all HIGH) and then float.
- Drive $\overline{LOCK}$ HIGH and then float.
- Tristate AD0-AD7, A8-A19, $\overline{S7}$, DT/$\overline{R}$
- Drive ALE LOW (ALE is never floated).
- Drive HLDA LOW.

INTERNAL PERIPHERAL INTERFACE

All the 80188 integrated peripherals are controlled via 16-bit registers contained within an internal 256-byte control block. This control block may be mapped into either memory or I/O space. Internal logic will recognize the address and respond to the bus cycle. During bus cycles to internal registers, the bus controller will signal the operation externally (i.e., the $\overline{RD}$, $\overline{WR}$, status, address, data, etc., lines will be driven as in a normal bus cycle), but D_{7-0}, SRDY, and ARDY will be ignored. The base address of the control block must be on an even 256-byte boundary (i.e., the lower 8 bits of the base address are all zeros). All of the defined registers within this control block may be read or written by the 80186 CPU at any time. The location of any register contained within the 256-byte control block is determined by the current base address of the control block.

The control block base address is programmed via a 16-bit relocation register contained within the control block at offset FEH from the base address of the control block (see Figure 8). It provides the upper 12 bits of the base address of the control block. Note that mapping the control register block into an address range corresponding to a chip-select range is not recommended (the chip select circuitry is discussed later in this date sheet). In addition, bit 12 of this register determines whether the control block will be mapped into I/O or memory space. If this bit is 1, the control block will be located in memory space, whereas if the bit is 0, the control block will be located in I/O space. If the control register block is mapped into I/O space, the upper 4 bits of the base address must be programmed as 0 (since I/O addresses are only 16 bits wide).

In addition to providing relocation information for the control block, the relocation register contains bits which place the interrupt controller into iRMX mode, and cause the CPU to interrupt upon encountering ESC instructions. At RESET, the relocation register is set to 20FFH. This causes the control block to start at FF00H in I/O space. An offset map of the 256-byte control register block is shown in Figure 9.

The integrated 80188 peripherals operate semiautonomously from the CPU. Access to them for the most part is via software read/write of the control and data locations in the control block. Most of these registers can be both read and written. A few dedicated lines, such as interrupts and DMA request provide real-time communication between the CPU and peripherals as in a more conventional system utilizing discrete peripheral blocks. The overall interaction and function of the peripheral blocks has not substantially changed.

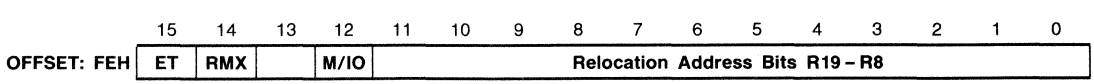
CHIP-SELECT/READY GENERATION LOGIC

The 80188 contains logic which provides programmable chip-select generation for both memories and peripherals. In addition, it can be programmed to provide READY (or WAIT state) generation. It can also provide latched address bits A1 and A2. The chip-select lines are active for all memory and I/O cycles in their programmed areas, whether they be generated by the CPU or by the integrated DMA unit.

Memory Chip Selects

The 80188 provides 6 memory chip select outputs for 3 address areas: upper memory, lower memory, and midrange memory. One each is provided for upper memory and lower memory, while four are provided for midrange memory.

The range for each chip select is user-programmable and can be set to 2K, 4K, 8K, 16K, 32K, 64K, 128K (plus 1K and 256K for upper and lower chip selects). In addition, the beginning or base address of the midrange memory chip select may also be selected. Only one chip select may be programmed to be active for any memory location at a time. All chip select sizes are in bytes, whereas 80188 memory is arranged in words. This means that if, for example, 16 64K x 1 memories are used, the memory block size will be 128K, not 64K.



ET = ESC Trap/No ESC Trap (1/0)
M/IO = Register block located in Memory / I/O Space (1/0)
RMX = Normal Interrupt Controller mode / IRMX compatible Interrupt Controller mode (0/1)

Figure 8. Relocation Register

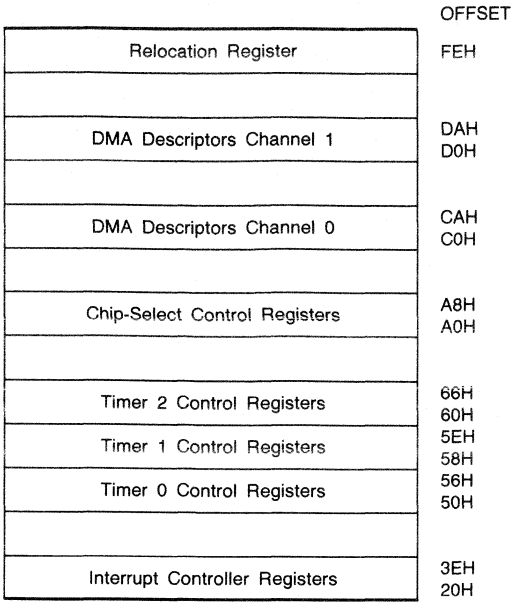


Figure 9. Internal Register Map

Upper Memory $\overline{CS}$

The 80188 provides a chip select, called $\overline{UCS}$, for the top of memory. The top of memory is usually used as the system memory because after reset the 80188 begins executing at memory location FFFF0H.

The upper limit of memory defined by this chip select is always FFFFFH, while the lower limit is programmable. By programming the lower limit, the size of the select block is also defined. Table 7 shows the relationship between the base address selected and the size of the memory block obtained.

Table 7. UMCS Programming Values

Starting Address (Base Address)	Memory Block Size	UMCS Value (Assuming R0 = R1 = R2 = 0)
FFC00	1K	FFF8H
FF800	2K	FFB8H
FF000	4K	FF38H
FE000	8K	FE38H
FC000	16K	FC38H
F8000	32K	F838H
F0000	64K	F038H
E0000	128K	E038H
C0000	256K	C038H

The lower limit of this memory block is defined in the UMCS register (see Figure 10). This register is at offset A0H in the internal control block. The legal values for bits 6-13 and the resulting starting address and memory block sizes are given in Table 7. Any combination of bits 6-13 not shown in Table 7 will result in undefined operation. After reset, the UMCS register is programmed for a 1K area. It must be reprogrammed if a larger upper memory area is desired.

Any internally generally 20-bit address whose upper 16-bits are greater than or equal to UMCS (with bits 0-5 "0") will cause UCS to be activated. UMCS bits R2-R0 are used to specify READY mode for the area of memory defined by this chip-select register, as explained below.

Lower Memory $\overline{CS}$

The 80188 provides a chip select for low memory called LCS. The bottom of memory contains the interrupt vector table, starting at location 00000H.

The lower limit of memory defined by this chip select is always 0H, while the upper limit is programmable. By programming the upper limit, the size of the memory block is also defined. Table 8 shows the relationship between the upper address selected and the size of the memory block obtained.

Table 8. LMCS Programming Values

Upper Address	Memory Block Size	LMCS Value (Assuming R0 = R1 = R2 = 0)
003FFH	1K	0038H
007FFH	2K	0078H
00FFFH	4K	00F8H
01FFFH	8K	01F8H
03FFFH	16K	03F8H
07FFFH	32K	07F8H
0FFFFH	64K	0FF8H
1FFFFH	128K	1FF8H
3FFFFH	256K	3FF8H

The upper limit of this memory block is defined in the LMCS register (see Figure 11). This register is at offset A2H in the internal control block. The legal values for bits 6-15 and the resulting upper address and memory block sizes are given in Table 8. Any combination of bits 6-15 not shown in Table 8 will result in undefined operation. After reset, the LMCS register value is undefined. However, the $\overline{LCS}$ chip-select line will not become active until the LMCS register is accessed.

Any internally generated 20-bit address whose upper 16 bits are less than or equal to LMCS (with bits 0-5 "1") will cause $\overline{LCS}$ to be active. LMCS register bits R2-R0 are used to specify the READY mode for the area of memory defined by this chip-select register.

Mid-Range Memory $\overline{CS}$

The 80188 provides four $\overline{MCS}$ lines which are active within a user-locatable memory block. This block can be located anywhere within the 80188 1M byte memory address space exclusive of the areas defined by $\overline{UCS}$ and $\overline{LCS}$. Both the base address and size of this memory block are programmable.

The size of the memory block defined by the midrange select lines, as shown in Table 9, is determined by bits 8-14 of the MPCS register (see Figure 12). This register is at location A8H in the internal control block. One and only one of bits 8-14 must be set at a time. Unpredictable operation of the $\overline{MCS}$ lines will otherwise occur. Each of the four chip-select lines is active for one of the four equal contiguous divisions of the mid-range block. Thus, if the total block size is 23K, each chip select is active for 8K of memory with $\overline{MCS0}$ being active for the first range and $\overline{MCS3}$ being active for the last range.

The EX and MS in MPCS relate to peripheral functionally as described in a later section.

Table 9. MMCS Programming Values

Total Block Size	Individual Select Size	MMCS Bits 14 – 8
8K	2K	0000001B
16K	4K	0000010B
32K	8K	0000100B
64K	16K	0001000B
128K	32K	0010000B
256K	64K	0100000B
512K	128K	1000000B

The base address of the mid-range memory block is defined by bits 15-9 of the MMCS register (see Figure 13). This register is at offset A6H in the internal control block. These bits correspond to bits A19-A13 of the 20-bit memory address. Bits A12-A0 of the base address area always 0. The base address may be set at any integer multiple of the size of the total memory block selected. For example, if the midrange block size is 32K (or the size of the block for which each $\overline{MCS}$ line is active is 8K), the block could be located at 10000H or 18000H, but not at 14000H, since the first few integer multiples of a 32K memory block are 0H, 8000H, 10000H, 18000H, etc. After reset, the contents of both of these registers is undefined. However, none of the $\overline{MCS}$ lines will be active until both the MMCS and MPCS registers are accessed.

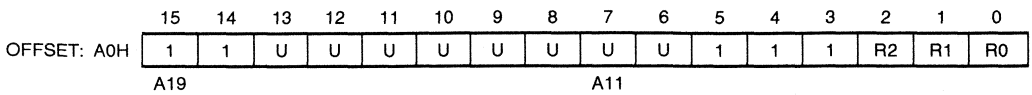


Figure 10. UMCS Register

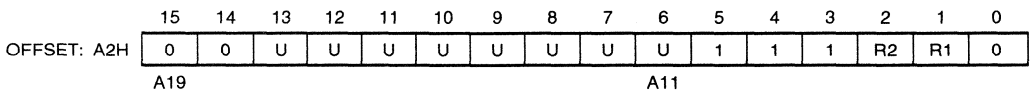


Figure 11. LMCS Register

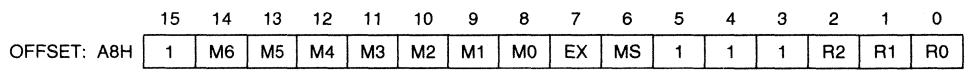


Figure 12. MPCS Register

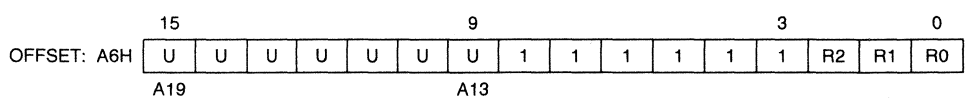


Figure 13. MMCS Register

MMCS bits R2-R0 specify READY mode of operation for all mid-range chip selects. All devices in mid-range memory must use the same number of WAIT states.

The 512K block size for the mid-range memory chip selects is a special case. When using 512K, the base address would have to be at either locations 00000H or 80000H. If it were to be programmed at 00000H when the $\overline{LCS}$ line was programmed, there would be an internal conflict between the $\overline{LCS}$ ready generation logic and the $\overline{MCS}$ ready generation logic. Likewise, if the base address were programmed at 80000H, there would be a conflict with the $\overline{UCS}$ ready generation logic. Since the $\overline{LCS}$ chip-select line does not become active until programmed, while the $\overline{UCS}$ line is active at reset, the memory base can be set only at 00000H. If this base address is selected, however, the $\overline{LCS}$ range must not be programmed.

Peripheral Chip Selects

The 80188 can generate chip selects for up to seven peripheral devices. These chip selects are active for seven contiguous blocks of 128 bytes above a programmable base address. This base address may be located in either memory or I/O space.

Seven $\overline{CS}$ lines called $\overline{PCS0}$-6 are generated by the 80188. The base address is user-programmable; however it can only be a multiple of 1K bytes, i.e., the least significant 10 bits of the starting address are always 0.

$\overline{PCS5}$ and $\overline{PCS6}$ can also be programmed to provide latched address bits A1, A2. If so programmed, they cannot be used as peripheral selects. These outputs can be connected directly to the A0, A1 pins used for selecting internal registers of 8-bit peripheral chips. This scheme simplifies the hardware interface because the 8-bit registers of peripherals are simply

treated as 16-bit registers located on even boundaries in I/O space or memory space where only the lower 8-bits of the register are significant: the upper 8-bits are "don't cares."

The starting address of the peripheral chip-select block is defined by the PACS register (see Figure 14). This register is located at offset A4H in the internal control block. Bits 15-6 of this register correspond to bits 19-10 of the 20-bit Programmable Base Address (PBA) of the peripheral chip-select block. Bits 9-0 of the PBA of the peripheral chip-select block are all zeros. If the chip-select block is located in I/O space, bits 12-15 must be programmed zero, since the I/O address is only 16 bits wide. Table 10 shows the address range of each peripheral chip select with respect to the PBA contained in PACS register.

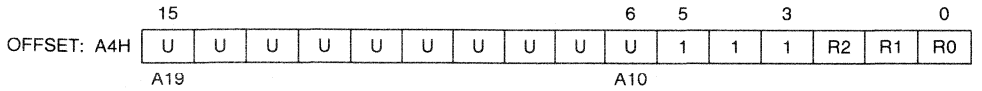


Figure 14. PACS Register

The user should program bits 15-6 to correspond to the desired peripheral base location. PACS bits 0-2 are used to specify READY mode for $\overline{PCS0}$-$\overline{PCS3}$.

Table 10. PCS Address Ranges

PCS Line	Active between Locations
PCS0	PBA — PBA + 127
PCS1	PBA + 128 — PBA + 255
PCS2	PBA + 256 — PBA + 383
PCS3	PBA + 384 — PBA + 511
PCS4	PBA + 512 — PBA + 639
PCS5	PBA + 640 — PBA + 767
PCS6	PBA + 768 — PBA + 895

The mode of operation of the peripheral chip selects is defined by the MPCS register (which is also used to set the size of the mid-range memory chip-select block, see Figure 15). This register is located at offset A8H in the internal control block. Bit 7 is used to select the function of $\overline{PCS5}$ and $\overline{PCS6}$, while bit 6 is used to select whether the peripheral chip selects are mapped into memory or I/O space. Table 11 describes the programming of these bits. After reset, the contents of both the MPCS and the PACS registers are undefined, however none of the PCS lines will be active until both of the MPCS and PACS registers are accessed.

Table 11. MS, EX Programming Values

Bit	Description
MS	1 = Peripherals mapped into memory space. 0 = Peripherals mapped into I/O space.
EX	0 = 5 $\overline{PCS}$ lines. A1, A2 provided. 1 = 7 $\overline{PCS}$ lines. A1, A2 are not provided.

MPCS bits 0-2 are used to specify READY mode for $\overline{PCS4}$-$\overline{PCS6}$ as outlined below.

READY Generation Logic

The 80188 can generate a "READY" signal internally for each of the memory or peripheral $\overline{CS}$ lines. The number of WAIT states to be inserted for each peripheral or memory is programmable to provide 0-3 wait states for all accesses to the area for which the chip select is active. In addition, the 80188 may be programmed to either ignore external READY for each chip- select range individually or to factor external READY with the integrated ready generator.

READY control consists of 3 bits for each $\overline{CS}$ line or group of lines generated by the 80188. The interpretation of the ready bits is shown in Table 12.

Table 12. READY Bits Programming

R2	R1	R0	Number of WAIT States Generated
0	0	0	0 wait states, external RDY also used.
0	0	1	1 wait state inserted, external RDY also used.
0	1	0	2 wait states inserted, external RDY also used.
0	1	1	3 wait states inserted, external RDY also used.
1	0	0	0 wait states, external RDY ignored.
1	0	1	1 wait state inserted, external RDY ignored.
1	1	0	2 wait states inserted, external RDY ignored.
1	1	1	3 wait states inserted, external RDY ignored.

The internal ready generator operates in parallel with external READY, not in series if the external READY is used (R2 = 0). This means, for example, if the internal generator is set to insert two wait states, but activity on the external READY lines will insert four wait states, the processor will only insert four wait states, not six. This is because the two wait states generated by the internal generator overlapped the first two wait states generated by the external ready signal. Note that the external ARDY and SRDY lines are always ignored during cycles accessing internal peripherals.

R2-R0 of each control word specifies the READY mode for the corresponding block, with the exception of the peripheral chip selects: R2-R0 of PACS set the $\overline{PCS0}$-3 READY mode, R2-R0 of MPCS set the $\overline{PCS4}$-6 READY mode.

Chip Select/Ready Logic and Reset

Upon reset, the Chip-Select/Ready Logic will perform the following actions:

- All chip-select outputs will be driven HIGH.
- Upon leaving RESET, the UCS line will be programmed to provide chip selects to a 1K block with the accompanying READY control bits set at 011 to allow the maximum number of internal wait states in conjunction with external Ready consideration (i.e. UMCS resets to FFFBH).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET: A8H	1	M6	M5	M4	M3	M2	M1	M0	EX	MS	1	1	1	R2	R1	R0

Figure 15. MPCS Register

- No other chip select or READY control registers have any predefined values after RESET. They will not become active until the CPU accesses their control registers. Both the PACS and MPCS registers must be accessed before the $\overline{PCS}$ lines will become active.

DMA CHANNELS

The 80188 DMA controller provides two independent high-speed DMA channels. Data transfers can occur between memory and I/O spaces (e.g., Memory to I/O) or within the same space (e.g., Memory to Memory or I/O to I/O). Data can be transferred either in bytes (8 bits) or in words (16 bits) to or from even or odd addresses. Each DMA channel maintains both a 20-bit source and destination pointer which can be optionally incremented or decremented after each data transfer (by one or two depending on byte or word transfers). Each data transfer consumes 2 bus cycles (a minimum of 8 clocks), one cycle to fetch data and the other to store data. This provides a maximum data transfer rate of one Mword/sec or 2 MBytes/sec.

DMA Operation

Each channel has six registers in the control block which define each channel's specific operation. The control registers consist of a 20-bit Source pointer (2 words), a 20-bit Destination pointer (2 words), a 16-bit Transfer Counter, and a 16-bit Control Word. The format of the DMA Control Blocks is shown in Table 13. The Transfer Count Register (TC) specifies the number of DMA transfers to be performed. Up to 64K byte or word transfers can be performed with automatic termination. The Control Word defines the channel's operation (see Figure 17). All registers may be modified or altered during any DMA activity. Anyl changes made to these registers will be reflected immediately in DMA operation.

Table 13. DMA Control Block Format

Register Name	Register Address	
	Ch. 0	Ch. 1
Control Word	CAH	DAH
Transfer Count	C8H	D8H
Destination Pointer (upper 4 bits)	C6H	D6H
Destination Pointer	C4H	D4H
Source Pointer (upper 4 bits)	C2H	D2H
Source Pointer	C0H	D0H

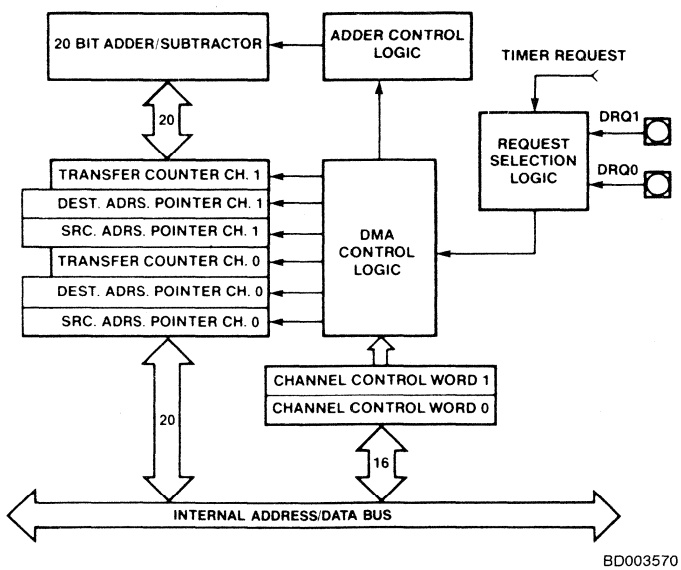


Figure 16. DMA Unit Block Diagram

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M/$\overline{IO}$	DESTINATION DEC	INC	M/$\overline{IO}$	SOURCE DEC	INC	TC	INT	SYN		P	TDRQ	X	CHG/$\overline{NOCHG}$	ST/$\overline{STOP}$	$\overline{B}$/W

X = DON'T CARE

Figure 17. DMA Control Register

DMA Channel Control Word Register

Each DMA Channel Control Word determines the mode of operation for the particular 80188 DMA channel. This register specifies:

- the mode of synchronization;
- whether bytes or words will be transferred;
- whether interrupts will be generated after the last transfer;
- whether DMA activity will cease after a programmed number of DMA cycles;
- the relative priority of the DMA channel with respect to the other DMA channel;
- whether the source pointer will be incremented, decremented, or maintained constant after each transfer;
- whether the source pointer addresses memory or I/O space;
- whether the destination pointer will be incremented, decremented, or maintained constant after each transfer; and
- whether the destination pointer will address memory or I/O space.

The DMA channel control registers may be changed while the channel is operating. However, any changes made during operation will affect the current DMA transfer.

DMA Control Word Bit Descriptions

$\overline{B}$/W: Byte/Word (0/1) Transfers.

ST/$\overline{STOP}$: Start/stop (1/0) Channel.

CHG/$\overline{NOCHG}$: Change/Do not change (1/0) ST/$\overline{STOP}$ bit. If this bit is set when writing to the control word, the ST/$\overline{STOP}$ bit will be programmed by the write to the control word. If this bit is cleared when writing the control word, the ST/$\overline{STOP}$ bit will not be altered. This bit is not stored; it will always be a 0 on read.

INT: Enable Interrupts to CPU on byte count termination.

TC: If set, DMA will terminate when the contents of the Transfer Count register reach zero. The ST/$\overline{STOP}$ bit will also be reset at this point if TC is set. If this bit is cleared, the DMA unit will decrement the transfer count register for each DMA cycle, but the DMA transfer will not stop when the contents of the TC register reach zero.

SYN: (2 bits)
- 00 No synchronization
  NOTE: The ST bit will be cleared automatically when the contents of the TC register reach zero regardless of the state of the bit.
- 01 Source synchronization.
- 10 Destination synchronization.
- 11 Unused.

SOURCE: INC Increment source pointer by 1 or 2 (depends on $\overline{B}$/W) after each transfer.

M/$\overline{IO}$ Source pointer is in M/IO space (1/0).

DEC Decrement source pointer by 1 or 2 (depends on $\overline{B}$/W) after each transfer.

DEST: INC Increment destination pointer by 1 or 2 ($\overline{B}$/W) after each transfer.

M/$\overline{IO}$ Destination pointer is in M/IO space (1/0).

DEC Decrement destination pointer by 1 or 2 (depending on $\overline{B}$/W) after each transfer.

P Channel priority-relative to other channel.
0 low priority.
1 high priority.
Channels will alternate cycles if both set at same priority level.

TDRQ 0: Disable DMA requests from timer 2.
1: Enable DMA requests from timer 2.

Bit 3 Bit 3 is not used.

If both INC and DEC are specified for the same pointer, the pointer will remain constant after each cycle.

DMA Destination and Source Pointer Registers

Each DMA channel maintains a 20-bit source and a 20-bit destination pointer. Each of these pointers takes up two full 16-bit registers in the peripheral control block. The lower four bits of the upper register contain the upper four bits of the 20-bit physical address (see Figure 18). These pointers may be individually incremented or decremented after each transfer. If word transfers are performed the pointer is incremented or decremented by two. Each pointer may point into either memory or I/O space. Since the DMA channels can perform transfers to or from odd addresses, there is no restriction on values for the pointer registers. Higher transfer rates can be obtained if all word transfers are performed to even ad-

dresses, since this will allow data to be accessed in a single memory access.

DMA Transfer Count Register

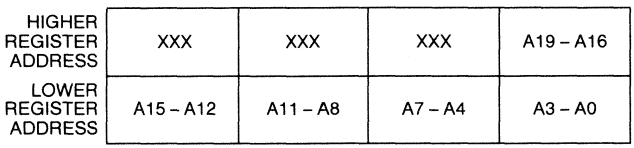
Each DMA channel maintains a 16-bit transfer count register (TC). This register is decremented after every DMA cycle, regardless of the state of the TC bit in the DMA Control Register. If the TC bit in the DMA control word is set, however, DMA activity will terminate when the transfer count register reaches zero.

DMA Requests

Data transfers may be either source or destination synchronized, that is either the source of the data or the destination of the data may request the data transfer. In addition, DMA transfers may be unsynchronized; that is, the transfer will take place continually until the correct number of transfers has occurred. When source or unsynchronized transfers are performed, the DMA channel may begin another transfer immediately after the end of a previous DMA transfer. This allows complete transfer to take place every 2 bus cycles or eight clock cycles (assuming no wait states). No prefetching occurs when destination synchronization is performed, however. Data will not be fetched from the source address until the destination device signatls that it is ready to receive it. When destination synchronized transfers are requested, the DMA controller will relinquish control of the bus after every transfer. If no other bus activity is initiated, another DMA cycle will begin after two processor clocks. This is done to allow the destination device time to remove its request if another transfer is not desired. Since the DMA controller will relinquish the bus, the CPU can initiate a bus cycle. As a result, a complete bus cycle will often be inserted between destination synchronized transfers. These lead to the maximum DMA transfer rates shown in Table 14.

Table 14. Maximum DMA Transfer Rates with 8 MHz 80188

Type of Synchronization Selected	CPU Running	CPU Halted
Unsynchronized	1 Mbytes/sec	1 Mbytes/sec
Source Synch	1 Mbytes/sec	1 Mbytes/sec
Destination Synch	.65 Mbytes/sec	.75 Mbytes/sec



XXX = DON'T CARE

Figure 18. DMA Memory Pointer Register Format

DMA Acknowledge

No explicit DMA acknowledge pulse is provided. Since both source and destination pointers are maintained, a read from a requesting source, or a write to a requesting destination, should be used as the DMA acknowledge signal. Since the chip-select lines can be programmed to be active for a given block of memory or I/O space, and the DMA pointers can be programmed to point to the same given block, a chip-select line could be used to indicate a DMA acknowledge.

DMA Priority

The DMA channels may be programmed such that one channel is always given priority over the other, or they may be programmed such as to alternate cycles when both have DMA requests pending. DMA cycles always have priority over internal CPU cycles except between locked memory accesses or word accesses the odd memory locations; however, an external bus hold takes priority over an internal DMA cycle. Because an interrupt request cannot suspend a DMA operation and the CPU cannot access memory during a DMA cycle, interrupt latency time will suffer during sequences of continuous DMA cycles. An NMI request, however, will cause all internal DMA activity to halt. This allows the CPU to quickly respond to the NMI request.

DMA Programming

DMA cycles will occur whenever the ST/STOP bit of the Control Register is set. If synchronized transfers are programmed, a DRQ must also have been generated. Therefore, the source and destination transfer pointers, and the transfer count register (if used) must be programmed before this bit is set.

Each DMA register may be modified while the channel is operating. If the CHG/NOCHG bit is cleared when the control register is written, the ST/STOP bit of the control register will not be modified by the write. If multiple channel registers are modified, it is recommended that a LOCKED string transfer be used to prevent a DMA transfer from occurring between updates to the channel registers.

DMA Channels and Reset

Upon RESET, the DMA channels will perform the following actions:

- The Start/Stop bit for each channel will be rest to STOP.
- Any transfer in progress is aborted.

TIMERS

The 80188 provides three internal 16-bit programmable timers (see Figure 19). Two of these are highly flexible and are connected to four external pins (2 per timer). They can be used to count external events, time external events, generate nonrepetitive waveforms, etc. The third timer is not connected to any external pins, and is useful for real-time coding and time delay applications. In addition, this third timer can be used as a prescaler to the other two, or as a DMA request source.

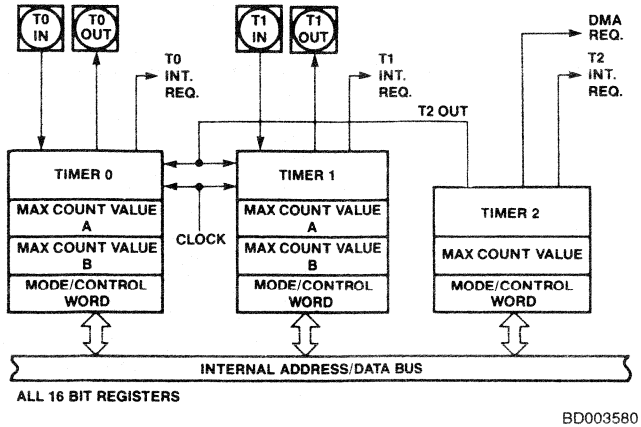


Figure 19. Timer Block Diagram

Timer Operation

The timers are controlled by 11 16-bit registers in the internal peripheral control block. The configuration of these registers is shown in Table 15. The count register contains the current value of the timer. It can be read or written at any time independent of whether the timer is running or not. The value of this register will be incremented for each timer event. Each of the timers is equipped with a MAX COUNT register, which defines the maximum count the timer will reach. After reaching the MAX COUNT register value, the timer count value will reset to zero during that same clock, i.e., the maximum count value is never stored in the count register itself. Timers 0 and 1 are, in addition, equipped with a second MAX COUNT register, which enables the timers to alternate their count between two different MAX COUNT values programmed by the user. If a single MAX COUNT register is used, the timer output pin will switch LOW for a single clock, 2 clocks after the maximum count value has been reached. In the dual MAX COUNT register mode, the output pin will indicate which MAX COUNT register is currently in use, thus allowing nearly complete freedom in selecting waveform duty cycles. For the timers with two MAX COUNT registers, the RIU bit in the control register determines which is used for the comparison.

Each timer gets serviced every fourth CPU-clock cycle, and thus can operate at speeds up to one-quarter the internal clock frequency (one-eighth the crystal rate). External clocking of the timers may be done at up to a rate of one-quarter of the internal CPU-clock rate (2 MHz for an 8 MHz CPU clock). Due to internal synchronization and pipelining of the timer circuitry, a timer output may take up to 6 clocks to respond to any individual clock or gate input.

Since the count registers and the maximum count registers are all 16-bits wide, 16 bits of resolution are provided. Any Read or Write access to the timers will add one wait state to the minimum four-clock bus cycle. However, this is needed to synchronize and coordinate the internal data flows between the internal timers and the internal bus.

The timers have several programmable options.

- All three timers can be set to halt or continue on a terminal count.
- Timers 0 and 1 can select between internal and external clocks, alternate between MAX COUNT registers and be set to retrigger on external events.
- The timers may be programmed to cause an interrupt on terminal count.

These options are selectable via the timer mode/control word.

Timer Mode/Control Register

The mode/control register (see Figure 20) allows the user to program the specific mode of operation or check the current programmed status for any of the three integrated timers.

Table 15. Timer Control Block format

Register Name	Register Offset		
	Tmr. 0	Tmr. 1	Tmr. 2
Mode/Control Word	56H	5EH	66H
Max Count B	54H	5CH	not present
Max Count A	52H	5AH	62H
Count Register	50H	58H	60H

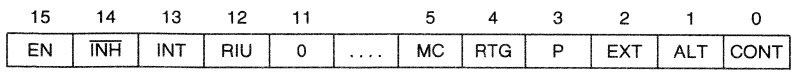


Figure 20. Timer Mode/Control Register

ALT:

The ALT bit determines which of two MAX COUNT registers is used for count comparison. If ALT = 0, register A for that timer is always used, while if ALT = 1, the comparison will alternate between register A and register B when each maximum count is reached. This alternation allows the user to change one MAX COUNT register while the other is being used, and thus provides a method of generating nonrepetitive waveforms. Square waves and pulse outputs of any duty cycle are a subset of available signals obtained by not changing the final count registers. The ALT bit also determines the function of the timer output pin. If ALT is zero, the output pin will go LOW for one clock, the clock after the maximum count is reached, If ALT is one, the output pin will reflect the current MAX COUNT register being used (0/1 for B/A).

CONT:

Setting the CONT bit causes the associated timer to run continuously, while resetting it causes the timer to halt upon maximum count. If CONT = 0 and ALT = 1, the timer will count to the MAX COUNT register A value, reset, count to the register B value, reset, and halt.

EXT:

The external bit selects between internal and external clocking for the timer. The external signal may be asynchronous with respect to the 80188 clock. If this bit is set, the timer will count LOW-to-HIGH transitions on the input pin. If cleared, it will count an internal clock while using the input pin for control. In this mode, the function of the external pin is defined by the RTG bit. The maximum input to output transition latency time may be as much as 6 clocks. However, clock inputs may be pipelined as closely together as every 4 clocks without losing clock pulses.

P:

The prescaler bit is ignored unless internal clocking has been selected (EXT = 0). If the P bit is a zero, the timer will count at one-fourth the internal CPU clock rate. If the P bit is a one, the output of timer 2 will be used as a clock for the timer. Note that the user must initialize and start timer 2 to obtain the prescaled clock.

RTG:

Retrigger bit is only active for internal clocking (EXT = 0). In this case it determines the control function provided by the input pin.

If RTG = 0, the input level gates the internal clock on and off. If the input pin is HIGH, the timer will count; if the input pin is LOW, the timer will hold its value. As indicated previously, the input signal may be asynchronous with respect to the 80188 clock.

When RTG = 1, the input pin detects LOW-to-HIGH transitions. The first such transition starts the timer running, clearing the timer value to zero on the first clock, and then incrementing thereafter. Further transitions on the input pin will again reset the timer to zero, from which it will start counting up again. If CONT = 0, when the timer has reached maximum count, the EN bit will be cleared, inhibiting further timer activity.

EN:

The enable bit provides programmer control over the timer's RUN/HALT status. When set, the timer is enabled to increment subject to the input pin constraints in the internal clock mode (discussed previously). When cleared, the timer will be inhibited from counting. All input pin transitions during the time EN is zero will be ignored. If CONT is zero, the EN bit is automatically cleared upon maximum count.

INH:

The inhibit bit allows for selective updating of the enable (EN) bit. If INH is a one during the write to the mode/control word, then the state of the EN bit will be modified by the write. If INH is a zero during the write, the EN bit will be unaffected by the operation. This bit is not stored; it will always be a 0 on a read.

INT:

When set, the INT bit enables interrupts from the timer, which will be generated on every terminal count. If the timer is configured in dual MAX COUNT register mode, an interrupt will be generated each time the value in MAX COUNT register A is reached, and each time the value in MAX COUNT register B is reached. If this enable bit is cleared after the interrupt request has been generated, but before a pending interrupt is serviced, the interrupt request will still be in force. (The request is latched in the Interrupt Controller.)

MC:

The Maximum Count bit is set whenever the timer reaches its final maximum count value. If the timer is configured in dual MAX COUNT register mode, this bit will be set each time the value in MAX COUNT register A is reached, and each time the value in MAX COUNT register B is reached. This bit is set regardless of the timer's interrupt-enable bit. The MC bit gives the user the ability to monitor timer status through software instead of through interrupts.

RIU:

The Register In Use bit indicates which MAX COUNT register is currently being used for comparison to the timer count value. A zero value indicates register A. The RIU bit cannot be written, i.e., its value is not affected when the control register is written. It is always cleared when the ALT bit is zero.

Not all mode bits are provided for timer 2. Certain bits are hardwired as indicated below:

ALT = 0, EXT = 0, P = 0, RTG = 0, RIU = 0

Count Registers

Each of the three timers has a 16-bit count register. The current contents of this register may be read or written by the processor at any time. If the register is written into while the timer is counting, the new value will take effect in the current count cycle.

Max Count Registers

Timers 0 and 1 have two MAX COUNT registers, while timer 2 has a single MAX COUNT register. These contain the number of events the timer will count. In timers 0 and 1, the MAX COUNT register used can alternate between the two max count values whenever the current maximum count is reached. The condition which causes a timer to reset is equivalent between the current count value and the max count being used. This means that if the count is changed to be above the max count value, or if the max count value is changed to be below the current value, the timer will not reset to zero, but rather will count to its maximum value, "wrap around" to zero, then count until the max count is reached.

Timers and Reset

Upon RESET, the Timers will perform the following actions:

- All EN (Enable) bits are reset preventing timer counting.
- All SEL (Select) bits are reset to zero. This selects MAX COUNT register A, resulting in the Timer Out pins going HIGH upon RESET.

INTERRUPT CONTROLLER

The 80188 can receive interrupts from a number of sources, both internal and external. The internal interrupt controller serves to merge these requests on a priority basis, for individual service by the CPU. Internal interrupt sources (Timers and DMA channels) can be disabled by their own control registers or by mask bits within the interrupt controller. The 80188 interrupt controller has its own control registers that set the mode of operation for the controller.

The interrupt controller will resolve priority among requests that are pending simultaneously. Nesting is provided so interrupt service routines for lower priority interrupts may themselves be interrupted by higher priority interrupts. A block diagram of the interrupt controller is shown in Figure 21.

The interrupt controller has a special iRMX 86 compatibility mode that allows the use of the 80188 within the iRMX 86 operating system interrupt structure. The controller is set in this mode by setting bit 14 in the peripheral control block relocation register (see iRMX 86 Compatibility Mode section). In this mode, the internal 80188 interrupt controller functions as a "slave" controller to an external "master" controller. Special initialization software must be included to properly set up the 80188 interrupt controller in iRMX 86 mode.

MASTER (NON-iRMX) MODE OPERATION

Interrupt Controller External Interface

For external interrupt sources, five dedicated pins are provided. One of these pins is dedicated to NMI, non-maskable interrupt. This is typically used for power-fail interrupts, etc. The other four pins may function either as four interrupt input lines with internally generated interrupt vectors, as an interrupt line and an interrupt acknowledge line (called the "cascade mode") along with two other input lines with internally generated interrupt vectors, or as two interrupt input lines and two dedicated interrupt acknowledge ouput lines. When the interrupt lines are configured in cascade mode, the 80188 interrupt controller will not generate internal interrupt vectors.

External sources in the cascade mode use externally generated interrupt vectors. When an interrupt is acknowledged, two $\overline{\text{INTA}}$ cycles are initiated and the vector is read into the 80188 on the second cycle. The capability to interface to external 8259A programmable interrupt controllers is thus provided when the inputs are configured in cascade mode.

Interrupt Controller Modes of Operation

The basic modes of operation of the interrupt controller in non-iRMX mode are similar to the 8259A. The interrupt controller responds identically to internal interrupts in all three modes: the difference is only in the interpretation of function of the four external interrupt pins. The interrupt controller is set into one of these three modes by programming the correct bits in the INT0 and INT1 control registers. The modes of interrupt controller operation are as follows:

Fully Nested Mode

When in the fully nested mode four pins are used as direct interrupt requests. The vectors for these four inputs are generated internally. An in-service bit is provided for every interrupt source. If a lower-priority device requests an interrupt while the in-service bit (IS) is set, no interrupt will be generated by the interrupt controller. In addition, if another interrupt request occurs from the same interrupt source while the inservice bit is set, no interrupt will be generated by the interrupt controller. This allows interrupt service routines to operate with interrupts enabled without being themselves interrupted by lower-priority interrupts. Since interrupts are enabled, higher-priority interrupts will be serviced.

When a service routine is completed, the proper IS bit must be reset by writing the proper pattern to the EOI register. This is required to allow subsequent interrupts from this interrupt source and to allow servicing of lower-priority interrupts. An EOI command is issued at the end of the service routine just before the issuance of the return from interrupt instruction. If the fully nested structure has been upheld, the next highest-priority source with its IS bit set is then serviced.

Cascade Mode

The 80188 has four interrupt pins and two of them have dual functions. In the fully nested mode the four pins are used as direct interrupt inputs and the corresponding vectors are generated internally. In the cascade mode, the four pins are configured into interrupt input-dedicated acknowledge signal pairs. The interconnection is shown in Figure 22. INT0 is an interrupt input interfaced to an 8259A, while INT2/$\overline{\text{INTA0}}$ serves as the dedicated interrupt acknowledge signal to that peripheral. The same is true for INT1 and INT3/$\overline{\text{INTA1}}$. Each pair can selectively be placed in the cascade or non-cascade mode by programming the proper value into INT0 and INT1 control registers. The use of the dedicated acknowledge signals eliminates the need for the use of external logic to generate $\overline{\text{INTA}}$ and device select signals.

The primary cascade mode allows the capability to serve up to 128 external interrupt sources through the use of external master and slave 8259As. Three levels of priority are created, requiring priority resolution in the 80188 interrupt controller, the master 8259As, and the slave 8259As. If an external interrupt is serviced, one IS bit is set at each of these levels. When the interrupt service routine is completed, up to three end-of-interrupt commands must be issued by the programmer.

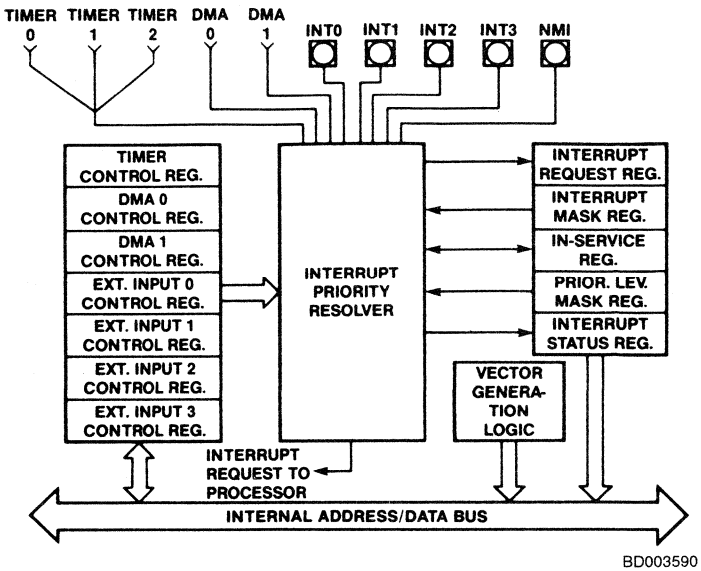


Figure 21. Interrupt Controller Block Diagram

Special Fully Nested Mode

This mode is entered by setting the SFNM bit in INT0 or INT1 control register. It enables complete nestability with external 8259A masters. Normally, an interrupt request from an interrupt source will not be recognized unless the in-service bit for that source is reset. If more than one interrupt source is connected to an external interrupt controller, all of the interrupts will be funneled through the same 80188 interrupt request pin. As a result, if the external interrupt controller receives a higher-priority interrupt, its interrupt will not be recognized by the 80188 controller until the 80188 in-service bit is reset. In special fully nested mode, the 80188 interrupt controller will allow interrupts from an external pin regardless of the state of the in-service bit for an interrupt source in order to allow multiple interrupts from a single pin. An in-service bit will continue to be set, however, to inhibit interrupts from other lower-priority 80188 interrupt sources.

Special procedures should be followed when resetting IS bits at the end of interrupt service routines. Software polling of the external master's IS register is required to determine if there is more than one bit set. If so, the IS bit in the 80188 remains active and the next interrupt service routine is entered.

Operation in a Polled Environment

The controller may be used in a polled mode if interrupts are undesirable. When polling, the processor disables interrupts and then polls the interrupt controller whenever it is convenient. Polling the interrupt controller is accomplished by reading the Poll Word (Figure 9). bit 15 in the poll word indicates to the processor that an interrupt of high enough priority is requesting service. Reading the Poll Word causes the In-Service bit of the highest-priority source to be set.

It is desirable to be able to read the Poll Word information without guaranteeing service of any pending interrupt, i.e., not set the indicated in-service bit. The 80188 provides a Poll Status Word in addition to the conventional Poll Word to allow this to be done. Poll Word information is duplicated in the Poll Status Word, but reading the Poll Status Word does not set the associated in-service bit. These words are located in two adjacent memory locations in the register file.

Master (NON-iRMX) Mode Features

Programmable Priority

The user can program the interrupt sources into any of eight different priority levels. The programming is done by placing a 3-bit priority level (0–7) in the control register of each interrupt source. (A source with a priority level of 4 has higher priority over all priority levels from 5 to 7. Priority registers containing values lower than 4 have greater priority.) All interrupt sources have preprogrammed default priority levels (see Table 4).

If two requests with the same programmed priority level are pending at once, the priority ordering scheme shown in Table 4 is used. If the serviced interrupt routine reenables interrupts, it allows other requests to be serviced.

End-of-Interrupt Command

The end-of-interrupt (EOI) command is used by the programmer to reset the In-Service (IS) bit when an interrupt service routine is completed. The EOI command is issued by writing the proper pattern to the EOI register. There are two types of EOI commands, specific and nonspecific. The nonspecific command does not specify which IS bit is reset. When issued, the interrupt controller automatically resets the IS bit of the highest priority source with an active service routine. A specific EOI command requires that the programmer send the interrupt vector type to the interrupt controller indicating which source's IS bit is to be reset. This command is used when the fully nested structure has been disturbed or the highest priority IS bit that was set does not belong to the service routine in progress.

Trigger Mode

The four external interrupt pins can be programmed in either edge- or level-trigger mode. The control register for each external source has a level-trigger mode (LTM) bit. All interrupt inputs are active HIGH. In the edge sense mode or the level-trigger mode the interrupt request must remain active (HIGH) until the interrupt request is acknowledged by the 80188 CPU. In the edge-sense mode, if the level remains high after the interrupt is acknowledged, the input is disabled and no further requests will be generated. The input level must go LOW for at least one clock cycle to reenable the input. In the level-trigger mode, no such provision is made: holding the interrupt input HIGH will cause continuous interrupt requests.

Interrupt Vectoring

The 80188 Interrupt Controller will generate interrupt vectors for the integrated DMA channels and the integrated Timers. In addition, the Interrupt Controller will generate interrupt vectors for the external interrupt lines if they are not configured in Cascade or Special Fully Nested Mode. The interrupt vectors generated are fixed and cannot be changed (see Table 4).

Interrupt Controller Registers

The Interrupt Controller register model is shown in Figure 23. It contains 15 registers. All registers can both be read or written unless specified otherwise.

In-Service Register

This register can be read from or written into. The format is shown in Figure 24. It contains the In-Service bit for each of the interrupt sources. The In-Service bit for each of the interrupt sources. The In-Service bit is set to indicate that a source's service routine is in progress. When an In-Service bit is set, the interrupt controller will not generate interrupts to the CPU when it receives interrupt requests from devices with a lower programmed priority level. The TMR bit is the In-Service bit for all three timers; the D0 and D1 bits are the In-Service bits for the two DMA channels; the 10–13 are the In-Service bits for the external interrupt pins. The IS bit is set when the processor acknowledges an interrupt request either by an interrupt acknowledge or by reading the poll register. The IS bit is reset at the end of the interrupt service routine by an end-of-interrupt command issued by the CPU.

Interrupt Request Register

The internal interrupt sources have interrupt request bits inside the interrupt controller. The format of this register is shown in Figure 24. A read from this register yields the status of these bits. The TMR bit is the logical OR of all timer interrupt requests. D0 and D1 are the interrupt request bits for the DMA channels.

The state of the external interrupt input pins is also indicated. The state of the external interrupt pins is not a stored condition inside the interrupt controller, therefore the external interrupt bits cannot be written. The external interrupt request bits show exactly when an interrupt request is given to the interrupt controller, so if edge-triggered mode is selected, the bit in the register will be HIGH only after an inactive-to-active transition. For internal interrupt sources, the register bits are set when a request arrives and are reset when the processor acknowledges the requests.

Mask Register

This is a 16-bit register that contains a mask bit for each interrupt source. The format for this register is shown in Figure 24. A one in a bit position corresponding to a particular source serves to mask the source from generating interrupts. These mask bits are the exact same bits which are used in the individual control registers; programming a mask bit using the mask register will also change this bit in the individual control registers, and vice versa.

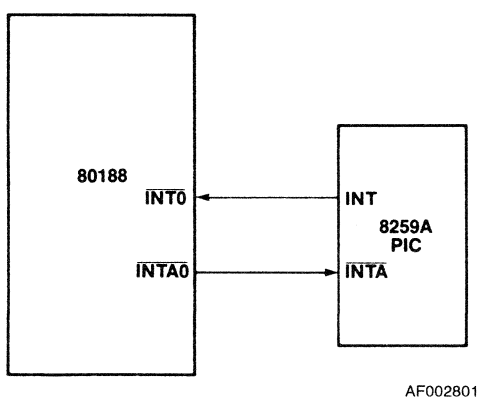


Figure 22. Cascade Mode Interrupt Connection

	OFFSET
INT3 CONTROL REGISTER	3EH
INT2 CONTROL REGISTER	3CH
INT1 CONTROL REGISTER	3AH
INT0 CONTROL REGISTER	38H
DMA 1 CONTROL REGISTER	36H
DMA 0 CONTROL REGISTER	34H
TIMER CONTROL REGISTER	32H
INTERRUPT CONTROLLER STATUS REGISTER	30H
INTERRUPT REQUEST REGISTER	2EH
IN-SERVICE REGISTER	2CH
PRIORITY MASK REGISTER	2AH
MASK REGISTER	28H
POLL STATUS REGISTER	26H
POLL REGISTER	24H
EOI REGISTER	22H

Figure 23. Interrupt Controller Registers (Non-iRMX 86 Mode)

Priority Mask Register

This register is used to mask all interrupts below particular interrupt priority levels. The format of this register is shown in Figure 25. The code in the lower three bits of this register inhibits interrupts of priority lower (a higher priority number) than the code specified. For example, 100 written into this register masks interrupts of level five (101), six (110), and seven (111). The register is reset to seven (111) upon RESET so all interrupts are unmasked.

Interrupt Status Register

This register contains general interrupt controller status information. The format of this register is shown in Figure 26. The bits in the status register have the following functions:

DHLT: DMA Halt Transfer; setting this bit halts all DMA transfers. It is automatically set whenever a non-maskable interrupt occurs, and it is reset when an IRET instruction is executed. The purpose of this bit is to allow prompt service of all non-maskable interrupts. This bit may also be set by the CPU.

IRTx: These three bits represent the individual timer interrupt request bits. These bits are used to differentiate the timer interrupts, since the timer IR bit in the interrupt request register is the "OR" function of all timer interrupt requests. Note that setting any one of these three bits initiates an interrupt request to the interrupt controller.

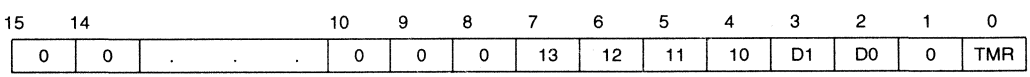


Figure 24. In-Service, Interrupt Request, and Mask Register Formats

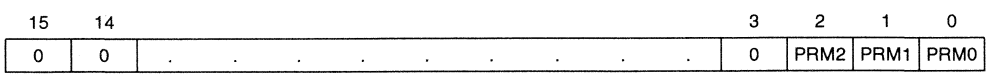


Figure 25. Priority Mask Register Format

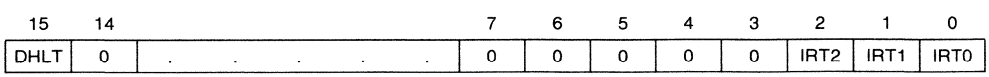


Figure 26. Interrupt Status Register Format

Timer, DMA 0, 1; Control Registers

These registers are the control words for all the internal interrupt sources. The format for these registers is shown in Figure 27. The three bit positions PR0, PR1, and PR2 represent the programmable priority level of the interrupt source. The MSK bit inhibits interrupt requests from the interrupt source. The MSK bits in the individual control registers are the exact same bits as are in the Mask Register; modifying them in the individual control registers will also modify them in the Mask Register, and vice versa.

INT0–INT3 Control Registers

These registers are the control words for the four external input pins. Figure 28 shows the format of the INT0 and INT1 Control registers; Figure 29 shows the format of the INT2 and INT3 Control registers. In cascade mode or special fully nested mode, the control words for INT2 and INT3 are not used.

The bits in the various control registers are encoded as follows:

PRO-2: Priority programming information. Highest priority = 000, lowest priority = 111.

LTM: Level-trigger mode bit. 1 = level-triggered; 0 = edge-triggered. Interrupt Input levels are active high. In level-triggered mode, an interrupt is generated whenever the external line is high. In edge-triggered mode, an interrupt will be generated only when this level is preceded by an inactive-to-active transition on the line. In both cases, the level must remain active until the interrupt is acknowledged.

MSK: Mask bit, 1 = mask; 0 = nonmask.

C: Cascade mode bit, 1 = cascade; 0 = direct

SFNM: Special fully nested mode bit, 1 = SFNM; 0 = normal nested mode.

EOI Register

The end of the interrupt register is a command register which can only be written into. The format of this register is shown in Figure 30. It initiates an EOI command when written to by the 80188 CPU.

The bits in the EOI register are encoded as follows:

S_X: Encoded information that specifies an interrupt source vector type as shown in Table 4. For example, to reset the In-Service bit for DMA channel 0, these bits should be set to 01010, since the vector type for DMA channel 0 is 10. Note that to reset the single In-Service bit for any of the three timers, the vector type for timer 0(8) should be written in this register.

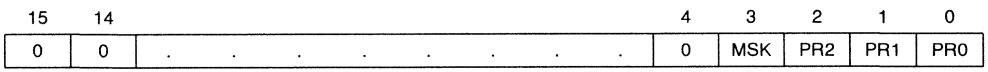


Figure 27. Timer/DMA Control Register Formats

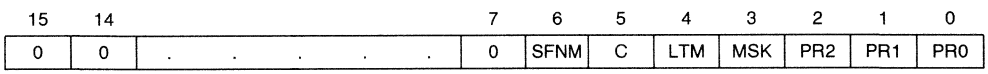


Figure 28. INT0/INT1 Control Register Formats

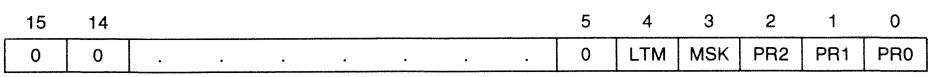


Figure 29. INT2/INT3 Control Register Formats

NSPEC/: A bit that determines the type of EOI command.

SPEC Nonspecific = 1, Specific = 0.

Poll and Poll Status Registers

These registers contain polling information. The format of these registers is shown in Figure 31. They can only be read. Reading the Poll register constitutes a software poll. This will set the IS bit of the highest priority pending interrupt. Reading the poll status register will not set the IS bit of the highest priority pending interrupt; only the status of pending interrupts will be provided.

Encoding of the Poll and Poll Status register bits are as follows:

S_X: Encoded information that indicates the vector type of the highest priority interrupting source. Valid only when INTREQ = 1.

INTREQ: This bit determines if an interrupt request is present. Interrupt Request = 1; no Interrupt Request = 0.

iRMX 86 COMPATIBILITY MODE

This mode allows iRMX 86–80188 compatibility. The interrupt model of iRMX 86 requires one master and multiple slave 8259As in cascaded fashion. When iRMX mode is used, the internal 80188 interrupt controller will be used as a slave controller to an external master interrupt controller. The internal 80188 resources will be monitored through the internal interrupt controller, while the external controller functions as the system master interrupt controller.

Upon reset, the 80188 interrupt controller will be in the non-iRMX 86 mode of operation. To set the controller in the iRMX 86 mode, bit 14 of the Relocation Register should be set.

Because of pin limitations caused by the need to interface to an external 8259A master, the internal interrupt controller will no longer accept external inputs. There are however, enough 80188 interrupt controller inputs (internally) to dedicate one to each timer. In this mode, each timer interrupt source has its own mask bit, IS bit, and control word.

The iRMX 86 operating system requires peripherals to be assigned fixed priority levels. This is incompatible with the normal operation of the 80188 interrupt controller. Therefore, the initialization software must program the proper priority levels for each source. The required priority levels for the internal interrupt sources in iRMX mode are shown in Table 16.

Table 16. Internal Source Priority Level

Priority Level	Interrupt Source
0	Timer 0
1	(reserved)
2	DMA 0
3	DMA 1
4	Timer 1
5	Timer 2

These level assignments must remain fixed in the iRMX 86 mode of operation.

iRMX 86 Mode External Interface

The configuration of the 80188 with respect to an external 8259A master is shown in Figure 32. The INT0 input is used as the 80188 CPU interrupt input. INT3 functions as an output to send the 80188 slave-interrupt-request to one of the 8 master-PIC-inputs.

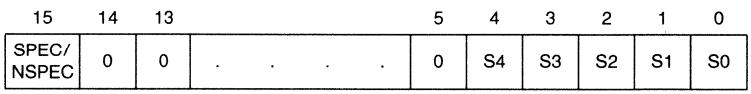


Figure 30. EOI Register Format

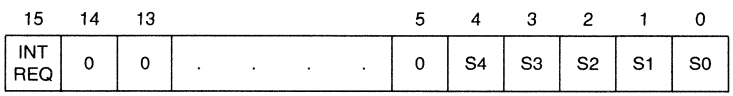


Figure 31. Poll Register Format

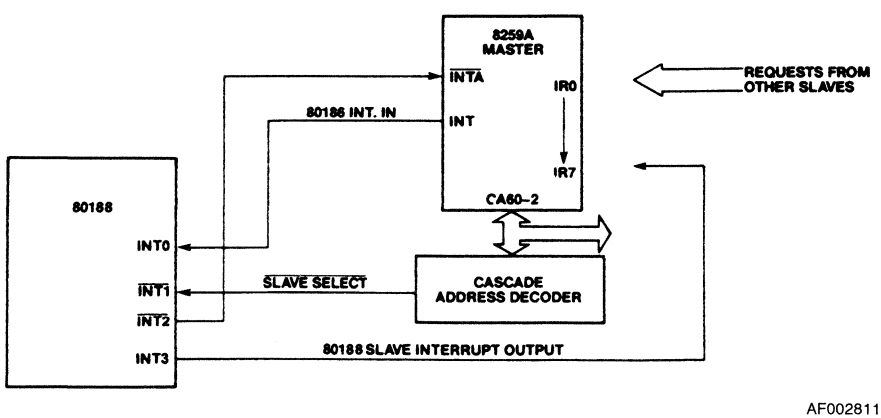


Figure 32. iRMX 86 Interrupt Controller Interconnection

Correct master-slave interface requires decoding of the slave addresses (CAS0–2). Slave 8259As do this internally. Because of pin limitations, the 80188 salve address will have to be decoded externally. $\overline{INT1}$ is used as a slave-select input. Note that the slave vector address is transferred internally, but the READY input must be supplied externally.

$\overline{INT2}$ is used as an acknowledge output, suitable to drive the $\overline{INTA}$ input of an 8259A.

Interrupt Nesting

iRMX 86 mode operation allows nesting of interrupt requests. When an interrupt is acknowledged, the priority logic masks off all priority levels except those with equal or higher priority.

Vector Generation in the iRMX 86 MODE

Vector generation in iRMX mode is exactly like that of an 8259A slave. The interrupt controller generates an 8-bit vector which the CPU multiplies by four and uses as an address into a vector table. The significant five bits of the vector are user-programmable while the lower three bits are generated by the priority logic. These bits represent the encoding of the priority level requesting service. The significant five bits of the vector are programmed by writing to the Interrupt Vector register at offset 20H.

Specific End-of-Interrupt

In iRMX mode the specific EOI command operates to reset an in-service bit of a specific priority. The user supplies a 3-bit priority-level value that points to an in-service bit to be reset. The command is executed by writing the correct value in the Specific EOI register at offset 22H.

Interrupt Controller Registers in the iRMX 86 Mode

All control and command registers are located inside the internal peripheral control block. Figure 33 shows the offsets of these registers.

End-of-Interrupt Register

The end-of-interrupt register is a command register which can only be written. The format of this register is shown in Figure 34. It initiates an EOI command when written by the 80188 CPU.

The bits in the EOI register are encoded as follows:

L_X: Encoded value indicating the priority of the IS bit to be reset.

In-Service Register

This register can be read from or written into. It contains the in-service bit for each of the internal sources. The format for this register is shown in Figure 35. Bit positions 2 and 3 correspond to the DMA channels; positions 0, 4, and 5 correspond to the integral timers. The source's IS bit is set when the processor acknowledges its interrupt request.

Interrupt Request Register

This register indicates which internal peripherals have interrupt requests pending. The format of this register is shown in Figure 35. The interrupt request bits are set when a request arrives from an internal source, and are rest when the processor acknowledges the request.

Mask Register

This register contains a mask bit for each interrupt source. The format for this register is shown in Figure 35. If the bit in this register corresponding to a particular interrupt source is set, any interrupts from that source will be masked. These mask bits are exactly the same bits which are used in the individual control registers, i.e., changing the state of a mask bit in this register will also change the state of the mask bit in the individual interrupt control register corresponding to the bit.

Control Registers

These registers are the control words for all the internal interrupt sources. The format of these registers is shown in Figure 36. Each of the timers and both of the DMA channels have their own Control Register.

The bits of the Control Registers are encoded as follows:

pr_X: 3-bit encoded field indicating a priority level for the source; note that each source must be programmed at specified levels.

msk: mask bit for the priority level indicated by pr_X bits.

	OFFSET
LEVEL 5 CONTROL REGISTER (TIMER 2)	3AH
LEVEL 4 CONTROL REGISTER (TIMER 1)	38H
LEVEL 3 CONTROL REGISTER (DMA 1)	36H
LEVEL 2 CONTROL REGISTER (DMA 0)	34H
LEVEL 0 CONTROL REGISTER (TIMER 0)	32H
INTERRUPT-REQUEST REGISTER	2EH
IN-SERVICE REGISTER	2CH
PRIORITY-LEVEL MASK REGISTER	2AH
MASK REGISTER	28H
SPECIFIC EOI REGISTER	22H
INTERRUPT VECTOR REGISTER	20H

Figure 33. Interrupt Controller Registers (iRMX 86 Mode)

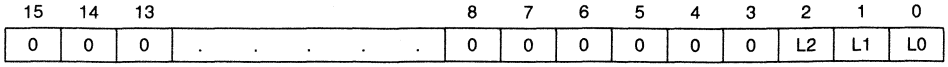


Figure 34. Specific EQI Register Format

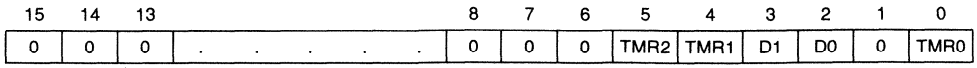


Figure 35. In-Service, Interrupt Request, and Mask Register Format

Interrupt Vector Register

This register provides the upper five bits of the interrupt vector address. The format of this register is shown in Figure 37. The interrupt controller itself provides the lower three bits of the interrupt vector as determined by the priority level of the interrupt request.

The format of the bits in this register is:

t_x: 5-bit field indicating the upper five bits of the vector address.

Priority-Level Mask Register

This register indicates the lowest priority-level interrupt which will be serviced.

The encoding of the bits in this register is:

m_x: 3-bit encoded field indication priority-level value. All levels of lower priority will be masked.

Interrupt Controller and Reset

Upon RESET, the interrupt controller will perform the following actions:

- All SFNM bits reset to 0, implying Fully Nested Mode.
- All PR bits in the various control registers set to 1. Tis places all sources at lowest priority (level 111).
- All LTM bits reset to 0, resulting in edge-sense mode.
- All Interrupt Service bits reset to 0.
- All Interrupt Request bits reset to 0.
- All MSK (Interrupt Mask) bits set to 1 (mask).
- All C (Cascade) bits reset to 0 (non-cascade).
- All PRM (Priority Mask) bits set to 1, implying no levels masked.
- Initialized to NON-iRMX 86 mode.

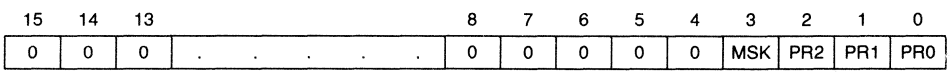


Figure 36. Control Word Format

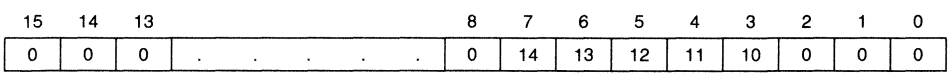


Figure 37. Interrupt Vector Register Format

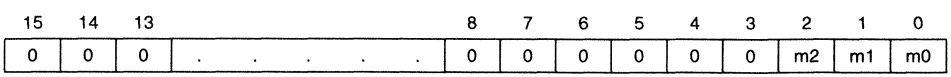


Figure 38. Priority Level Mask Register

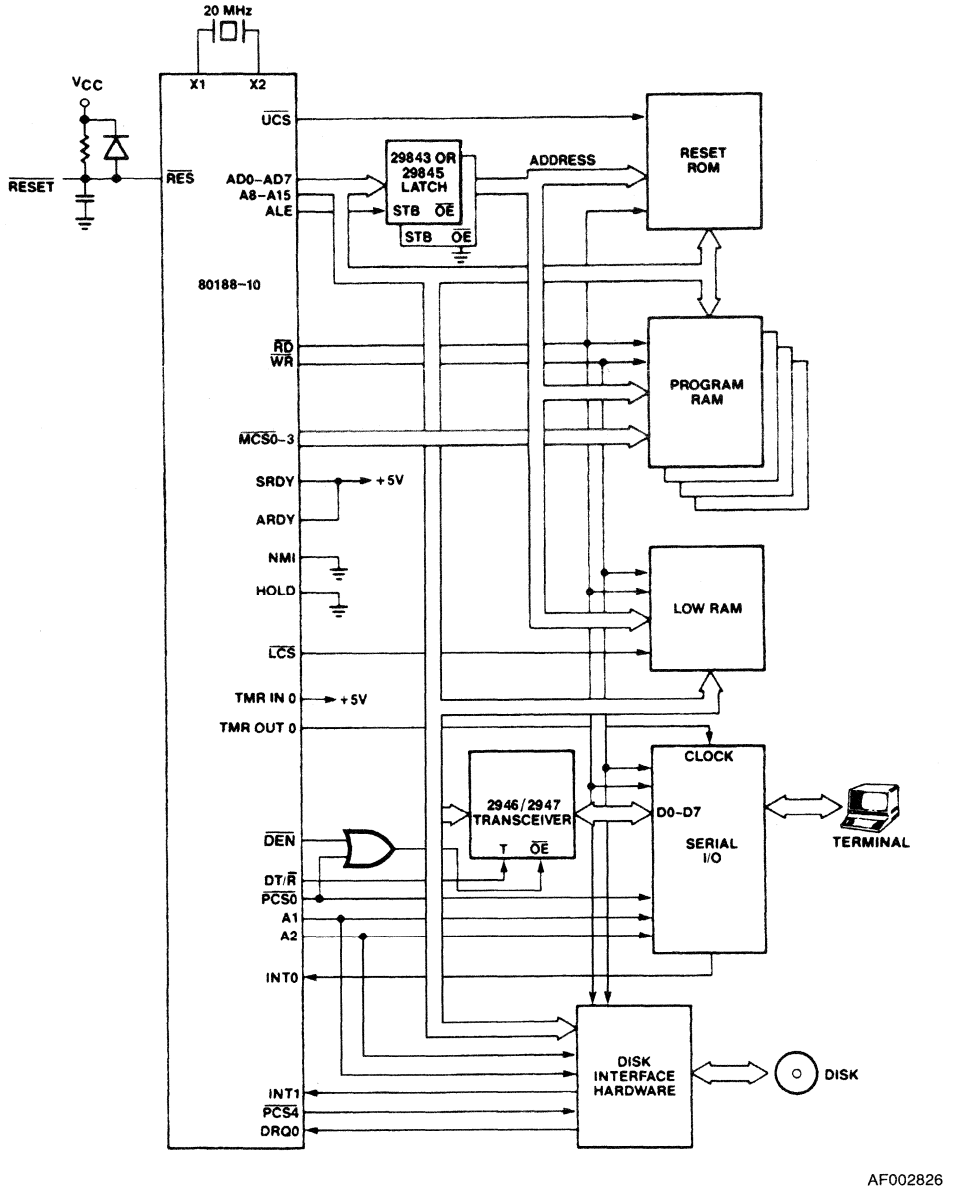


Figure 39. Typical 80188 Computer

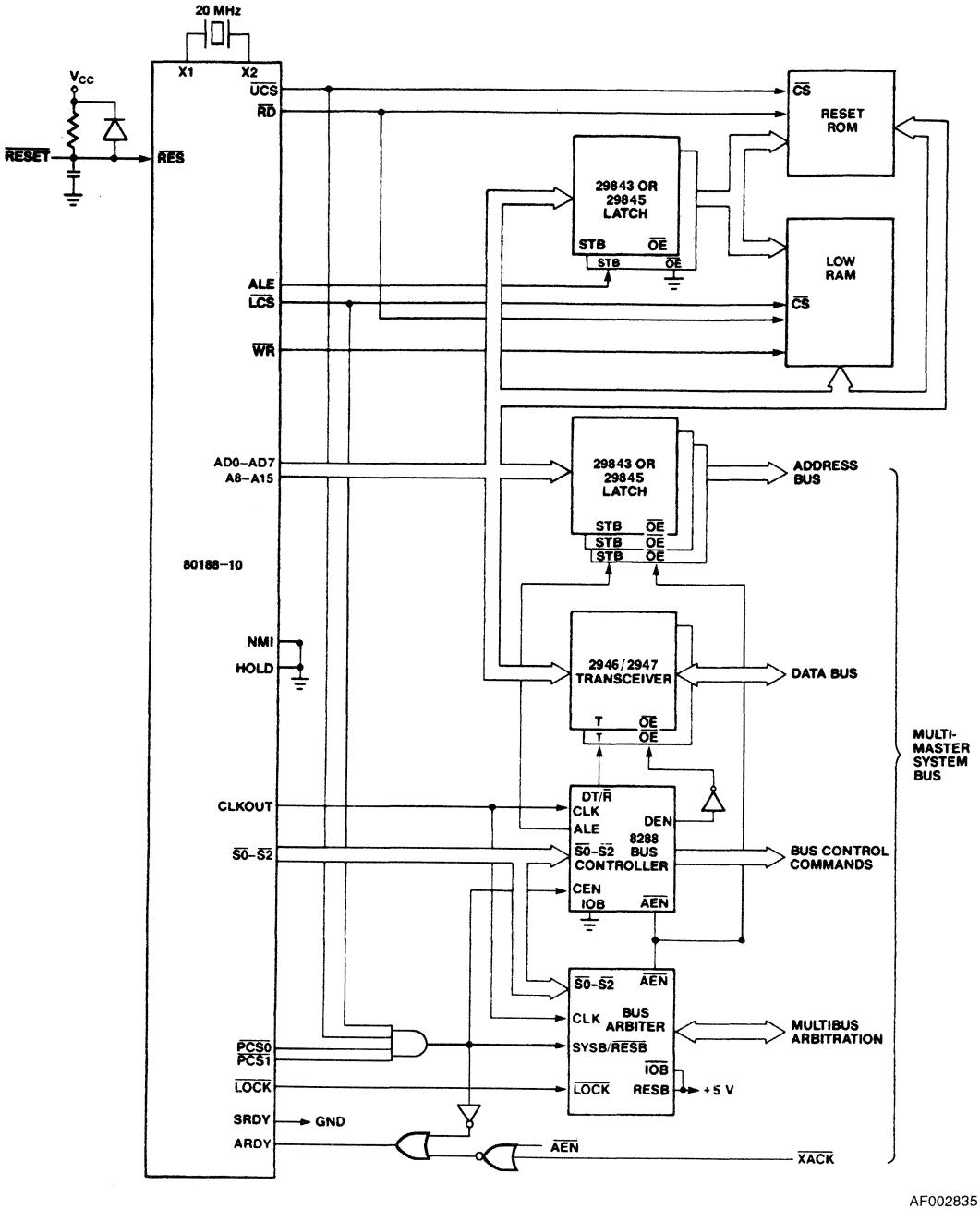


Figure 40. Typical 80188 Multi-Master Bus Interface

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground..... −1.0 V to +7 V
Power Dissipation (steady state 70°C).....2.0 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A).....0 to +70°C
(T_C).....0 to +110°C
Supply Voltage (V_{CC}) ..... 5 V ± 10%

Industrial (I) Devices
Temperature (T_A).....−40 to +85°C
Supply Voltage (V_{CC}) .....5 V ± 5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS (over operating range unless otherwise specified)

Parameters	Description	Test Conditions	Min.	Max.	Units
V_{IL}	Input LOW Voltage		−0.5	+0.8	Volts
V_{IH}	Input HIGH Voltage (All except X1 and $\overline{RES}$)		2.0	V_{CC} + 0.5	Volts
V_{IH1}	Input HIGH Voltage ($\overline{RES}$)		3.0	V_{CC} + 0.5	Volts
V_{OL}	Output LOW Voltage	I_A = 2.5 mA for $\overline{S0}$ - $\overline{S2}$ I_A = 2.0 mA for all other outputs		0.45	Volts
V_{OH}	Output HIGH Voltage	I_{OA} = −400 μA	2.4		Volts
I_{CC}	Power Supply Current	T_A = −40°C		600**	mA
		T_A = 0°C		500	
		T_A = 70°C		375	
I_{LI}	Input Leakage Current	0 V < V_{IN} < V_{CC}		±10	μA
I_{LO}	Output Leakage Current	0.45 V < V_{OUT} < V_{CC}		±10	μA
V_{CLO}	Clock Output LOW	I_A = 4.0 mA		0.6	Volts
V_{CHO}	Clock Output HIGH	I_{OA} = −200 μA	4.0		Volts
V_{CLI}	Clock Input LOW Voltage		−0.5	0.6	Volts
V_{CHI}	Clock Input HIGH Voltage		3.9	V_{CC} + 1.0	Volts
C_{IN}	Input Capacitance			10	pF
C_{IO}	I/O Capacitance			20	pF

SWITCHING CHARACTERISTICS

PIN TIMING

80188 Timing Requirements All Timings Measured At 1.5 Volts Unless Otherwise Noted

			80188-10 (10 MHz)		80188 (8 MHz)		
Parameters	**Description**	**Test Conditions**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
T_{DVCL}	Data in Setup (A/D)		15		20		ns
T_{CLDX}	Data in Hold (A/D)		8		10		ns
T_{ARYHCH}	Asynchronous Ready (AREADY) active setup time*		15		20		ns
T_{ARYLCL}	AREADY inactive setup time		25		35		ns
T_{CHARYX}	AREADY hold time		15		15		ns
T_{ARYCHL}	Asynchronous Ready inactive hold time		15		15		ns
T_{SRYCL}	Synchronous Ready (SREADY) transition setup time		20		20		ns
T_{CLSRY}	SREADY transition hold time		15		15		ns
T_{HVCL}	HOLD Setup*		20		25		ns
T_{INVCH}	INTR, NMI, $\overline{TEST}$, TIMERIN, Setup*		25		25		ns
T_{INVCL}	DRQ0, DRQ1, Setup*		20		25		ns

*To guarantee recognition at next clock.
**For Industrial Grade Parts only.

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS (Cont'd.)

80188 Master Interface Timing Responses

Parameters	Description	Test Conditions	80188-10 (10 MHz)		80188 (8 MHz)		Units
			Min.	Max.	Min.	Max.	
T_{CLAV}	Address Valid Delay	C_L = 20 – 200 pF all outputs	5	44	5	55	ns
T_{CLAX}	Address Hold		10		10		ns
T_{CLAZ}	Address Float Delay		TCLAX	30	TCLAX	35	ns
T_{CHCZ}	Command Lines Float Delay			40		45	ns
T_{CHCV}	Command Lines Valid Delay (after float)			45		55	ns
T_{LHLL}	ALE Width		TCLCL-30		TCLCL-35		ns
T_{CHLH}	ALE Active Delay			30		35	ns
T_{CHLL}	ALE Inactive Delay			30		35	ns
T_{LLAX}	Address Hold to ALE Inactive		TCHCL-20		TCHCL-25		ns
T_{CLDV}	Data Valid Delay		10	40	10	44	ns
T_{CLDOX}	Data Hold Time		10		10		ns
T_{WHDX}	Data Hold after WR		TCLCL-34		TCLCL-40		ns
T_{CVCTV}	Control Active Delay1		5	40	5	50	ns
T_{CHCTV}	Control Active Delay2		10	44	10	55	ns
T_{CVCTX}	Control Inactive Delay		5	44	5	55	ns
T_{CVDEX}	$\overline{DEN}$ Inactive Delay (Non-Write Cycle)		10	56	10	70	ns
T_{AZRL}	Address Float to $\overline{RD}$ Active		0		0		ns
T_{CLRL}	$\overline{RD}$ Active Delay		10	56	10	70	ns
T_{CLRH}	$\overline{RD}$ Inactive Delay		10	44	10	55	ns
T_{RHAV}	$\overline{RD}$ Inactive to Address Active		TCLCL-40		TCLCL-40		ns
T_{CLHAV}	HLDA Valid Delay		5	40	5	50	ns
T_{RLRH}	$\overline{RD}$ Width		2TCLCL-46		2TCLCL-50		ns
T_{WLWH}	$\overline{WR}$ Width		2TCLCL-34		2TCLCL-40		ns
T_{AVAL}	Address Valid to ALE LOW		TCLCH-19		TCLCH-25		ns
T_{CHSV}	Status Active Delay		10	45	10	55	ns
T_{CLSH}	Status Inactive Delay		10	50	10	65	ns
T_{CLTMV}	Timer Output Delay	100 pF max		48		60	ns
T_{CLRO}	Reset Delay			48		60	ns
T_{CHQSV}	Queue Status Delay			28		35	ns
T_{CHDX}	Status Hold Time		10		10		ns
T_{AVCH}	Address Valid to Clock HIGH		10		10		ns
T_{CLLV}	LOCK Valid/Invalid Delay		5	60	5	65	ns

80188 Chip-Select Timing Responses

Parameters	Description	Test Conditions	80188-10 (10 MHz)		80188 (8 MHz)		Units
			Min.	Max.	Min.	Max.	
T_{CLCSV}	Chip-Select Active Delay			45		66	ns
T_{CXCSX}	Chip-Select Hold from Command Inactive		35		35		ns
T_{CHCSX}	Chip-Select Inactive Delay		5	32	5	35	ns

SWITCHING CHARACTERISTICS (Cont'd.)

80188 CLKIN Requirements

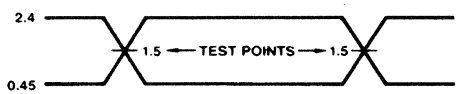
Parameters	Description	Test Conditions	80188-10 (10 MHz)		80188 (8 MHz)		Units
			Min.	Max.	Min.	Max.	
T_{CKIN}	CLKIN Period		50	250	62.5	250	ns
T_{CKHL}	CLKIN Fall Time	3.5 to 1.0 volts		10		10	ns
T_{CKLH}	CLKIN Rise Time	1.0 to 3.5 volts		10		10	ns
T_{CLCK}	CLKIN LOW Time	1.5 volts	20		25		ns
T_{CHCK}	CLKIN HIGH Time	1.5 volts	20		25		ns

80188 CLKOUT Timing (200 pF load)

Parameters	Description	Test Conditions	Min.	Max.	Min.	Max.	Units
T_{CICO}	CLKIN to CLKOUT Skew			25		50	ns
T_{CLCL}	CLKOUT Period		100	500	125	500	ns
T_{CLCH}	CLKOUT LOW Time	1.5 volts	$\frac{1}{2}T_{CLCL}$-6.0		$\frac{1}{2}T_{CLCL}$-7.5		ns
T_{CHCL}	CLKOUT HIGH Time	1.5 volts	$\frac{1}{2}T_{CLCL}$-6.0		$\frac{1}{2}T_{CLCL}$-7.5		ns
T_{CH1CH2}	CLKOUT Rise Time	1.0 to 3.5 volts		12		15	ns
T_{CL2CL1}	CLKOUT Fall Time	3.5 to 1 volts		12		15	ns

All timings measured at 1.5 volts unless otherwise noted.

SWITCHING TEST INPUT/OUTPUT WAVEFORM

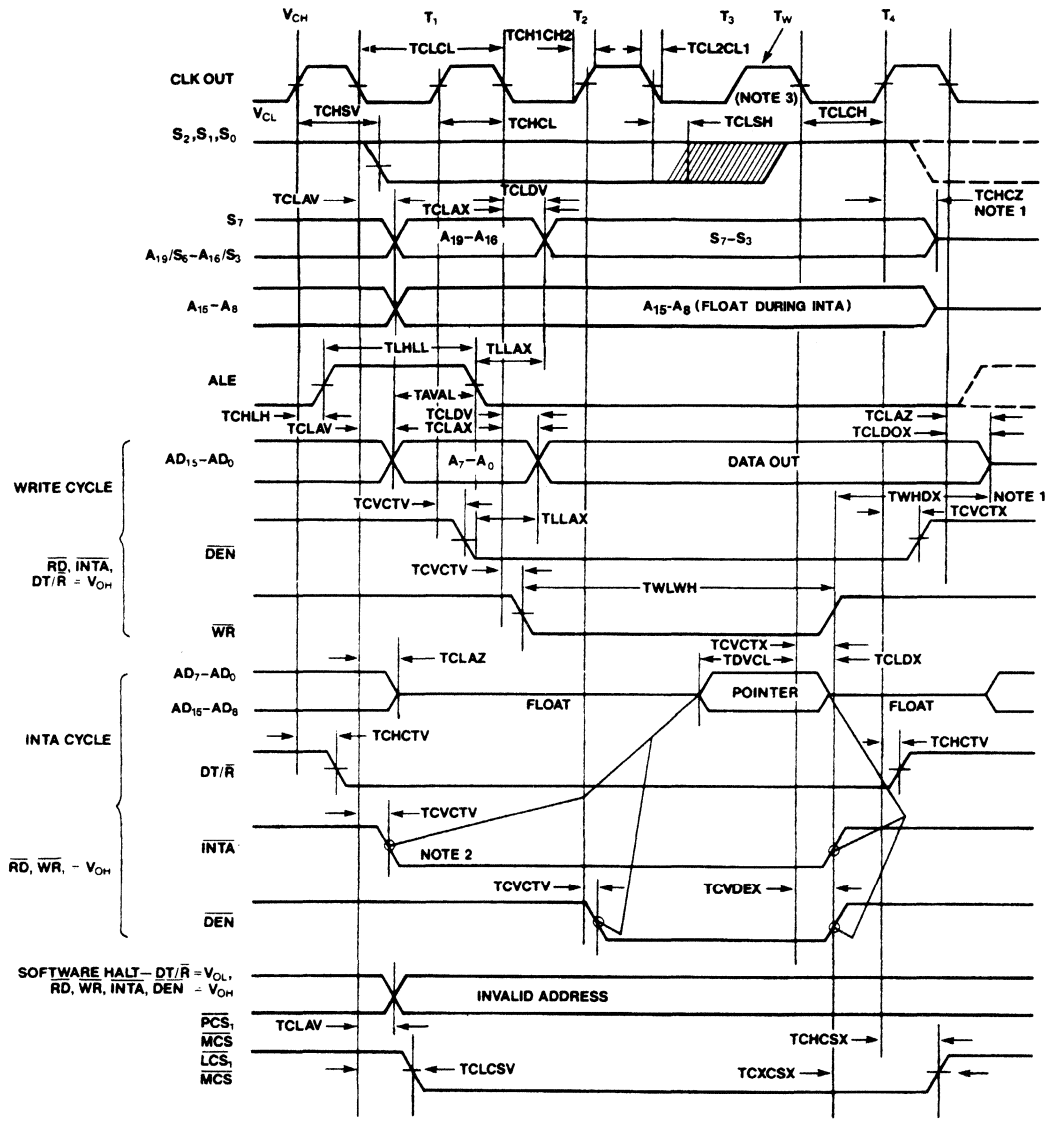


WF001870

AC testing inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0." The clock is driven at 4.3 V and 0.25 V. Timing measurements are made at 1.5 V for both a logic "1" and "0."

SWITCHING WAVEFORMS

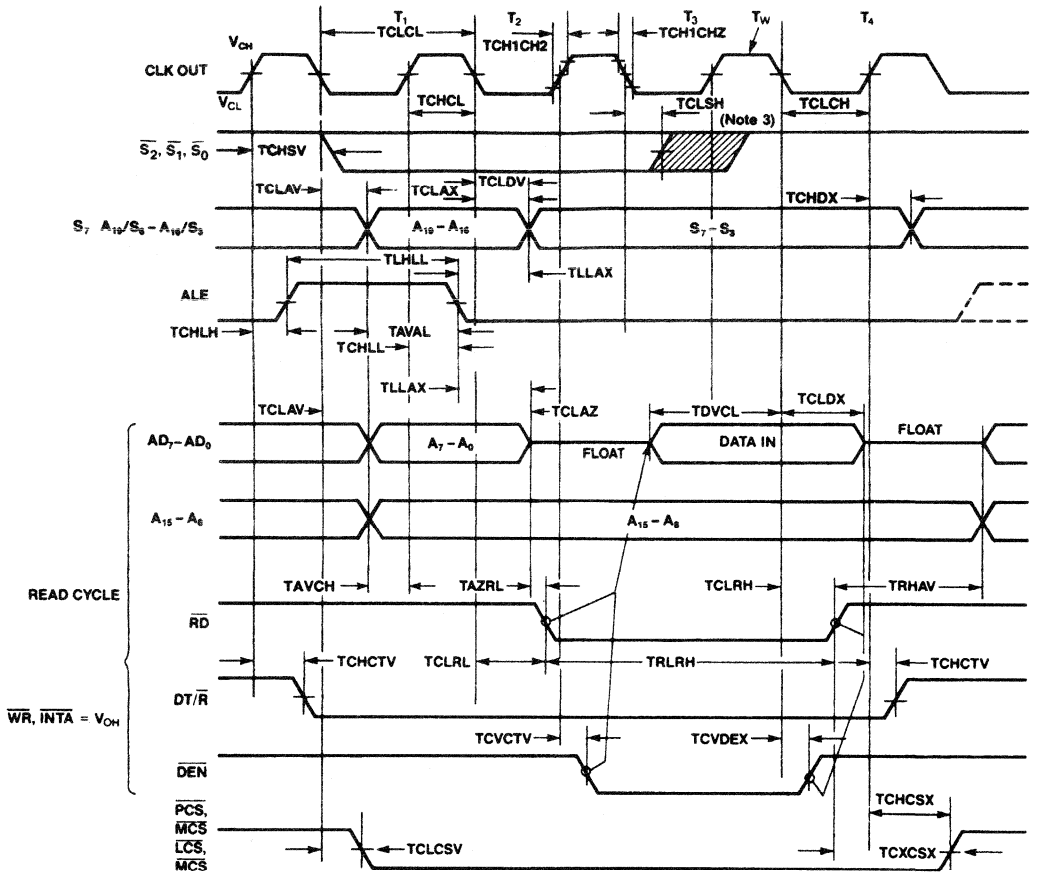
MAJOR CYCLE TIMING



WF006214

SWITCHING WAVEFORMS (Cont'd.)

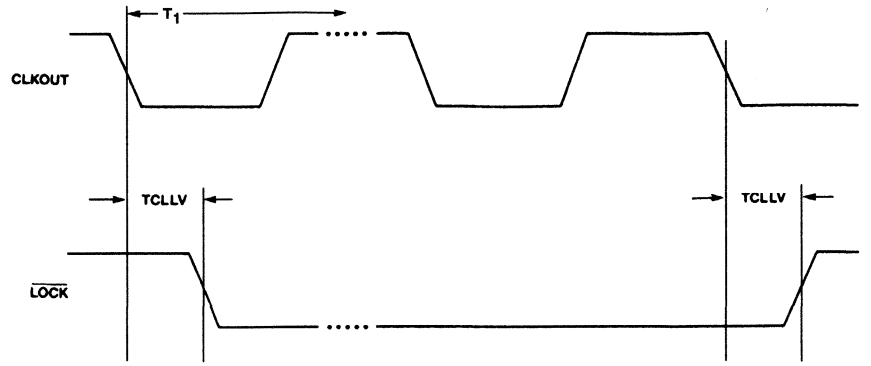
MAJOR CYCLE TIMING (Cont'd.)



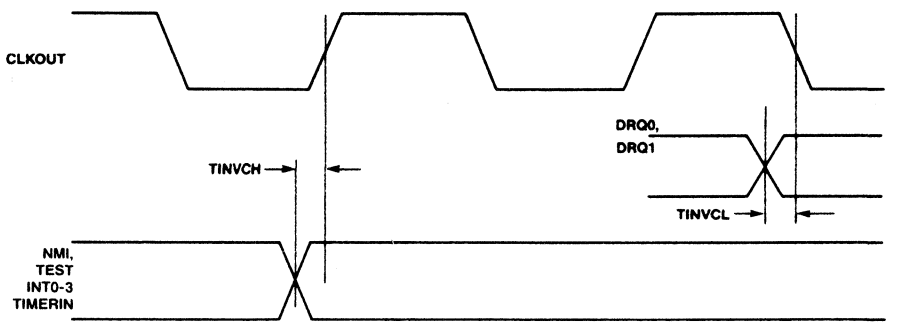
WF006226

Notes:
1. Following a Write cycle, the Local Bus is floated by the 80188 only when the 80188 enters a "Hold Acknowledge" state.
2. INTA occurs one clock later in RMX-mode.
3. Status inactive just prior to T_4.

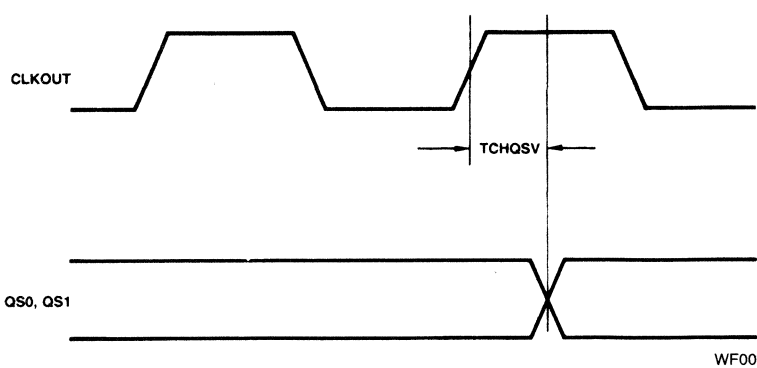
SWITCHING WAVEFORMS (Cont'd.)



WF006232

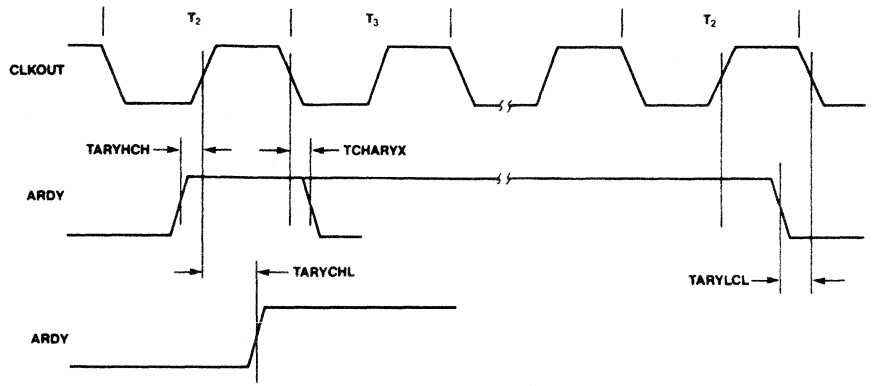


WF006240

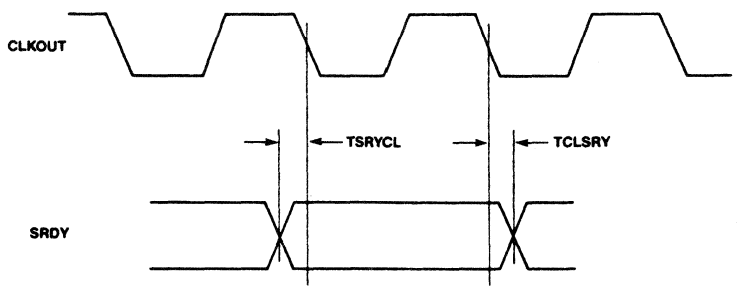


WF006250

SWITCHING WAVEFORMS (Cont'd.)

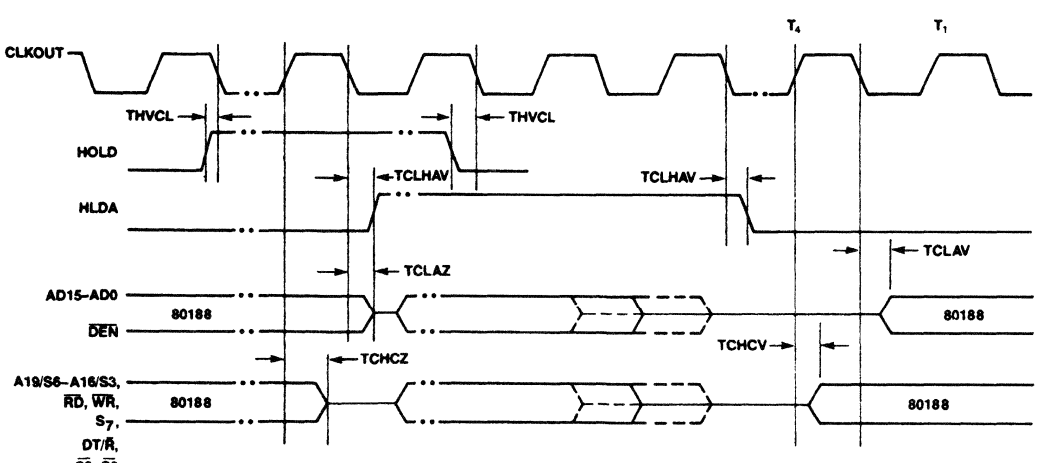


WF006262



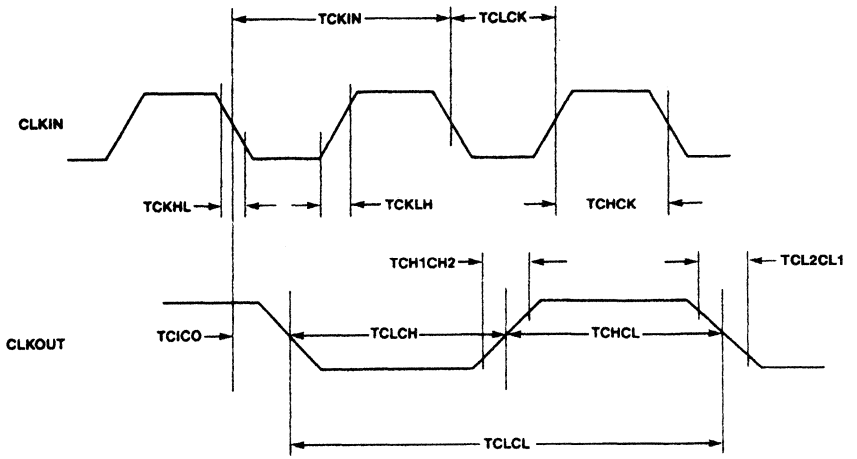
WF006270

HOLD-HLDA TIMING



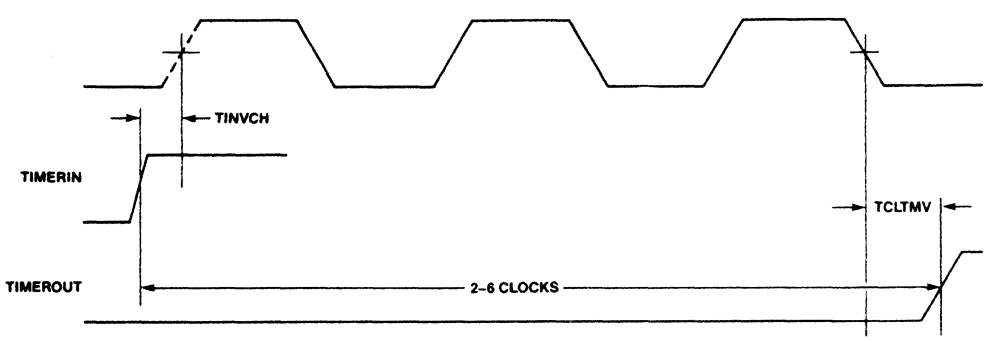
WF006282

SWITCHING WAVEFORMS (Cont'd.)



WF006293

TIMER ON 80188



WF006300

80188 INSTRUCTION TIMINGS

The following instruction timings represent the minimum execution time in clock cycles for each instruction. The timings given are based on the following assumptions:

- The opcode, along with any data or displacement required for execution of a particular instruction, has been pre-fetched and resides in the queue at the time it is needed.
- No wait states or bus HOLDS occur.
- All word-data is located on even-address boundaries.

All jumps and calls include the time required to fetch the opcode of the next instruction at the destination address.

All instructions which involve memory reference can require one (and in some cases, two) additional clocks above the minimum timings shown. This is due to the asynchronous nature of the handshake between the BIU and the Execution unit.

INSTRUCTION SET SUMMARY

FUNCTION	FORMAT	Clock Cycles	Comments
DATA TRANSFER			
MOV = Move:			
Register to Register/Memory	1 0 0 0 1 0 0 w | mod reg r/m	2/12	
Register/memory to register	1 0 0 0 1 0 1 w | mod reg r/m	2/9	
Immediate to register/memory	1 1 0 0 0 1 1 w | mod 0 0 0 r/m | data | data if w = 1	12 – 13	8/16-bit
Immediate to register	1 0 1 1 w reg | data | data if w = 1	3 – 4	8/16-bit
Memory to accumulator	1 0 1 0 0 0 0 w | addr-low | addr-high	9	
Accumulator to memory	1 0 1 0 0 0 1 w | addr-low | addr-high	8	
Register/memory to segment register	1 0 0 0 1 1 1 0 | mod 0 reg r/m	2/9	
Segment register to register/memory	1 0 0 0 1 1 0 0 | mod 0 reg r/m	2/11	
PUSH = Push:			
Memory	1 1 1 1 1 1 1 1 | mod 1 1 0 r/m	16	
Register	0 1 0 1 0 reg	10	
Segment register	0 0 0 reg 1 1 0	9	
Immediate	0 1 1 0 1 0 s 1 | data | data if s = 0	10	
PUSHA = Push All	0 1 1 0 0 0 0 0	36	
POP = Pop:			
Memory	1 0 0 0 1 1 1 1 | mod 0 0 0 r/m	20	
Register	0 1 0 1 1 reg	10	
Segment register	0 0 0 reg 1 1 1 | (reg ≠ 01)	8	
POPA = Pop All	0 1 1 0 0 0 0 1	51	
XCHG = Exchange:			
Register/memory with register	1 0 0 0 0 1 1 w | mod reg r/m	4/17	
Register with accumulator	1 0 0 1 0 reg	3	
IN = Input from:			
Fixed port	1 1 1 0 0 1 0 w | port	10	
Variable port	1 1 1 0 1 1 0 w	8	
OUT = Output to:			
Fixed port	1 1 1 0 0 1 1 w | port	9	
Variable port	1 1 1 0 1 1 1 w	7	
XLAT = Translate byte to AL	1 1 0 1 0 1 1 1	11	
LEA = Load EA to register	1 0 0 0 1 1 0 1 | mod reg r/m	6	
LDS = Load pointer to DS	1 1 0 0 0 1 0 1 | mod reg r/m | (mod ≠ 11)	18	
LES = Load pointer to ES	1 1 0 0 0 1 0 0 | mod reg r/m | (mod ≠ 11)	18	
LAHF = Load AH with flags	1 0 0 1 1 1 1 1	2	
SAHF = Store AH into flags	1 0 0 1 1 1 1 0	3	
PUSHF = Push flags	1 0 0 1 1 1 0 0	9	
POPF = Pop flags	1 0 0 1 1 1 0 1	8	

Shaded areas indicate new 80188 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Cont'd.)

FUNCTION	FORMAT	Clock Cycles	Comments
ARITHMETIC			
ADD = Add:			
Reg/memory with register to either	0 0 0 0 0 0 d w | mod reg r/m	3/10	
Immediate to register / memory	1 0 0 0 0 0 s w | mod 0 0 0 r/m | data | data if s w = 01	4/16	
Immediate to accumulator	0 0 0 0 0 1 0 w | data | data if w = 1	3/4	8/16-bit
ADC = Add with carry:			
Reg/memory with register to either	0 0 0 1 0 0 d w | mod reg r/m	3/10	
Immediate to register/memory	1 0 0 0 0 0 s w | mod 0 1 0 r/m | data | data if s w = 01	4/16	
Immediate to accumulator	0 0 0 1 0 1 0 w | data | data if w = 1	3/4	8/16-bit
INC = Increment:			
Register/memory	1 1 1 1 1 1 1 w | mod 0 0 0 r/m	3/15	
Register	0 1 0 0 0 reg	3	
SUB = Subtract:			
Reg/memory and register to either	0 0 1 0 1 0 d w | mod reg r/m	3/10	
Immediate from register/memory	1 0 0 0 0 0 s w | mod 1 0 1 r/m | data | data if s w = 01	4/16	
Immediate from accumulator	0 0 1 0 1 1 0 w | data | data if w = 1	3/4	8/16-bit
SBB = Subtract with borrow:			
Reg/memory and register to either	0 0 0 1 1 0 d w | mod reg r/m	3/10	
Immediate from register/memory	1 0 0 0 0 0 s w | mod 0 1 1 r/m | data | data if s w = 01	4/16	
Immediate from accumulator	0 0 0 1 1 1 0 w | data | data if w = 1	3/4	8/16-bit
DEC = Decrement:			
Register/memory	1 1 1 1 1 1 1 w | mod 0 0 1 r/m	3/15	
Register	0 1 0 0 1 reg	3	
CMP = Compare:			
Register/memory with register	0 0 1 1 1 0 1 w | mod reg r/m	3/10	
Register with register/memory	0 0 1 1 1 0 0 w | mod reg r/m	3/10	
Immediate with register/memory	1 0 0 0 0 0 s w | mod 1 1 1 r/m | data | data if s w = 01	3/10	
Immediate with accumulator	0 0 1 1 1 1 0 w | data | data if w = 1	3/4	8/16-bit
NEG = Change sign	1 1 1 1 0 1 1 w | mod 0 1 1 r/m	3	
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1	8	
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1	4	
AAS = ASCII adjust for subtract	0 0 1 1 1 1 1 1	7	
DAS = Decimal adjust for subtract	0 0 1 0 1 1 1 1	4	
MUL = Mulitiply (unsigned)	1 1 1 1 0 1 1 w | mod 1 0 0 r/m		
Register-Byte		26 – 28	
Register-Word		35 – 37	
Memory-Byte		32 – 34	
Memory-Word		41 – 43	
IMUL = Integer multiply (signed):	1 1 1 1 0 1 1 w | mod 1 0 1 r/m		
Register-Byte		25 – 28	
Register-Word		34 – 37	
Memory-Byte		31 – 34	
Memory-Word		40 – 43	
IMUL = Integer immediate multiply (signed)	0 1 1 0 1 0 s 1 | mod reg r/m | data | data if s = 0	22 – 25/29 – 32	
DIV = Divide (unsigned):	1 1 1 1 0 1 1 w | mod 1 1 0 r/m		
Register-Byte		29	
Register-Word		38	
Memory-Byte		35	
Memory-Word		44	

Shaded areas indicate new 80188 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Cont'd.)

FUNCTION	FORMAT	Clock Cycles	Comments
ARITHMETIC (Continued):			
IDIV = Integer divide (signed):	1 1 1 1 0 1 1 w | mod 1 1 1 r/m	44 – 52	
Register-Byte		53 – 61	
Register-Word		50 – 58	
Memory-Byte		59 – 67	
Memory-Word			
AAM = ASCII adjust for multiply	1 1 0 1 0 1 0 0 | 0 0 0 0 1 0 1 0	19	
AAD = ASCII adjust for divide	1 1 0 1 0 1 0 1 | 0 0 0 0 1 0 1 0	15	
CBW = Convert byte to word	1 0 0 1 1 0 0 0	2	
CWD = Convert word to double word	1 0 0 1 1 0 0 1	4	
LOGIC			
Shift/Rotate Instructions:			
Register/Memory by 1	1 1 0 1 0 0 0 w | mod TTT r/m	2/15	
Register/Memory by CL	1 1 0 1 0 0 1 w | mod TTT r/m		
Register/Memory by Count	1 1 0 0 0 0 0 w | mod TTT r/m | count	5 + n/17 + n	

TTT	Instruction
0 0 0	ROL
0 0 1	ROR
0 1 0	RCL
0 1 1	RCR
1 0 0	SHL/SAL
1 0 1	SHR
1 1 1	SAR

AND = And:					
Reg/memory and register to either	0 0 1 0 0 0 d w	mod reg r/m		3/10	
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1	4/16
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w = 1		3/4 8/16-bit
TEST = And function to flags, no result:					
Register/memory and register	1 0 0 0 0 1 0 w	mod reg r/m		3/10	
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1	4/10
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w = 1		3/4 8/16-bit
OR = Or:					
Reg/memory and register to either	0 0 0 0 1 0 d w	mod reg r/m		3/10	
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w = 1	4/16
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w = 1		3/4 8/16-bit
XOR = Exclusive or:					
Reg/memory and register to either	0 0 1 1 0 0 d w	mod reg r/m		3/10	
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1	4/16
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w = 1		3/4 8/16-bit
NOT = Invert register/memory	1 1 1 1 0 1 1 w	mod 0 1 0 r/m		3	
STRING MANIPULATION:					
MOVS = Move byte/word	1 0 1 0 0 1 0 w			8 + 8n	
CMPS = Compare byte/word	1 0 1 0 0 1 1 w			5 + 22n	
SCAS = Scan byte/word	1 0 1 0 1 1 1 w			5 + 15n	
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w			6 + 11n	
STOS = Stor byte/wd from AL/A	1 0 1 0 1 0 1 w			6 + 9n	
INS = Input byte/wd from DX port	0 1 1 0 1 1 0 w			8	
OUTS = Output byte/wd to DX port	0 1 1 0 1 1 1 w			7	

Shaded areas indicate new 80188 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Cont'd.)

FUNCTION	FORMAT	Clock Cycles	Comments
STRING MANIPULATION (Continued):			
Repeated by count in CX			
MOVS = Move string	1 1 1 1 0 0 1 0 | 1 0 1 0 0 1 0 w	14	
CMPS = Compare string	1 1 1 1 0 0 1 z | 1 0 1 0 0 1 1 w	22	
SCAS = Scan string	1 1 1 1 0 0 1 z | 1 0 1 0 1 1 1 w	15	
LODS = Load string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 1 0 w	12	
STOS = Store string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 0 1 w	10	
INS = Input string	1 1 1 1 0 0 1 0 | 0 1 1 0 1 1 0 w	8 + 8n/14	Repeated/ Not Repeated
OUTS = Output string	1 1 1 1 0 0 1 0 | 0 1 1 0 1 1 1 w	8 + 8n/14	Repeated/ Not Repeated
CONTROL TRANSFER			
CALL = Call:			
Direct within segment	1 1 1 0 1 0 0 0 | disp-low | disp-high	14	
Register memory indirect within segment	1 1 1 1 1 1 1 1 | mod 0 1 0 r/m	13/19	
Direct intersegment	1 0 0 1 1 0 1 0 | segment offset | segment selector	23	
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 0 1 1 r/m | (mod ≠ 11)	38	
JMP = Unconditional jump:			
Short/long	1 1 1 0 1 0 1 1 | disp-low	13	
Direct within segment	1 1 1 0 1 0 0 1 | disp-low | disp-high	13	
Register/memory indirect within segment	1 1 1 1 1 1 1 1 | mod 1 0 0 r/m	11/17	
Direct intersegment	1 1 1 0 1 0 1 0 | segment offset | segment selector	13	
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 1 0 1 r/m | (mod ≠ 11)	26	
RET = Return from CALL:			
Within segment	1 1 0 0 0 0 1 1	16	
Within seg adding immed to SP	1 1 0 0 0 0 1 0 | data-low | data-high	18	
Intersegment	1 1 0 0 1 0 1 1	22	
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0 | data-low | data-high	25	

Shaded areas indicate new 80188 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Cont'd.)

FUNCTION	FORMAT	Clock Cycles	Comments
CONTROL TRANSFER (Continued):			
JE/JZ = Jump on equal zero	0 1 1 1 0 1 0 0 | disp	4/13	13 if JMP taken
JL/JNGE = Jump on less not greater or equal	0 1 1 1 1 1 0 0 | disp	4/13	4 if JMP not taken
JLE/JNG = Jump on less or equal not greater	0 1 1 1 1 1 1 0 | disp	4/13	
JB/JNAE = Jump on below not above or equal	0 1 1 1 0 0 1 0 | disp	4/13	
JBE/JNA = Jump on below or equal not above	0 1 1 1 0 1 1 0 | disp	4/13	
JP/JPE = Jump on parity parity even	0 1 1 1 1 0 1 0 | disp	4/13	
JO = Jump on overflow	0 1 1 1 0 0 0 0 | disp	4/13	
JS = Jump on sign	0 1 1 1 1 0 0 0 | disp	4/13	
JNE/JNZ = Jump on not equal not zero	0 1 1 1 0 1 0 1 | disp	4/13	
JNL/JGE = Jump on not less greater or equal	0 1 1 1 1 1 0 1 | disp	4/13	
JNLE/JG = Jump on not less or equal greater	0 1 1 1 1 1 1 1 | disp	4/13	
JNB/JAE = Jump on not below above or equal	0 1 1 1 0 0 1 1 | disp	4/13	
JNBE/JA = Jump on not below or equal above	0 1 1 1 0 1 1 1 | disp	4/13	
JNP/JPO = Jump on not par / par odd	0 1 1 1 1 0 1 1 | disp	4/13	
JNO = Jump on not overflow	0 1 1 1 0 0 0 1 | disp	4/13	
JNS = Jump on not sign	0 1 1 1 1 0 0 1 | disp	4/13	
LOOP = Loop CX times	1 1 1 0 0 0 1 0 | disp	5/15	
LOOPZ/LOOPE = Loop while zero equal	1 1 1 0 0 0 0 1 | disp	6/16	
LOOPNZ/LOOPNE = Loop while not zero equal	1 1 1 0 0 0 0 0 | disp	6/16	
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1 | disp	16 5	JMP taken/ JMP not taken
ENTER = Enter Procedure	1 1 0 0 1 0 0 0 | data-low | data-high | L		
L = 0		15	
L = 1		25	
L > 1		22 + 16(n – 1)	
LEAVE = Leave Procedure	1 1 0 0 1 0 0 1	8	
INT = Interrupt:			
Type specified	1 1 0 0 1 1 0 1 | type	47	
Type 3	1 1 0 0 1 1 0 0	45	if INT. taken/
INTO = Interrupt on overflow	1 1 0 0 1 1 1 0	48/4	if INT. not taken
IRET = Interrupt return	1 1 0 0 1 1 1 1	28	
BOUND = Detect value out of range	0 1 1 0 0 0 1 0 | mod reg r/m	33 – 35	

Shaded areas indicate new 80188 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Cont'd.)

FUNCTION	FORMAT	Clock Cycles	Comments
PROCESSOR CONTROL			
CLC = Clear carry	1 1 1 1 1 0 0 0	2	
CMC = Complement carry	1 1 1 1 0 1 0 1	2	
STC = Set carry	1 1 1 1 1 0 0 1	2	
CLD = Clear direction	1 1 1 1 1 1 0 0	2	
STD = Set direction	1 1 1 1 1 1 0 1	2	
CLI = Clear interrupt	1 1 1 1 1 0 1 0	2	
STI = Set interrupt	1 1 1 1 1 0 1 1	2	
HLT = Halt	1 1 1 1 0 1 0 0	2	
WAIT = Wait	1 0 0 1 1 0 1 1	6	if $\overline{\text{test}}$ = 0
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0	2	
ESC = Processor Extension Escape	1 0 0 1 1 T T T | mod LLL r/m (TTT LLL are opcode to processor extension)	6	

FOOTNOTES

The effective Address (EA) of the memory operand is computed according to the mod and r/m fields:

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

SEGMENT OVERRIDE PREFIX

0 0 1 reg 1 1 0

reg is assigned according to the following:

reg	Segment Register
00	ES
01	CS
10	SS
11	DS

REG is assigned according to the following table:

16-Bit (w = 1)	8-Bit (w = 0)
000 AX	000 AL
001 CX	001 CL
010 DX	010 DL
011 BX	011 BL
100 SP	100 AH
101 BP	101 CH
110 SI	110 DH
111 DI	111 BH

The physical addresses of all operands addressed by the BP register are computed using the SS segment register. The physical addresses of the destination operands of the string primitive operations (those addressed bythe DI register) are computed using the ES segment, which may not be overridden.

80286

High-Performance Microprocessor with
Memory Management and Protection

DISTINCTIVE CHARACTERISTICS

- High performance processor (up to 13.3 times iAPX 86 when using the 16 MHz 80286)
- Large address space
  - 16 megabytes physical
  - 1 gigabyte virtual memory per task
- Integrated memory management, four-level memory protection and support for virtual memory and operating systems
- Two iAPX 86 upward compatible operating modes
  - iAPX 86 real address mode
  - Protected virtual address mode
- High bandwidth bus interface (16 megabyte/sec)
- Range of clock rates
  - 8 MHz 80286-8
  - 10 MHz for 80286-10 (Preliminary)
  - 12 MHz 80286-12 (Preliminary)
  - 16 MHz 80286-16 (Advance Info.)

GENERAL DESCRIPTION

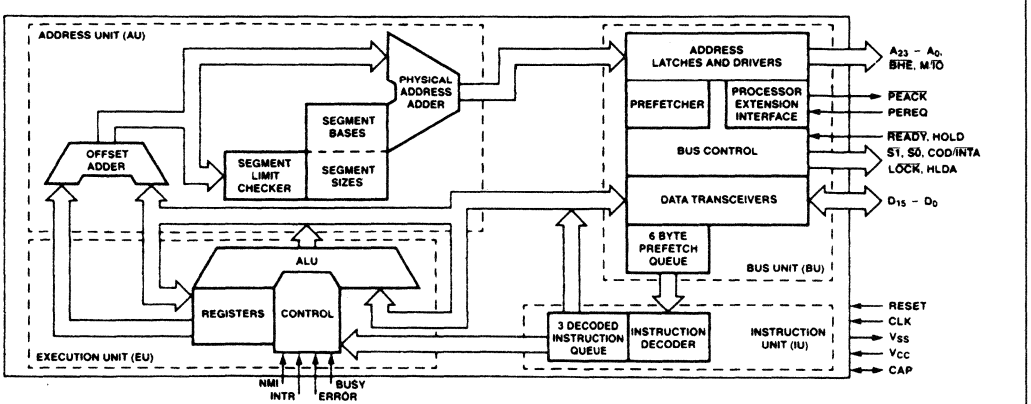
The 80286 is an advanced, high performance microprocessor with specially optimized capabilities for multiple user and multi-tasking systems. The 80286 has built-in memory protection that supports operating system and task isolation as well as program and data privacy within tasks. A 12 MHz 80286 provides up to ten times greater throughput than the standard 5 MHz 8086. The 80286 includes memory management capabilities that map up to 2^{30} bytes (one gigabyte) of virtual address space per task into 2^{24} bytes (16 megabytes) of physical memory.

The 80286 is upward compatible with iAPX 86 and 88 software. Using iAPX 86 real address mode, the 80286 is object code compatible with existing iAPX 86, 88 software.

In protected virtual address mode, the 80286 is source code compatible with iAPX 86, 88 software and may require upgrading to use virtual addresses supported by the 80286's integrated memory management and protection mechanism. Both modes operate at full 80286 performance and execute a superset of the iAPX 86 and 88 instructions.

The 80286 provides special operations to support the efficient implementation and execution of operating systems. For example, one instruction can end execution of one task, save its state, switch to a new task, load its state, and start execution of the new task. The 80286 also supports virtual memory systems by providing a segment-not-present exception and restartable instructions.

BLOCK DIAGRAM



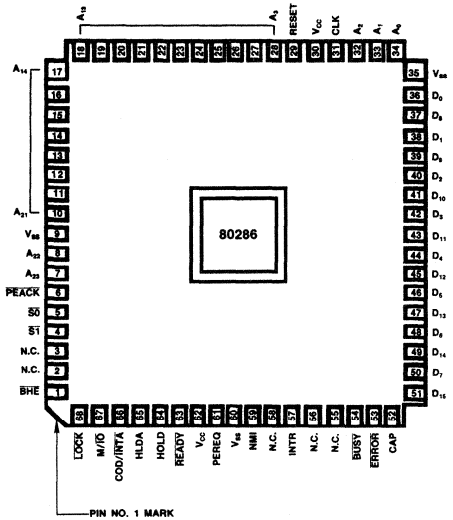
BD003960

Publication # 03552 | Rev. E | Amendment /0
Issue Date: May 1987

CONNECTION DIAGRAMS

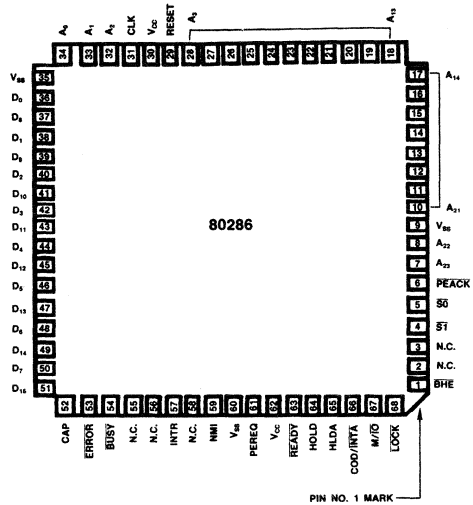
LCC

Component Pad Views –
as viewed from underside of component on the P.C. board.



CD005613

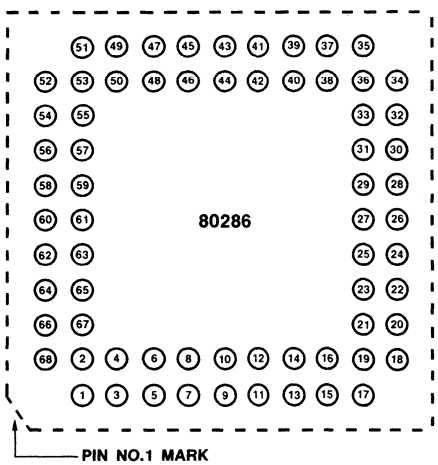
P.C. Board Views –
as viewed from the component side of the P.C. board.



CD005902

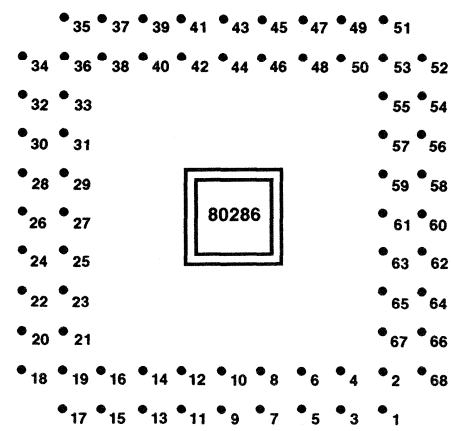
There are no electrical connections on the bottom of this package.

PGA



CD005794

Pins pointing away from viewer

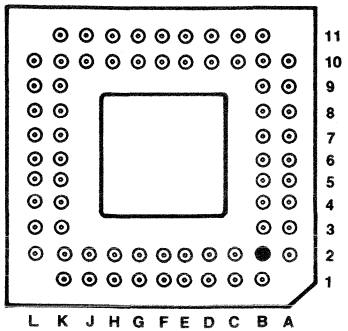


CD005802

Pins pointing toward viewer

PGA (continued)

Bottom View



CD005911

NAME	PAD	PIN
$\overline{BHE}$	1	B1
NC	2	B2
NC	3	C1
$\overline{S1}$	4	C2
$\overline{S0}$	5	D1
$\overline{PEACK}$	6	D2
A_{23}	7	E1
A_{22}	8	E2
V_{SS}	9	F1
A_{21}	10	F2
A_{20}	11	G1
A_{19}	12	G2
A_{18}	13	H1
A_{17}	14	H2
A_{16}	15	J1
A_{15}	16	J2
A_{14}	17	K1
A_{13}	18	L2
A_{12}	19	K2
A_{11}	20	L3
A_{10}	21	K3
A_9	22	L4
A_8	23	K4
A_7	24	L5
A_6	25	K5
A_5	26	L6
A_4	27	K6
A_3	28	L7
RESET	29	K7
V_{CC}	30	L8
CLK	31	K8
A_2	32	L9
A_1	33	K9
A_0	34	L10

NAME	PAD	PIN
V_{SS}	35	K11
D_0	36	K10
D_8	37	J11
D_1	38	J10
D_9	39	H11
D_2	40	H10
D_{10}	41	G11
D_3	42	G10
D_{11}	43	F11
D_4	44	F10
D_{12}	45	E11
D_5	46	E10
D_{13}	47	D11
D_6	48	D10
D_{14}	49	C11
D_7	50	C10
D_{15}	51	B11
CAP	52	A10
$\overline{ERROR}$	53	B10
$\overline{BUSY}$	54	A9
NC	55	B9
NC	56	A8
INTR	57	B8
NC	58	A7
NMI	59	B7
V_{SS}	60	A6
PEREQ	61	B6
V_{CC}	62	A5
$\overline{READY}$	63	B5
HOLD	64	A4
HLDA	65	B4
COD/$\overline{INTA}$	66	A3
M/$\overline{IO}$	67	B3
$\overline{LOCK}$	68	A2

RELATED AMD PRODUCTS

Part No.	Description
82C288*	Bus Controller
82284*	Clock Driver
82C54	Programmable Interval Timer
8259A	Interrupt Controller
Am9517A	DMA Controller

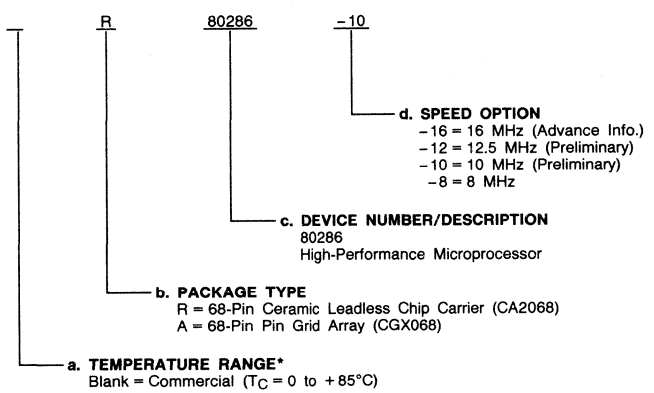
*For 12.5-MHz operation, see the "82284 and 82288 Emulation in an IBM PC/AT Computer Using Two AmPAL**16R8B Devices" Application Note, PID #08927A.

ORDERING INFORMATION

Commodity Products

AMD products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Temperature Range
b. Package Type
c. Device Number
d. Speed Option (if applicable)
e. Optional Processing



Valid Combinations	
R, A	80286-12
	80286-10
	80286-8

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device will also be available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for preliminary electrical performance characteristics.

PIN DESCRIPTION

Active State	Name	I/O	Description
Active HIGH	CLK	I	System Clock provides the fundamental timing for 80286 systems. It is divided by two inside the 80286 to generate the processor clock. The internal divide-by-two circuitry can be synchronized to an external clock generator by a LOW-to-HIGH transition on the RESET input.
Active HIGH	$D_0 - D_{15}$	I/O	Data Bus inputs data during memory, I/O, and interrupt acknowledge read cycles; outputs data during memory and I/O write cycles. The data bus is active HIGH and floats to three-state OFF during bus hold acknowledge.
Active HIGH	$A_{23} - A_0$	O	Address Bus outputs physical memory and I/O port addresses. A_0 is LOW when data is to be transferred on pins D_{7-0}. $A_{23} - A_{16}$ are LOW during I/O transfers. The address bus is active HIGH and floats to three-state OFF during bus hold acknowledge.
Active LOW	$\overline{BHE}$	O	Bus High Enable indicates transfer of data on the upper byte of the data bus D_{15-8}. Eight-bit oriented devices assigned to the upper byte of the data bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is active LOW and floats to three-state OFF during bus hold acknowledge.

$\overline{BHE}$ and A_0 Encodings

$\overline{BHE}$ Value	A_0 Value	Function
0	0	Word Transfer
0	1	Byte transfer on upper half of data bus (D_{15-8})
1	0	Byte transfer on lower half of data bus (D_{7-0})
1	1	Reserved

Active LOW	$\overline{S1}$, $\overline{S0}$	O	Bus Cycle Status indicates initiation of a bus cycle and, along with M/$\overline{IO}$ and COD/$\overline{INTA}$, defines the type of bus cycle. The bus is in a T_S state whenever one or both are LOW. $\overline{S1}$ and $\overline{S0}$ are active LOW and float to three-state OFF during bus hold acknowledge.

80286 Bus Cycle Status Definition

COD/$\overline{INTA}$	M/$\overline{IO}$	$\overline{S1}$	$\overline{S0}$	Bus cycle initiated
0 (LOW)	0	0	0	Interrupt acknowledge
0	0	0	1	Reserved
0	0	1	0	Reserved
0	0	1	1	None; not a status cycle
0	1	0	0	IF $A_1 = 1$ then halt; else shutdown
0	1	0	1	Memory data read
0	1	1	0	Memory data write
0	1	1	1	None; not a status cycle
1 (HIGH)	0	0	0	Reserved
1	0	0	1	I/O read
1	0	1	0	I/O write
1	0	1	1	None; not a status cycle
1	1	0	0	Reserved
1	1	0	1	Memory instruction read
1	1	1	0	Reserved
1	1	1	1	None; not a status cycle

	M/$\overline{IO}$	O	Memory/IO Select distinguishes memory access from I/O access. If HIGH during T_S, a memory cycle or a halt/shutdown cycle is in progress. If LOW, an I/O cycle or an interrupt acknowledge cycle is in progress. M/$\overline{IO}$ floats to three-state OFF during bus hold acknowledge.
	COD/$\overline{INTA}$	O	Code/Interrupt Acknowledge distinguishes instruction fetch cycles from memory data read cycles. Also distinguishes interrupt acknowledge cycles from I/O cycles. COD/$\overline{INTA}$ floats to three-state OFF during bus hold acknowledge.
Active LOW	$\overline{LOCK}$	O	Bus Lock indicates that other system bus masters are not to gain control of the system bus following the current bus cycle. The $\overline{LOCK}$ signal may be activated explicitly by the "LOCK" instruction prefix or automatically by 80286 hardware during memory XCHG instructions, interrupt acknowledge, or descriptor table access. $\overline{LOCK}$ is active LOW and floats to three-state OFF during bus hold acknowledge.
Active LOW	$\overline{READY}$	I	Bus Ready terminates a bus cycle. Bus cycles are extended without limit until terminated by $\overline{READY}$ LOW. $\overline{READY}$ is an active LOW synchronous input requiring set-up and hold times relative to the system clock be met for correct operation. $\overline{READY}$ is ignored during bus hold acknowledge.
Active HIGH	HOLD, HLDA	I O	Bus Hold Request and Hold Acknowledge control ownership of the 80286 local bus. The HOLD input allows another local bus master to request control of the local bus. When control is granted, the 80286 will float its bus drivers to three-state OFF and then active HLDA, thus entering the bus hold acknowledge condition. The local bus will remain granted to the requesting master until HOLD becomes inactive which results in the 80286 deactivating HLDA and regaining control of the local bus. This terminates the bus hold acknowledge condition. HOLD may be asynchronous to the system clock. These signals are active HIGH.

PIN DESCRIPTION (Cont.)

Active State	Name	I/O	Description
Active HIGH	INTR	I	Interrupt Request requests the 80286 to suspend its current program execution and service a pending external request. Interrupt requests are masked whenever the interrupt enable bit in the flag word is cleared. When the 80286 responds to an interrupt request, it performs two interrupt acknowledge bus cycles to read an 8-bit interrupt vector that identifies the source of the interrupt. To assure program interruption, INTR must remain active until the first interrupt acknowledge cycle is completed. INTR is sampled at the beginning of each processor cycle and must be active HIGH at least two processor cycles before the current instruction ends in order to interrupt before the next instruction. INTR is level sensitive, active HIGH, and may be asynchronous to the system clock.
Active HIGH	NMI	I	Non-maskable Interrupt Request interrupts the 80286 with an internally supplied vector value of 2. No interrupt acknowledge cycles are performed. The interrupt enable bit in the 80286 flag word does not affect this input. The NMI input is active HIGH, may be asynchronous to the system clock, and is edge triggered after internal synchronization. For proper recognition, the input must have been previously LOW for at least four system clock cycles and remain HIGH for at least four system clock cycles.
	PEREQ, PEACK	I O	Processor Extension Operand Request and Acknowledge extended the memory management and protection capabilities of the 80286 to processor extensions. The PEREQ input requests the 80286 to perform a data operand transfer for a processor extension. The PEACK output signals the processor extension when the requested operand is being transferred. PEREQ is active HIGH and may be asynchronous to the system clock. PEACK is active LOW.
Active LOW	BUSY, ERROR	I I	Processor Extension Busy and Error indicate the operating condition of a processor extension to the 80286. An active BUSY input stops 80286 program execution on WAIT and some ESC instructions until BUSY becomes inactive (HIGH). The 80286 may be interrupted while waiting for BUSY to become inactive. An active ERROR input causes the 80286 to perform a processor extension interrupt when executing WAIT or some ESC instructions. These inputs are active LOW and may be asynchronous to the system clock.
Active HIGH	RESET	I	System Reset clears the internal logic of the 80286 and is active HIGH. The 80286 may be reinitialized at any time with a LOW-to-HIGH transition on RESET which remains active for more than 16 system clock cycles. During RESET active, the output pins of the 80286 enter the state shown below: **80286 Pin State During Reset** **Pin Value** | **Pin Names** 1 (HIGH) | $\overline{S0}$, $\overline{S1}$, $\overline{PEACK}$, $A_{23} - A_0$, $\overline{BHE}$, $\overline{LOCK}$ 0 (LOW) | M/$\overline{IO}$, COD/$\overline{INTA}$, HLDA three-state OFF | $D_{15} - D_0$ Operation of the 80286 begins after a HIGH-to-LOW transition on RESET. The HIGH-to-LOW transition of RESET must be synchronous to the system clock. Approximately 50 system clock cycles are required by the 80286 for internal initializations before the first bus cycle to fetch code from the power-on execution address is performed. A LOW-to-HIGH transition of RESET synchronous to the system clock, will begin a new processor cycle at the next HIGH-to-LOW transition of the system clock. The LOW-to-HIGH transition of RESET may be asynchronous to the system clock; however, in this case it cannot be predetermined which phase of the processor clock will occur during the next system clock period. Synchronous LOW-to-HIGH transitions of RESET are only required for systems where the processor clock must be phase synchronous to another clock.
Active HIGH	V_{SS}	I	System Ground: 0 VOLTS.
Active HIGH	V_{CC}	I	System Power: +5 Volt Power Supply.
Active HIGH	CAP	I	Substrate Filter Capacitor: a 0.047μf ±20% 12 V capacitor must be connected between this pin and ground. This capacitor filters the output of the internal substrate bias generator. A maximum DC leakage current of 1 μa is allowed through the capacitor. For correct operation of the 80286, the substrate bias generator must charge this capacitor to its operating voltage. The capacitor charge-up time is 5 milliseconds (max.) after V_{CC} and CLK reach their specified AC and DC parameters. RESET may be applied to prevent spurious activity by the CPU during this time. After this time, the 80286 processor clock can be phase synchronized to another clock by pulsing RESET LOW synchronous to the system clock.

FUNCTIONAL DESCRIPTION

Introduction

The 80286 is an advanced, high-performance microprocessor with specially optimized capabilities for multiple user and multi-tasking systems. Depending on the application, the 80286's performance is up to six times faster than the standard 5 MHz 8086's, while providing complete upward software compatibility with AMD's iAPX 86, 88, and 186 family of CPU's.

The 80286 operates in two modes: iAPX 86 real address mode and protected virtual address mode. Both modes execute a superset of the iAPX 86 and 88 instruction set.

In iAPX 86 real address mode programs use real addresses with up to one megabyte of address space. Programs use virtual addresses in protected virtual address mode, also called protected mode. In protected mode, the 80286 CPU automatically maps 1 gigabyte of virtual addresses per task into a 16 megabyte real address space. This mode also provides memory protection to isolate the operating system and ensure privacy of each task's programs and data. Both modes provide the same base instruction set, registers, and addressing modes.

The following pages describe first, the base 80286 architecture common to both modes; second, iAPX 86 real address mode; and third, protected mode.

80286 Base Architecture

The iAPX 86, 88, 186, and 286 CPU family all contain the same basic set of registers, instructions, and addressing modes. The 80286 processor is upward compatible with the 8086, 8088, and 80186 CPU's.

Register Set

The 80286 base architecture has fifteen registers as shown in Figure 1. These registers are grouped into the following four categories:

General Registers: Eight 16-bit general purpose registers used to contain arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used either in their entirety as 16-bit words or split into pairs of separate 8-bit registers.

Segment Registers: Four 16-bit special purpose registers select, at any given time, the segments of memory that are immediately addressable for code, stack, and data. (For usage, refer to Memory Organization.)

Base and Index Registers: Four of the general purpose registers may also be used to determine offset addresses of operands in memory. These registers may contain base addresses or indexes to particular locations within a segment. The addressing mode determines the specific registers used for operand address calculations.

Status and Control Registers: Three 16-bit special purpose registers record or control certain aspects of the 80286 processor state. These include the Instruction Pointer, which contains the offset address of the next sequential instruction to be executed.

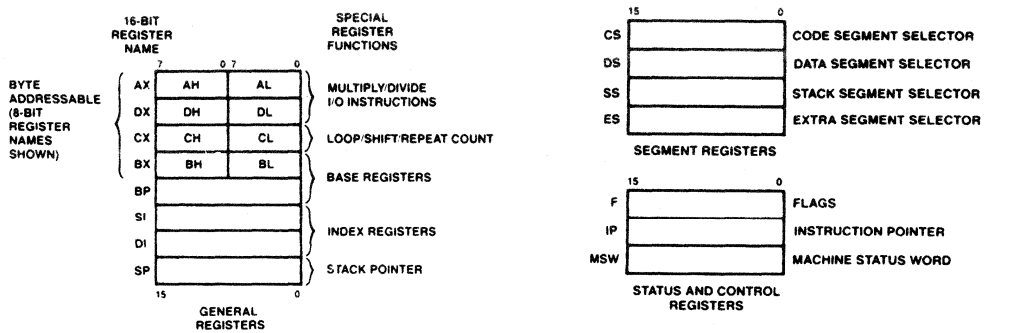


Figure 1. Register Set

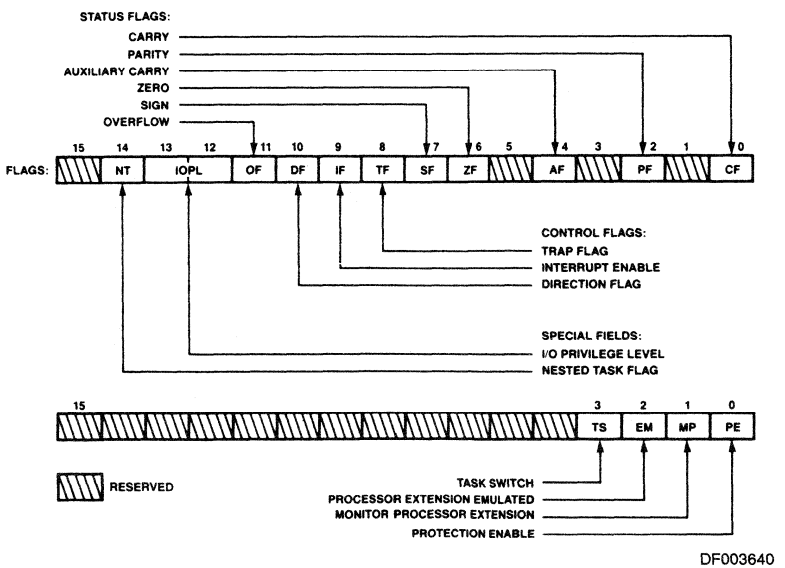


Figure 2. Status and Control Register Bit Functions

Flags Word Description

The Flags word (Flags) records specific characteristics of the result of logical and arithmetic instructions (bits 0, 2, 4, 6, 7, and 11) and controls the operation of the 80286 within a given operating mode (bits 8 and 9). Flags is a 16-bit register. The function of the flag bits is given in Table 1.

Table 1. Flags Word Bit Functions

Bit Position	Name	Function
0	CF	Carry Flag — Set on high-order bit carry or borrow; cleared otherwise
2	PF	Parity Flag — Set if low-order 8 bits of result contain an even number of 1 bits; cleared otherwise
4	AF	Set on carry-from or borrow-to the low-order four bits of AL; cleared otherwise
6	ZF	Zero Flag — Set if result is zero; cleared otherwise
7	SF	Sign Flag — Set equal to high-order bit of result (0 if positive, 1 if negative)
11	OF	Overflow Flag — Set if result is a too-large large positive number or a too-small negative number (excluding sign-bit) to fit in destination operand; cleared otherwise
8	TF	Single Step Flag — Once set, a single step interrupt occurs after the next instruction executes. TF is cleared by the single step interrupt.
9	IF	Interrupt-Enable Flag — When set, maskable interrupts will cause the CPU to transfer control to an interrupt vector specified location
10	DF	Direction Flag — Causes string instructions to auto decrement the appropriate index registers when set. Clearing DF causes auto increment.

Instruction Set

The instruction set is divided into seven categories: data transfer, arithmetic, shift/rotate/logical, string manipulation, program transfer, high-level instructions, and processor control. These categories are summarized in Figures 3-9.

An 80286 instruction can reference zero, one, or two operands where an operand resides in a register, in the instruction itself, or in memory. Zero-operand instructions (e.g., NOP and HLT) are usually one byte long. One-operand instructions (e.g., INC and DEC) are usually two bytes long, but some are encoded in only one byte. One-operand instructions may reference a register or memory location. Two-operand instructions permit the following six types of instruction operations:

— Register to Register

— Memory to Register

— Immediate to Register

— Memory to Memory

— Register to Memory

— Immediate to Memory

Two-operand instructions (e.g., MOV and ADD) are usually three to six bytes long. Memory to memory operations are provided by a special class of string instructions requiring one to three bytes. For detailed instruction formats and encodings refer to the instruction set summary at the end of this document.

GENERAL PURPOSE	
MOV	Move byte or word
PUSH	Push word onto stack
POP	Pop word off stack
PUSHA	Push all registers on stack
POPA	Pop all registers from stack
XCHG	Exchange byte or word
XLAT	Translate byte
INPUT/OUTPUT	
IN	Input byte or word
OUT	Output byte or word
ADDRESS OBJECT	
LEA	Load effective address
LDS	Load pointer using DS
LES	Load pointer using ES
FLAG TRANSFER	
LAHF	Load AH register from flags
SAHF	Store AH register in flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack

Figure 3. Data Transfer Instructions

ADDITION	
ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
SUBTRACTION	
SUB	Subtract byte or word
SBB	Subtract byte or word with borrow
DEC	Decrement byte or word by 1
NEG	Negate byte or word
CMP	Compare byte or word
AAS	ASCII adjust for subtraction
DAS	Decimal adjust for subtraction
MULTIPLICATION	
MUL	Multiply byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII adjust for multiply
DIVISION	
DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
CBW	Convert byte to word
CWD	Convert word to doubleword

Figure 4. Arithmetic Instructions

MOVS	Move byte or word string
INS	Input bytes or word string
OUTS	Output bytes or word string
CMPS	Compare byte or word string
SCAS	Scan byte or word string
LODS	Load byte or word string
STOS	Store byte or word string
REP	Repeat
REPE/REPZ	Repeat while equal/zero
REPNE/REPNZ	Repeat while not equal/not zero

Figure 5. String Instructions

LOGICALS	
NOT	"Not" byte or word
AND	"And" byte or word
OR	"Inclusive or" byte or word
XOR	"Exclusive or" byte or word
TEST	"Test" byte or word
SHIFTS	
SHL/SAL	Shift logical/arithmetic left byte or word
SHR	Shift logical right byte or word
SAR	Shift arithmetic right byte or word
ROTATES	
ROL	Rotate left byte or word
ROR	Rotate right byte or word
RCL	Rotate through carry left byte or word
RCR	Rotate through carry right byte or word

Figure 6. Shift/Rotate/Logical Instructions

CONDITIONAL TRANSFERS		UNCONDITIONAL TRANSFERS	
JA/JNBE	Jump if above/not below nor equal	CALL	Call procedure
JAE/JNB	Jump if above or equal/not below	RET	Return from procedure
JB/JNAE	Jump if below/not above nor equal	JMP	Jump
JBE/JNA	Jump if below or equal/not above		
JC	Jump if carry	**ITERATION CONTROLS**	
JE/JZ	Jump if equal/zero		
JG/JNLE	Jump if greater/not less nor equal	LOOP	Loop
JGE/JNL	Jump if greater or equal/not less	LOOPE/LOOPZ	Loop if equal/zero
JL/JNGE	Jump if less/not greater nor equal	LOOPNE/LOOPNZ	Loop if not equal/not zero
JLE/JNG	Jump if less or equal/not greater	JCXZ	Jump if register CX = 0
JNC	Jump if not carry		
JNE/JNZ	Jump if not equal/not zero	**INTERRUPTS**	
JNO	Jump if not overflow		
JNP/JPO	Jump if not parity/parity odd	INT	Interrupt
JNS	Jump if not sign	INTO	Interrupt if overflow
JO	Jump if overflow	IRET	Interrupt return
JP/JPE	Jump if parity/parity even		
JS	Jump if sign		

Figure 7. Program Transfer Instructions

FLAG OPERATIONS	
STC	Set carry flag
CLC	Clear carry flag
CMC	Complement carry flag
STD	Set direction flag
CLD	Clear direction flag
STI	Set interrupt enable flag
CLI	Clear interrupt enable flag
EXTERNAL SYNCHRONIZATION	
HLT	Halt until interrupt or reset
WAIT	Wait for $\overline{\text{BUSY}}$ not active
ESC	Escape to extension processor
LOCK	Lock bus during next instruction
NO OPERATION	
NOP	No operation
EXECUTION ENVIRONMENT CONTROL	
LMSW	Load machine status word
SMSW	Store machine status word

Figure 8. Processor Control Instructions

ENTER	Format stack for procedure entry
LEAVE	Restore stack for procedure exit
BOUND	Detects values outside prescribed range

Figure 9. High-Level Instructions

Memory Organization

Memory is organized as sets of variable length segments. Each segment is a linear contiguous sequence of up to 64K(2^{16}) 8-bit bytes. Memory is addressed using a two-component address (a pointer) that consists of a 16-bit segment selector and a 16-bit offset. The segment selector indicates the desired segment in memory. The offset component indicates the desired byte address within the segment.

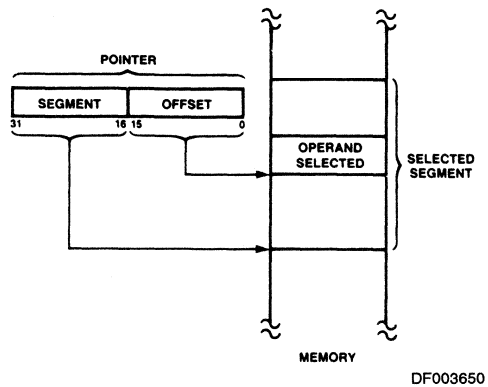


Figure 10. Two-Component Address

Table 2. Segment Register Selection Rules

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Instructions	Code (CS)	Automatic with instruction prefetch
Stack	Stack (SS)	All stack pushes and pops. Any memory reference which uses BP as a base register.
Local Data	Data (DS)	All data references except when relative to stack or string destination
External (Global) Data	Extra (ES)	Alternate data segment and destination of string operation

All instructions that address operands in memory must specify the segment and the offset. For speed and compact instruction encoding, segment selectors are usually stored in the high speed segment registers. An instruction need specify only the desired segment register and an offset to address a memory operand.

Most instructions need not explicitly specify which segment register is used. The correct segment register is automatically chosen according to the rules of Table 2. These rules follow the way programs are written (see Figure 11) as independent modules that require areas for code and data, a stack, and access to external data areas.

Special segment override instruction prefixes allow the implicit segment register selection rules to be overridden for special cases. The stack, data, and extra segments may coincide for simple programs. To access operands that do not reside in one of the four immediately available segments, either a full 32-bit pointer can be used or a new segment selector must be loaded.

Addressing Modes

The 80286 provides a total of eight addressing modes for instructions to specify operands. Two addressing modes are provided for instructions that operate on register or immediate operands:

Register Operand Mode: The operand is located in one of the 8- or 16-bit general registers.

Immediate Operand Mode: The operand is included in the instruction.

Six modes are provided to specify the location of an operand in a memory segment. A memory operand address consists of two 16-bit components: segment selector and offset. The segment selector is supplied by a segment register either implicitly chosen by the addressing mode or explicitly chosen by a segment override prefix. The offset is calculated by summing any combination of the following three address elements:

the displacement (an 8- or 16-bit immediate value contained in the instruction)

the base (contents of either the BX or BP base registers)

the index (contents of either the SI or DI index registers)

Any carry out from the 16-bit addition is ignored. Eight-bit displacements are sign extended to 16-bit values.

Combinations of these three address elements define the six memory addressing modes, here described.

Direct Mode: The operand's offset is contained in the instruction as an 8- or 16-bit displacement element.

Register Indirect Mode: The operand's offset is in one of the registers SI, DI, BX, or BP.

Based Mode: The operand's offset is the sum of an 8- or 16-bit displacement and the contents of a base register (BX or BP).

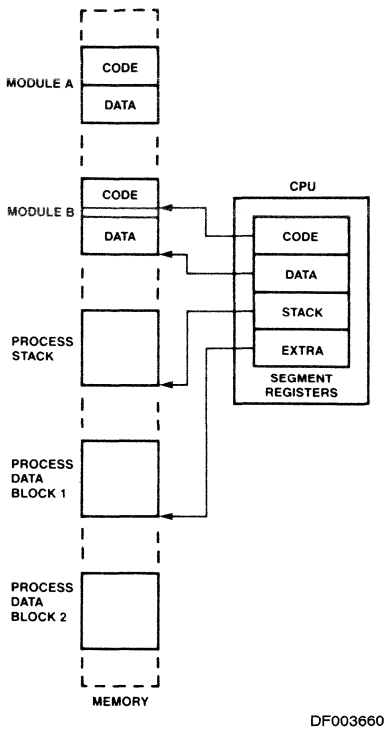


Figure 11. Segmented Memory Helps Structure Software

Indexed Mode: The operand's offset is the sum of an 8- or 16-bit displacement and the contents of an index register (SI or DI).

Based Indexed Mode: The operand's offset is the sum of the contents of a base register and an index register.

Based Indexed Mode with Displacement: The operand's offset is the sum of a base register's contents, an index register's contents, and an 8- or 16-bit displacement.

Data Types

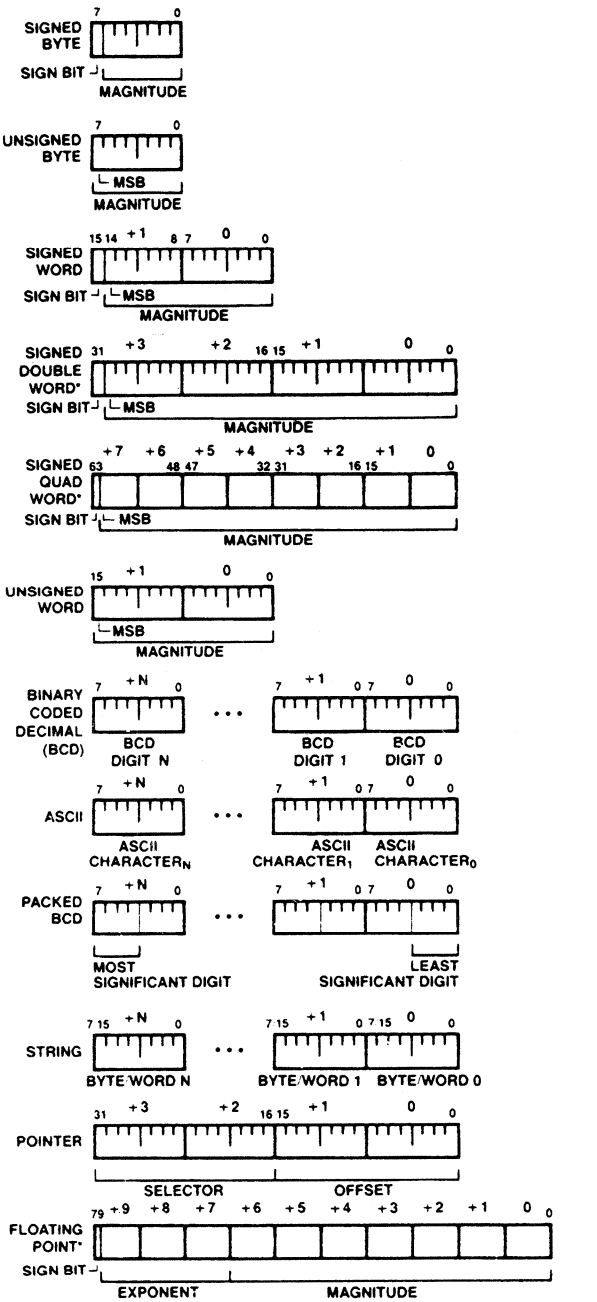
The 80286 directly supports the following data types:

Integer:	A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a 2's complement representation. Signed 32- and 64-bit integers are supported using the 80287 Numeric Data Processor.
Ordinal:	An unsigned binary numeric value contained in an 8-bit byte or 16-bit word.
Pointer:	A 32-bit quantity, composed of a segment selector component and an offset component. Each component is a 16-bit word.
String:	A contiguous sequence of bytes or words. A string may contain from 1 byte to 64K bytes.
ASCII:	A byte representation of alphanumeric and control characters using the ASCII standard of character representation.
BCD:	A byte (unpacked) representation of the decimal digits 0 – 9.
Packed BCD:	A byte (packed) representation of two decimal digits 0 – 9 storing one digit in each nibble of the byte.
Floating Point:	A signed 32-, 64-, or 80-bit real number representation. (Floating point operands are supported using the iAPX 287 Numeric Processor configuration.)

Figure 12 graphically represents the data types supported by the 80286.

I/O Space

The I/O space consists of 64K 8-bit or 32K 16-bit ports. I/O instructions address the I/O space with either an 8-bit port address, specified in the instruction, or a 16-bit port address in the DX register. 8-bit port addresses are zero extended such that $A_{15} - A_8$ are LOW. I/O port addresses 00F8(H) through 00FF(H) are reserved.



DF003670

*Support by iAPX 286/287 Numeric Data Processor Configuration

Figure 12. 80286 Supported Data Types

Table 3. Interrupt Vector Assignments

Function	Interrupt Number	Related Instructions	Return Address Before Instruction Causing Exception?
Divide error exception	0	DIV, IDIV	Yes
Single step interrupt	1	All	
NMI interrupt	2	All	
Breakpoint interrupt	3	INT	
INTO detected overflow exception	4	INTO	No
BOUND range exceeded exception	5	BOUND	Yes
Invalid opcode exception	6	Any undefined opcode	Yes
Processor extension not available exception	7	ESC or WAIT	Yes
Reserved	8-15		
Processor extension error input	16	ESC or WAIT	
Reserved	17-31		
User defined	32-255		

Interrupts

An interrupt transfers execution to a new program location. The old program address (CS:IP) and machine state (Flags) are saved on the stack to allow resumption of the interrupted program. Interrupts fall into three classes: hardware initiated, INT instructions, and instruction exceptions. Hardware initiated interrupts occur in response to an external input and are classified as non-maskable or maskable. Programs may cause an interrupt with an INT instruction. Instruction exceptions occur when an unusual condition, which prevents further instruction processing, is detected while attempting to execute an instruction. The return address from an exception will always point at the instruction causing the exception and include any leading instruction prefixes.

A table containing up to 256 pointers defines the proper interrupt service routine for each interrupt. Interrupts 0 – 31, some of which are used for instruction exceptions, are reserved. For each interrupt, an 8-bit vector must be supplied to the 80286 which identifies the appropriate table entry. Exceptions supply the interrupt vector internally. INT instructions contain or imply the vector and allow access to all 256 interrupts. Maskable hardware initiated interrupts supply the 8-bit vector to the CPU during an interrupt acknowledge bus sequence. Non-maskable hardware interrupts use a predefined internally supplied vector.

Maskable Interrupt (INTR)

The 80286 provides a maskable hardware interrupt request pin, INTR. Software enables this input by setting the interrupt flag bit (IF) in the flag word. All 224 user-defined interrupt sources can share this input, yet they can retain separate interrupt handlers. An 8-bit vector read by the CPU during the interrupt acknowledge sequence (discussed in System Interface section) identifies the source of the interrupt.

Further maskable interrupts are disabled while servicing an interrupt by resetting the IF but as part of the response to an interrupt or exception. The saved flag word will reflect the enable status of the processor prior to the interrupt. Until the flag word is restored to the flag register, the interrupt flag will be zero unless specifically set. The interrupt return instruction includes restoring the flag word, thereby restoring the original status of IF.

Non-Maskable Interrupt Request (NMI)

A non-maskable interrupt input (NMI) is also provided. NMI has higher priority than INTR. A typical use of NMI would be to activate a power failure routine. The activation of this input causes an interrupt with an internally supplied vector value of 2. No external interrupt acknowledge sequence is performed.

While executing the NMI servicing procedure, the 80286 will not service further NMI requests, INTR requests, or the processor extension segment overrun interrupt until an interrupt return (IRET) instruction is executed or the CPU is reset. If NMI occurs while currently servicing an NMI, its presence will be saved for servicing after executing the first IRET instruction. IF is cleared at the beginning of an NMI interrupt to inhibit INTR interrupts.

Single Step Interrupt

The 80286 has an internal interrupt that allows programs to execute one instruction at a time. It is called the single step interrupt and is controlled by the single step flag bit (TF) in the flag word. Once this bit is set, an internal single step interrupt will occur after the next instruction has been executed. The interrupt clears the TF bit and uses an internally supplied vector of 1. The IRET instruction is used to set the TF bit and transfer control to the next instruction to be single stepped.

Interrupt Priorities

When simultaneous interrupt requests occur, they are processed in a fixed order as shown in Table 4. Interrupt processing involves saving the flags, return address, and setting CS:IP to point at the first instruction of the interrupt handler. If other interrupts remain enabled, they are processed before the first instruction of the current interrupt handler is executed. The last interrupt processed is therefore the first one serviced.

Table 4. Interrupt Processing Order

Order	Interrupt
1	INT instruction or exception
2	Single step
3	NMI
4	Processor extension segment overrun
5	INTR

Initialization and Processor Reset

Processor initialization or start up is accomplished by driving the RESET input pin HIGH. RESET forces the 80286 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as RESET is active. After RESET becomes inactive and an internal processing interval elapses, the 80286 begins execution in real address mode with the instruction at physical location FFFFF0(H). RESET also sets some registers to predefined values as shown in Table 5.

Table 5. 80286 Initial Register State after RESET

Flag word	0002(H)
Machine Status Word	FFF0(H)
Instruction pointer	FFF0(H)
Code segment	F000(H)
Data segment	0000(H)
Extra segment	0000(H)
Stack segment	0000(H)

Machine Status Word Description

The machine status word (MSW) records when a task switch takes place and controls the operating mode of the 80286. It is a 16-bit register of which the lower four bits are used. One bit places the CPU into protected mode, while the other three bits, as shown in Table 6, control the processor extension interface. After RESET, this register contains FFF0(H) which places the 80286 in iAPX 86 real address mode.

Table 6. MSW Bit Functions

Bit Position	Name	Function
0	PE	Protected mode Enable places the 80286 into protected mode and cannot be cleared except by RESET.
1	MP	Monitor Processor extension allows WAIT instructions to cause a processor extension not present exception (number 7).
2	EM	Emulate processor extension causes a processor extension not present exception (number 7) on ESC instructions to allow emulating a processor extension.
3	TS	Task Switched indicates the next instruction using a processor extension will cause exception 7, allowing software to test whether the current processor extension context belongs to the current task.

The LMSW and SMSW instructions can load and store the MSW in real address mode. The recommended use of TS, EM, and MP is shown in Table 7.

Halt

The HLT instruction stops program execution and prevents the CPU from using the local bus until restarted. Either NMI, INTR with IF = 1, or RESET will force the 80286 out of halt. If interrupted, the saved CS:IP will point to the next instruction after the HLT.

iAPX 86 Real Address Mode

The 80286 executes a fully upward-compatible superset of the 8086 instruction set in real address mode. In real address mode the 80286 is object code compatible with 8086 and 8088 software. The real address mode architecture (registers and addressing modes) is exactly as described in the 80286 Base Architecture section.

Table 7. Recommended MSW Encodings For Processor Extension Control

TS	MP	EM	Recommended Use	Instructions Causing Exception
0	0	0	iAPX 86 real address mode only. Initial encoding after RESET. 80286 operation is identical to iAPX 86, 88.	None
0	0	1	No processor extension is available. Software will emulate its function.	ESC
1	0	1	No processor extension is available. Software will emulate its function. The current processor extension context may belong to another task.	ESC
0	1	0	A processor extension exists.	None
1	1	0	A processor extension exists. The current processor extension context may belong to another task. The exception on WAIT allows software to test for an error pending from a previous processor extension operation.	ESC or WAIT

Memory Size

Physical memory is a contiguous array of up to 1,048,576 bytes (one megabyte) addressed by pins A_0 through A_{19} and $\overline{BHE}$. A_{20} through A_{23} are ignored.

Memory Addressing

In real address mode the processor generates 20-bit physical addresses directly from a 20-bit segment base address and a 16-bit offset.

The selector portion of a pointer is interpreted as the upper 16 bits of a 20-bit segment address. The lower four bits of the 20-bit segment address are always zero. Segment addresses, therefore, begin on multiples of 16 bytes. See Figure 13 for a graphic representation of address formation.

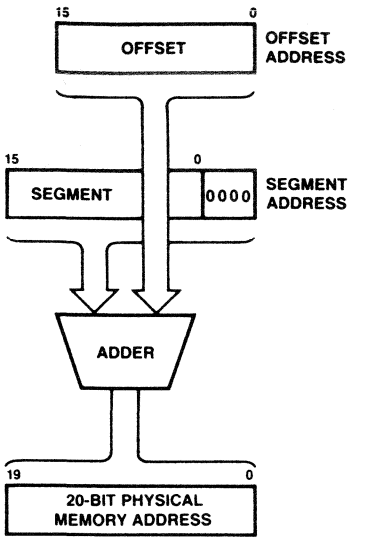


Figure 13. iAPX 86 Real Address Mode Address Calculation

All segments in real address mode are 64K bytes in size and may be read, written, or executed. An exception or interrupt can occur if data operands or instructions attempt to wrap around the end of a segment (e.g. a word with its low order byte at offset FFFF(H) and its high order byte at offset 0000(H)). If, in real address mode, the information contained in a segment does not use the full 64K bytes, the unused end of the segment may be overlayed by another segment to reduce physical memory requirements.

Reserved Memory Locations

The 80286 reserves two fixed areas of memory in real address mode (see Figure 14): system initialization area and interrupt table area. Locations from addresses FFFF0(H) through FFFFF(H) are reserved for system initialization. Initial execution begins at location FFFF0(H). Locations 00000(H) through 003FF(H) are reserved for interrupt vectors.

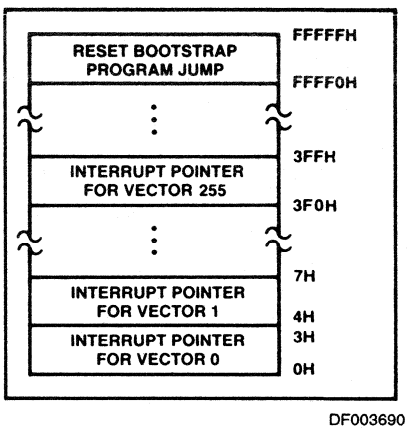


Figure 14. iAPX 86 Real Address Mode Initially Reserved Memory Locations

Table 8. Real Address Mode Addressing Interrupts

Function	Interrupt Number	Related Instructions	Return Address Before Instruction?
Interrupt table limit too small exception	8	INT vector is not within table limit	Yes
Processor extension segment overrun interrupt	9	ESC with memory operand extending beyond offset FFFF(H)	No
Segment overrun exception	13	Word memory reference with offset = FFFF(H) or an attempt to execute past the end of a segment	Yes

Interrupts

Table 8 shows the interrupt vectors reserved for exceptions and interrupts which indicate an addressing error. The exceptions leave the CPU in the state existing before attempting to execute the failing instruction (except for PUSH, POP, PUSHA, or POPA). Refer to the next section on protected mode initialization for a discussion on exception 8.

Protected Mode Initialization

To prepare the 80286 for protected mode, the LIDT instruction is used to load the 24-bit interrupt table base and 16-bit limit for the protected mode interrupt table. This instruction can also set a base and limit for the interrupt vector table in real address mode. After reset, the interrupt table base is initialized to 000000(H) and its size set to 03FF(H). These values are compatible with iAPX 86, 88 software. LIDT should only be executed in preparation for the protected mode.

Shutdown

Shutdown occurs when a severe error is detected that prevents further instruction processing by the CPU. Shutdown and halt are externally signalled via a halt bus operation. They can be distinguished by A_1 HIGH for halt and A_1 LOW for shutdown. In real address mode, shutdown can occur under two conditions:

- Exceptions 8 or 13 happen and the IDT limit does not include the interrupt vector.
- A CALL, INT, or POP instruction attempts to wrap around the stack segment when SP is not even.

An NMI input can bring the CPU out of shutdown if the IDT limit is at least 000F(H) and SP is greater than 0005(H); otherwise, shutdown can only be exited via the RESET input.

Protected Virtual Address Mode

The 80286 executes a fully upward-compatible superset of the 8086 instruction set in protected virtual address mode (protected mode). Protected mode also provides memory management and protection mechanisms and associated instructions.

The 80286 enters protected virtual address mode from real address mode by setting the PE (Protection Enable) bit of the machine status word with the Load Machine Status Word (LMSW) instruction. Protected mode offers extended physical and virtual memory address space, memory protection mechanisms, and new operations to support operating systems and virtual memory.

All registers, instructions, and addressing modes described in the 80286 Base Architecture section remain the same. Programs for the iAPX 86, 88, 186, and real address mode 80286 can be run in protected mode; however, embedded constants for segment selectors are different.

Memory Size

The protected mode 80286 provides a 1 gigabyte virtual address space per task mapped into a 16-megabyte physical address space defined by the address pin $A_{23} - A_0$ and $\overline{BHE}$. The virtual address space may be larger than the physical address space since any use of an address that does not map to a physical memory location will cause a restartable exception.

Memory Addressing

As in real address mode, protected mode uses 32-bit pointers, consisting of 16-bit selector and offset components. The selector, however, specifies an index into a memory resident table rather than the upper 16-bits of a real memory address. The 24-bit base address of the desired segment is obtained from the tables in memory. The 16-bit offset is added to the segment base address to form the physical address as shown in Figure 15. The tables are automatically referenced by the CPU whenever a segment register is loaded with a selector. All 80286 instructions which load a segment register will reference the memory based tables without additional software. The memory based tables contain 8 byte values called descriptors.

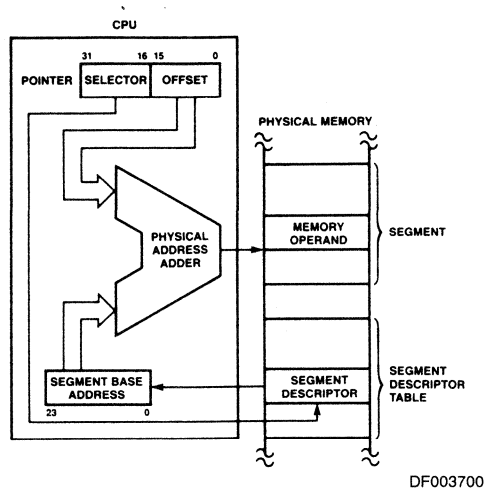


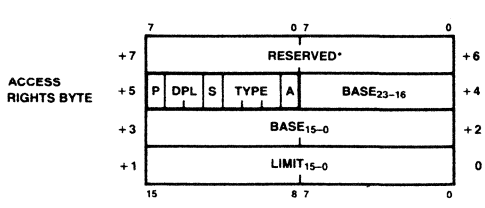
Figure 15. Protected Mode Memory Addressing

Descriptors

Descriptors define the use of memory. Special types of descriptors also define new functions for transfer of control and task switching. The 80286 has segment descriptors for code, stack and data segments, and system control descriptors for special system data segments and control transfer operations. Descriptor accesses are performed as locked bus operations to assure descriptor integrity in multi-processor systems.

Code and Data Segment Descriptors (S = 1)

Besides segment base addresses, code and data descriptors contain other segment attributes including segment size (1 to 64K bytes), access rights (read only, read/write, execute only, and execute/read), and presence in memory (for virtual memory systems)(see Figure 16). Any segment usage violating a segment attribute indicated by the segment descriptor will prevent the memory cycle and cause an exception or interrupt.



*Must be set to 0 for compatability with iAPX 386.

Access Rights Byte Definition

	Bit Position	Name	Function	
	7	Present (P)	P = 1 Segment is mapped into physical memory. P = 0 No mapping to physical memory exists; base and limit are not used.	
	6-5	Descriptor Privilege Level (DPL)	Segment privilege attribute used in privilege tests.	
	4	Segment Descriptor (S)	S = 1 Code or Data segment descriptor S = 0 Non-segment descriptor	
Type Field Definition	3	Executable (E)	E = 0 Data segment descriptor type is:	Data Segment
	2	Expansion Direction (ED)	ED = 0 Grow up segment, offsets must be ≤ limit. ED = 1 Grow down segment, offsets must be > limit.	Data Segment
	1	Writable (W)	W = 0 Data segment may not be written into. W = 1 Data segment may be written into.	Data Segment
	3	Executable (E)	E = 1 Code Segment Descriptor type is:	Code Segment
	2	Conforming (C)	C = 1 Code segment may only be executed when CPL ≥ DPL.	Code Segment
	1	Readable (R)	R = 0 Code segment may not be read. R = 1 Code segment may be read.	Code Segment
	0	Accessed (A)	A = 0 Segment has not been accessed. A = 1 Segment selector has been loaded into segment register or used by selector test instructions.	

Figure 16. Code and Data Segment Descriptors

Code and data are stored in two types of segments: code segments and data segments. Both types are identified and defined by segment descriptors. Code segments are identified by the executable (E) bit set to 1 in the descriptor access rights byte. The access rights byte of both code and data segment descriptor types have three fields in common: present (P) bit, Descriptor Privilege Level (DPL), and accessed (A) bit. If P = 0, any attempted use of this segment will cause a not-present exception. DPL specifies the privilege level of the segment descriptor. DPL controls when the descriptor may be used by a task (refer to privilege discussion). The A bit shows whether the segment has been previously accessed for usage profiling, a necessity for virtual memory systems. The CPU will always set this bit when accessing the descriptor.

Data segments (S = 1, E = 0) may be either read-only or read-write as controlled by the W bit of the access rights byte. Read-only (W = 0) data segments may not be written into. Data segments may grow in two directions, as determined by the Expansion Direction (ED) bit: upwards (ED = 0) for data segments, and downwards (ED = 1) for a segment containing a stack. The limit field for a data segment descriptor is interpreted differently depending on the ED bit (see Figure 16).

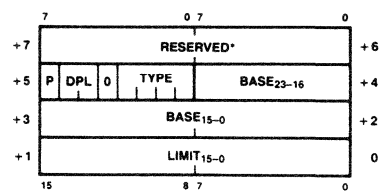
A code segment (S = 1, E = 1) may be execute-only or execute/read as determined by the Readable (R) bit. Code segments may never be written into and execute-only code segments (R = 0) may not be read. A code segment may also have an attribute called Conforming (C). A conforming code segment may be shared by programs that execute at different privilege levels. The DPL of a conforming code segment defines the range of privilege levels at which the segment may be executed (refer to privilege discussion).

System Segment Descriptors
(S = 0, TYPE 1-3)

In addition to code and data segment descriptors, the protected mode 80286 defines system segment descriptors. These descriptors define special system data segments which contain a table of descriptors (Local Descriptor Table Descriptor) or segments which contain the execution state of a task (Task State Segment Descriptor).

Figure 17 gives the formats for the special system data segment descriptors. The descriptors contain a 24-bit base address of the segment and a 16-bit limit. The access byte defines the type of descriptor, its state and privilege level. The descriptor contents are valid and the segment is in physical memory if P = 1. If P = 0, the segment is not valid. The DPL field is only used in Task State Segment descriptors and indicates the privilege level at which the descriptor may be used (see Privilege). Since the Local Descriptor Table descriptor may only be used by a special privileged instruction, the DPL field is not used. Bit 4 of the access byte is 0 to indicate that it is a system control descriptor. The Type field specifies the descriptor type as indicated in Figure 17.

System Segment Descriptor



TB000088

*Must be set to 0 for compatability with iAPX 386.

System Segment Descriptor Fields

Name	Value	Description
TYPE	1 2 3	Available Task State Segment Local Descriptor Table Descriptor Busy Task State Segment
P	0 1	Descriptor contents are not valid Descriptor contents are valid
DPL	0 – 3	Descriptor Privilege Level
BASE	24-bit number	Base Address of special system data segment in real memory
LIMIT	16-bit number	Offset of last byte in segment

Figure 17. System Segment Format

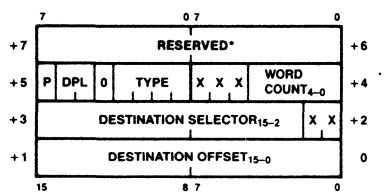
Gate Descriptors
(S = 0, TYPE = 4-7)

Gates are used to control access to entry points within the target code segment. The gate descriptors are call gates, task gates, interrupt gates and trap gates. Gates provide a level of indirection between the source and destination of the control transfer. This indirection allows the CPU to automatically perform protection checks and control the entry point of the destination. Call gates are used to change privilege levels (see Privilege); task gates are used to perform a task switch; and interrupt and trap gates are used to specify interrupt service routines. The interrupt gate disables interrupts (resets IF) while the trap gate does not.

Figure 18 shows the format of the gate descriptors. The descriptor contains a destination pointer that points to the descriptor of the target segment and the entry point offset. The destination selector in an interrupt gate, trap gate, and call gate must refer to a code segment descriptor. These gate descriptors contain the entry point to prevent a program from constructing and using an illegal entry point. Task gates may only refer to a task state segment. Since task gates invoke a task switch, the destination offset is not used in the task gate.

Exception 13 is generated when the gate is used if a destination selector does not refer to the correct descriptor type. The Word Count field is used in the call gate descriptor to indicate the number of parameters (0 – 31 words) to be automatically copied from the caller's stack to the stack of the called routine when a control transfer changes privilege levels. The Word Count field is not used by any other gate descriptor.

Gate Descriptor



TB000086

*Must be set to 0 for compatability with iAPX 386.

Gate Descriptor Fields

Name	Value	Description
TYPE	4 5 6 7	–Call Gate –Task Gate –Interrupt Gate –Trap Gate
P	0 1	–Descriptor Contents are not valid –Descriptor Contents are valid
DPL	0 – 3	Descriptor Privilege Level
WORD COUNT	0 – 31	Number of words to copy from callers stack to called procedures stack. Only used with call gate.
DESTINATION SELECTOR	16-bit selector	Selector to the target code segment (Call, Interrupt or Trap Gate) Selector to the target task state segment (Task Gate)
DESTINATION OFFSET	16-bit offset	Entry point within the target code segment

Figure 18. Gate Descriptor Format

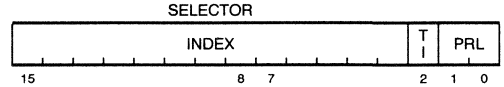
The access byte format is the same for all gate descriptors. P = 1 indicates that the gate contents are valid. P = 0 indicates the contents are not valid and causes exception 11 if referenced. DPL is the Descriptor Privilege Level and specifies when this descriptor may be used by a task (refer to privilege discussion). Bit 4 must equal 0 to indicate a system control descriptor. The Type field specifies the descriptor type as indicated in Figure 18.

Segment Descriptor Cache Registers

A segment descriptor cache register is assigned to each of the four segment registers (CS, SS, DS, ES). Segment descriptors are automatically loaded (cached) into a segment descriptor cache register (Figure 20) whenever the associated segment register is loaded with a selector. Only segment descriptors may be loaded into segment descriptor cache registers. Once loaded, all references to that segment of memory use the cached descriptor information instead of reaccessing memory. The descriptor cache registers are not visible to programs. No instructions exist to store their contents. They only change when a segment register is loaded.

Selector Fields

A protected mode selector has three fields: descriptor entry index, local or global descriptor table indicator (TI), and selector privilege (RPL) as shown in Figure 19. These fields select one of two memory based tables of descriptors, select the appropriate table entry and allow high-speed testing of the selector's privilege attribute (refer to privilege discussion).



Bits	Name	Function
1 – 0	REQUESTED PRIVILEGE LEVEL (RPL)	INDICATES SELECTOR PRIVILEGE LEVEL DESIRED
2	TABLE INDICATOR (TI)	TI = 0 USE GLOBAL DESCRIPTOR TABLE (GDT) TI = 1 USE LOCAL DESCRIPTOR TABLE (LDT)
15 – 3	INDEX	SELECT DESCRIPTOR ENTRY IN TABLE

Figure 19. Selector Fields

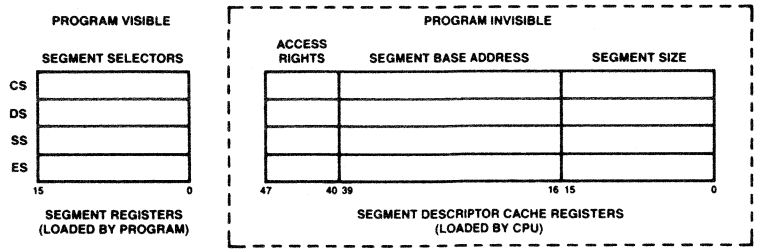


Figure 20. Descriptor Cache Registers

Local and Global Descriptor Tables

Two tables of descriptors, called descriptor tables, contain all descriptors accessible by a task at any given time. A descriptor table is a linear array of up to 8192 descriptors. The upper 13 bits of the selector value are an index into a descriptor table. Each table has a 24-bit base register to locate the descriptor table in physical memory and a 16-bit limit register that confines descriptor access to the defined limits of the table as shown in Figure 21. A restartable exception (13) will occur if an attempt is made to reference a descriptor outside the table limits.

One table, called the Global Descriptor Table (GDT), contains descriptors available to all tasks. The other table, called the Local Descriptor Table (LDT), contains descriptors that can be private to a task. Each task may have its own private LDT. The GDT may contain all descriptor types except interrupt and trap descriptors. The LDT may contain only segment, task gate, and call gate descriptors. A segment cannot be accessed by a task if its segment descriptor does not exist in either descriptor table at the time of access.

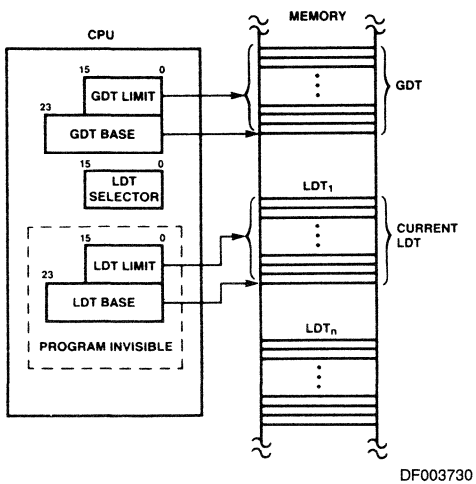


Figure 21. Local and Global Descriptor Table Definition

The LGDT and LLDT instructions load the base and limit of the global and local descriptor tables. LGDT and LLDT are protected. They may only be executed by trusted programs operating at level 0. The LGDT instruction loads a six byte field containing the 16-bit table limit and 24-bit base address of the Global Descriptor Table as shown in Figure 22. The LLDT instruction loads a selector which refers to a descriptor in the Local Descriptor Table. This descriptor contains the base address and limit for an LDT, as shown in Figure 17.

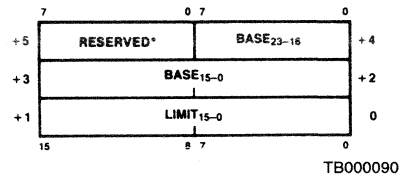


Figure 22. Global Descriptor Table and Interrupt Descriptor Data Type

*Must be set to 0 for compatibility with iAPX 386.

Interrupt Descriptor Table

The protected mode 80286 has a third descriptor table, called the Interrupt Descriptor Table (IDT) (see Figure 23), used to define up to 256 interrupts. It may contain only task gates, interrupt gates and trap gates. The IDT (Interrupt Descriptor Table) has a 24-bit base and 16-bit limit register in the CPU. The protected LIDT instruction loads these registers with a six-byte value of identical form to that of the LGDT instruction (see Figure 22 and Protected Mode Initialization).

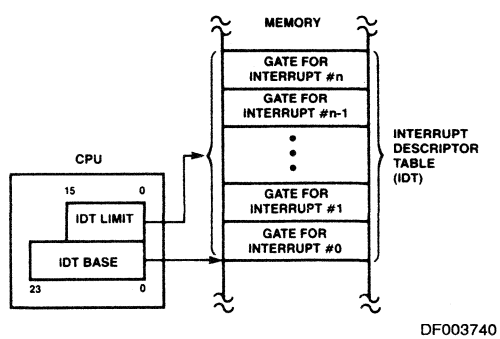


Figure 23. Interrupt Descriptor Table Definition

References to IDT entries are made via INT instructions, external interrupt vectors, or exceptions. The IDT must be at least 256 bytes in size to allocate space for all reserved interrupts.

Privilege

The 80286 has a four-level hierarchical privilege system which controls the use of privileged instructions and access to descriptors (and their associated segments) within a task. Four-level privilege, as shown in Figure 24, is an extension of the user/supervisor mode commonly found in minicomputers. The privilege levels are numbered 0 through 3. Level 0 is the most privileged level. Privilege levels provide protection within a task. (Tasks are isolated by providing private LDT's for each task.) Operating system routines, interrupt handlers, and other system software can be included and protected within the virtual address space of each task using the four levels of privilege. Tasks may also have a separate stack for each privilege level.

Tasks, descriptors, and selectors have a privilege level attribute that determines whether the descriptor may be used. Task privilege effects the use of instructions and descriptors. Descriptor and selector privilege only effect access to the descriptor.

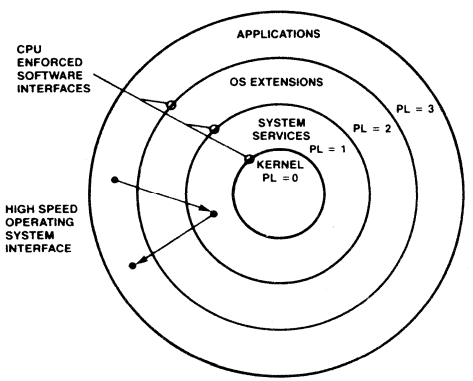


Figure 24. Hierarchical Privilege Levels

Task Privilege

The task always executes at one of the four privilege levels. A task privilege level at any specific instant is called the Current Privilege Level (CPL) and is defined by the lower two bits of the CS register. CPL cannot change during execution in a single code segment. A task's CPL may only be changed by control transfers through gate descriptors to a new code segment (See Control Transfer). Tasks begin executing at the CPL value specified by the code segment when the task is initiated via a task switch operation. A task executing at Level 0 can access all data segments defined in the GDT and the task's LDT and is considered the most trusted level. A task executed at Level 3 has the most restricted access to data and is considered the least trusted level.

Descriptor Privilege

Descriptor privilege is specified by the Descriptor Privilege Level (DPL) field of the descriptor access byte. DPL specifies the least trusted privilege level (CPL) at which a task may access the descriptor. Descriptors with DPL = 0 are the most protected. Only tasks executing at privilege level 0 (CPL = 0) may access them. Descriptors with DPL = 3 are the least protected (i.e. have the least restricted access) since tasks can access them when CPL = 0, 1, 2, or 3. This rule applies to all descriptors, except LDT descriptors.

Selector Privilege

Selector privilege is specified by the Requested Privilege Level (RPL) field in the least significant two bits of a selector. Selector RPL may establish a less trusted privilege level than the current privilege level for the use of a selector. This level is called the task's effective privilege level (EPL). RPL can only reduce the scope of a task's access to data with this selector. A task's effective privilege is the numeric maximum of RPL and CPL. A selector with RPL = 0 imposes no additional restriction on its use while a selector with RPL = 3 can only refer to segments at privilege Level 3 regardless of the task's CPL. RPL is generally used to verify that pointer parameters passed to a more trusted procedure are not allowed to use data at a more privileged level than the caller (refer to pointer testing instructions).

Descriptor Access and Privilege Validation

Determining the ability of a task to access a segment involves the type of segment to be accessed, the instruction used, the type of descriptor used and CPL, RPL, and DPL. The two basic types of segment accesses are control transfer (selectors loaded into CS) and data (selectors loaded into DS, ES or SS).

Data Segment Access

Instructions that load selectors into DS and ES must refer to a data segment descriptor or readable code segment descriptor. The CPL of the task and the RPL of the selector must be the same as or more privileged (numerically equal to or lower than) than the descriptor DPL. In general, a task can only access data segments at the same or less privileged levels than the CPL or RPL (whichever is numerically higher) to prevent a program from accessing data it cannot be trusted to use.

An exception to the rule is a readable conforming code segment. This type of code segment can be read from any privilege level.

If the privilege checks fail (e.g. DPL is numerically less than the maximum of CPL and RPL) or an incorrect type of descriptor is referenced (e.g. gate descriptor or execute only code segment), exception 13 occurs. If the segment is not present, exception 11 is generated.

Instructions that load selectors into SS must refer to data segment descriptors for writable data segments. The descriptor privilege (DPL) and RPL must equal CPL. All other descriptor types or privilege level violation will cause exception 13. A not present fault causes exception 12.

Control Transfer

Four types of control transfer can occur when a selector is loaded into CS by a control transfer operation (see Table 10). Each transfer type can only occur if the operation which loaded the selector references the correct descriptor type. Any violation of these descriptor usage rules (e.g. JMP through a call gate or RET to a Task State Segment) will cause exception 13.

The ability to reference a descriptor for control transfer is also subject to rules of privilege. A CALL or JUMP instruction may only reference a code segment descriptor with DPL equal to the task CPL or a conforming segment with DPL of equal or greater privilege than CPL. The RPL of the selector used to reference the code descriptor must have as much privilege as CPL.

RET and IRET instructions may only reference code segment descriptors with descriptor privilege equal to or less privileged than the task CPL. The selector loaded into CS is the return address from the stack. After the return, the selector RPL is

the task's new CPL. If CPL changes, the old stack pointer is popped after the return address.

When a JMP or CALL references a Task State Segment descriptor, the descriptor DPL must be the same or less privileged than the task's CPL. Reference to a valid Task State Segment descriptor causes a task switch (see Task Switch Operation). Reference to a Task State Segment descriptor at a more privileged level than the task's CPL generates exception 13.

When an instruction or interrupt references a gate descriptor, the gate DPL must have the same or less privilege than the task CPL. If DPL is at a more privileged level than CPL, exception 13 occurs. If the destination selector contained in the gate references a code segment descriptor, the code segment descriptor DPL must be the same or more privileged than the task CPL. If not, Exception 13 is issued. After the control transfer, the code segment descriptor DPL is the task's new CPL. If the destination selector in the gate references a task state segment, a task switch is automatically performed (see Task Switch Operation).

The privilege rules on control transfer require:

– JMP or CALL direct to a code segment (code segment descriptor) can only be to a conforming segment with DPL of equal or greater privilege than CPL or a non-conforming segment at the same privilege level.

– interrupts within the task or calls that may change privilege levels can only transfer control through a gate at the same or a less privileged level than CPL to a code segment at the same or more privileged level than CPL.

– return instructions that don't switch tasks can only return control to a code segment at the same or less privileged level.

– task switch can be performed by a call, a jump or an interrupt which references either a task gate or task state segment at the same or less privileged level.

Table 9. Descriptor Types Used for Control Transfer

Control Transfer Types	Operation Types	Descriptor Referenced	Descriptor Table
Intersegment within the same privilege level	JMP, CALL, RET, IRET*	Code Segment	GDT/LDT
Intersegment to the same or higher privilege level Interrupt within task may change CPL.	CALL	Call Gate	GDT/LDT
	Interrupt Instruction, Exception, External Interrupt	Trap or Interrupt Gate	IDT
Intersegment to a lower privilege level (changes task CPL)	RET, IRET*	Code Segment	GDT/LDT
Task Switch	CALL, JMP	Task State Segment	GDT
	CALL, JMP	Task Gate	GDT/LDT
	IRET** Interrupt Instruction, Exception, External Interrupt	Task Gate	IDT

*NT (Nested Task bit of flag word) = 0
**NT (Nested Task bit of flag word) = 1

Privilege Level Changes

Any control transfer that changes CPL within the task causes a change of stacks as part of the operation. Initial values of SS:SP for privilege levels 0, 1, and 2 are kept in the task state segment (refer to Task Switch Operation). During a JMP or CALL control transfer, the new stack pointer is loaded into the SS and SP registers and the previous stack pointer is pushed onto the new stack.

When returning to the original privilege level, its stack is restored as part of the RET or IRET instruction operation. For subroutine calls that pass parameters on the stack and cross privilege levels, a fixed number of words, as specified in the gate, are copied from the previous stack to the current stack. The intersegment RET instruction with a stack adjustment value will correctly restore the previous stack pointer upon return.

Protection

The 80286 includes mechanisms to protect critical instructions that affect the CPU execution state (e.g. HLT) and code or data segments from improper usage. These mechanisms are grouped under the term "protection" and have three forms:

Restricted usage of segments (e.g. no write allowed to read-only data segments). The only segments available for use are defined by descriptors in the Local Descriptor Table (LDT) and Global Descriptor Table (GDT).

Restricted access to segments via the rules of privilege and descriptor usage.

Privileged instructions or operations that may only be executed at certain privilege levels as determined by the CPL and I/O Privilege Level (IOPL). The IOPL is defined by bits 14 and 13 of the flag word.

These checks are performed for all instructions and can be split into three categories: segment load checks (Table 10), operand reference checks (Table 11), and privileged instruction checks (Table 12). Any violation of the rules shown will result in an exception. A not-present exception related to the stack segment causes exception 12.

The IRET and POPF instructions do not perform some of their defined functions if CPL is not of sufficient privilege (numerically small enough). Precisely these are:

The IF bit is not changed if CPL > IOPL.
The IOPL field of the flag word is not changed if CPL > 0.

No exceptions or other indication are given when these conditions occur.

Table 10. Segment Register Load Checks

Error Description	Exception Number
Descriptor table limit exceeded	13
Segment descriptor not-present	11 or 12
Privilege rules violated	13
Invalid descriptor/segment type segment register load: —Read only data segment load to SS —Special control descriptor load to DS, ES, SS —Execute only segment load to DS, ES, SS —Data segment load to CS —Read/Execute code segment load to SS	13

Table 11. Operand Reference Checks

Error Description	Exception Number
Write into code segment	13
Read from execute-only code segment	13
Write to read-only data segment	13
Segment limit exceeded[1]	12 or 13

Note 1: Carry out in offset calculations is ignored.

Table 12. Privileged Instruction Checks

Error Description	Exception Number
CPL ≠ 0 when executing the following instructions: LIDT, LLDT, LGDT, LTR, LMSW, CTS, HLT	13
CPL > IOPL when executing the following instructions: INS, IN, OUTS, OUT, STI, CLI, LOCK	13

Exceptions

The 80286 detects several types of exceptions and interrupts in protected mode (see Table 13). Most are restartable after the exceptional condition is removed. Interrupt handlers for most exceptions receive an error code, pushed on the stack after the return address, that identifies the selector involved (0 if none). The return address normally points to the failing instruction, including all leading prefixes. For a processor extension segment overrun exception, the return address will not point at the ESC instruction that caused the exception; however, the processor extension registers may contain the address of the failing instruction.

Table 13. Protected Mode Exceptions

Interrupt Vector	Function	Return Address At Failing Instruction?	Always Restartable?	Error Code on Stack?
8	Double exception detected	Yes	No[2]	Yes
9	Processor extension segment overrun	No	No[2]	No
10	Invalid task state segment	Yes	Yes	Yes
11	Segment not present	Yes	Yes	Yes
12	Stack segment overrun or segment not present	Yes	Yes[1]	Yes
13	General protection	Yes	No[2]	Yes

Notes 1. When a PUSHA or POPA instruction attempts to wrap around the stack segment, the machine state after the exception will not be restartable because stack segment wrap around is not permitted. This condition is identified by the value of the saved SP being either 0000(H), 0001(H), FFFE(H), or FFFF(H).
2. These exceptions indicate a violation to privilege rules or usage rules has occurred. Restart is generally not attempted under those conditions.
3. All these checks are performed for all instructions and can be split into three categories: Segment Load Checks (Table 10), Operand Reference Checks (Table 11), and Privileged Instruction Checks (Table 12). Any violation of the rules shown will result in an exception. A not-present exception causes exception 11 or 12 and is restartable.

Special Operations

Task Switch Operation

The 80286 provides a built-in task switch operation which saves the entire 80286 execution state (registers, address space, and a link to the previous task), loads a new execution state, and commences execution in the new task. Like gates, the task switch operation is invoked by executing an inter-segment JMP or CALL instruction which refers to a Task State Segment (TSS) or task gate descriptor in the GDT or LDT. An INT n instruction, exception, or external interrupt may also invoke the task switch operation by selecting a task gate descriptor in the associated IDT descriptor entry.

The TSS descriptor points at a segment (see Figure 25) containing the entire 80286 execution state while a task gate descriptor contains a TSS selector. The limit field must be > 002B(H).

Each task must have a TSS associated with it. The current TSS is identified by a special register in the 80286 called the Task Register (TR). This register contains a selector referring to the task state segment descriptor that defines the current TSS. A hidden base and limit register associated with TR are loaded whenever TR is loaded with a new selector.

The IRET instruction is used to return control to the task that called the current task or was interrupted. Bit 14 in the flag register is called the Nested Task (NT) bit. It controls the function of the IRET instruction. If NT = O, the IRET instruction performs the regular current task return; when NT = 1, IRET performs a task switch operation back to the previous task.

When a CALL or INT instruction initiates a task switch, the old and new TSS will be marked busy and the back link field of the new TSS set to the old TSS selector. The NT bit of the new task is set by CALL or INT initiated task switches. An interrupt that does not cause a task switch will clear NT. NT may also be set or cleared by POPF or IRET instructions.

The task state segment is marked busy by changing the descriptor type field from Type 1 to Type 3. Use of a selector

that references a busy task state segment causes Exception 13.

Processor Extension Context Switching

The context of a processor extension is not changed by the task switch operation. A processor extension context need only be changed when a different task attempts to use the processor extension (which still contains the context of a previous task). The 80286 detects the first use of a processor extension after a task switch by causing the processor extension not present exception (7). The interrupt handler may then decide whether a context change is necessary.

Whenever the 80286 switches tasks, it sets the Task Switched (TS) bit of the MSW. TS indicates that a processor extension context may belong to a different task than the current one. The processor extension not present exception (7) will occur when attempting to execute an ESC or WAIT instruction if TS = 1 and a processor extension is present (MP = 1 in MSW).

Pointer Testing Instructions

The 80286 provides several instructions to speed pointer testing and consistency checks for maintaining system integrity (see Table 14). These instructions use the memory management hardware to verify that a selector value refers to an appropriate segment without risking an exception. A condition flag indicates whether use of the selector or segment will cause an exception.

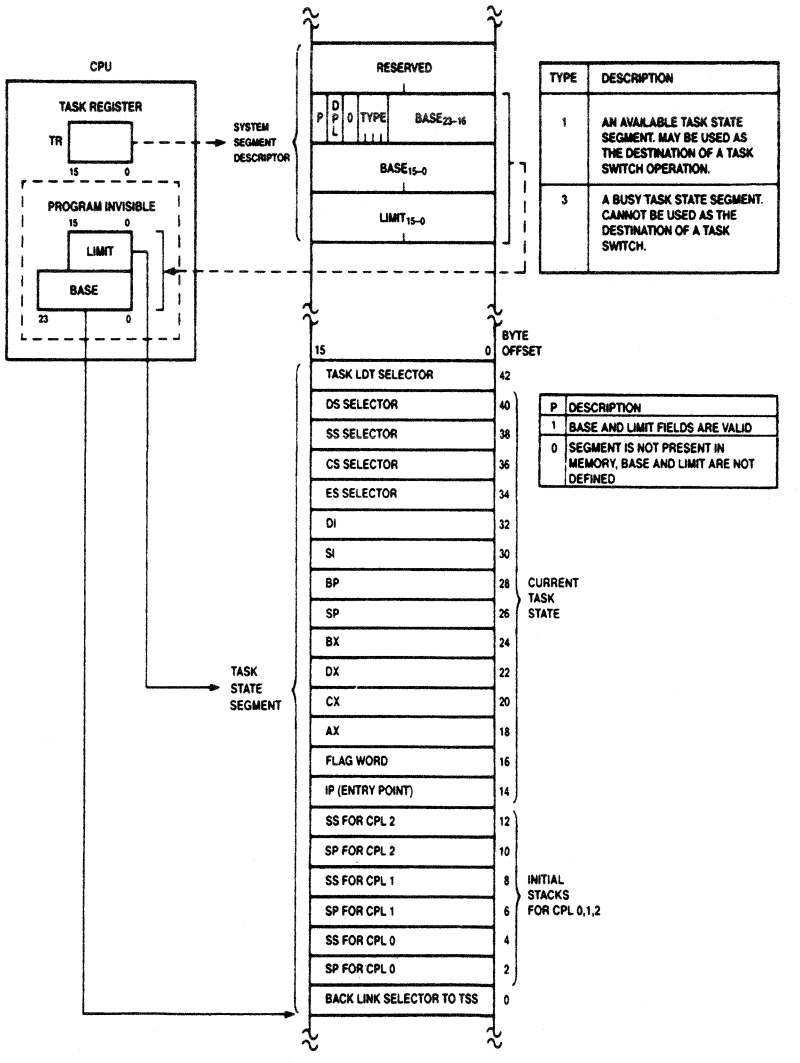


Figure 25. Task State Segment and TSS Registers

Table 14. Pointer Test Instructions

Instruction	Operands	Function
ARPL	Selector, Register	Adjust Requested Privilege Level: adjusts the RPL of the selector to the numeric maximum of current selector RPL value and the RPL value in the register. Set zero flag if selector RPL was changed.
VERR	Selector	VERify for Read: sets the zero flag if the segment referred to by the selector can be read.
VERW	Selector	VERify for Write: sets the zero flag if the segment referred to by the selector can be written.
LSL	Register, Selector	Load Segment Limit: reads the segment limit into the register if privilege rules and descriptor type allow. Set zero flag if successful.
LAR	Register, Selector	Load Access Rights: reads the descriptor access rights byte into the register if privilege rules allow. Set zero flag if successful.

Double Fault and Shutdown

If two separate exceptions are detected during a single instruction execution, the 80286 performs the double fault exception (8). If an exception occurs during processing of the double fault exception, the 80286 will enter shutdown. During shutdown no further instructions or exceptions are processed. Either NMI (CPU remains in protected mode) or RESET (CPU exits protected mode) can force the 80286 out of shutdown. Shutdown is externally signalled via a HALT bus operation with A_1 HIGH.

Protected Mode Initialization

The 80286 initially executes in real address mode after RESET. To allow initialization code to be placed at the top of physical memory, A_{23-20} will be HIGH when the 80286 performs memory references relative to the CS register, until CS is changed. A_{23-20} will be zero for references to the DS, ES, or SS segments. Changing CS in real address mode will force $A_{23} - A_{20}$ LOW whenever using CS thereafter. The initial CS:IP value of FFOO:FFFO provides 64K bytes of code space for initialization code without changing CS.

Before placing the 80286 into protected mode, several registers must be initialized. The GDT and IDT base registers must refer to a valid GDT and IDT. After executing the LMSW instruction to set PE, the 80286 must immediately execute an intrasegment JMP instruction to clear the instruction queue of instructions decoded in real address mode.

To force the 80286 CPU registers to match the initial protected mode state assumed by software, execute a JMP instruction with a selector referring to the initial TSS used in the system. This will load the task register, local descriptor table register, segment registers and initial general register state. The TR should point at a valid TSS since a task switch operation involves saving the current task state.

System Interface

The 80286 system interface appears in two forms: a local bus and a system bus. The local bus consists of address, data, status, and control signals at the pins of the CPU. A system bus is any buffered version of the local bus. A system bus may also differ from the local bus in terms of coding of status and control lines and/or timing and loading of signals. The 80286 family includes several devices to generate standard system buses such as the IEEE 796 Standard Multibus™.

Bus Interface Signals and Timing

The 80286 microsystem local bus interfaces the 80286 to local memory and I/O components. The interface has 24 address lines, 16 data lines, and 8 status and control signals.

The 80286 CPU, 82284 clock generator, 82C288 bus controller, 82289 bus arbiter, 8286/7 transceivers, and 8282/3 latches provide a buffered and decoded system bus interface. The 82284 generates the system clock and synchronizes $\overline{\text{READY}}$ and RESET. The 82C288 converts bus operation status encoded by the 80286 into command and bus control signals. These components can provide the timing and electrical power drive levels required for most system bus interfaces including the multibus.

Physical Memory and I/O Interface

A maximum of 16 megabytes of physical memory can be addressed in protected mode. One megabyte can be addressed in real address mode. Memory is accessible as bytes or words. Words consist of any two consecutive bytes addressed with the least significant byte stored in the lowest address.

Byte transfers occur on either half of the 16-bit local data bus. Even bytes are accessed over D_{7-0} while odd bytes are transferred over D_{15-8}. Even-addressed words are transferred over D_{15-0} in one bus cycle, while odd-addressed words require two bus operations. The first transfers data on D_{15-8}, and the second transfers data on D_{7-0}. Both byte data transfers occur automatically, transparent to software.

Two bus signals, A_0 and $\overline{\text{BHE}}$, control transfers over the lower and upper halves of the data bus. Even address byte transfers are indicated by A_0 LOW and $\overline{\text{BHE}}$ HIGH. Odd address byte transfers are indicated by A_0 HIGH and $\overline{\text{BHE}}$ LOW. Both A_0 and $\overline{\text{BHE}}$ are LOW for even address word transfers.

The I/O address space contains 64K addresses in both modes. The I/O space is accessible as either bytes or words, as is memory. Byte wide peripheral devices may be attached to either the upper or lower byte of the data bus. Byte-wide I/O devices attached to the upper data byte (D_{15-8}) are accessed with odd I/O addresses. Devices on the lower data byte are accessed with even I/O addresses. An interrupt controller such as the 8259A must be connected to the lower data byte (D_{7-0}) for proper return of the interrupt vector.

Bus Operation

The 80286 uses a double frequency system clock (CLK input) to control bus timing. All signals on the local bus are measured relative to the system CLK input. The CPU divides the system clock by 2 to produce the internal processor clock, which determines bus state. Each processor clock is composed of two system clock cycles named phase 1 and phase 2. The 82284 clock generator output (PCLK) identifies the next phase of the processor clock. (See Figure 26.)

Multibus is a registered trademark of Intel Corporation.

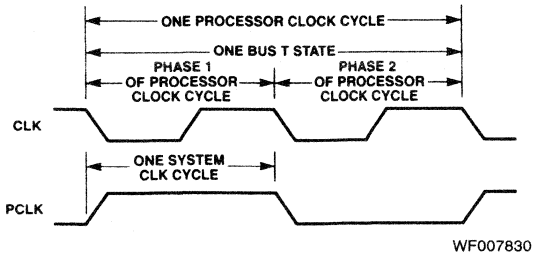


Figure 26. System and Processor Clock Relationships

Six types of bus operations are supported: memory read, memory write, I/O read, I/O write, interrupt acknowledge, and halt/shutdown. Data can be transferred at a maximum rate of one word per two processor clock cycles.

The 80286 bus has three basic states: idle (T_i), send status (T_s), and perform command (T_c). The 80286 CPU also has a fourth local bus state called hold (T_h). T_h indicates that the 80286 has surrendered control of the local bus to another bus master in response to a HOLD request.

Each bus state is one processor clock long. Figure 27 shows the four 80286 local bus states and allowed transitions.

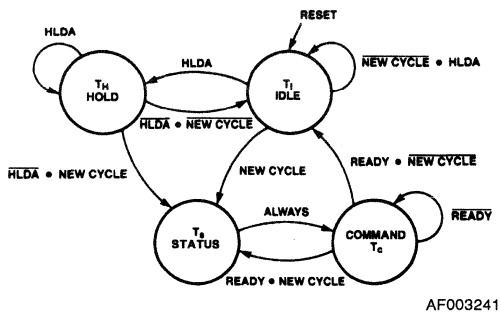


Figure 27. 80286 Bus States

Bus States

The idle (T_i) state indicates that no data transfers are in progress or requested. The first active state, T_s, is signalled by either status line $\overline{S1}$ or $\overline{S0}$ going LOW also identifying phase 1 of the processor clock. During T_s, the command encoding, the address, and data (for a write operation) are available on the 80286 output pins. The 82C288 bus controller decodes the status signals and generates Multibus compatible read/write command and local transceiver control signals.

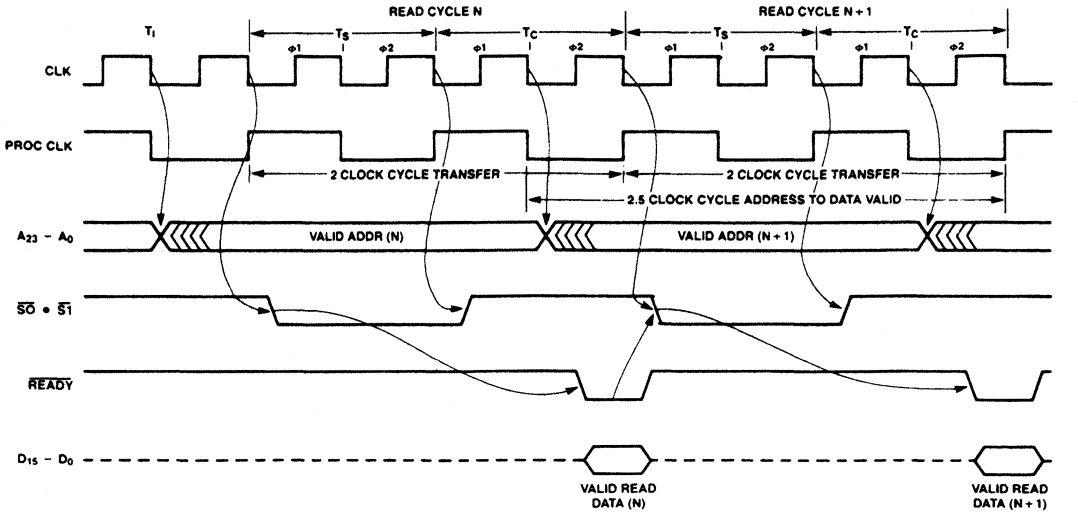
After T_s, the perform command (T_c) state is entered. Memory or I/O devices respond to the bus operation during T_c, either transferring read data to the CPU or accepting write data. T_c states may be repeated as often as necessary to assure sufficient time for the memory or I/O device to respond. The $\overline{READY}$ signal determines whether T_c is repeated. A repeated T_c state is called a wait state.

During hold (T_h), the 80286 will float all address, data, and status output pins, enabling another bus master to use the local bus. The 80286 HOLD input signal is used to place the 80286 into the T_h state. The 80286 HLDA output signal indicates that the CPU has entered T_h.

Pipelined Addressing

The 80286 uses a local bus interface with pipelined timing to allow as much time as possible for data access. Pipelined timing allows bus operations to be performed in two processor cycles, while allowing each individual bus operation to last for three processor cycles.

The timing of the address outputs is pipelined such that the address of the next bus operation becomes available during the current bus operation. Or in other words, the first clock of the next bus operation is overlapped with the last clock of the current bus operation. Therefore, address decode and routing logic can operate in advance of the next bus operation. External address latches may hold the address stable for the entire bus operation and provide additional AC and DC buffering.



WF007840

Pipelining: valid address (N + 1) available in last phase of bus cycle (N).

Figure 28. Basic Bus Cycle

The 80286 does not maintain the address of the current bus operation during all T_C states. Instead, the address for the next bus operation may be emitted during phase 2 of any T_C. The address remains valid during phase 1 of the first T_C to guarantee hold time, relative to ALE, for the address latch inputs.

Bus Control Signals

The 82C288 bus controller provides control signals: address latch enable (ALE), Read/Write commands, data transmit/receive (DT/$\overline{R}$), and data enable (DEN) that control the address latches, data transceivers, write enable, and output enable for memory and I/O systems.

The Address Latch Enable (ALE) output determines when the address may be latched. ALE provides at least one system CLK period of address hold time from the end of the previous bus operation until the address for the next bus operation appears at the latch outputs. This address hold time is required to support Multibus® and common memory systems.

The data bus transceivers are controlled by 82C288 outputs Data Enable (DEN) and Data Transmit/Receive (DT/$\overline{R}$). DEN enables the data transceivers while DT/$\overline{R}$ controls transceiver direction. DEN and DT/$\overline{R}$ are timed to prevent bus contention between the bus master, data bus transceivers, and system data bus transceivers.

Command Timing Controls

Two system timing customization options, command extension and command delay, are provided on the 80286 local bus.

Command extension allows additional time for external devices to respond to a command and is analogous to inserting wait states on the 8086. External logic can control the duration of any bus operation such that the operation is only as long as necessary. The $\overline{READY}$ input signal can extend any bus operation for as long as necessary.

Command delay allows an increase of address or write data set-up time to system bus command active for any bus operation by delaying when the system bus command becomes active. Command delay is controlled by the 82C288 CMDLY input. After T_S, the bus controller samples CMDLY at each failing edge of CLK. If CMDLY is HIGH, the 82C288 will not activate the command signal. When CMDLY is LOW, the 82C288 will activate the command signal. After the command becomes active, the CMDLY input is not sampled.

When a command is delayed, the available response time from command active to return read data or accept write data is less. To customize system bus timing, an address decoder can determine which bus operations require delaying the command. The CMDLY input does not affect the timing of ALE, DEN, or DT/$\overline{R}$.

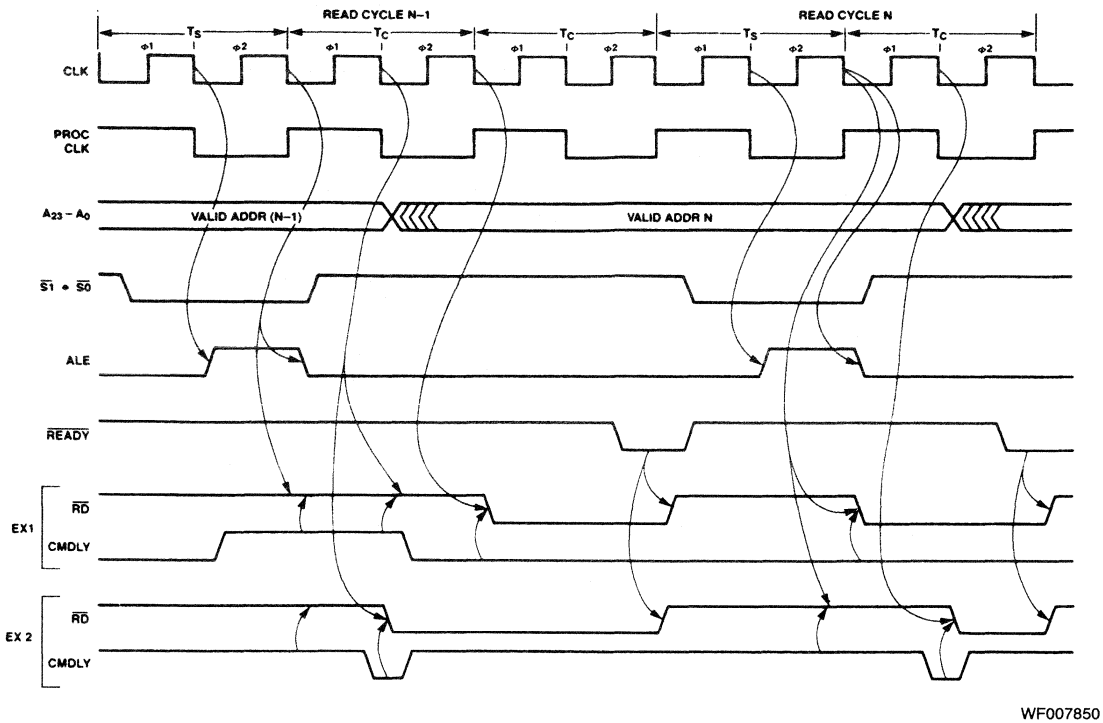


Figure 29. CMDLY Controls and Leading Edge of the Command

Figure 29 illustrates four uses of CMDLY. Example 1 shows delaying the read command two system CLKs for cycle N-1 and no delay for cycle N, and example 2 shows delaying the read command one system CLK for cycle N-1 and one system CLK delay for cycle N.

Bus Cycle Termination

At maximum transfer rates, the 80286 bus alternates between the status and command states. The bus status signals become inactive after T_S so that they may correctly signal the start of the next bus operation after the completion of the current cycle. No external indication of T_C exists on the 80286 local bus. The bus master and bus controller enter T_C directly after T_S and continue executing T_C cycles until terminated by $\overline{READY}$.

$\overline{READY}$ Operation

The current bus master and 82C288 bus controller terminate each bus operation simultaneously to achieve maximum bus bandwidth. Both are informed in advance by $\overline{READY}$ active which identifies the last T_C cycle of the current bus operation. The bus master and bus controller must see the same sense of the $\overline{READY}$ signal, thereby requiring $\overline{READY}$ be synchronous to the system clock.

Synchronous Ready

The 82284 clock generator provides $\overline{READY}$ synchronization from both synchronous and asynchronous sources (see Figure 30). The synchronous ready input ($\overline{SRDY}$) of the clock generator is sampled with the falling edge of CLK at the end of phase 1 of each T_C. The state of $\overline{SRDY}$ is then broadcast to the bus master and bus controller via the $\overline{READY}$ output line.

Asynchronous Ready

Many systems have devices or subsystems that are asynchronous to the system clock. As a result, their ready outputs cannot be guaranteed to meet the 82284 $\overline{SRDY}$ set-up and hold time requirements. The 82284 asynchronous ready input ($\overline{ARDY}$) is designed to accept such signals. The $\overline{ARDY}$ input is sampled at the beginning of each T_C cycle by 82284 synchronization logic. This provides a system CLK cycle time to resolve its value before broadcasting it to the bus master and bus controller.

$\overline{ARDY}$ or $\overline{ARDYEN}$ must be HIGH at the end of T_S. $\overline{ARDY}$ cannot be used to terminate bus cycle with no wait status.

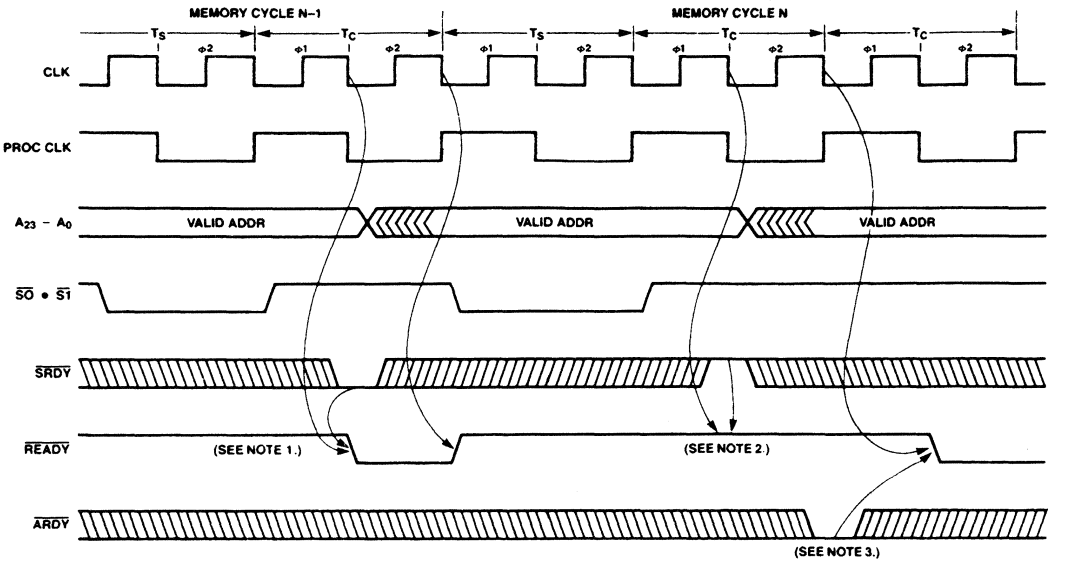


Figure 30. Synchronous and Asynchronous Ready

Notes: 1. $\overline{\text{SRDYEN}}$ is active LOW.
2. If $\overline{\text{SRDYEN}}$ is HIGH, the state of $\overline{\text{SRDY}}$ will not effect $\overline{\text{READY}}$.
3. $\overline{\text{ARDYEN}}$ is active LOW.

Each ready input of the 82284 has an enable pin ($\overline{\text{SRDYEN}}$ and $\overline{\text{ARDYEN}}$) to select whether the current bus operation will be terminated by the synchronous or asynchronous ready. Either of the ready inputs may terminate a bus operation. These enable inputs are active low and have the same timing as their respective ready inputs. Address decode logic usually selects whether the current bus operation should be terminated by $\overline{\text{ARDY}}$ or $\overline{\text{SRDY}}$.

Data Bus Control

Figures 31, 32, and 33 show how the DT/$\overline{\text{R}}$, DEN, data bus, and address signals operate for different combinations of read, write, and idle bus operations. DT/$\overline{\text{R}}$ goes active (LOW) for a read operation. DT/$\overline{\text{R}}$ remains HIGH before, during, and between write operations.

The data bus is driven with write data during the second phase of T_S. The delay in write data timing allows the read data drivers, from a previous read cycle, sufficient time to enter three-state OFF before the 80286 CPU begins driving the local data bus for write operations. Write data will always remain valid for one system clock past the last T_C to provide sufficient hold time for Multibus or other similar memory or I/O systems. During write-read or write-idle sequences the data bus enters three-state OFF during the second phase of the processor cycle after the last T_C. In a write-write sequence the data bus does not enter three-state OFF between T_C and T_S.

Bus Usage

The 80286 local bus may be used for several functions: instruction data transfers, data transfers by other bus masters, instruction fetching, processor extension data transfers, interrupt acknowledge, and halt/shutdown. This section describes local bus activities which have special signals or requirements.

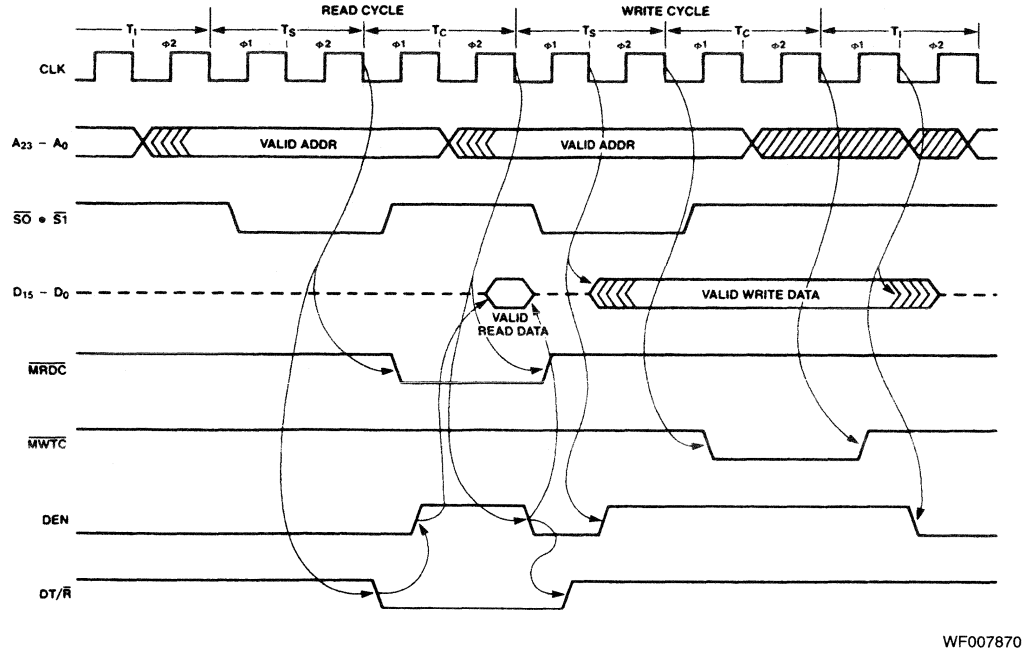


Figure 31. Back-to-Back Read-Write Cycles

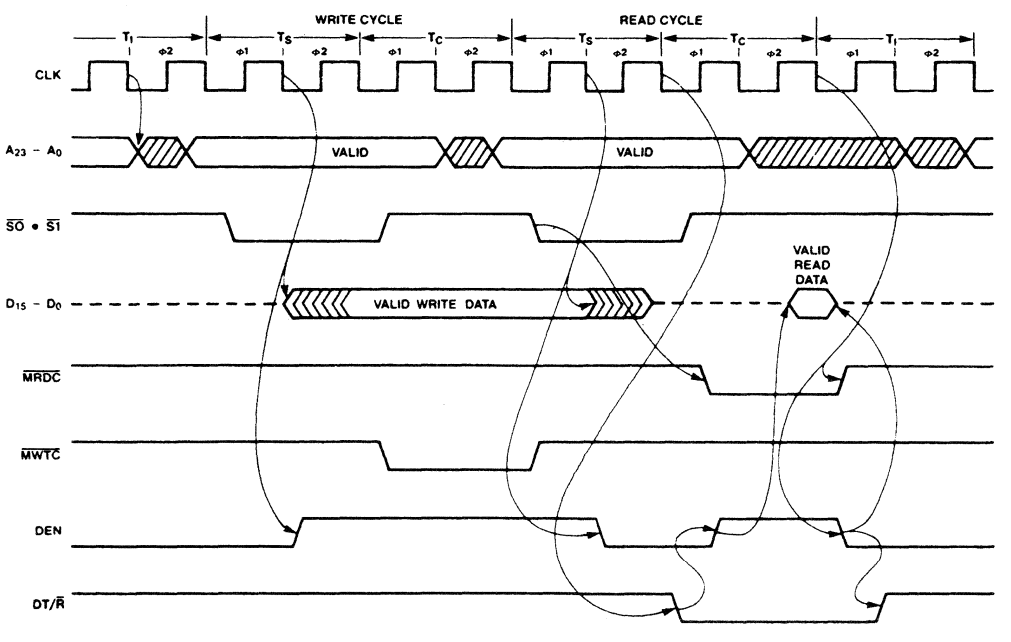
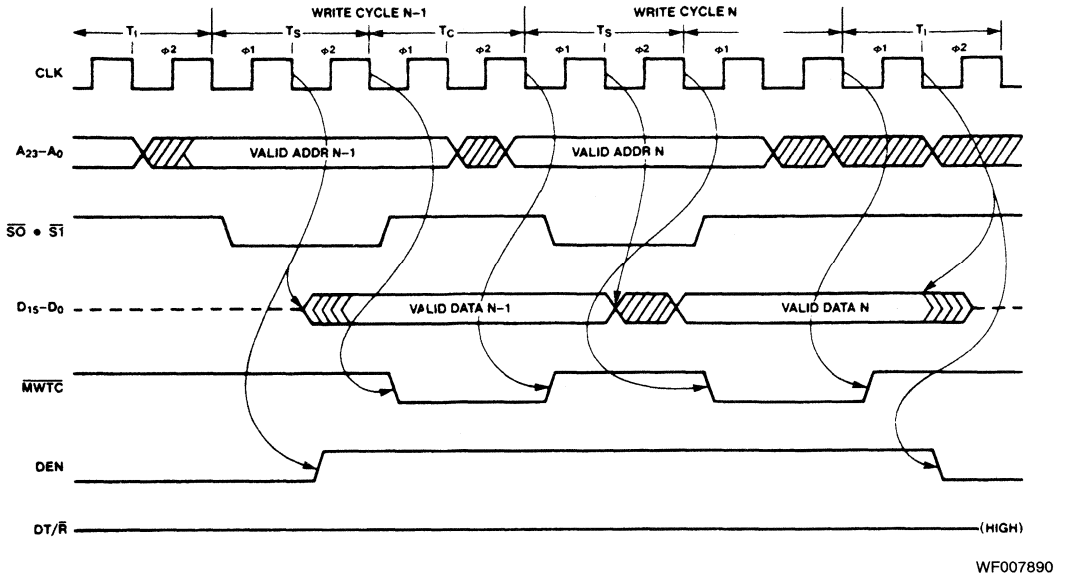


Figure 32. Back-to-Back Write-Read Cycles



WF007890

Figure 33. Back-to-Back Write-Write Cycles

HOLD and HLDA

HOLD and HLDA allow another bus master to gain control of the local bus by placing the 80286 bus into the T_h state. The sequence of events required to pass control between the 80286 and another local bus master are shown in Figure 34.

In this example, the 80286 is initially in the T_h, state as signaled by HLDA being active. Upon leaving T_h, as signaled by HLDA going inactive, a write operation is started. During the write operation another local bus master requests the local bus from the 80286 as shown by the HOLD signal. After completing the write operation, the 80286 performs one T_i bus cycle, to guarantee write data hold time, then enters T_h as signaled by HLDA going active.

The CMDLY signal and $\overline{ARDY}$ ready are used to start and stop the write bus command, respectively. Note that $\overline{SRDY}$ must be inactive or disabled by $\overline{SRDYEN}$ to guarantee $\overline{ARDY}$ will terminate the cycle.

HOLD must not be active during the time from the leading edge of RESET until 34 CLKs following the trailing edge of RESET unless the 80286 is in the Halt condition. To ensure that the 80286 remains in the Halt condition until the processor Reset operation is complete, no interrupts should occur after the execution of HLT until 34 CLKs after the trailing edge of the RESET pulse.

Lock

The CPU asserts an active lock signal during Interrupt-Acknowledge cycles, the XCHG instruction, and during some descriptor accesses. Lock is also asserted when the LOCK prefix is used. The LOCK prefix may be used with the following ASM-286 assembly instructions; MOVS, INS, and OUTS. For bus cycles other than Interrupt-Acknowledge cycles, Lock will be active for the first and subsequent cycles of a series of cycles to be locked. Lock will not be shown active during the last cycle to be locked. For the next-to-last cycle, Lock will become inactive at the end of the first T_C regardless of the number of wait-states inserted. For Interrupt-Acknowledge cycles, Lock will be active for each cycle, and will become inactive at the end of the first T_C for each cycle regardless of the number of wait-states inserted.

Instruction Fetching

The 80286 Bus Unit (BU) will fetch instructions ahead of the current instruction being executed. This activity is called prefetching. It occurs when the local bus would otherwise be idle and obeys the following rules:

A prefetch bus operation starts when at least two bytes of the 6-byte prefetch queue are empty.

The prefetcher normally performs word prefetches independent of the byte alignment of the code segment base in physical memory.

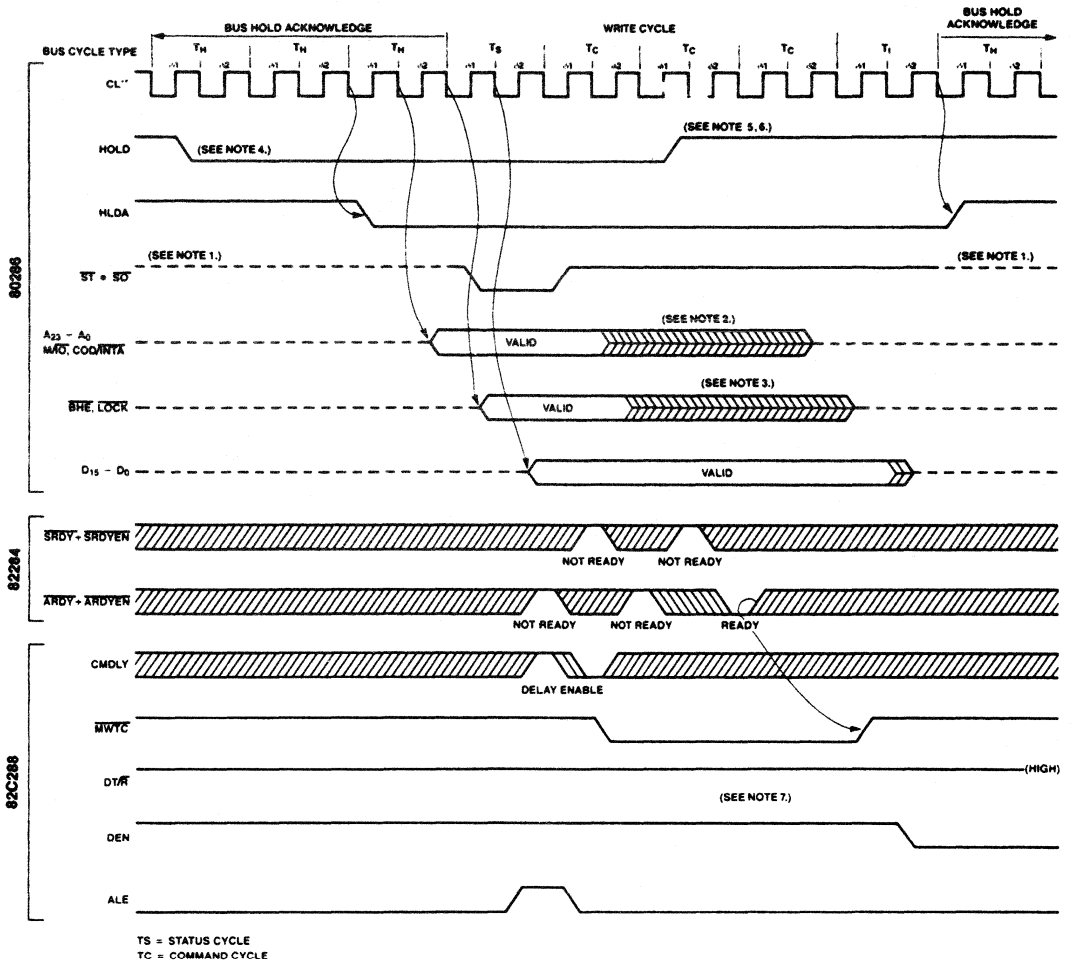
The prefetcher will perform only a byte code fetch operation for control transfers to an instruction beginning on a numerically odd physical address.

Prefetching stops whenever a control transfer or HLT instruction is decoded by the IU and placed into the instruction queue.

In real address mode, the prefetcher may fetch up to 5 bytes beyond the last control transfer or HLT instruction in a code segment.

In protected mode, the prefetcher will never cause a segment overrun exception. The prefetcher stops at the last physical memory word of the code segment. Exception 13 will occur if the program attempts to execute beyond the last full instruction in the code segment.

If the last byte of a code segment appears on an even physical memory address, the prefetcher will read the next physical byte of memory (perform a word code fetch). The value of this byte is ignored and any attempt to execute it causes exception 13.



WF007901

Figure 34. Multibus Write Terminated by Asynchronous Ready with Bus Hold

Notes:
1. Status lines are not driven by 80286, yet remain high due to pull-up resistors in 82C288 and 82289 during HOLD state.
2. Address, M/$\overline{IO}$ and COD/$\overline{INTA}$ may start floating during any TC depending on when internal 80286 bus arbiter decides to release bus to external HOLD. The float starts in $\phi 2$ of TC.
3. $\overline{BHE}$ and $\overline{LOCK}$ may start floating after the end of any TC depending on when internal 80286 bus arbiter decides to release bus to external HOLD.
4. The minimum HOLD ↓ to HLDA ↓ time is shown. Max.imum is one T_H longer.
5. The earliest HOLD ↑ time is shown which will always allow a subsequent memory cycle if pending.
6. The minimum HOLD ↑ to HLDA ↑ time is shown. Max.imum is a function of the instruction, type of bus cycle and other machine status (i.e., Interrupts, Waits, Lock, etc.)
7. Asynchronous ready allows termination of the cycle. Synchronous ready does not signal ready in this example. Synchronous ready state is ignored after ready is signaled via the asynchronous input.

Processor Extension Transfers

The processor extension interface uses I/O port addresses 00F8(H), 00FA(H), and 00FC(H) which are part of the I/O port address range and is a reserved area. An ESC instruction with EM = 0 and TS = 0 will perform I/O bus operations to one or more of these I/O port addresses independent of the value of IOPL and CPL.

ESC instructions with memory references enable the CPU to accept PEREQ inputs for processor extension operand transfers. The CPU will determine the operand starting address and read/write status of the instruction. For each operand transfer, two or three bus operations, one word transfer with I/O port address 00FA(H), and one or two bus operations with memory are performed. Three bus operations are required for each word operand aligned on an odd byte address.

Interrupt Acknowledge Sequence

Figure 35 illustrates an interrupt acknowledge sequence performed by the 80286 in response to an INTR input. An interrupt acknowledge sequence consists of two INTA bus operations. The first allows a master 8259A Programmable Interrupt Controller (PIC) to determine which if any of its slaves should return the interrupt vector. An eight bit vector is read by the 80286 during the second INTA bus operation to select an interrupt handler routine from the interrupt table.

The Master Cascade Enable (MCE) signal of the 82C288 is used to enable the cascade address drivers, during INTA bus operations (see Figure 35), onto the local address bus for distribution to slave interrupt controllers via the system address bus. The 80286 emits the $\overline{LOCK}$ signal (active LOW) during T_S of the first INTA bus operation. A local bus "hold" request will not be honored until the end of the second INTA bus operation.

Three idle processor clocks are provided by the 80286 between INTA bus operations to allow for the minimum INTA to INTA time and CAS (cascade address) out delay of the 8259A. The second INTA bus operation must always have at least one extra T_C state added via logic controlling $\overline{READY}$. $A_{23} - A_0$ are in three-state OFF until after the first T_C state of the second INTA bus operation. This prevents bus contention between the cascade address drivers and CPU address drivers. The extra T_C state allows time for the 80286 to resume driving the address lines for subsequent bus operations.

Local Bus Usage Priorities

The 80286 local bus is shared among several internal units and external HOLD requests. In case of simultaneous requests, their relative priorities are:

(Highest) Any transfers which assert $\overline{LOCK}$ either explicitly (via the LOCK instruction prefix) or implicitly (i.e., segment descriptor access, interrupt acknowledge sequence, or an XCHG with memory).

The second of the two byte bus operations required for an odd aligned word operand.

Local bus request via HOLD input.

Processor extension data operand transfer via PEREQ input.

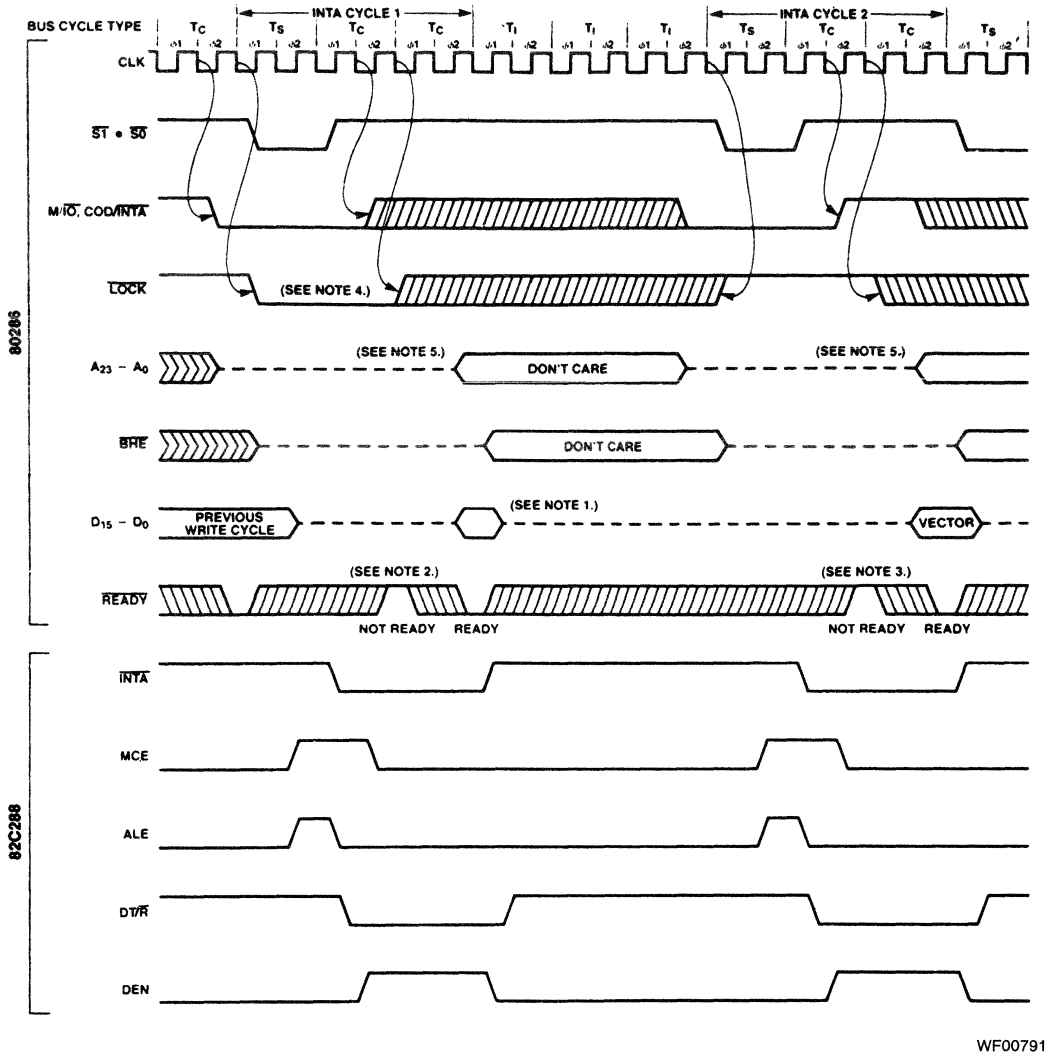
Data transfer performed by EU as part of an instruction.

(Lowest) An instruction prefetch request from BU. The EU will inhibit prefetching two processor clocks in advance of any data transfers to minimize waiting by EU for a prefetch to finish.

Halt or Shutdown Cycles

The 80286 externally indicates halt or shutdown conditions as a bus operation. These conditions occur due to a HLT instruction or multiple protection exceptions while attempting to execute one instruction. A halt or shutdown bus operation is signalled when $\overline{S1}$, $\overline{S0}$ and COD/$\overline{INTA}$ are LOW and M/$\overline{IO}$ is HIGH. A_1 HIGH indicates halt, and A_1 LOW indicates shutdown. The 82C288 bus controller does not issue ALE, nor is $\overline{READY}$ required to terminate a halt or shutdown bus operation.

During halt or shutdown, the 80286 may service PEREQ or HOLD requests. A processor extension segment overrun exception during shutdown will inhibit further service of PEREQ. Either NMI or RESET will force the 80286 out of either halt or shutdown. An INTR, if interrupts are enabled, or a processor extension segment overrun exception will also force the 80286 out of halt.



WF007911

Figure 35. Interrupt Acknowledge Sequence

Notes:
1. Data is ignored.
2. First INTA cycle should have at least one wait state inserted to meet 8259A minimum INTA pulse width.
3. Second INTA cycle must have at least one wait state inserted since the CPU will not drive $A_{23} - A_0$, $\overline{BHE}$, and $\overline{LOCK}$ until after the first TC state.
   The CPU imposed one/clock delay prevents bus contention between cascade address buffer being disabled by MCE ↓ and address outputs.
   Without the wait state, the 80286 address will not be valid for a memory cycle started immediately after the second INTA cycle. The 8259A also requires one wait state for minimum INTA pulse width.
4. $\overline{LOCK}$ is active for the first INTA cycle to prevent the 82289 from releasing the bus between INTA cycles in a multi-master system.
5. $A_{23} - A_0$ exits three-state OFF during $\phi 2$ of the second T_C in the INTA cycle.

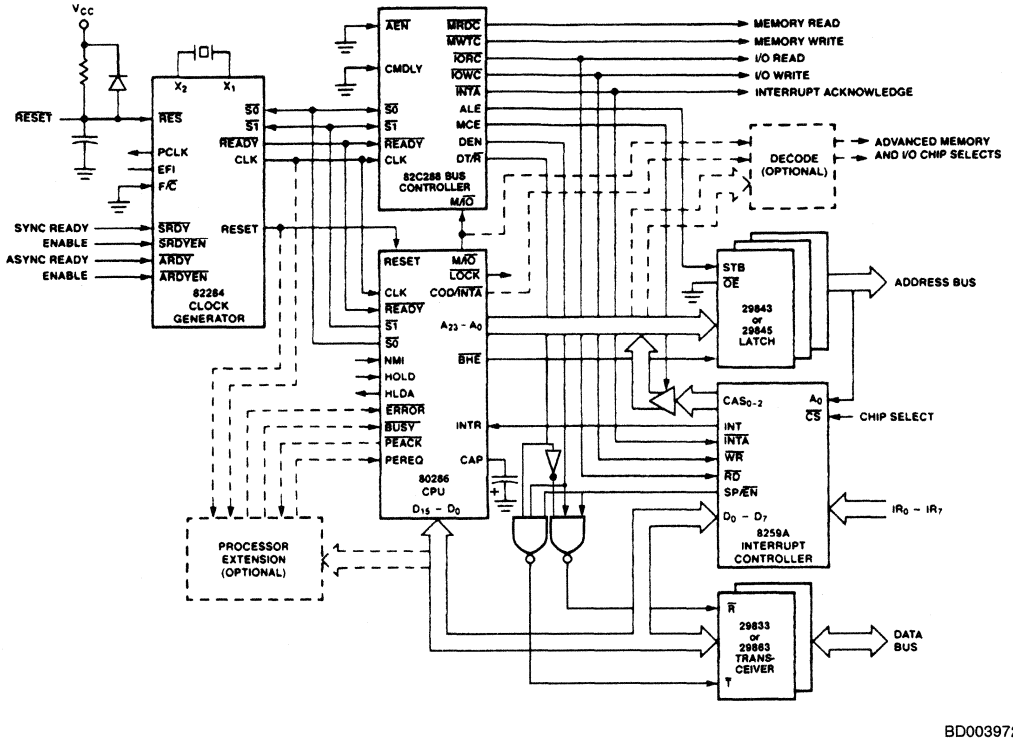


Figure 36. Basic 80286 System Configuration

System Configurations

The versatile bus structure of the 80286 microsystem, with a full complement of support chips, allows flexible configuration of a wide range of systems. The basic configuration, shown in Figure 36, is similar to an iAPX 86 maximum mode system. It includes the CPU plus an 8259A interrupt controller, 82284 clock generator, and the 82C288 Bus Controller. The iAPX 86 latches (29843 and 29845) and transceivers (29833 and 29863) may be used in an 80286 microsystem.

As indicated by the dashed lines in Figure 36, the ability to add processor extensions is an integral feature of 80286 microsystems. The processor extension interface allows external hardware to perform special functions and transfer data concurrent with CPU execution of other instructions. Full system integrity is maintained because the 80286 supervises all data transfers and instruction execution for the processor extension.

The 80286 with the 80287 numeric processor extension (NPX) uses this interface. The iAPX 286/287 has all the instructions and data types of an iAPX 86/87 or iAPX 88/87. The 80287 NPX can perform numeric calculations and data transfers concurrently with CPU program execution. Numerics code and data have the same integrity as all other information protected by the 80286 protection mechanism.

The 80286 can overlap chip select decoding and address propagation during the data transfer for the previous bus operation. This information is latched into the 29843/45's by ALE during the middle of a T_S cycle. The latched chip select and address information remains stable during the bus operation while the next cycle's address is being decoded and propagated into the system. Decode logic can be implemented with a high-speed bipolar PROM.

The optional decode logic shown in Figure 36 takes advantage of the overlap between address and data of the 80286 bus cycle to generate advanced memory and I/O-select signals. This minimizes system performance degradation caused by address propagation and decode delays. In addition to selecting memory and I/O, the advanced selects may be used with configurations supporting local and system buses to enable the appropriate bus interface for each bus cycle. The COD/$\overline{\text{INTA}}$ and M/$\overline{\text{IO}}$ signals are applied to the decode logic to distinguish between interrupt, I/O, code and data bus cycles.

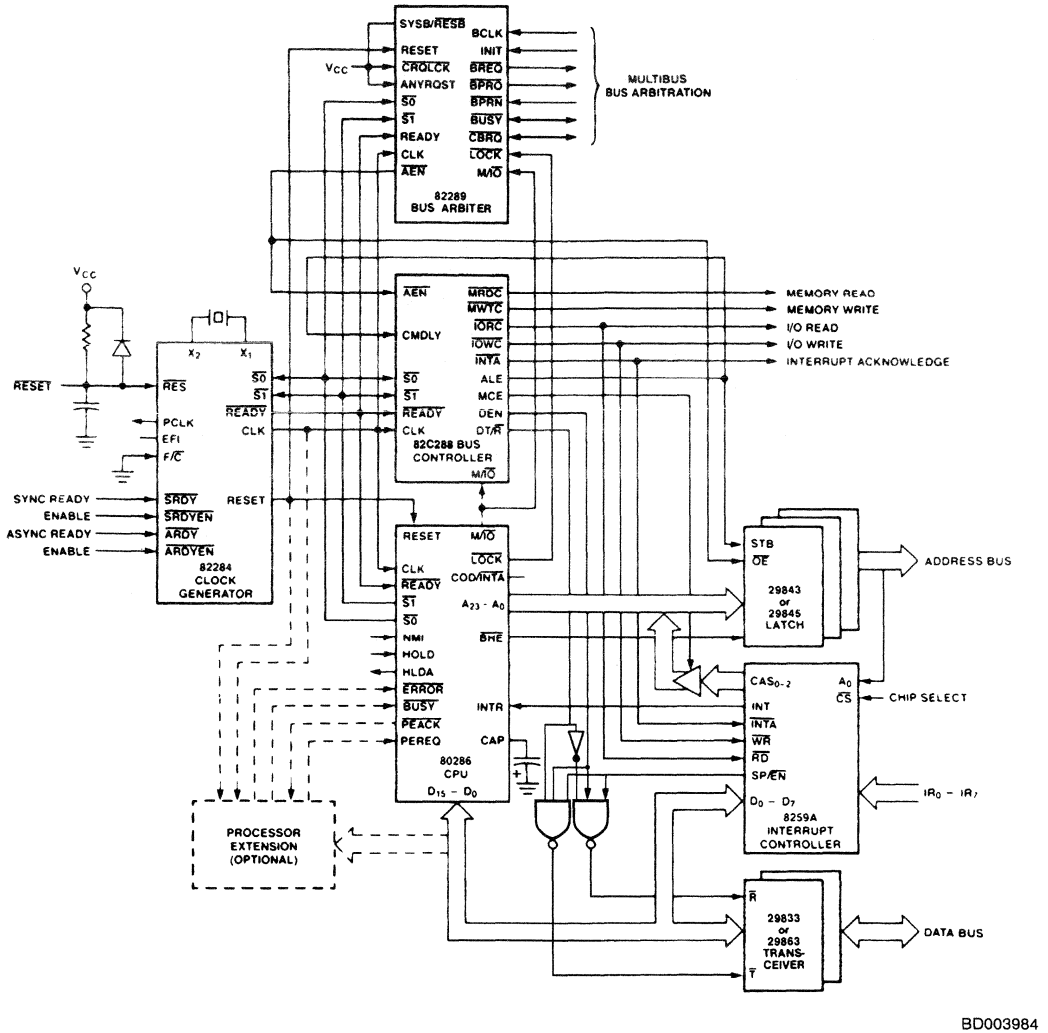


Figure 37. Multibus System Bus Interface

By adding the 82289 bus arbiter chip, the 80286 provides a Multibus system bus interface as shown in Figure 37. The ALE output of the 82C288 for the Multibus bus is connected to its CMDLY input to delay the start of commands one system CLK as required to meet Multibus address and write data set-up times. This arrangement will add at least one extra T_C state to each bus operation which uses the Multibus.

A second 82C288 bus controller and additional latches and transceivers could be added to the local bus of Figure 37. This configuration allows the 80286 to support an on-board bus for local memory and peripherals and the Multibus for system bus interfacing.

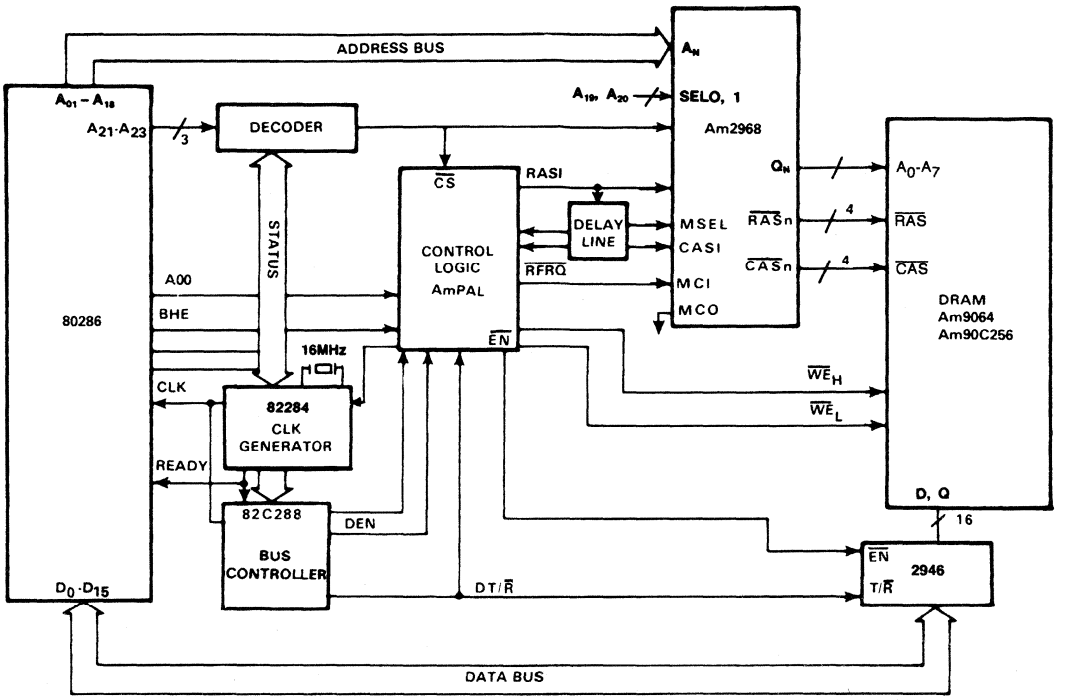


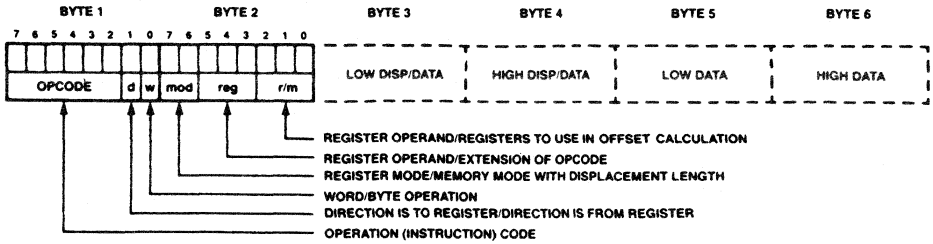
Figure 38. 80286 Interface with the Am2968 Dynamic Memory Controller

Figure 38 shows the interface of the 80286 with the Am2968 Dynamic Memory Controller. The interface is a timing controller which consists of some control logic and a delay line. The timing controller runs asynchronously to the CPU. It arbitrates between memory requests and refresh requests by generating the proper signals to the dynamic memory controller and memory. The design described is a simple, cost-effective solution to interfacing the 80286 with the Am2968. A further description about DRAM selection based on processor speed may be found in the Am2968 Application Note.

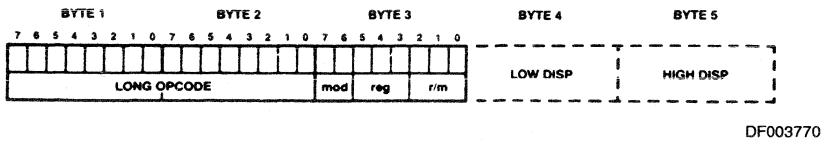
Table 15. 80286 Systems Recommended Pull-up Resistor Values

80286 Pin and Name	Pull-up Value	Purpose
4 — $\overline{S1}$	20KΩ±10%	Pull $\overline{S0}$, $\overline{S1}$, and $\overline{PEACK}$ inactive during 80286 hold periods.
5 — $\overline{S0}$		
6 — $\overline{PEACK}$		
53 — $\overline{ERROR}$	20KΩ±10%	Pull $\overline{ERROR}$ and $\overline{BUSY}$ inactive when 80287 not present (or temporarily removed from socket).
54 — $\overline{BUSY}$		
63 — $\overline{READY}$	910Ω±5%	Pull $\overline{READY}$ inactive within required minimum time (C_L = 150pF, $I_R \leqslant$ 7mA).

Two-operand instructions (e.g. MOV and ADD) are usually three to six bytes long. Memory-to-memory operations are provided by a special class of string instructions requiring one to three bytes.



A. SHORT OPCODE FORMAT EXAMPLE



B. LONG OPCODE FORMAT EXAMPLE

Figure 39. 80286 Instruction Format Examples

80286 INSTRUCTION SET SUMMARY

Instruction Timing Notes

The instruction clock counts listed below establish the maximum execution rate of the 80286. With no delays in bus cycles, the actual clock count of an 80286 program will average 5% more than the calculated clock count, due to instruction sequences which execute faster than they can be fetched from memory.

To calculate elapsed times for instruction sequences, multiply the sum of all instruction clock counts, as listed in the table below, by the processor clock period. An 8 MHz processor clock has a clock period of 125 nanoseconds and requires an 80286 system clock (CLK input) of 16 MHz.

Instruction Clock Count Assumptions

1. The instruction has been prefetched, decoded, and is ready for execution. Control transfer instruction clock counts include all time required to fetch, decode, and prepare the next instruction for execution.
2. Bus cycles do not require wait states.
3. There are no processor extension data transfer or local bus HOLD requests.
4. No exceptions occur during instruction execution.

Instruction Set Summary Notes

Addressing displacements selected by the MOD field are not shown. If necessary they appear after the instruction fields shown.

Above/below refers to unsigned value
Greater refers to positive signed value
Less refers to less positive (more negative) signed values

if d = 1 then to register; if d = 0 then from register

if w = 1 then word instruction; if w = 0 then byte instruction

if s = 0 then 16-bit immediate data to form the operand

if s = 1 then an immediate data byte is sign-extended to form the 16-bit operand

x = don't care

z = used for string primitives for comparison with ZF FLAG

If two clock counts are given, the smaller refers to a register operand and the larger refers to a memory operand.

* = add one clock if offset calculation requires summing 3 elements

n = number of times repeated

m = number of bytes of code in next instruction

Level (L)—Lexical nesting level of the procedure

The following comments describe possible exceptions, side effects, and allowed usage for instructions in both operating modes of the 80286.

Real Address Mode Only

1. This is a protected mode instruction. Attempted execution in real address mode will result in an undefined opcode exception (6).
2. A segment overrun exception (13) will occur if a word operand reference at offset FFFF(H) is attempted.
3. This instruction may be executed in real address mode to initialize the CPU for protected mode.
4. The IOPL and NT fields will remain 0.
5. Processor extension segment overrun interrupt (9) will occur if the operand exceeds the segment limit.

Either Mode

6. An exception may occur, depending on the value of the operand.

7. $\overline{\text{LOCK}}$ is automatically asserted regardless of the presence or absence of the LOCK instruction prefix.

8. $\overline{\text{LOCK}}$ does not remain active between all operand transfers.

Protected Virtual Address Mode Only

9. A general protection exception (13) will occur if the memory operand cannot be used due to either a segment limit or access rights violation. If a stack segment limit is violated, a stack segment overrun exception (12) occurs.

10. For segment load operations, the CPL, RPL, and DPL must agree with privilege rules to avoid an exception. The segment must be present to avoid a not-present exception (11). If the SS register is the destination, and a segment not-present violation occurs, a stack exception (12) occurs.

11. All segment descriptor accesses in the GDT or LDT made by this instruction will automatically assert $\overline{\text{LOCK}}$ to maintain descriptor integrity in multiprocessor systems.

12. JMP, CALL, INT, RET, IRET instructions referring to another code segment will cause a general protection exception (13) if any privilege rule is violated.

13. A general protection exception (13) occurs if CPL $\neq 0$.

14. A general protection exception (13) occurs if CPL > IOPL.

15. The IF field of the flag word is not updated if CPL > IOPL. The IOPL field is updated only if CPL = 0.

16. Any violation of privilege rules as applied to the selector operand do not cause a protection exception; rather, the instruction does not return a result and the zero flag is cleared.

17. If the starting address of the memory operand violates a segment limit, or an invalid access is attempted, a general protection exception (13) will occur before the ESC instruction is executed. A stack segment overrun exception (12) will occur if the stack limit is violated by the operand's starting address. If a segment limit is violated during an attempted data transfer then a processor extension segment overrun exception (9) occurs.

18. The destination of an INT, JMP, CALL, RET, or IRET instruction must be in the defined limit of a code segment or a general protection exception (13) will occur.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin with
Respect to Ground..... −1.0 to +7.0 V
Power Dissipation ..... 3.3 Watts

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_C)..... 0 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ± 5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS (V_{CC} = 5 V ±5%, T_{CASE} = 0 to +85°C)

Parameters	Description	Test Conditions	Min.	Max.	Units
V_{IL}	Input LOW Voltage		−.5	.8	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC} + .5	V
V_{ILC}	CLK Input LOW Voltage		−.5	.6	V
V_{IHC}	CLK Input HIGH Voltage		3.8	V_{CC} + .5	V
V_{OL}	Output LOW Voltage	I_{OL} = 2.0 mA		.45	V
V_{OH}	Output HIGH Voltage	I_{OH} = −400 µA	2.4		V
I_{LI}	Input Leakage Current	$0 V \leq V_{IN} \leq V_{CC}$		±10	µA
I_{LO}	Output Leakage Current	$.45 V \leq V_{OUT} \leq V_{CC}$		±10	µA
I_{CC}	Supply Current (turn on, 0°C)	Note 1		600	mA
C_{CLK}	CLK Input Capacitance	F_C = 1 MHz		20	pF
C_{IN}	Other Input Capacitance	F_C = 1 MHz		10	pF
C_O	Input/Output Capacitance	F_C = 1 MHz		20	pF
I_{LO}	Output Leakage Current	$0 V \leq V_{out} < .045 V$		±1	mA
I_{IL}	Input Sustaining Current on $\overline{BUSY}$ and $\overline{ERROR}$ pins	V_{in} = 0 V	30	500	µA
I_{LCR}	Input CLK Leakage Current	$0.45 \leq V_{IN} \leq V_{CC}$		±10	µA
I_{LCR}	Input CLK Leakage Current	$0 V \leq V_{IN} \leq 0.45 V$		±1	mA

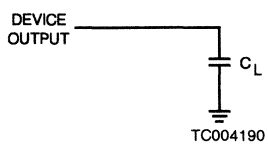
Notes: 1. Low temperature is worst case.

SWITCHING CHARACTERISTICS (V_{CC} = 5 V ± 5%, T_{CASE} = 0 to +85°C)

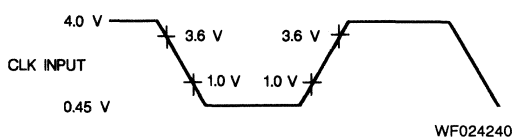
AC timings are referenced to 0.8 V and 2.0 V points of signals as illustrated in datasheet waveforms, unless otherwise noted.

Parameters	Description	Test Conditions	8 MHz		10 MHz		12.5 MHz		16 MHz		Units
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
1	System Clock (CLK) Period		62	250	50	250	40	250	31	250	ns
2	System Clock (CLK) LOW Time	at 1.0 V	15	225	12	232	11	237	10	239	ns
3	System Clock (CLK) HIGH Time	at 3.6 V	25	235	16	239	13	239	12	243	ns
17	System Clock (CLK) Rise Time	1.0 V to 3.6 V		10		8		8		5	ns
18	System Clock (CLK) Fall Time	3.6 V to 1.0 V		10		8		8		4	ns
4	Asynch. Inputs Set-up Time	Note 1	20		20		15		11		ns
5	Asynch. Inputs Hold Time	Note 1	20		20		15		11		ns
6	RESET Set-up Time		28		23		18		14		ns
7	RESET Hold Time		5		5		5		3		ns
8	Read Data Set-up Time		10		8		5		5		ns
9	Read Data Hold Time		8		8		6		5		ns
10	$\overline{\text{READY}}$ Set-up Time		38		26		22		15		ns
11	$\overline{\text{READY}}$ Hold Time		25		25		20		15		ns
12	Status/$\overline{\text{PEACK}}$ Valid Delay	Note 2, Note 3	1	40	-	-	-	-	-	-	ns
12a	Status/$\overline{\text{PEACK}}$ Active Delay	Note 2, Note 3	-	-	1	22	3	18	1	18	ns
12b	Status/$\overline{\text{PEACK}}$ Inactive Delay	Note 2, Note 3	-	-	1	30	3	20	1	20	ns
13	Address Valid Delay	Note 2, Note 3	1	60	1	35	1	32	1	29	ns
14	Write Data Valid Delay	Note 2, Note 3	0	50	0	30	0	30	0	22	ns
15	Address/Status/Data Float Delay	Note 2, Note 4	0	50	0	47	0	32	0	29	ns
16	HLDA Valid Delay	Note 2, Note 3	0	50	0	47	0	25	0	25	ns
19	Address Valid To Status Valid Setup Time	Note 3, Note 5, Note 6	38		27		22		22		ns

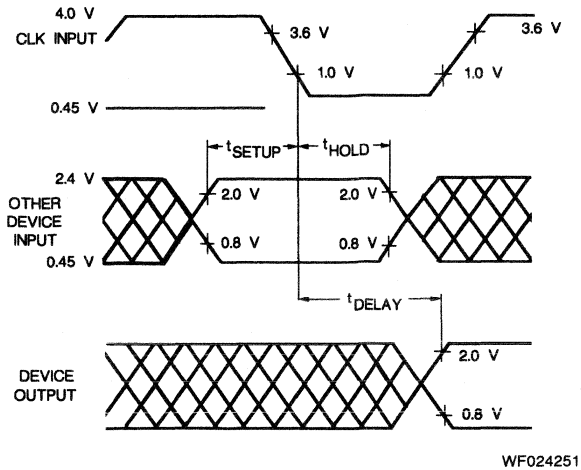
Notes:
1. Asynchronous inputs are INTR, NMI, HOLD PEREQ, $\overline{\text{ERROR}}$, and $\overline{\text{BUSY}}$. This specification is given only for testing purposes to assure recognition at a specific CLK edge.
2. Delay from 1.0 V on the CLK to 0.8 V or 2.0 V or float on the output as appropriate for valid or floating condition.
3. Output load: C_L = 100 pF.
4. Float condition occurs when output current is less than I_{LO} in magnitude.
5. Delay measured from address either reaching 0.8 V or 2.0 V (valid) to status going active reaching 2.0 V or status going inactive reaching 0.8 V.
6. For load capacitance of 10 pF on STATUS/$\overline{\text{PEACK}}$ lines, subtract typically 7 ns for 8 MHz spec, and maximum 7 ns for 10 MHz spec.



NOTE 7:
AC Test Loading on Outputs



NOTE 8:
AC Drive and Measurement Points — CLK Input



WF024251

NOTE 9:
AC Setup, Hold and Delay Time Measurement — General

82284 Timing Requirements

Parameters	Description	Test Conditions	82284-8		82284-10		Units
			Min.	Max.	Min.	Max.	
11	$\overline{SRDY}/\overline{SRDYEN}$ Set-up Time		17		15		ns
12	$\overline{SRDY}/\overline{SRDYEN}$ Hold Time		0		0		ns
13	$\overline{ARDY}/\overline{ARDYEN}$ Set-up Time	Note 1	0		0		ns
14	$\overline{ARDY}/\overline{ARDYEN}$ Hold Time	Note 1	30		30		ns
19	PCLK Delay	C_L = 75 pF I_{OL} = 5 mA I_{OH} = −1 mA	0	45	0	35	ns

Note 1. These times are given for testing purposes to assure a predetermined action.

82C288 Timing Requirements

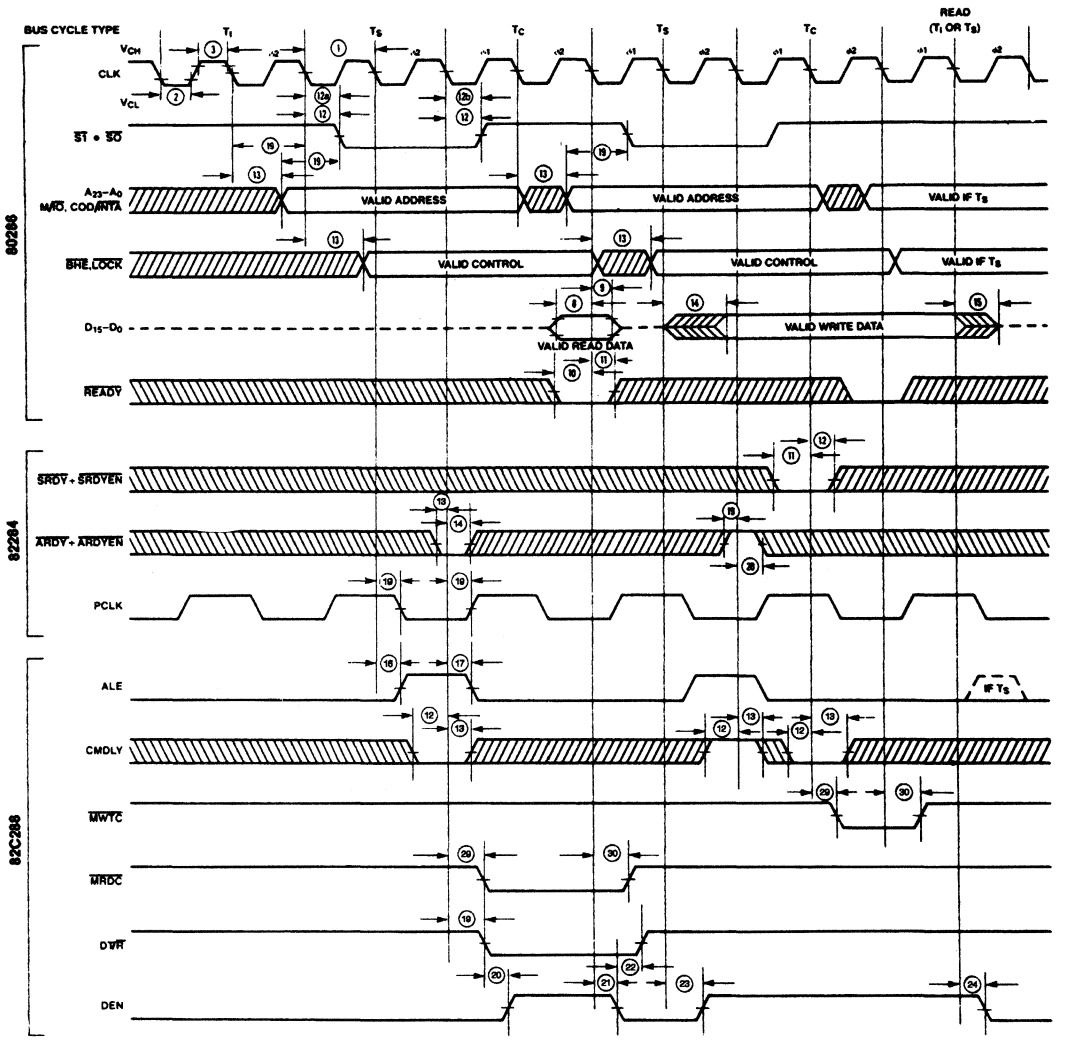
Parameters	Description		Test Conditions	82C288-8		82C288-10		Units
				Min.	Max.	Min.	Max.	
12	CMDLY Set-up Time			20		15		ns
13	CMDLT Hold Time			1		1		ns
30	Command Delay from CLK	Command Inactive	C_L = 300 pF Max. I_{OL} = 32 mA Max. I_{OH} = 5 mA Max.	5	25	5	20	ns
29		Command Active		3	25	3	21	
16	ALE Active Delay		C_L = 150 pF I_{OL} = 16 mA Max. I_{OH} = −1 mA Max.	3	20	3	16	ns
17	ALE Inactive Delay				25		19	ns
19	DT/$\overline{R}$ Read Active Delay				25		23	ns
22	DT/$\overline{R}$ Read Inactive Delay			5	35	5	20	ns
20	DEN Read Active Delay			5	35	5	21	ns
21	DEN Read Inactive Delay			3	35	3	21	ns
23	DEN Write Active Delay				30		23	ns
24	DEN Write Inactive Delay			3	30	3	19	ns

SWITCHING WAVEFORMS

MAJOR CYCLE TIMING

Read Cycle illustrated with zero wait states

Write Cycle illustrated with one wait state

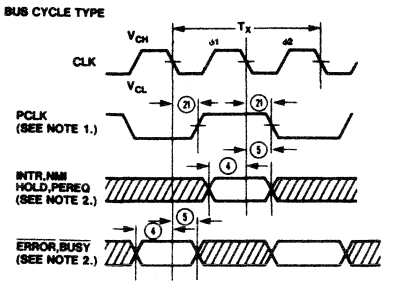


WF007982

Note: 1. The modified timing is due to the $\overline{\text{CMDLY}}$ signal being active.

SWITCHING WAVEFORMS (Cont'd.)

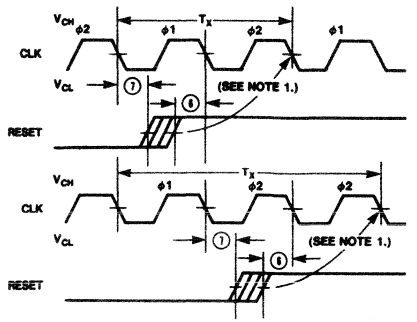
80286 ASYNCHRONOUS INPUT SIGNAL TIMING



WF009930

Notes:
1. PCLK indicates which processor cycle phase will occur on the next CLK. PCLK may not indicate the correct phase until the first bus cycle is performed.
2. These inputs are asynchronous. The set-up and hold times shown assure recognition for testing purposes.

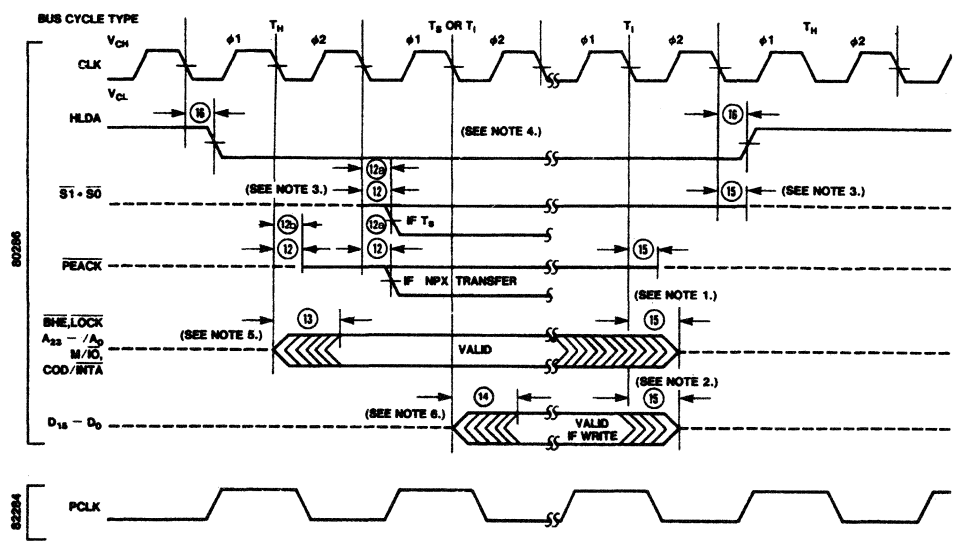
80286 RESET INPUT TIMING AND SUBSEQUENT PROCESSOR CYCLE PHASE



WF007930

Note 1: When RESET meets the set-up time shown, the next CLK will start or repeat $\phi 1$ of a processor cycle.

EXITING AND ENTERING HOLD

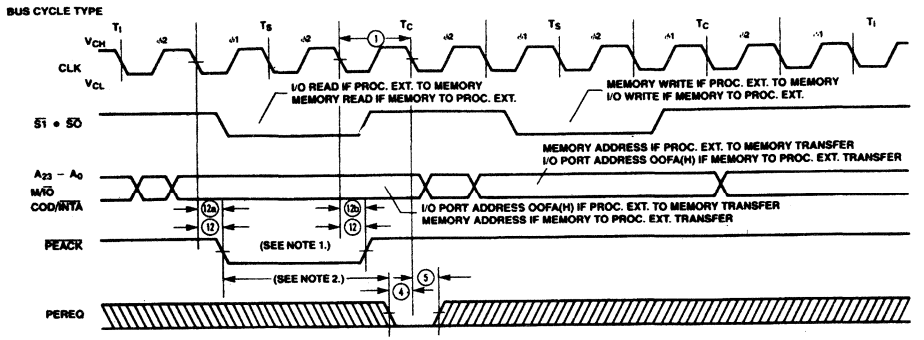


WF009942

Notes: 1. These signals may not be driven by the 80286 during the time shown. The worst case in terms of latest float time is shown.
2. The data bus will be driven as shown if the last cycle before T_I in the diagram was a write T_C.
3. The 80286 floats its status pins during T_H. External 20 kΩ resistors keep these signals high (see Table 15).
4. For HOLD request set-up to HLDA, refer to Figure 34.
5. $\overline{BHE}$ and $\overline{LOCK}$ are driven at this time but will not become valid until T_S.
6. The data bus will remain in three-state OFF if a read cycle is performed.

SWITCHING WAVEFORMS (Cont'd.)

80286 PEREQ/$\overline{\text{PEACK}}$ TIMING REQUIRED PEREQ TIMING FOR ONE TRANSFER ONLY



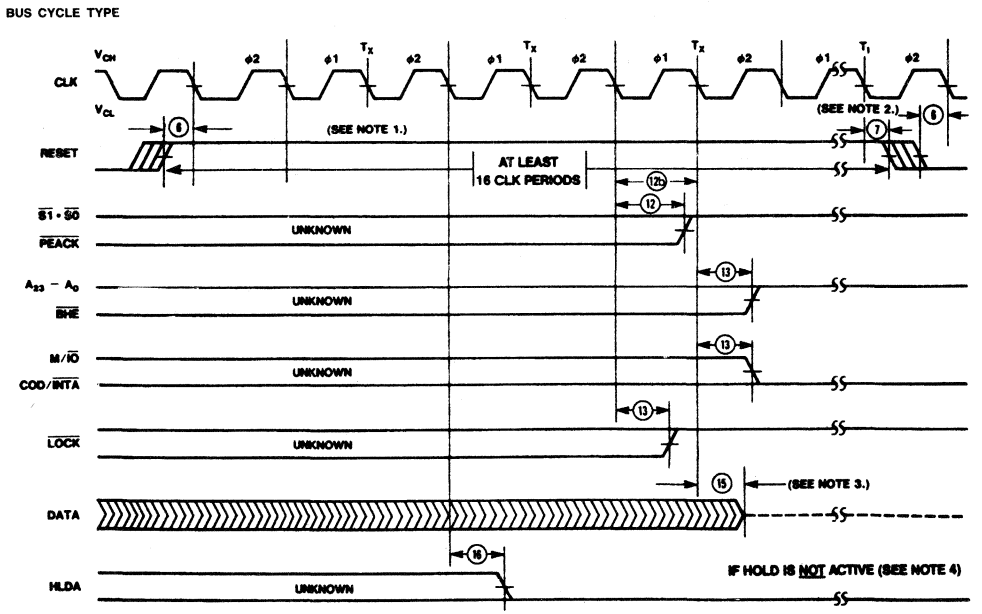
WF007953

ASSUMING WORD-ALIGNED MEMORY OPERAND; IF ODD ALIGNED, 80286 TRANSFERS TO/FROM MEMORY BYTE-AT-A-TIME WITH TWO MEMORY CYCLES.

Notes: 1. $\overline{\text{PEACK}}$ always goes active during the first bus operation of a processor extension data operand transfer sequence. The first bus operation will be either a memory read at operand address or I/O read at port address OOFA(H).

2. To prevent a second processor extension data operand transfer, the worst case maximum time (shown above) is: $3 \times ① - ⑪_{max} - ④_{min}$. The actual, configuration dependent, maximum time is: $3 \times ① - ⑪_{max} - ④_{min} + A \times 2 \times ①$. A is the number of extra T_C states added to either the first or second bus operation of the processor extension data operand transfer sequence.

INITIAL 80286 PIN STATE DURING RESET



WF007962

Notes: 1. Set-up time for RESET ↑ may be violated with the consideration that $\phi 1$ of the processor clock may begin one system CLK period later.

2. Set-up and hold times for RESET ↓ must be met for proper operation, but RESET ↓ may occur during $\phi 1$ or $\phi 2$.

3. The data bus is only guaranteed to be in three-state OFF at the time shown.

4. HOLD is acknowledged during RESET, causing HLDA to go active and the appropriate pins to float. If HOLD remains active while RESET goes inactive, the 80286 remains in HOLD state and will not perform any bus accesses until HOLD is deactivated.

80286 INSTRUCTION SET SUMMARY

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
DATA TRANSFER					
MOV = Move:					
Register to Register/Memory	1 0 0 0 1 0 0 w | mod reg r/m	2,3*	2,3*	2	9
Register/memory to register	1 0 0 0 1 0 1 w | mod reg r/m	2,5*	2,5*	2	9
Immediate to register/memory	1 1 0 0 0 1 1 w | mod 0 0 0 r/m | data | data if w = 1	2,3*	2,3*	2	9
Immediate to register	1 0 1 1 w reg | data | data if w = 1	2	2		
Memory to accumulator	1 0 1 0 0 0 0 w | addr-low | addr-high	5	5	2	9
Accumulator to memory	1 0 1 0 0 0 1 w | addr-low | addr-high	3	3	2	9
Register/memory to segment register	1 0 0 0 1 1 1 0 | mod 0 reg r/m	2,5*	17,19*	2	9,10,11
Segment register to register/memory	1 0 0 0 1 1 0 0 | mod 0 reg r/m	2,3*	2,3*	2	9
PUSH = Push:					
Memory	1 1 1 1 1 1 1 1 | mod 1 1 0 r/m	5*	5*	2	9
Register	0 1 0 1 0 reg	3	3	2	9
Segment register	0 0 0 reg 1 1 0	3	3	2	9
Immediate	0 1 1 0 1 0 s 0 | data | data if s = 0	3	3	2	9
PUSHA = Push All	0 1 1 0 0 0 0 0	17	17	2	9
POP = Pop:					
Memory	1 0 0 0 1 1 1 1 | mod 0 0 0 r/m	5*	5*	2	9
Register	0 1 0 1 1 reg	5	5	2	9
Segment register	0 0 0 reg 1 1 1 (reg ≠ 01)	5	20	2	9,10,11
POPA = Pop All	0 1 1 0 0 0 0 1	19	19	2	9
XCHG = Exchange:					
Register/memory with register	1 0 0 0 0 1 1 w | mod reg r/m	3,5*	3,5*	2,7	7,9
Register with accumulator	1 0 0 1 0 reg	3	3		
IN = Input from:					
Fixed port	1 1 1 0 0 1 0 w | port	5	5		14
Variable port	1 1 1 0 1 1 0 w	5	5		14
OUT = Output to:					
Fixed port	1 1 1 0 0 1 1 w | port	3	3		14
Variable port	1 1 1 0 1 1 1 w	3	3		14
XLAT = Translate byte to AL	1 1 0 1 0 1 1 1	5	5		9
LEA = Load EA to register	1 0 0 0 1 1 0 1 | mod reg r/m	3*	3*		
LDS = Load pointer to DS	1 1 0 0 0 1 0 1 | mod reg r/m (mod ≠ 11)	7*	21*	2	9,10,11
LES = Load pointer to ES	1 1 0 0 0 1 0 0 | mod reg r/m (mod ≠ 11)	7*	21*	2	9,10,11
LAHF = Load AH with flags	1 0 0 1 1 1 1 1	2	2		
SAHF = Store AH into flags	1 0 0 1 1 1 1 0	2	2		
PUSHF = Push flags	1 0 0 1 1 1 0 0	3	3	2	9
POPF = Pop flags	1 0 0 1 1 1 0 1	5	5	2,4	9,15

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
ARITHMETIC					
ADD = Add:					
Reg/memory with register to either	0 0 0 0 0 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register / memory	1 0 0 0 0 0 s w | mod 0 0 0 r/m | data | data if s:w = 01	3,7*	3,7*	2	9
Immediate to accumulator	0 0 0 0 0 1 0 w | data | data if w = 1	3	3		
ADC = Add with carry:					
Reg/memory with register to either	0 0 0 1 0 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 s w | mod 0 1 0 r/m | data | data if s:w = 01	3,7*	3,7*	2	9
Immediate to accumulator	0 0 0 1 0 1 0 w | data | data if w = 1	3	3		
INC = Increment:					
Register/memory	1 1 1 1 1 1 1 w | mod 0 0 0 r/m	2,7*	2,7*	2	9
Register	0 1 0 0 0 reg	2	2		
SUB = Subtract:					
Reg/memory and register to either	0 0 1 0 1 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate from register/memory	1 0 0 0 0 0 s w | mod 1 0 1 r/m | data | data if s:w = 01	3,7*	3,7*	2	9
Immediate from accumulator	0 0 1 0 1 1 0 w | data | data if w = 1	3	3		
SBB = Subtract with borrow:					
Reg/memory and register to either	0 0 0 1 1 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate from register/memory	1 0 0 0 0 0 s w | mod 0 1 1 r/m | data | data if s:w = 01	3,7*	3,7*	2	9
Immediate from accumulator	0 0 0 1 1 1 0 w | data | data if w = 1	3	3		
DEC = Decrement:					
Register/memory	1 1 1 1 1 1 1 w | mod 0 0 1 r/m	2,7*	2,7*	2	9
Register	0 1 0 0 1 reg	2	2		
CMP = Compare:					
Register/memory with register	0 0 1 1 1 0 1 w | mod reg r/m	2,6*	2,6*	2	9
Register with register/memory	0 0 1 1 1 0 0 w | mod reg r/m	2,7*	2,7*	2	9
Immediate with register/memory	1 0 0 0 0 0 s w | mod 1 1 1 r/m | data | data if s:w = 01	3,6*	3,6*	2	9
Immediate with accumulator	0 0 1 1 1 1 0 w | data | data if w = 1	3	3		
NEG = Change sign	1 1 1 1 0 1 1 w | mod 0 1 1 r/m	2	7*	2	7
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1	3	3		
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1	3	3		
AAS = ASCII adjust for subtract	0 0 1 1 1 1 1 1	3	3		
DAS = Decimal adjust for subtract	0 0 1 0 1 1 1 1	3	3		
MUL = Mulitiply (unsigned)	1 1 1 1 0 1 1 w | mod 1 0 0 r/m				
Register-Byte		13	13		
Register-Word		21	21		
Memory-Byte		16*	16*	2	9
Memory-Word		24*	24*	2	9
IMUL = Integer multiply (signed):	1 1 1 1 0 1 1 w | mod 1 0 1 r/m				
Register-Byte		13	13		
Register-Word		21	21		
Memory-Byte		16*	16*	2	9
Memory-Word		24*	24*	2	9

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
ARITHMETIC (Continued)					
IMUL = Integer immediate multiply (signed)	0 1 1 0 1 0 s 1 | mod reg r/m | data | data if s = 0	21,24*	21,24*	2	9
DIV = Divide (unsigned):	1 1 1 1 0 1 1 w | mod 1 1 0 r/m				
Register-Byte		14	14		
Register-Word		22	22		
Memory-Byte		17*	17*	2,6	6,9
Memory-Word		25*	25*	2,6	6,9
IDIV = Integer divide (signed):	1 1 1 1 0 1 1 w | mod 1 1 1 r/m				
Register-Byte		17	17		
Register-Word		25	25		
Memory-Byte		20*	20*	2	9
Memory-Word		28*	28*	2	9
AAM = ASCII adjust for multiply	1 1 0 1 0 1 0 0 | 0 0 0 0 1 0 1 0	16	16		
AAD = ASCII adjust for divide	1 1 0 1 0 1 0 1 | 0 0 0 0 1 0 1 0	14	14		
CBW = Convert byte to word	1 0 0 1 1 0 0 0	2	2		
CWD = Convert word to double word	1 0 0 1 1 0 0 1	2	2		
LOGIC					
Shift/Rotate Instructions:					
Register/Memory by 1	1 1 0 1 0 0 0 w | mod TTT r/m	2,7*	2,7*	2	9
Register/Memory by CL	1 1 0 1 0 0 1 w | mod TTT r/m	5 + n,8 + n*	5 + n,8 + n*	2	9
Register/Memory by Count	1 1 0 0 0 0 0 w | mod TTT r/m | count	5 + n,8 + n*	5 + n,8 + n*	2	9
	TTT / **Instruction**: 0 0 0 ROL; 0 0 1 ROR; 0 1 0 RCL; 0 1 1 RCR; 1 0 0 SHL/SAL; 1 0 1 SHR; 1 1 1 SAR				
AND = And:					
Reg/memory and register to either	0 0 1 0 0 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 0 w | mod 1 0 0 r/m | data | data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0 0 1 0 0 1 0 w | data | data if w = 1	3	3		
TEST = And function to flags, no result:					
Register/memory and register	1 0 0 0 0 1 0 w | mod reg r/m	2,6*	2,6*	2	9
Immediate data and register/memory	1 1 1 1 0 1 1 w | mod 0 0 0 r/m | data | data if w = 1	3,6*	3,6*	2	9
Immediate data and accumulator	1 0 1 0 1 0 0 w | data | data if w = 1	3	3		
OR = Or:					
Reg/memory and register to either	0 0 0 0 1 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 0 w | mod 0 0 1 r/m | data | data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0 0 0 0 1 1 0 w | data | data if w = 1	3	3		
XOR = Exclusive or:					
Reg/memory and register to either	0 0 1 1 0 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 0 w | mod 1 1 0 r/m | data | data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0 0 1 1 0 1 0 w | data | data if w = 1	3	3		
NOT = Invert register/memory	1 1 1 1 0 1 1 w | mod 0 1 0 r/m	2,7*	2,7*	2	9

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
STRING MANIPULATION:					
MOVS = Move byte/word	1 0 1 0 0 1 0 w	5	5	2	9
CMPS = Compare byte/word	1 0 1 0 0 1 1 w	8	8	2	9
SCAS = Scan byte/word	1 0 1 0 1 1 1 w	7	7	2	9
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w	5	5	2	9
STOS = Stor byte/wd from AL/A	1 0 1 0 1 0 1 w	3	3	2	9
INS = Input byte/wd from DX port	0 1 1 0 1 1 0 w	5	5	2	9,14
OUTS = Output byte/wd to DX port	0 1 1 0 1 1 1 w	5	5	2	9,14
Repeated by count in CX					
MOVS = Move string	1 1 1 1 0 0 1 0 | 1 0 1 0 0 1 0 w	5 + 4n	5 + 4n	2	9
CMPS = Compare string	1 1 1 1 0 0 1 z | 1 0 1 0 0 1 1 w	5 + 9n	5 + 9n	2	9
SCAS = Scan string	1 1 1 1 0 0 1 z | 1 0 1 0 1 1 1 w	5 + 8n	5 + 8n	2	9
LODS = Load string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 1 0 w	5 + 4n	5 + 4n	2	9
STOS = Store string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 0 1 w	4 + 3n	4 + 3n	2	9
INS = Input string	1 1 1 1 0 0 1 0 | 0 1 1 0 1 1 0 w	5 + 4n	5 + 4n	2	9,14
OUTS = Output string	1 1 1 1 0 0 1 0 | 0 1 1 0 1 1 1 w	5 + 4n	5 + 4n	2	9,14
CONTROL TRANSFER					
CALL = Call:					
Direct within segment	1 1 1 0 1 0 0 0 | disp-low | disp-high	7 + m	7 + m	2	8
Register memory indirect within segment	1 1 1 1 1 1 1 1 | mod 0 1 0 r/m	7 + m,11 + m*	7 + m,11 + m*	2	8,9
Direct intersegment	1 0 0 1 1 0 1 0 | segment offset | segment selector	13 + m	26 + m	2	8,11,12
Protected Mode Only (Direct Intersegment):					
Via call gate to same privilege level			41 + m		8,11,12
Via call gate to different privilege level, no parameters			82 + m		8,11,12
Via call gate to different privilege level, x parameters			86 + 4x + m		8,11,12
Via TSS			177 + m		8,11,12
Via task gate			182 + m		8,11,12
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 0 1 1 r/m (mod ≠ 11)	16 + m	29 + m*	2	8,9,11,12
Protected Mode Only (Indirect Intersegment):					
Via call gate to same privilege level			44 + m*		8,9,11,12
Via call gate to different privilege level, no parameters			83 + m*		8,9,11,12
Via call gate to different privilege level, x parameters			90 + 4x + m*		8,9,11,12
Via TSS			180 + m*		8,9,11,12
Via task gate			185 + m*		8,9,11,12
JMP = Unconditional jump:					
Short/long	1 1 1 0 1 0 1 1 | disp-low	7 + m	7 + m		8
Direct within segment	1 1 1 0 1 0 0 1 | disp-low | disp-high	7 + m	7 + m		8
Register/memory indirect within segment	1 1 1 1 1 1 1 1 | mod 1 0 0 r/m	7 + m,11 + m*	7 + m,11 + m*	2	8,9
Direct intersegment	1 1 1 0 1 0 1 0 | segment offset | segment selector	11 + m	23 + m		8,11,12
Protected Mode Only (Direct Intersegment):					
Via call gate to same privilege level			38 + m		8,11,12
Via TSS			175 + m		8,11,12
Via task gate			180 + m		8,11,12
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 1 0 1 r/m (mod ≠ 11)	15 + m*	26 + m*	2	8,9,11,12

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
CONTROL TRANSFER (Continued):					
Protected Mode Only (Indirect Intersegment):					
Via call gate to same privilege level			41 + m*		8,9,11,12
Via TSS			178 + m*		8,9,11,12
Via task gate			183 + m*		8,9,11,12
RET = Return from CALL:					
Within segment	1 1 0 0 0 0 1 1	11 + m	11 + m	2	8,9
Within seg adding immed to SP	1 1 0 0 0 0 1 0 | data-low | data-high	11 + m	11 + m	2	8,9
Intersegment	1 1 0 0 1 0 1 1	15 + m	25 + m	2	8,9,11,12
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0 | data-low | data-high	15 + m		2	8,9,11,12
Protected Mode Only (RET):					
To different privilege level			55 + m		
JE/JZ = Jump on equal zero	0 1 1 1 0 1 0 0 | disp	7 + m or 3	7 + m or 3		8
JL/JNGE = Jump on less not greater or equal	0 1 1 1 1 1 0 0 | disp	7 + m or 3	7 + m or 3		8
JLE/JNG = Jump on less or equal not greater	0 1 1 1 1 1 1 0 | disp	7 + m or 3	7 + m or 3		8
JB/JNAE = Jump on below not above or equal	0 1 1 1 0 0 1 0 | disp	7 + m or 3	7 + m or 3		8
JBE/JNA = Jump on below or equal not above	0 1 1 1 0 1 1 0 | disp	7 + m or 3	7 + m or 3		8
JP/JPE = Jump on parity/parity even	0 1 1 1 1 0 1 0 | disp	7 + m or 3	7 + m or 3		8
JO = Jump on overflow	0 1 1 1 0 0 0 0 | disp	7 + m or 3	7 + m or 3		8
JS = Jump on sign	0 1 1 1 1 0 0 0 | disp	7 + m or 3	7 + m or 3		8
JNE/JNZ = Jump on not equal not zero	0 1 1 1 0 1 0 1 | disp	7 + m or 3	7 + m or 3		8
JNL/JGE = Jump on not less greater or equal	0 1 1 1 1 1 0 1 | disp	7 + m or 3	7 + m or 3		8
JNLE/JG = Jump on not less or equal greater	0 1 1 1 1 1 1 1 | disp	7 + m or 3	7 + m or 3		8
JNB/JAE = Jump on not below above or equal	0 1 1 1 0 0 1 1 | disp	7 + m or 3	7 + m or 3		8
JNBE/JA = Jump on not below or equal above	0 1 1 1 0 1 1 1 | disp	7 + m or 3	7 + m or 3		8
JNP/JPO = Jump on not par / par odd	0 1 1 1 1 0 1 1 | disp	7 + m or 3	7 + m or 3		8
JNO = Jump on not overflow	0 1 1 1 0 0 0 1 | disp	7 + m or 3	7 + m or 3		8
JNS = Jump on not sign	0 1 1 1 1 0 0 1 | disp	7 + m or 3	7 + m or 3		8
LOOP = Loop CX times	1 1 1 0 0 0 1 0 | disp	8 + m or 4	8 + m or 4		8
LOOPZ/LOOPE = Loop while zero equal	1 1 1 0 0 0 0 1 | disp	8 + m or 4	8 + m or 4		8
LOOPNZ/LOOPNE = Loop while not zero equal	1 1 1 0 0 0 0 0 | disp	8 + m or 4	8 + m or 4		8
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1 | disp	8 + m or 4	8 + m or 4		8
ENTER = Enter Procedure	1 1 0 0 1 0 0 0 | data-low | data-high | L				
L = 0		11	11	2	9
L = 1		15	15	2	9
L > 1		16 - 4(L - 1)	16 - 4(L - 1)	2	9
LEAVE = Leave Procedure	1 1 0 0 1 0 0 1	5	5	2	9

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
CONTROL TRANSFER (Continued):					
INT = Interrupt:					
Type specified	1 1 0 0 1 1 0 1 | type	23 + m		2	
Type 3	1 1 0 0 1 1 0 0	23 + m		2	
INTO = Interrupt on overflow	1 1 0 0 1 1 1 0	24 – m or 3 (3 if no) (Interrupt)	24 – or 3 (3 if no) (Interrupt)	2	
Protected Mode Only:					
Via interrupt or trap gate to same privilege level			40 + m		8,11,12
Via interrupt or trap gate to fit different privilege level			78 + m		8,11,12
Via Task Gate			167 + m		8,11,12
IRET = Interrupt return	1 1 0 0 1 1 1 1	17 + m	31 + m	2,4	8,9,11,12,15
Protected Mode Only:					
To different privilege level			55 + m		8,9,11,12,15
To different task (NT = 1)			169 + m		8,9,11,12
BOUND = Detect value out of range	0 1 1 0 0 0 1 0 | mod reg r/m	13*	13* (Use INT clock count if exception 5)	2,6	6,8,9,11,12
PROCESSOR CONTROL					
CLC = Clear carry	1 1 1 1 1 0 0 0	2	2		
CMC = Complement carry	1 1 1 1 0 1 0 1	2	2		
STC = Set carry	1 1 1 1 1 0 0 1	2	2		
CLD = Clear direction	1 1 1 1 1 1 0 0	2	2		
STD = Set direction	1 1 1 1 1 1 0 1	2	2		
CLI = Clear interrupt	1 1 1 1 1 0 1 0	3	3		14
STI = Set interrupt	1 1 1 1 1 0 1 1	2	2		14
HLT = Halt	1 1 1 1 0 1 0 0	2	2		13
WAIT = Wait	1 0 0 1 1 0 1 1	3	3		
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0	0	0		14
CTS = Clear task switched flag	0 0 0 0 1 1 1 1 | 0 0 0 0 0 1 1 0	2	2	3	13
ESC = Processor Extension Escape	1 0 0 1 1 T T T | mod LLL r/m (TTT LLL are opcode to processor extension)	9-20*	9-20*	5	17
SEG = Segment override prefix	0 0 1 reg 1 1 0	0	0		

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
PROTECTION CONTROL					
LGDT = Load global descriptor table register	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 1 | mod 0 1 0 r/m	11*	11*	2,3	9,13
SGDT = Store global descriptor table register	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 1 | mod 0 0 0 r/m	11*	11*	2,3	9
LIDT = Load interrupt descriptor table register	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 1 | mod 0 1 1 r/m	12*	12*	2,3	9,13
SIDT = Store interrupt descriptor table register	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 1 | mod 0 0 1 r/m	12*	12*	2,3	9
LLDT = Load local descriptor table register from register memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 0 | mod 0 1 0 r/m		17,19*	1	9,11,13
SLDT = Store local descriptor table register to register/memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 0 | mod 0 0 0 r/m		2,3*	1	9
LTR = Load task register from register/memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 0 | mod 0 1 1 r/m		17,19*	1	9,11,13
STR = Store task register to register memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 0 | mod 0 0 1 r/m		2,3*	1	9,11,13
LMSW = Load machine status word from register/memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 1 | mod 1 1 0 r/m	3,6*	3,6*	2,3	9,13
SMSW = Store machine status word	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 1 | mod 1 0 0 r/m	2,3*	2,3*	2,3	9
LAR = Load access rights from register/ memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 1 0 | mod reg r/m		14,16*	1	9,16
LSL = Load segment limit from register/ memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 1 1 | mod reg r/m		14,16*	1	9,16
ARPL = Adjust requested privilege level: from register/memory	0 1 1 0 0 0 1 1 | mod reg r/m		10*,11*	2	9
VERR = Verify read access: register/memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 0 | mod 1 0 0 r/m		14,16*	1	9,16
VERR = Verify write access:	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 0 | mod 1 0 1 r/m		14,16*	1	9,16

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

Footnotes

The effective Address (EA) of the memory operand is computed according to the mod and r/m fields:

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

SEGMENT OVERRIDE PREFIX

0 0 1 reg 1 1 0

REG is assigned according to the following:

REG	Segment Register
00	ES
01	CS
10	SS
11	DS

REG is assigned according to the following table:

16-Bit (w = 1)		8-Bit (w = 0)	
000	AX	000	AL
001	CX	001	CL
010	DX	010	DL
011	BX	011	BL
100	SP	100	AH
101	BP	101	CH
110	SI	110	DH
111	DI	111	BH

The physical addresses of all operands addressed by the BP register are computed using the SS segment register. The physical addresses of the destination operands of the string primitive operations (those addressed by the DI register) are computed using the ES segment, which may not be overridden.

80L286

Low-Power High-Performance Microprocessor with
Memory Management and Protection

DISTINCTIVE CHARACTERISTICS

- High-performance processor (up to ten times iAPX 86 when using the 12 MHz 80L286).
- Identical to the 80286 except consumes 37% less power.
- Available in cost-effective Plastic Leaded Chip Carrier (PLCC) package.
- Socketed PLCC footprint is compatible with socketed LCC and PGA footprints.
- Surface-mountable PLCC for high density board utilization.
- 8, 10 and 12.5 MHz operation
- Large address space
  - 16 megabytes physical
  - 1 gigabyte virtual memory per task
- Integrated memory management, four-level memory protection and support for virtual memory and operating systems.

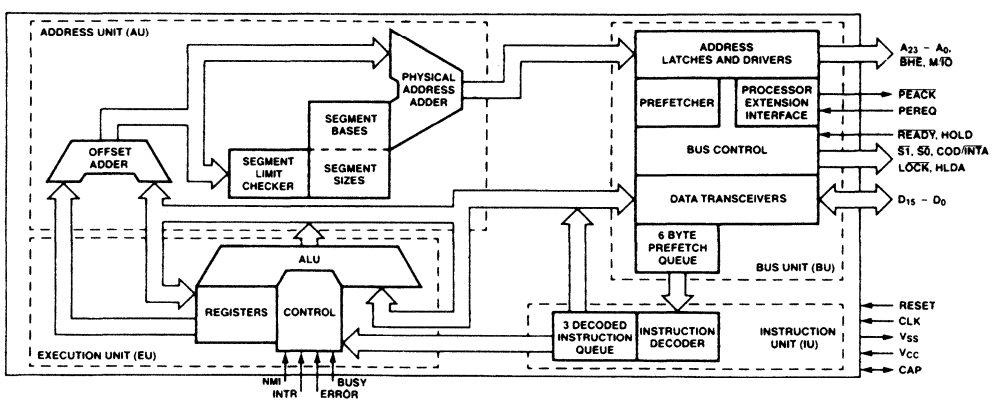
GENERAL DESCRIPTION

The 80L286 is an advanced, high performance microprocessor, identical to the 80286, except consumes much less power. The 80L826 uses less than 2 watts at 55°C — 37% below the 80286. Its reduced power enables the 80L826 to be packaged in low-cost, Plastic Leaded Chip Carrier (PLCC) without a heat sink or heat spreader. Cooler operation also enhances reliability. The PLCC package can be surface-mounted or socketed. The footprint of the socketed PLCC package is identical to the socketed LCC or PGA packages so no board layout change is needed. The 80L826 is available in 8, 10 and 12 MHz speeds and is fully compatible with the 82C288 Bus Controller and the 82284 Clock Driver.

The 80L286 is upward compatible with iAPX 86 and 88 software. Using iAPX 86 real address mode, the 80L286 is object code compatible with existing iAPX 86, 88 software. In protected virtual address mode, the 80L286 is source code compatible with iAPX 86, 88 software and may require upgrading to use virtual addresses supported by the 80L286's integrated memory management and protection mechanism. Both modes operate at full 80L286 performance and execute a superset of the iAPX 86 and 88 instructions.

The 80L286 provides special operations to support the efficient implementation and execution of operating systems. For example, one instruction can end execution of one task, save its state, switch to a new task, load its state, and start execution of the new task. The 80L286 also supports virtual memory systems by providing a segment-not-present exception and restartable instructions.

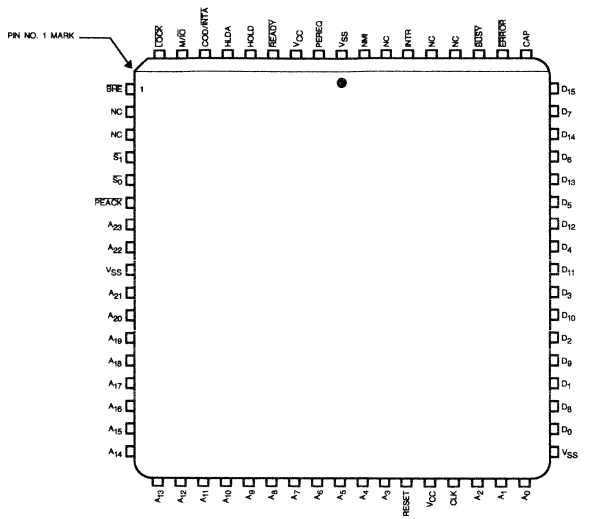
BLOCK DIAGRAM



BD003960

Publication # 08511 | Rev. C | Amendment /0
Issue Date: June 1987

CONNECTION DIAGRAM
Top View



CD010641

As viewed from top of package (PC side of component board)

RELATED AMD PRODUCTS

Part No.	Description
82C288*	Bus Controller
82284*	Clock Driver
82C54	Programmable Interval Timer
8259A	Interrupt Controller
Am9517A	DMA Controller

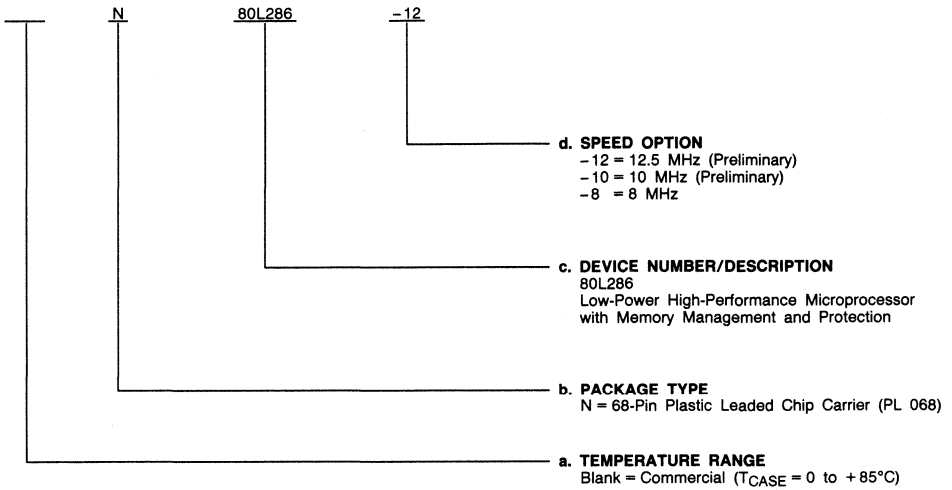
*For 12.5-MHz operation, see the "82284 and 82288 Emulation in an IBM PC/AT Computer Using Two AmPAL** 16 R8B Devices" Application Note, PID #08927A.

**PAL is a registered trademark of and is used under license form Monolithic Memories, Inc.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
N	80L286-12
	80L286-10
	80L286-8

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Active State	Name	I/O	Description
Active HIGH	CLK	I	System Clock provides the fundamental timing for 80L286 systems. It is a 16 MHz signal divided by two inside the 80L286 to generate the 8 MHz processor clock. The internal divide-by-two circuitry can be synchronized to an external clock generator by a LOW-to-HIGH transition on the RESET input.
Active HIGH	$D_0 - D_{15}$	I/O	Data Bus inputs data during memory, I/O, and interrupt acknowledge read cycles; outputs data during memory and I/O write cycles. The data bus is active HIGH and floats to three-state OFF during bus hold acknowledge.
Active HIGH	$A_{23} - A_0$	O	Address Bus outputs physical memory and I/O port addresses. A_0 is LOW when data is to be transferred on pins D_{7-0}. $A_{23} - A_{16}$ are LOW during I/O transfers. The address bus is active HIGH and floats to three-state OFF during bus hold acknowledge.
Active LOW	$\overline{BHE}$	O	Bus High Enable indicates transfer of data on the upper byte of the data bus D_{15-8}. Eight-bit oriented devices assigned to the upper byte of the data bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is active LOW and floats to three-state OFF during bus hold acknowledge.

$\overline{BHE}$ and A_0 Encodings

$\overline{BHE}$ Value	A_0 Value	Function
0	0	Word Transfer
0	1	Byte transfer on upper half of data bus (D_{15-8})
1	0	Byte transfer on lower half of data bus (D_{7-0})
1	1	Reserved

Active LOW	$\overline{S1}$, $\overline{S0}$	O	Bus Cycle Status indicates initiation of a bus cycle and, along with M/$\overline{IO}$ and COD/$\overline{INTA}$, defines the type of bus cycle. The bus is in a T_S state whenever one or both are LOW. $\overline{S1}$ and $\overline{S0}$ are active LOW and float to three-state OFF during bus hold acknowledge.

80L286 Bus Cycle Status Definition

COD/$\overline{INTA}$	M/$\overline{IO}$	$\overline{S1}$	$\overline{S0}$	Bus cycle initiated
0 (LOW)	0	0	0	Interrupt acknowledge
0	0	0	1	Reserved
0	0	1	0	Reserved
0	0	1	1	None; not a status cycle
0	1	0	0	IF $A_1 = 1$ then halt; else shutdown
0	1	0	1	Memory data read
0	1	1	0	Memory data write
0	1	1	1	None; not a status cycle
1 (HIGH)	0	0	0	Reserved
1	0	0	1	I/O read
1	0	1	0	I/O write
1	0	1	1	None; not a status cycle
1	1	0	0	Reserved
1	1	0	1	Memory instruction read
1	1	1	0	Reserved
1	1	1	1	None; not a status cycle

	M/$\overline{IO}$	O	Memory/IO Select distinguishes memory access from I/O access. If HIGH during T_S, a memory cycle or a halt/shutdown cycle is in progress. If LOW, an I/O cycle or an interrupt acknowledge cycle is in progress. M/$\overline{IO}$ floats to three-state OFF during bus hold acknowledge.
	COD/$\overline{INTA}$	O	Code/Interrupt Acknowledge distinguishes instruction fetch cycles from memory data read cycles. Also distinguishes interrupt acknowledge cycles from I/O cycles. COD/$\overline{INTA}$ floats to three-state OFF during bus hold acknowledge.
Active LOW	$\overline{LOCK}$	O	Bus Lock indicates that other system bus masters are not to gain control of the system bus following the current bus cycle. The $\overline{LOCK}$ signal may be activated explicitly by the "LOCK" instruction prefix or automatically by 80L286 hardware during memory XCHG instructions, interrupt acknowledge, or descriptor table access. $\overline{LOCK}$ is active LOW and floats to three-state OFF during bus hold acknowledge.
Active LOW	$\overline{READY}$	I	Bus Ready terminates a bus cycle. Bus cycles are extended without limit until terminated by $\overline{READY}$ LOW. $\overline{READY}$ is an active LOW synchronous input requiring set-up and hold times relative to the system clock be met for correct operation. $\overline{READY}$ is ignored during bus hold acknowledge.
Active HIGH	HOLD, HLDA	I O	Bus Hold Request and Hold Acknowledge control ownership of the 80L286 local bus. The HOLD input allows another local bus master to request control of the local bus. When control is granted, the 80L286 will float its bus drivers to three-state OFF and then active HLDA, thus entering the bus hold acknowledge condition. The local bus will remain granted to the requesting master until HOLD becomes inactive which results in the 80L286 deactivating HLDA and regaining control of the local bus. This terminates the bus hold acknowledge condition. HOLD may be asynchronous to the system clock. These signals are active HIGH.

PIN DESCRIPTION (Cont'd.)

Active State	Name	I/O	Description
Active HIGH	INTR	I	Interrupt Request requests the 80L286 to suspend its current program execution and service a pending external request. Interrupt requests are masked whenever the interrupt enable bit in the flag word is cleared. When the 80L286 responds to an interrupt request, it performs two interrupt acknowledge bus cycles to read an 8-bit interrupt vector that identifies the source of the interrupt. To assure program interruption, INTR must remain active until the first interrupt acknowledge cycle is completed. INTR is sampled at the beginning of each processor cycle and must be active HIGH at least two processor cycles before the current instruction ends in order to interrupt before the next instruction. INTR is level sensitive, active HIGH, and may be asynchronous to the system clock.
Active HIGH	NMI	I	Non-maskable Interrupt Request interrupts the 80L286 with an internally supplied vector value of 2. No interrupt acknowledge cycles are performed. The interrupt enable bit in the 80L286 flag word does not affect this input. The NMI input is active HIGH, may be asynchronous to the system clock, and is edge triggered after internal synchronization. For proper recognition, the input must have been previously LOW for at least four system clock cycles and remain HIGH for at least four system clock cycles.
	PEREQ, $\overline{\text{PEACK}}$	I O	Processor Extension Operand Request and Acknowledge extended the memory management and protection capabilities of the 80L286 to processor extensions. The PEREQ input requests the 80L286 to perform a data operand transfer for a processor extension. The $\overline{\text{PEACK}}$ output signals the processor extension when the requested operand is being transferred. PEREQ is active HIGH and may be asynchronous to the system clock. $\overline{\text{PEACK}}$ is active LOW.
Active LOW	$\overline{\text{BUSY}}$, $\overline{\text{ERROR}}$	I I	Processor Extension Busy and Error indicate the operating condition of a processor extension to the 80L286. An active $\overline{\text{BUSY}}$ input stops 80L286 program execution on WAIT and some ESC instructions until $\overline{\text{BUSY}}$ becomes inactive (HIGH). The 80L286 may be interrupted while waiting for $\overline{\text{BUSY}}$ to become inactive. An active $\overline{\text{ERROR}}$ input causes the 80L286 to perform a processor extension interrupt when executing WAIT or some ESC instructions. These inputs are active LOW and may be asynchronous to the system clock.
Active HIGH	RESET	I	System Reset clears the internal logic of the 80L286 and is active HIGH. The 80L286 may be reinitialized at any time with a LOW-to-HIGH transition on RESET which remains active for more than 16 system clock cycles. During RESET active, the output pins of the 80L286 enter the state shown below: **80L286 Pin State During Reset** **Pin Value** — **Pin Names** 1 (HIGH) — $\overline{\text{S0}}$, $\overline{\text{S1}}$, $\overline{\text{PEACK}}$, $A_{23}-A_0$, $\overline{\text{BHE}}$, $\overline{\text{LOCK}}$ 0 (LOW) — M/$\overline{\text{IO}}$, COD/$\overline{\text{INTA}}$, HLDA three-state OFF — $D_{15}-D_0$ Operation of the 80L286 begins after a HIGH-to-LOW transition on RESET. The HIGH-to-LOW transition of RESET must be synchronous to the system clock. Approximately 50 system clock cycles are required by the 80L286 for internal initializations before the first bus cycle to fetch code from the power-on execution address is performed. A LOW-to-HIGH transition of RESET synchronous to the system clock, will begin a new processor cycle at the next HIGH-to-LOW transition of the system clock. The LOW-to-HIGH transition of RESET may be asynchronous to the system clock; however, in this case it cannot be predetermined which phase of the processor clock will occur during the next system clock period. Synchronous LOW-to-HIGH transitions of RESET are only required for systems where the processor clock must be phase synchronous to another clock.
Active HIGH	V_{SS}	I	System Ground: 0 VOLTS.
Active HIGH	V_{CC}	I	System Power: +5 Volt Power Supply.
Active HIGH	CAP	I	Substrate Filter Capacitor: a 0.047μf ±20% 12 V capacitor must be connected between this pin and ground. This capacitor filters the output of the internal substrate bias generator. A maximum DC leakage current of 1 μa is allowed through the capacitor. For correct operation of the 80L286, the substrate bias generator must charge this capacitor to its operating voltage. The capacitor charge-up time is 5 milliseconds (max.) after V_{CC} and CLK reach their specified AC and DC parameters. RESET may be applied to prevent spurious activity by the CPU during this time. After this time, the 80L286 processor clock can be phase synchronized to another clock by pulsing RESET LOW synchronous to the system clock.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin with
Respect to Ground..... −1.0 to +7.0 V
Power Dissipation (8, 10 MHz) ..... 2.0 Watts
(12 MHz) .....2.25 Watts

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_{CASE})..... 0 to +85°C
Supply Voltage (V_{CC}) .....5 V ± 5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS (T_{CASE} = 0 to 85°C, V_{CC} = 5 V ±5%)

Parameters	Description	Test Conditions	8 MHz		10 MHz		12 MHZ		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
V_{IL}	Input LOW Voltage		−.5	.8	−.5	.8	−.5	.8	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC} + .5	2.0	V_{CC} + .5	2.0	V_{CC} + .5	V
V_{ILC}	CLK Input LOW Voltage		−.5	.6	−.5	.6	−.5	.6	V
V_{IHC}	CLK Input HIGH Voltage		3.8	V_{CC} + .5	3.8	V_{CC} + .5	3.8	V_{CC} + .5	V
V_{OL}	Output LOW Voltage	I_{OL} = 2.0 mA		.45		.45		.45	V
V_{OH}	Output HIGH Voltage	I_{OH} = −400 µA	2.4		2.4		2.4		V
I_{LI}	Input Leakage Current	0 V ≤ V_{IN} ≤ V_{CC}		±10		±10		±10	µA
I_{LO}	Output Leakage Current	.45 V ≤ V_{OUT} ≤ V_{CC}		±10		±10		±10	µA
I_{CC}	Supply Current	T_A = 0°C		450		450		500	mA
		T_A = 55°C		375		375		425	mA
C_{CLK}	CLK Input Capacitance	F_C = 1 MHz		20		20		20	pF
C_{IN}	Other Input Capacitance	F_C = 1 MHz		10		10		10	pF
C_O	Input/Output Capacitance	F_C = 1 MHz		20		20		20	pF
I_{LO}	Output Leakage Current	0 V ≤ V_{out} < .045 V		±1		±1		±1	mA
I_{IL}	Input Sustaining Current on $\overline{BUSY}$ and $\overline{ERROR}$ pins	V_{in} = 0 V	30	500	30	500	30	500	µA

Notes: 1. Low temperature is worst case.

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS (T_{CASE} = 0 to +85°C)

AC timings are referenced to 0.8 V and 2.0 V points of signals as illustrated in datasheet waveforms, unless otherwise noted.

Parameters	Description	Test Conditions	8 MHz		10 MHz		12 MHz		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
1	System Clock (CLK) Period		62	250	50	250	40	250	ns
2	System Clock (CLK) LOW Time	at 1.0 V	15	225	12	232	11	237	ns
3	System Clock (CLK) HIGH Time	at 3.6 V	25	235	16	239	13	239	ns
17	System Clock (CLK) Rise Time	1.0 V to 3.6 V		10		8		8	ns
18	System Clock (CLK) Fall Time	3.6 V to 1.0 V		10		8		8	ns
4	Asynch. Inputs Setup Time	Note 1	20		20		15		ns
5	Asynch. Inputs Hold Time	Note 1	20		20		15		ns
6	RESET Setup Time		28		23		18		ns
7	RESET Hold Time		5		5		5		ns
8	Read Data Setup Time		10		8		5		ns
9	Read Data Hold Time		8		8		6		ns
10	$\overline{\text{READY}}$ Setup Time		38		26		22		ns
11	$\overline{\text{READY}}$ Hold Time		25		25		20		ns
12	Status/$\overline{\text{PEACK}}$ Valid Delay	Note 2, Note 3	1	40	-	-	-	-	ns
12a	Status/$\overline{\text{PEACK}}$ Active Delay	Note 2, Note 3	-	-	1	22	3	18	ns
12b	Status/$\overline{\text{PEACK}}$ Inactive Delay	Note 2, Note 3	-	-	1	30	3	20	ns
13	Address Valid Delay	Note 2, Note 3	1	60	1	35	1	32	ns
14	Write Data Valid Delay	Note 2, Note 3	0	50	0	30	0	30	ns
15	Address/Status/Data Float Delay	Note 2, Note 4	0	50	0	47	0	32	ns
16	HLDA Valid Delay	Note 2, Note 3	0	50	0	47	0	25	ns
19	Address Valid To Status Valid Setup Time	Note 3, Note 5, Note 6	38		27		22		ns

Notes:
1. Asynchronous inputs are INTR, NMI, HOLD PEREQ, $\overline{\text{ERROR}}$, and $\overline{\text{BUSY}}$. This specification is given only for testing purposes to assure recognition at a specific CLK edge.
2. Delay from 0.8 V on the CLK to 0.8 V or 2.0 V or float on the output as appropriate for valid or floating condition.
3. Output load: C_L = 100 pF.
4. Float condition occurs when output current is less than I_{LO} in magnitude.
5. Delay measured from address either reaching 0.8 V or 2.0 V (valid) to status going active reaching 2.0 V or status going inactive reaching 0.8 V.
6. For load capacitance of 10 pF on STATUS/$\overline{\text{PEACK}}$ lines, subtract typically 7 ns for 8 MHz spec, and maximum 7 ns for 10 MHz.

82284 Timing Requirements

Parameters	Description	Test Conditions	82284-8		82284-10		Units
			Min.	Max.	Min.	Max.	
11	$\overline{\text{SRDY}}$/$\overline{\text{SRDYEN}}$ Setup Time		17		15		ns
12	$\overline{\text{SRDY}}$/$\overline{\text{SRDYEN}}$ Hold Time		0		0		ns
13	$\overline{\text{ARDY}}$/$\overline{\text{ARDYEN}}$ Setup Time	Note 1	0		0		ns
14	$\overline{\text{ARDY}}$/$\overline{\text{ARDYEN}}$ Hold Time	Note 1	30		30		ns
19	PCLK Delay	C_L = 75 pF I_{OL} = 5 mA I_{OH} = −1 mA	0	45	0	35	ns

Note 1. These times are given for testing purposes to assure a predetermined action.

SWITCHING CHARACTERISTICS (Cont'd.)

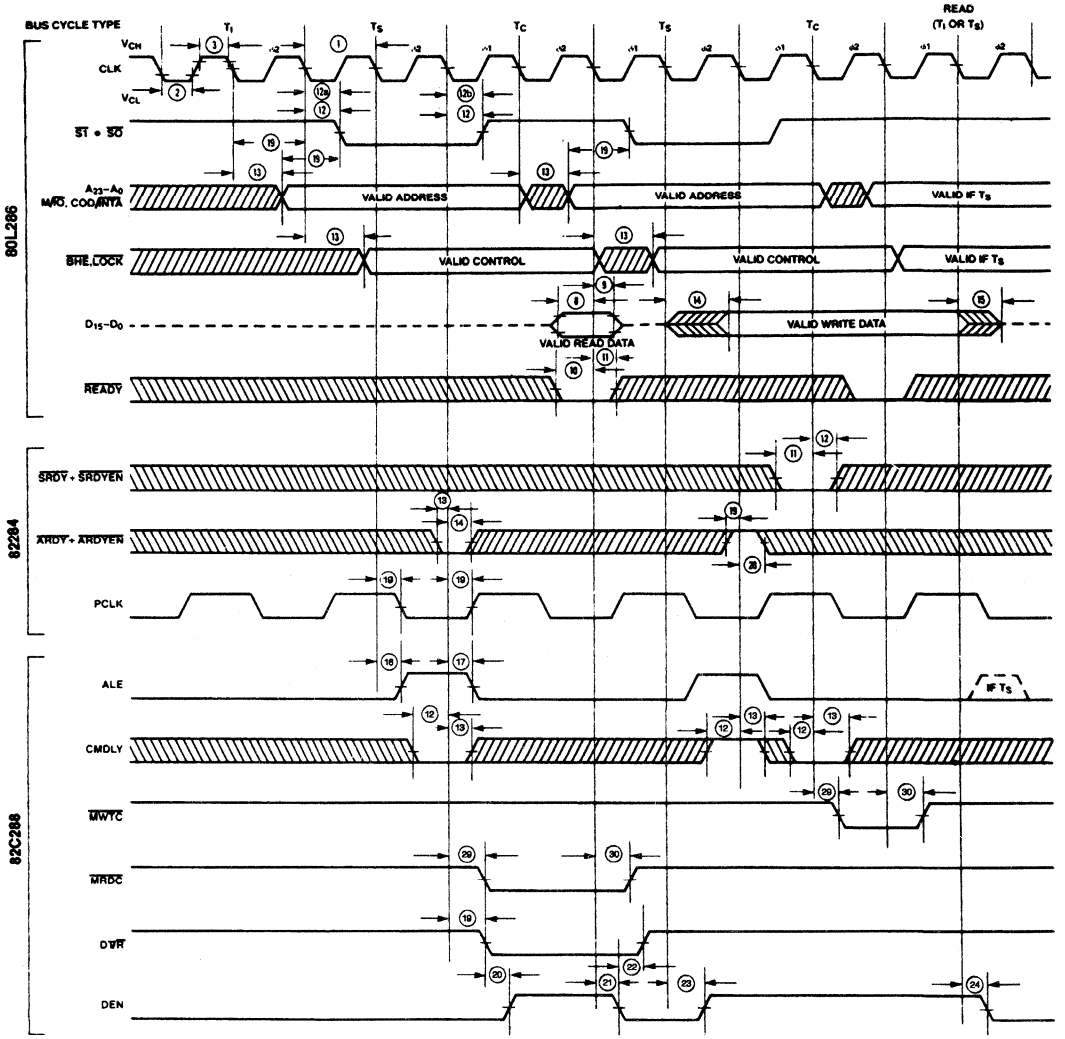
82C288 Timing Requirements*

Parameters	Description		Test Conditions	82C288-8		82C288-10		Units
				Min.	Max.	Min.	Max.	
12	CMDLY Setup Time			20		15		ns
13	CMDLT Hold Time			1		1		ns
30	Command Delay from CLK	Command Inactive	C_L = 300 pF Max. I_{OL} = 32 mA Max. I_{OH} = 5 mA Max.	5	25	5	20	ns
29		Command Active		3	25	3	21	
16	ALE Active Delay		C_L = 150 pF I_{OL} = 16 mA Max. I_{OH} = −1 mA Max.	3	20	3	16	ns
17	ALE Inactive Delay				25		19	ns
19	DT/$\overline{R}$ Read Active Delay				25		23	ns
22	DT/$\overline{R}$ Read Inactive Delay			5	35	5	20	ns
20	DEN Read Active Delay			5	35	5	21	ns
21	DEN Read Inactive Delay			3	35	3	21	ns
23	DEN Write Active Delay				30		23	ns
24	DEN Write Inactive Delay			3	30	3	19	ns

*For 12.5-MHz operation, see the "82284 and 82288 Emulation in an IBM PC/AT Computer Using Two AmPAL16R8B Devices" Application Note, PID #08927A.

SWITCHING WAVEFORMS

MAJOR CYCLE TIMING

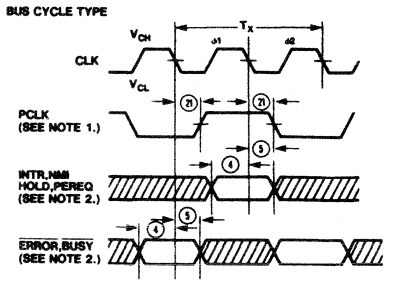


WF007983

Note: 1. $\overline{\text{MWTC}}$ is valid at this point only if CMDLY is LOW.

SWITCHING WAVEFORMS (Cont'd.)

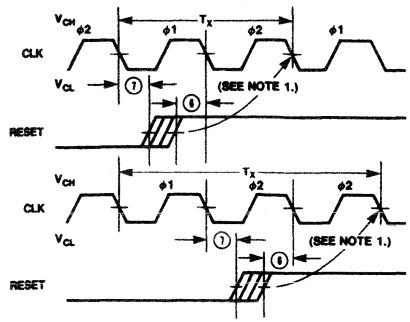
80L286 ASYNCHRONOUS INPUT SIGNAL TIMING



WF009930

Notes:
1. PCLK indicates which processor cycle phase will occur on the next CLK. PCLK may not indicate the correct phase until the first bus cycle is performed.
2. These inputs are asynchronous. The set-up and hold times shown assure recognition for testing purposes.

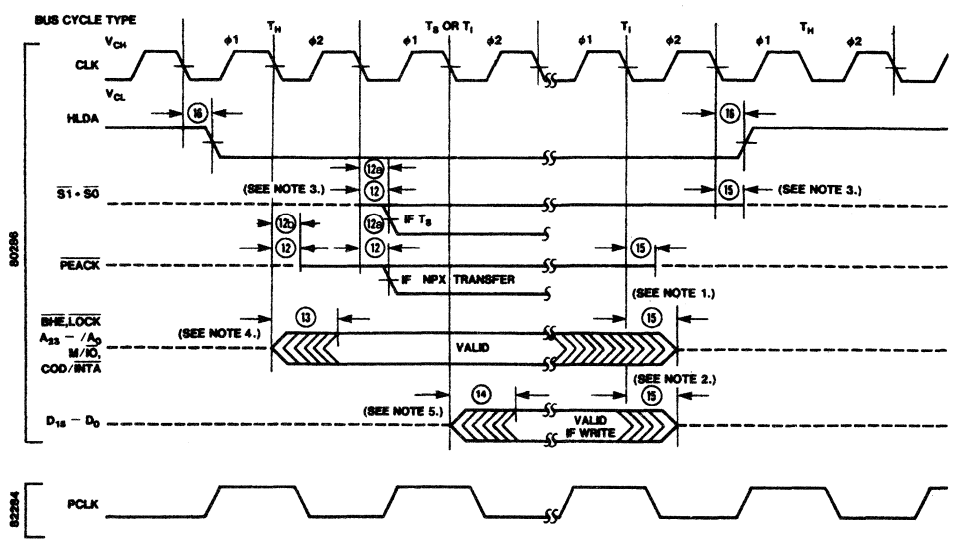
80L286 RESET INPUT TIMING AND SUBSEQUENT PROCESSOR CYCLE PHASE



WF007930

Note 1: When RESET meets the set-up time shown, the next CLK will start or repeat $\phi 1$ of a processor cycle.

EXITING AND ENTERING HOLD

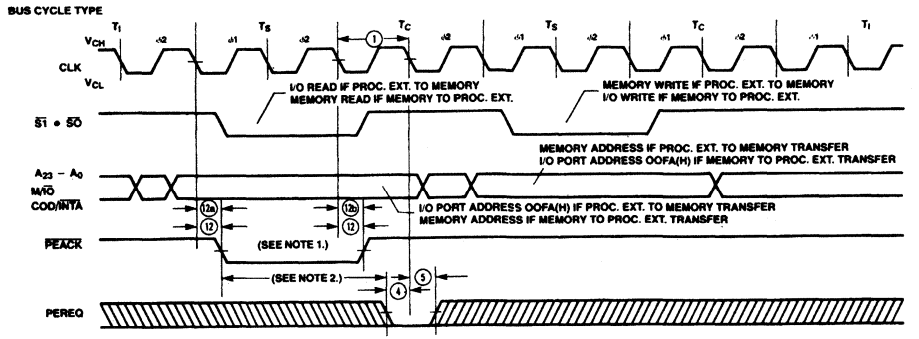


WF009943

Notes: 1. These signals may not be driven by the 80L286 during the time shown. The worst case in terms of latest float time is shown.
2. The data bus will be driven as shown if the last cycle before T_I in the diagram was a write T_C.
3. The 80L286 floats its status pins during T_H. External 20 kΩ resistors keep these signals high.
4. $\overline{BHE}$ and $\overline{LOCK}$ are driven at this time but will not become valid until T_S.
5. The data bus will remain in three-state OFF if a read cycle is performed.

SWITCHING WAVEFORMS (Cont'd.)

80L286 PEREQ/$\overline{PEACK}$ TIMING REQUIRED PEREQ TIMING FOR ONE TRANSFER ONLY



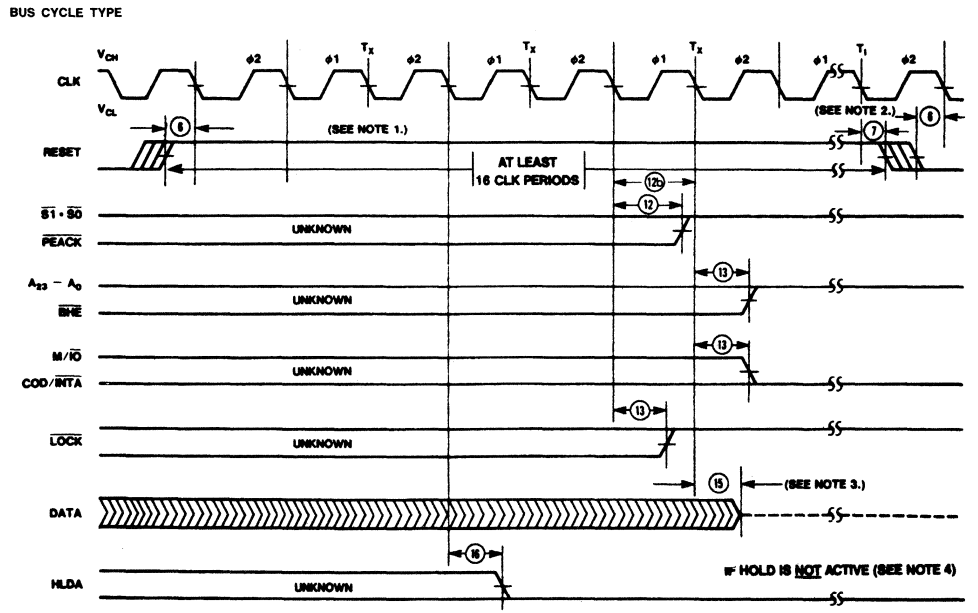
WF007953

ASSUMING WORD-ALIGNED MEMORY OPERAND; IF ODD ALIGNED, 80L286 TRANSFERS TO/FROM MEMORY BYTE-AT-A-TIME WITH TWO MEMORY CYCLES.

Notes: 1. $\overline{PEACK}$ always goes active during the first bus operation of a processor extension data operand transfer sequence. The first bus operation will be either a memory read at operand address or I/O read at port address OOFA(H).

2. To prevent a second processor extension data operand transfer, the worst case maximum time (shown above) is: $3 \times ① - ⑰_{max} - ④_{min}$. The actual, configuration dependent, maximum time is: $3 \times ① - ⑰_{max} - ④_{min} + A \times 2 \times ①$. A is the number of extra T_C states added to either the first or second bus operation of the processor extension data operand transfer sequence.

INITIAL 80L286 PIN STATE DURING RESET



WF007962

Notes: 1. Setup time for RESET ↑ may be violated with the consideration that $\phi 1$ of the processor clock may begin one system CLK period later.

2. Setup and hold times for RESET ↓ must be met for proper operation, but RESET ↓ may occur during $\phi 1$ or $\phi 2$.

3. The data bus is only guaranteed to be in three-state OFF at the time shown.

4. HOLD is acknowledged during RESET, causing HLDA to go active and the appropriate pins to float. If HOLD remains active while RESET goes inactive, the 80L286 remains in HOLD state and will not perform any bus accesses until HOLD is deactivated.

82284 and 82288 EMULATION IN AN IBM PC/AT COMPUTER USING TWO AmPAL16R8 DEVICES

Two AmPAL16R8 devices and four TTL packages generate the signals necessary to run an 80286 system at speeds up to 16 MHz. This reduces the cost and improves the speed of the 80286 micro-system. The first PAL device, AmPAL284, is used to emulate the 82284 Clock Driver and Ready Interface functions. The second PAL device, AmPAL288, is used to emulate the 82288 Bus Controller functions. This application generates the signals used in most non-multibus 80286 systems. The design has been tested in an AT computer. These PAL devices are not pin- for-pin compatible with the parts they emulate nor do they generate all the signals that an 80286 can use including some Multibus signals. However, applications requiring additional signals can be designed by adding additional circuitry.

The simplified Block Diagram (Figure 1) shows the relationship of the PAL devices to the AT computer. Figure 2 is the schematic diagram of the emulator and Table I shows the AT interface wiring chart.

The PAL device equations are written in ABEL and are included in this application note. The source code, the reduced equations, and a fuse map are included for each PAL device.

The AmPAL284 Description

A crystal oscillator is used to generate a clock at twice the 80286's internal clock speed. The PAL device is clocked with the inverted CPU clock. The Output Enable pin is grounded so that it always generates the output signals. The Reset signal is buffered with a Schmitt trigger 74LS14 so that an RC circuit can be used to provide a time delay. Following the Schmitt trigger, a 74F74 flip-flop is used to synchronize to the inverted CPU clock. The Ready Enables of the 82284 are connected together on the AT design and are connected to the $\overline{\text{RDYEN}}$ pin of the AmPAL284. The $\overline{\text{ARDY}}$ signal is synchronized to the inverted 80286 clock. The latched ready $\overline{\text{LARDY}}$ and the $\overline{\text{SRDY}}$ are sampled to generate the $\overline{\text{READY}}$ to the 80286.

The state machine shown in Figure 3 is used to generate the TS and the TC states. This state machine has four states, IDLE, TS2, TC1, and TC2. The state machine requires registered outputs Q0 and Q1 to implement the four states. The state machine goes to TS2 state when either S0 or S1 goes LOW, signalling the start of a bus cycle (refer to Table 2 and Figure 4). The ALE signal goes active (HIGH) for TS2 state and then goes inactive. On the next clock the state machine goes to TC1 state making $\overline{\text{DEN}}$ and DT/$\overline{\text{R}}$ signals active. The next clock causes the state machine to go to state TC2. The state machine either goes to IDLE, making $\overline{\text{DEN}}$ and DT/$\overline{\text{R}}$ inactive if $\overline{\text{RDY}}$ is active, or to TC1 if RDY is not active. The state machine must return to TC1, if $\overline{\text{RDY}}$ is not active, to keep an even number of clock cycles for each inserted TC state.

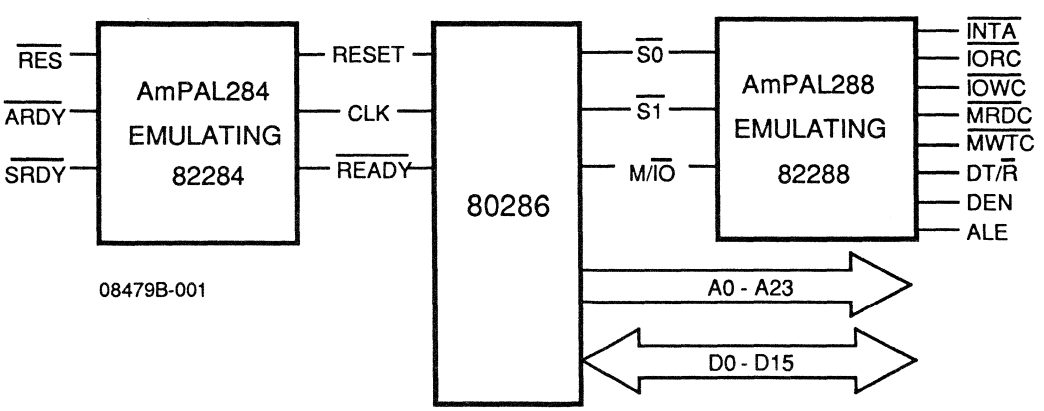
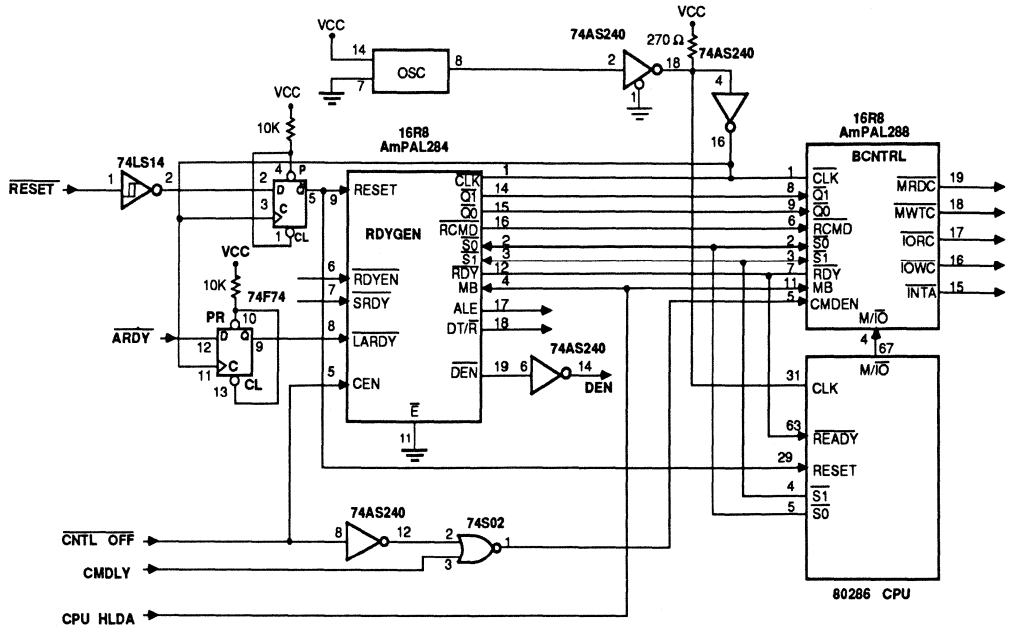


Figure 1. Block Diagram



08479B-002

Figure 2. Schematic Diagram of the 82284 and 82288 Emulator

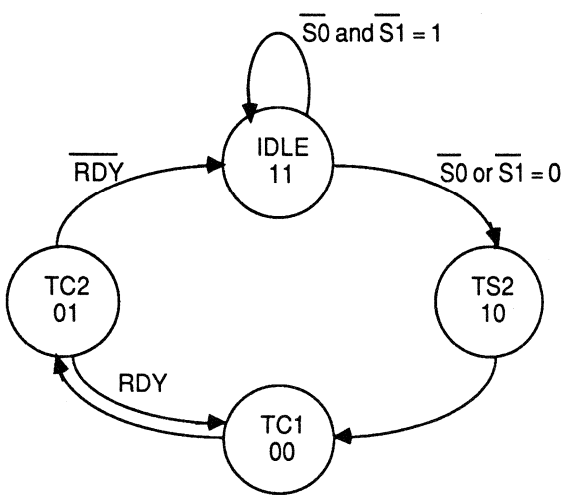
Table 1. Emulator to IBM PC/AT Interface Wiring Chart

Socket Pins on AT Board	Emulator Board Devices (Pin Nos.)				
	AmPAL284	AmPAL288	74F74	74AS240	74S02
82284 (Pin No.)					
$\overline{ARDY}$ (1)			D (12)		
$\overline{SRDY}$ (2)	$\overline{SRDY}$ (7)				
$\overline{ARDYEN}$ (3)	$\overline{RDYEN}$ (6)				
$\overline{READY}$ (4)	$\overline{READY}$ (12)	$\overline{RDY}$ (7)			
CLK (10)				Out (18)	
RES. (11)			D (2)		
RESET (12)			Q (5)		
82288 (Pin No.)					
$\overline{S1}$ (3)	$\overline{S1}$ (3)	$\overline{S1}$ (3)			
ALE (5)	ALE (17)				
MB (6)	MB (4)	MB (11)			
CMDLY (7)					In (3)
$\overline{MRDC}$ (8)		$\overline{MRDC}$ (19)			
$\overline{MWTC}$ (9)		$\overline{MWTC}$ (18)			
$\overline{IOWC}$ (11)		$\overline{IOWC}$ (16)			
$\overline{IORC}$ (12)		$\overline{IORC}$ (17)			
$\overline{INTA}$ (13)		$\overline{INTA}$ (15)			
CEN (15)	CEN (5)			In (8)	
DEN (16)				Out (14)	
DT/$\overline{R}$ (17)	DT/$\overline{R}$ (18)				
M/$\overline{IO}$ (18)		M/$\overline{IO}$ (4)			
$\overline{S0}$ (19)	$\overline{S0}$ (2)	$\overline{S0}$ (2)			

3.6.7 TBL1-A

Table 2. Command and Control Outputs for Each Type of Bus Cycle

Type of Bus Cycle	M/$\overline{IO}$	$\overline{S1}$	$\overline{S0}$	Command Activated	DT/$\overline{R}$ State	ALE, DEN Issued
Interrupt Acknowledge	0	0	0	$\overline{INTA}$	LOW	YES
I/O Read	0	0	1	$\overline{IORC}$	LOW	YES
I/O Write	0	1	0	$\overline{IOWC}$	HIGH	YES
None; idle	0	1	1	None	HIGH	NO
Halt/ Shutdown	1	0	0	None	HIGH	NO
Memory Read	1	0	1	$\overline{MRDC}$	LOW	YES
Memory Write	1	1	0	$\overline{MWTC}$	HIGH	YES
None; idle	1	1	1	None	HIGH	NO



	Q1	Q0
IDLE	1	1
TS2	1	0
TC1	0	0
TC2	0	1

08479B-003

Figure 3. AmPAL State Machine

The $\overline{RCMD}$ signal goes active when the $\overline{S0}$ and $\overline{S1}$ signals decode a Read or Interrupt Acknowledge cycle. This signal is used by the DT/$\overline{R}$ signal to control the trannsmit or the receive direction of the data transceivers and by the AmPAL288. The Command and Control outputs for each type of bus cycle are given in Table 2. Figure 3 shows the cycle timing.

The AmPAL284 source program listing, the reduced equations, and the JEDEC fuse map are shown in Figures 6 , 7 and 8.

The AmPAL288 Description

The AmPAL288 uses the Q0, Q1, $\overline{S0}$, $\overline{S1}$, and M/IO inputs to latch the state of the current cycle. As shown in the PAL equations, these signals are internal signals and are not used by the AT. The $\overline{RCMD}$ signal is an input and $\overline{WCMD}$, $\overline{MEM}$, and $\overline{INT}$ are internally generated. With these latched signals and CMDEN, the following five commands are generated IOWC, IORC, INTA, MRDC, and MWTC. These commands are only enabled when the MB signal is LOW. When MB goes HIGH the commands are high impedance (off).

The AmPAL288 source program listing, reduced equations, and the JEDEC fuse map are shown in Figures 9, 10 and 11.

PAL Device Selection

AmPAL16R8B speed devices are required for operation at 16 MHz and12.5 MHz. For 10 MHz and slower operations, AmPAL16R8A speed devices are adequate. The specifications on any faster 80286 are not established at this time, but the PAL device solution can support faster operations with faster PAL devices.

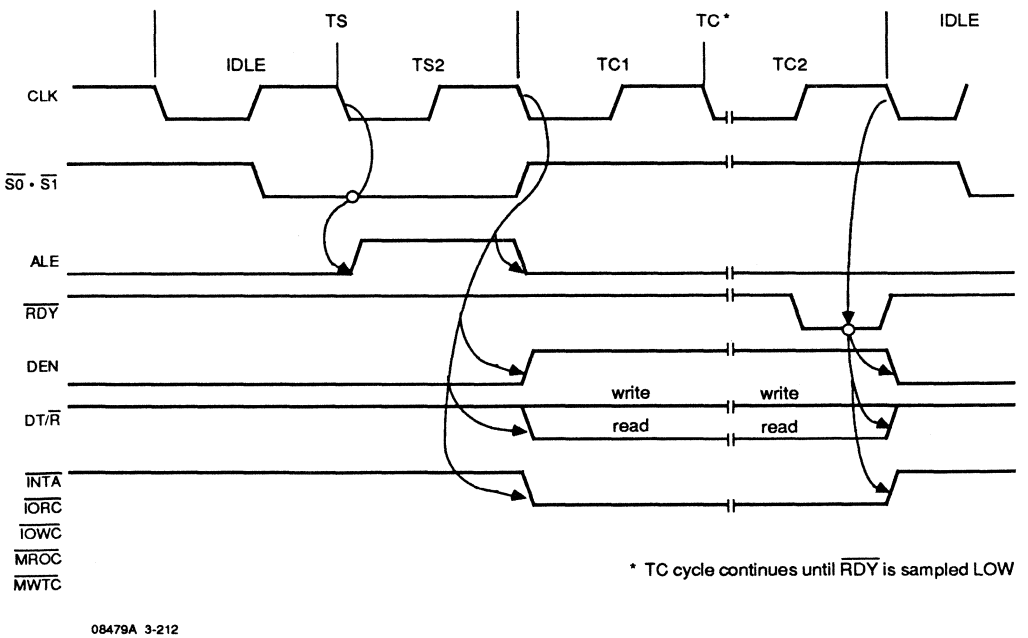


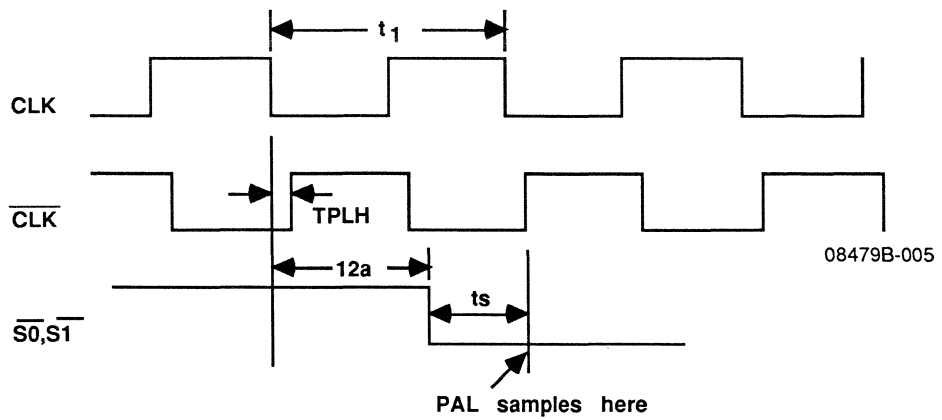
Figure 4. Cycle Timing Diagram

Critical Timing

The 16 MHz 80286 systems Clock Period (t_1) is 31ns and AmPAL16R8 setup time is 13ns (B-speed) or 10ns (D-speed). Therefore, Status Active Delay (t_{12a}) is the parameter that dictates which PAL device to use. To guarantee the proper setup time, use B-speed PAL devices. A 74AS240 or 74F240 must also be used in this design to invert the clock and generate a 2ns delay. Figure 5 shows the timing of this design. If, however, Status Active Delay (t_{12a}) were to increase to >20ns, then D-speed PAL devices would be required for this design.

Design Capabilities

This PAL device design provides the signals necessary for an IBM PC/AT system, but does not fully emulate the 82284 and the 82288. An oscillator is not included, therefore, an external crystal oscillator must be used. The clock requires a pull-up resistor to meet the VOH of the 80286. Multibus operation is not performed. PCLK and MCE signals are not generated. The READY signal is not open-collector. The ARDYEN and SRDYEN signals would have to be externally gated, if they are required.



t_1 – 80286 System Clock Period

t_{12a} – 80286 Status Active Delay

t_s – AmPAL16R8 Setup Time

TPLH – 74AS240 Propagation from LOW input to HIGH output.

General equation:

$$t_1 = t_{12a} + t_s - TPLH$$

Assuming status active delay = 18ns

31ns = 18ns + t_s(PAL device) – 2ns(min)

t_s = 31ns – 18ns + 2ns

t_s=15ns

or less is required for the AmPAL16R8 setup time.

Figure 5. Timing Diagram

```
flag '-R2';
title
'PAL16R8                                        PAL  DEVICE  LOGIC EQUATION
82284 PAL device emulation for the AT
COPYRIGHT 1986 ADVANCED MICRO DEVICES, INC.                    Doug Kern 8/13/86'

PAL284 device 'P16R8';

"declarations

        TRUE,FALSE = 1,0;
        HIGH,LOW = 1,0;
        X,Z,C = .X.,.Z.,.C.;

        GND,VCC
                pin  10,20;

                CLK,S0,S1,MB,CEN,RDYEN,SRDY,LARDY,RESET,EN
                pin1,2,3,4,5,6,7,8,9,11;

                DEN,DT_R,ALE,RCMD,Q0,Q1,AR,RDY
                pin      19,18,17,16,15,14,13,12;

                QSTATE = [Q1,Q0];"STATE MACHINE REGISTERS
"STATE ASSIGNMENTS

        IDLE= ^B11;
        TS2= ^B10;
        TC1= ^B00
        TC2= ^B01;

STATE_DIAGRAM QSTATE

        STATE IDLE:     CASE        !S0          :TS2;
                                    !S1          :TS2;
                                    S1 & S0      :IDLE;
                        ENDCASE;

        STATE TS2:      GOTO        TC1;

        STATE TC1:      GOTO        TC2;

        STATE TC2:      CASE        RDY          :TC1;
                                    !RDY         :IDLE;
                        ENDCASE;
```

Figure 6. AmPAL284 Source Program Listing
(contunied on next page)

(continued)

```
EQUATIONS
        !DT_R   :=!RCMD & Q1 & !Q0
                # !DT_R & !Q1 & !Q0
                # !DT_R & !Q1 & Q0 & RDY;

        !RCMD   := Q1 & Q0 & !S1 & !S0  "INTA
                # Q1 & Q0 & !S1 & S0 "READ
                # !RCMD & Q1 & !Q0
                # !RCMD & !Q1 & !Q0
                # !RCMD & !Q1 &Q0 & RDY;

        !RDY    := !SRDY & !RDYEN & S1 & S0
                # !LARDY & !RDYEN & S1 & S0
                # RESET;

        !DEN    := Q1  & !Q0 & !MB & CEN
                # !Q1 & !Q0 & !MB & CEN
                # !Q1 & Q0 & !MB & CEN & RDY
                # Q1  & !Q0 & MB & !CEN
                # !Q1 & !Q0 & MB & !CEN
                # !Q1 & Q0 & MB & !CEN & RDY;

        ALE     := Q1 & Q0 & !S1
                #  Q1 & Q0 & !S0;
```

Figure 6. AmPAL284 Source Program Listing

ABEL(tm) Version 1.00 - Document Generator 26-Aug-86
PAL16R8 PAL DEVICE LOGIC EQUATION
82284 PAL device emulation for the AT
COPYRIGHT 1986 ADVANCED MICRO DEVICES, INC. Doug Kern 8/13/86
Equations for Module AMD

Device PAL284

Reduced Equations:

```
Q1 :=     !((RDY & !Q1 & Q0 # !Q0));

Q0 :=     !((!S1 & Q1
           # (!S0 & Q1
           # (RDY & !Q1 & Q0
          # Q1 & !Q0))));

DT_R :=   !((RDY & !Q1 & !DT_R
           # (!RCMD & Q1 & !Q0
           # !Q1 & !Q0 & !DT_R)));

RCMD :=   !((!S1 & Q1 & Q0
          # (RDY & !RCMD & !Q1
          # !RCMD & !Q0)));

RDY :=    !((!SRDY & S1 & S0 & !RDYEN
          # (S1 & S0 & !RDYEN & !ARDY
           # RESET)));

DEN :=    !((RDY & !Q1 & !MB & CEN
           # (RDY & !Q1 & MB & !CEN
           # (!Q0 & !MB & CEN
           # !Q0 & MB & !CEN))));

ALE :=    !((S1 & S0 # (!Q1 # !Q0)));
```

Figure 7. AmPAL284 Reduced Equations

```
ABEL(tm) Version 1.00
JEDEC file for: P16R8
Created on: 26-Aug-86
PAL16R8                                          PAL DEVICE LOGIC EQUATION
82284 PAL Device emulation for the AT
COPYRIGHT 1986 ADVANCED MICRO DEVICES, INC.      Doug Kern 8/13/86*
QP20* QF2048*
L0000
11111111101101111111111011111101  11111111111111111101111011111101
11111111011110111111111011111101  11111111111111111110110111111111
11111111101101111110111111111111  00000000000000000000000000000000
11111111011110111111011111111111  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  11111111111111111101111011111101
00000000000000000000000000000000  11111111111111111110111111111111
11111110111111111111111011111101  00000000000000000000000000000000
11111111111111101110110111111111  00000000000000000000000000000000
11111110111111111110111011111111  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
01110111111111111111111111111111  00000000000000000000000000000000
11111111111111111111111011111111  00000000000000000000000000000000
11111111111111111110111111111111  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  01110111111111111011101111111111
00000000000000000000000000000000  01110111111111111011111110111111
11111101111111111110111011111111  11111111111111111111111111110111
11111111111111110111111011111101  00000000000000000000000000000000
11111111111111101110111111111111  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000*
00000000000000000000000000000000  C4D0B*
00000000000000000000000000000000  C8A0
11111011111111111111110111111111
10111111111111111111110111111111
```

Figure 8. AmPAL284 JEDEC Fuse Map

```
flag '-R2';

title
'PAL16R8                                                    PAL DEVICE LOGIC EQUATION
82288 emulation PAL device for the AT
COPYRIGHT 1986 ADVANCED MICRO DEVICES, INC.                 Doug Kern 8/13/86'

PAL288 device 'P16R8';

"declarations

        TRUE,FALSE = 1,0;
        HIGH,LOW = 1,0;
        X,Z,C = .X.,.Z.,.C.;

        GND,VCC
                pin  10,20;

                CLK,S0,S1,M_IO,CMDEN,RCMD,RDY,Q1,Q0,MB
                pin  1,2,3,4,5,6,7,8,9,11;

                MRDC,MWTC,IORC,IOWC,INTRA,MEM,INT,WCMD
                pin  19,18,17,16,15,14,13,12;

EQUATIONS

        !MRDC       := !MEM & !RCMD & Q1 & !Q0 & CMDEN
                    #  !MEM & !RCMD & !Q1 & !Q0 & CMDEN
                    #  !MEM & !RCMD & !Q1 & Q0 & RDY & CMDEN;

        !MWTC       := !MEM & !WCMD & Q1 & !Q0 & CMDEN
                    #  !MEM & !WCMD & !Q1 & !Q0 & CMDEN
                    # !MEM & !WCMD & !Q1 & Q0 & RDY & CMDEN;

        !IORC       := MEM & INT & !RCMD & Q1 & !Q0 & CMDEN
                    #  MEM & INT & !RCMD & !Q1 & !Q0 & CMDEN
                    #  MEM & INT & !RCMD & !Q1 & Q0 & RDY & CMDEN;

        !IOWC       := MEM & !WCMD & Q1 & !Q0 & CMDEN
                    #  MEM & !WCMD & !Q1 & !Q0 & CMDEN
                    #  MEM & !WCMD & !Q1 & Q0 & RDY & CMDEN;

        !INTRA      := !INT & Q1 & !Q0 & CMDEN
                    # !INT & !Q1 & !Q0 & CMDEN
                    # !INT & !Q1 & Q0 & RDY & CMDEN;


        !MEM        := Q1 & Q0 & M_IO & !S1 & S0
                    #  Q1 & Q0 & M_IO & S1 & !S0
                    # !MEM & Q1 & !Q0
                    # !MEM & !Q1 & !Q0
                    # !MEM & !Q1 & Q0 & RDY;
```

Figure 9. AmPAL288 Source Program Listing

(continued next page)

```
!INT      := Q1 & Q0 & !M_IO & !S1 & !S0
           # !INT & Q1 & !Q0
           # !INT & !Q1 & !Q0
           # !INT & !Q1 & Q0 & RDY;

!WCMD     := Q1 & Q0 & S1 & !S0
           # !WCMD & Q1 & !Q0
           # !WCMD & !Q1 & !Q0
           # !WCMD & !Q1 & Q0 & RDY;
```

Figure 9. AmPAL288 Source Program Listing (continued)

Reduced Equations:

```
MRDC := !((RDY & !RCMD & !Q1 & !MEM & CMDEN
         # !RCMD & !Q0 & !MEM & CMDEN));

MWTC := !((!WCMD & RDY & !Q1 & !MEM & CMDEN
         # !WCMD & !Q0 & !MEM & CMDEN));

IORC  := !((RDY & !RCMD & !Q1 & MEM & INT & CMDEN
          # !RCMD & !Q0 & MEM & INT & CMDEN));

IOWC  := !((!WCMD & RDY & !Q1 & MEM & CMDEN
          # !WCMD & !Q0 & MEM & CMDEN));

INTRA := !((RDY & !Q1 & !INT & CMDEN # !Q0 & !INT & CMDEN));

MEM   := !((!S1 & S0 & Q1 & Q0 & M_IO
         # (S1 & !S0 & Q1 & Q0 & M_IO
         # (RDY & !Q1 & !MEM
         # !Q0 & !MEM))));

INT   := !((!S1 & !S0 & Q1 & Q0 & !M_IO
         # (RDY & !Q1 & !INT
          # !Q0 & !INT)));

WCMD := !((S1 & !S0 & Q1 & Q0
         # (!WCMD & RDY & !Q1
         # !WCMD & !Q0)));
```

Figure 10. AmPAL288 Reduced Equations

```
ABEL(tm) Version 1.00
JEDEC file for: P16R8
Created on: 26-Aug-86
PAL16R8                                          PAL DEVICE LOGIC EQUATION
82288 emulation PAL device for the AT
COPYRIGHT 1986 ADVANCED MICRO DEVICES, INC.                    Doug Kern 8/13/86*
QP20* QF2048*
L0000
11111111111101111011011010111111  00000000000000000000000000000000
11111111111101111011111011111011  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  01111011011111111111111101110111
00000000000000000000000000000000  10110111011111111111111101110111
11111111111101111111011010111110  11111111111111111111011010111111
11111111111101111111111011111010  11111111111111111111111011111011
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  10111011101111111111111101110111
00000000000000000000000000000000  11111111111111111111011110101111
11111111111101111011010110011111  11111111111111111111111111101011
11111111111101111011110111011011  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  10110111111111111111111101110111
00000000000000000000000000000000  11111111111111111111011110111110
11111111111101111111010110111110  11111111111111111111111111111010
11111111111101111111110111111010  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000
00000000000000000000000000000000  00000000000000000000000000000000*
00000000000000000000000000000000  C45A1*
00000000000000000000000000000000  C83F
11111111111101111111011110101111
11111111111101111111111111101011
```

Figure 11. AmPAL288 JEDEC Fuse Map

8080A/Am9080A

8-Bit Microprocessor

DISTINCTIVE CHARACTERISTICS

- High-speed version with 1.3μsec instruction cycle
- Military temperature range operation to 1.5μsec
- Ion-implanted, n-channel, silicon-gate MOS technology
- 3.2mA of output drive at 0.4V (two full TTL loads)
- 700mV of high, 400mV of low level noise immunity
- 820mW maximum power dissipation at ±5% power

GENERAL DESCRIPTION

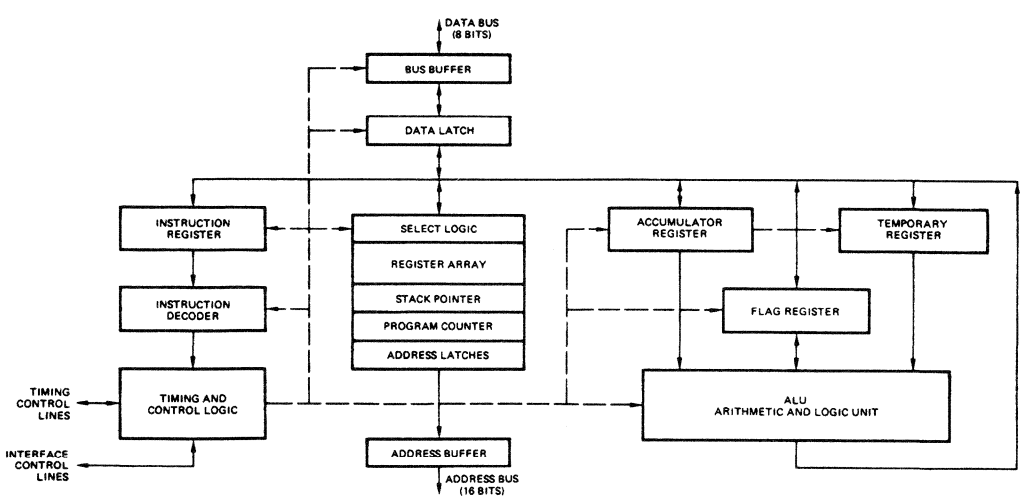
The 8080A products are complete, general-purpose, single-chip digital processors. They are fixed instruction set, parallel, 8-bit units fabricated with Advanced N-Channel Silicon Gate MOS technology. When combined with external memory and peripheral devices, powerful microcomputer systems are formed. The 8080A may be used to perform a wide variety of operations, ranging from complex arithmetic calculations to character handling to bit control. Several versions are available offering a range of performance options.

The processor has a 16-bit address bus that may be used to directly address up to 64K bytes of memory. The memory may be any combination of read/write and read-only. Data are transferred into or out of the processor on a bi-directional 8-bit data bus that is separate from the address lines. The data bus transfers instructions, data and status information between system devices. All transfers are handled using asynchronous handshaking controls so that any speed memory or I/O device is easily accommodated.

An accumulator plus six general registers are available to the programmer. The six registers are each 8 bits long and may be used singly or in pairs for both 8- and 16-bit operations. The accumulator forms the primary working register and is the destination for many of the arithmetic and logic operations.

A general purpose push-down stack is an important part of the processor architecture. The contents of the stack reside in R/W memory and the control logic, including a 16-bit stack pointer, is located on the processor chip. Subroutine call and return instructions automatically use the stack to store and retrieve the contents of the program counter. Push and Pop instructions allow direct use of the stack for storing operands, passing parameters and saving the machine state.

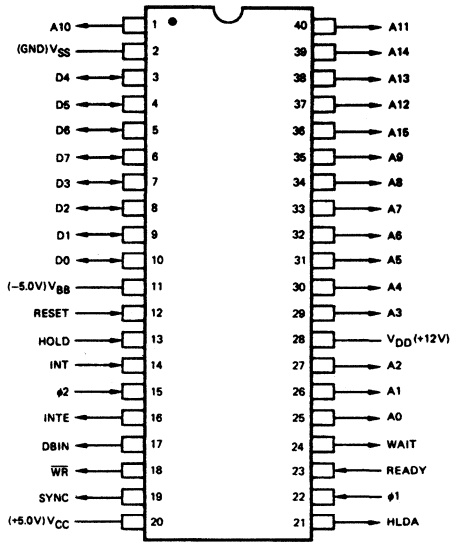
BLOCK DIAGRAM



BD003800

Publication # 04123 | Rev. B | Amendment /0
Issue Date: May 1987

CONNECTION DIAGRAM
Top View
DIPs

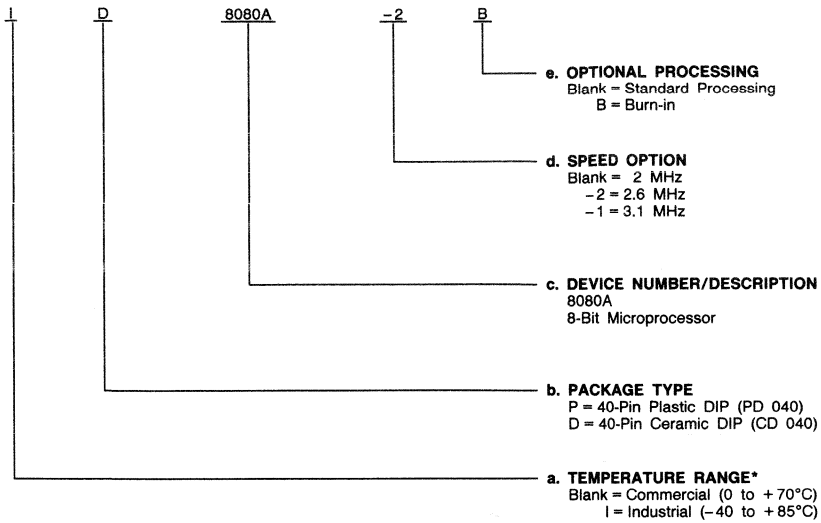


CD005573

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION – 8080A

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D	8080A
	8080A-2
	8080A-1
	8080AB
	8080A-2B
	8080A-1B
ID	8080AB
	8080A-2B

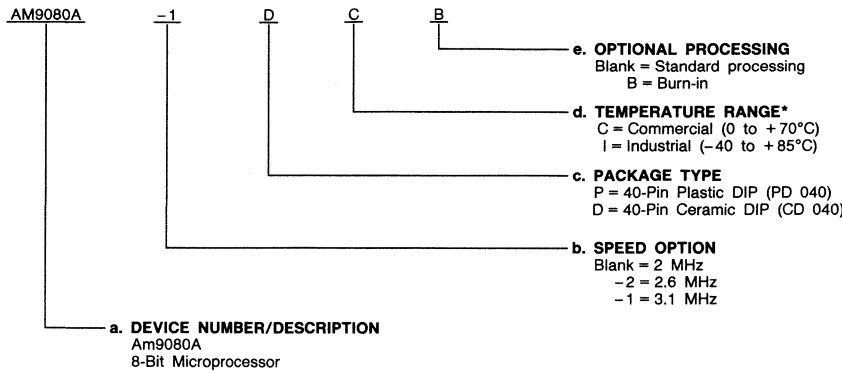
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Handbook (Order #09275A/0) for electrical performance characteristics.

ORDERING INFORMATION – Am9080A

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM9080A	PC, DC,
AM9080A-2	DCB, DIB
AM9080A-1	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Handbook (Order #09275A/0) for electrical performance characteristics.

PIN DESCRIPTION

TYPE	PINS	ABBREVIATION	SIGNAL
INPUT	1	V_{SS}	Ground
INPUT	3	V_{DD}, V_{CC}, V_{BB}	+12V, +5V, −5V Supplies
INPUT	2	ϕ_1, ϕ_2	Clocks
INPUT	1	RESET	Reset
INPUT	1	HOLD	Hold
INPUT	1	INT	Interrupt
INPUT	1	READY	Ready
IN/OUT	8	D_0-D_7	Data Bus
OUTPUT	16	A_0-A_{15}	Address
OUTPUT	1	INTE	Interrupt Enable
OUTPUT	1	DBIN	Data Bus In Control
OUTPUT	1	$\overline{WR}$	Write Not
OUTPUT	1	SYNC	Cycle Synchronization
OUTPUT	1	HLDA	Hold Acknowledge
OUTPUT	1	WAIT	Wait

Pin No.	Names	I/O	Description
22, 15	ϕ_1, ϕ_2	I	The Clock inputs provide basic timing generation for all internal operations. They are non-overlapping two phase, high level signals. All other inputs to the processor are TTL compatible.
12	RESET	I	The Reset input initializes the processor by clearing the program counter, the instruction register, the interrupt enable flip-flop and the hold acknowledge flip-flop. The Reset signal should be active for at least three clock periods. The general registers are not cleared.
13	HOLD	I	The Hold input allows an external signal to cause the processer to relinquish control over the address lines and the data bus. When Hold goes active, the processor completes its current operation, activates the HLDA output, and puts the 3-state address and data lines into their high-impedance state. The Holding device can then utilize the address and data busses without interference.
23	READY	I	The Ready input synchronizes the processor with external units. When Ready is absent, indicating the external operation is not complete, the processor will enter the Wait state. It will remain in the Wait state until the clock cycle, following the appearance of Ready.
14	INT	I	The Interrupt input signal provides a mechanism for external devices to modify the instruction flow of the program in progress. Interrupt requests are handled efficiently with the vectored interrupt procedure and the general purpose stack. Interrupt processing is described in more detail on the next page.
10-7, 3-6	D_0-D_7	I/O	The Data Bus is comprised of 8 bidirectional signal lines for transferring data, instructions and status information between the processor and all external units.
25-27, 29-35, 1, 40, 37-39, 36	A_0-A_{15}	O	The Address Bus is comprised of 16 output signal lines used to address memory and peripheral devices.
19	SYNC	O	The Sync output indicates the start of each processor cycle and the presence of processor status information on the data bus.
17	DBIN	O	The Data Bus In output signal indicates that the bidirectional data bus is in the input mode and incoming data may be gated onto the Data Bus.
24	WAIT	O	The Wait output indicates that the processor has entered the Wait state and is prepared to accept a Ready from the current external operation.
18	$\overline{WR}$	O	The Write output indicates the validity of output on the data bus during a write operation.
21	HLDA	O	The Hold Acknowledge output signal is a response to a Hold input. It indicates that processor activity has been suspended and the Address and Data Bus signals will enter their high-impedance state.
16	INTE	O	The Interrupt Enable output signal shows the status of the interrupt enable flip-flop, indicating whether or not the processor will accept interrupts.

8080A/Am9080A INSTRUCTION SET

The instructions executed by the 8080A are variable length and may be one, two or three bytes long. The length is determined by the nature of the operation being performed and the addressing mode being used.

The instruction summary shows the number of successive memory bytes occupied by each instruction, the number of clock cycles required for the execution of the instruction, the binary coding of the first byte of each instruction, the mnemonic coding used by assemblers and a brief description of each operation. Some branch-type instructions have two execution times depending on whether the conditional branch is taken or not. Some fields in the binary code are labeled with alphabetic abbreviations. That shown as vvv is the address pointer used in the one-byte Call instruction (RST). Those shown as ddd or sss designate destination and source register fields that may be filled as follows:

111 A register
000 B register
001 C register
010 D register
011 E register
100 H register
101 L register
110 Memory

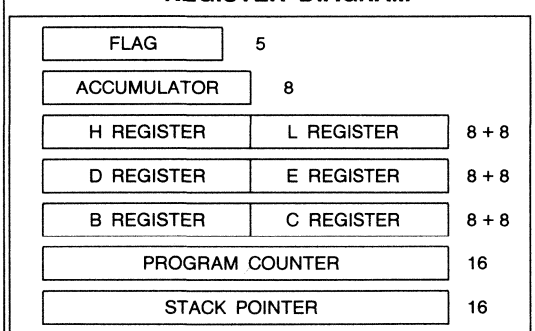
The register diagram shows the internal registers that are directly available to the programmer. The accumulator is the primary working register for the processor and is a specified or implied operand in many instructions. All I/O operations take place via the accumulator. Registers H, L, D, E, B and C may be used singly or in the indicated pairs. The H and L pair is the implied address pointer for many instructions.

The Flag register stores the program status bits used by the conditional branch instructions: carry, zero, sign and parity. The fifth flag bit is the intermediate carry bit. The flags and the accumulator can be stored on or retrieved from the stack with a single instruction. Bit positions in the flag register when pushed onto the stack (PUSH PSW) are:

7	6	5	4	3	2	1	0
S	Z	0	CY1	0	P	1	CY2

Where S = sign, Z = zero, CY1 = intermediate carry, P = parity, CY2 = carry.

REGISTER DIAGRAM



During Sync time at the beginning of each instruction cycle, the data bus contains operation status information that describes the machine cycle being executed. Positions for the status bits are:

7	6	5	4	3	2	1	0
MEMR	INP	M1	OUT	HLTA	STK	$\overline{WO}$	INTA

STATUS DEFINITION:

INTA Interrupt Acknowledge. Occurs in response to an Interrupt input and indicates that the processor will be ready for an interrupt instruction on the data bus when DBIN goes true.

$\overline{WO}$ Write or Output indicated when signal is LOW. When HIGH, a Read or Input will occur.

STK Stack indicates that the content of the stack pointer is on the address bus.

HLTA Halt Acknowledge.

OUT Output instruction is being executed.

M1 First instruction byte is being fetched.

INP Input instruction is being executed.

MEMR Memory Read operation.

INTERRUPT PROCESSING

When the processor interrupt mechanism is enabled (INTE = 1), interrupt signals from external devices will be recognized unless the processor is in the Hold State. In handling an interrupt, the processor will complete the execution of the current instruction, disable further interrupts and respond with INTA status instead of executing the next sequential instruction in the interrupted program.

The interrupting device should supply an instruction opcode to the processor during the next DBIN time after INTA status appears.

Any opcode may be used except XTHL. If the instruction supplied is a single byte instruction, it will be executed. (The usual single byte instruction utilized is RST.) If the interrupt instruction is two or three bytes long, the next one or two processor cycles, as indicated by the DBIN signal, should be used by the external device to supply the succeeding byte(s) of the interrupt instruction. Note that INTA status from the processor is not present during these operations.

If the interrupt instruction is not some form of CALL, it is executed normally by the processor except that the Program Counter is not incremented. The next instruction in the interrupted program is then fetched and executed. Notice that the interrupt mechanism must be re-enabled by the processor before another interrupt can occur.

If the interrupt instruction is some form of CALL, it is executed normally. The Program Counter is stored and control transferred to the interrupt service subroutine. The routine has responsibility for saving and restoring the machine state and for re-enabling interrupts if desired. When the interrupt service is complete, a RETURN instruction will transfer control back to the interrupted program.

INSTRUCTION SET SUMMARY

DATA TRANSFER

Op Code 7|6|5|4|3|2|1|0	No. of Bytes	Clock Cycles	Assembly Mnemonic	Instruction Description
0 1 d d d s s s	1	5	MOVr, r	Move register to register
0 1 1 1 0 s s s	1	7	MOVm, r	Move register to memory
0 1 d d d 1 1 0	1	7	MOVr, m	Move memory to register
0 0 d d d 1 1 0	2	7	MVI, r	Move to register, immediate
0 0 1 1 0 1 1 0	2	10	MVI, m	Move to memory, immediate
0 0 1 1 1 0 1 0	3	13	LDA	Load Acc, direct
0 0 0 0 1 0 1 0	1	7	LDAX B	Load Acc, indirect via B & C
0 0 0 1 1 0 1 0	1	7	LDAX D	Load Acc, indirect via D & E
0 0 1 0 1 0 1 0	3	16	LHLD	Load H & L, direct
0 0 1 0 0 0 0 1	3	10	LXI H	Load H & L, immediate
0 0 0 1 0 0 0 1	3	10	LXI D	Load D & E, immediate
0 0 0 0 0 0 0 1	3	10	LXI B	Load B & C, immediate
0 0 1 1 0 0 0 1	3	10	LXI SP	Load stack pointer, immediate
0 0 1 0 0 0 1 0	3	16	SHLD	Store H&L, direct
0 0 1 1 0 0 1 0	3	13	STA	Store Acc, direct
0 0 0 0 0 0 1 0	1	7	STAX B	Store Acc, indirect via B & C
0 0 0 1 0 0 1 0	1	7	STAX D	Store Acc, indirect via D & E
1 1 1 1 1 0 0 1	1	5	SPHL	Transfer H & L to stack pointer
1 1 1 0 1 0 1 1	1	4	XCHG	Exchange D & E with H & L
1 1 1 0 0 0 1 1	1	18	XTHL	Exchange top of stack with H & L
1 1 0 1 1 0 1 1	2	10	IN	Input to Acc
1 1 0 1 0 0 1 1	2	10	OUT	Output from Acc

CONTROL

0 1 1 1 0 1 1 0	1	7	HLT	Halt and enter wait state
0 0 1 1 0 1 1 1	1	4	STC	Set carry flag
0 0 1 1 1 1 1 1	1	4	CMC	Complement carry flag
1 1 1 1 1 0 1 1	1	4	EI	Enable interrupts
1 1 1 1 0 0 1 1	1	4	DI	Disable interrupts
0 0 0 0 0 0 0 0	1	4	NOP	No operation

BRANCH

1 1 0 0 0 0 1 1	3	10	JMP	Jump unconditionally
1 1 0 1 1 0 1 0	3	10	JC	Jump on carry
1 1 0 1 0 0 1 0	3	10	JNC	Jump on no carry
1 1 0 0 1 0 1 0	3	10	JZ	Jump on zero
1 1 0 0 0 0 1 0	3	10	JNZ	Jump on not zero
1 1 1 1 0 0 1 0	3	10	JP	Jump on positive
1 1 1 1 1 0 1 0	3	10	JM	Jump on minus
1 1 1 0 1 0 1 0	3	10	JPE	Jump on parity even
1 1 1 0 0 0 1 0	3	10	JPO	Jump on parity odd
1 1 0 0 1 1 0 1	3	17	CALL	Call unconditionally
1 1 0 1 1 1 0 0	3	17-11	CC	Call on carry
1 1 0 1 0 1 0 0	3	17-11	CNC	Call on no carry
1 1 0 0 1 1 0 0	3	17-11	CZ	Call on zero
1 1 0 0 0 1 0 0	3	17-11	CNZ	Call on not zero
1 1 1 1 0 1 0 0	3	17-11	CP	Call on positive
1 1 1 1 1 1 0 0	3	17-11	CM	Call on minus
1 1 1 0 1 1 0 0	3	17-11	CPE	Call on parity even
1 1 1 0 0 1 0 0	3	17-11	CPO	Call on parity odd
1 1 0 0 1 0 0 1	1	10	RET	Return unconditionally
1 1 0 1 1 0 0 0	1	11-5	RC	Return on carry
1 1 0 1 0 0 0 0	1	11-5	RNC	Return on no carry
1 1 0 0 1 0 0 0	1	11-5	RZ	Return on zero
1 1 0 0 0 0 0 0	1	11-5	RNZ	Return on not zero
1 1 1 1 0 0 0 0	1	11-5	RP	Return on positive
1 1 1 1 1 0 0 0	1	11-5	RM	Return on minus
1 1 1 0 1 0 0 0	1	11-5	RPE	Return on parity even
1 1 1 0 0 0 0 0	1	11-5	RPO	Return on parity odd
1 1 1 0 1 0 0 1	1	5	PCHL	Jump unconditionally, indirect via H & L
1 1 V V V 1 1 1	1	11	RST	Restart

ARITHMETIC

1 0 0 0 0 s s s	1	4	ADDr	Add register to Acc
1 0 0 0 1 s s s	1	4	ADCr	Add with carry register to Acc
1 0 0 0 0 1 1 0	1	7	ADDm	Add memory to Acc
1 0 0 0 1 1 1 0	1	7	ADCm	Add with carry memory to Acc
1 1 0 0 0 1 1 0	2	7	ADI	Add to Acc, immediate
1 1 0 0 1 1 1 0	2	7	ACI	Add with carry to Acc, immediate
0 0 0 0 1 0 0 1	1	10	DAD B	Double add B & C to H & L
0 0 0 1 1 0 0 1	1	10	DAD D	Double add D & E to H & L
0 0 1 0 1 0 0 1	1	10	DAD H	Double add H & L to H & L
0 0 1 1 1 0 0 1	1	10	DAD SP	Double add stack pointer to H & L
1 0 0 1 0 s s s	1	4	SUBr	Subtract register from Acc
1 0 0 1 1 s s s	1	4	SBBr	Subtract with borrow register from Acc
1 0 0 1 0 1 1 0	1	7	SUBm	Subtract memory from Acc
1 0 0 1 1 1 1 0	1	7	SBBm	Subtract with borrow memory from Acc
1 1 0 1 0 1 1 0	2	7	SUI	Subtract from Acc, immediate
1 1 0 1 1 1 1 0	2	7	SBI	Subtract with borrow from Acc, immediate
0 0 1 0 0 1 1 1	1	4	DAA	Decimal adjust Acc

STACK OPERATIONS

1 1 0 0 0 1 0 1	1	11	PUSH B	Push registers B & C on stack
1 1 0 1 0 1 0 1	1	11	PUSH D	Push registers D & E on stack
1 1 1 0 0 1 0 1	1	11	PUSH H	Push registers H & L on stack
1 1 1 1 0 1 0 1	1	11	PUSH PSW	Push Acc and flags on stack
1 1 0 0 0 0 0 1	1	10	POP B	Pop registers B & C off stack
1 1 0 1 0 0 0 1	1	10	POP D	Pop registers D & E off stack
1 1 1 0 0 0 0 1	1	10	POP H	Pop registers H & L off stack
1 1 1 1 0 0 0 1	1	10	POP PSW	Pop Acc and flags off stack

LOGICAL

1 0 1 0 0 s s s	1	4	ANA r	And register with Acc
1 0 1 0 0 1 1 0	1	7	ANA m	And memory with Acc
1 1 1 0 0 1 1 0	2	7	ANI	And with Acc, immediate
1 0 1 0 1 s s s	1	4	XRA r	Exclusive or register with Acc
1 0 1 0 1 1 1 0	1	7	XRA m	Exclusive Or memory with Acc
1 1 1 0 1 1 1 0	2	7	XRI	Exclusive Or with Acc, immediate
1 0 1 1 0 s s s	1	4	ORA r	Inclusive Or register with Acc
1 0 1 1 0 1 1 0	1	7	ORA m	Inclusive Or memory with Acc
1 1 1 1 0 1 1 0	2	7	ORI	Inclusive Or with Acc, immediate
1 0 1 1 1 s s s	1	4	CMP r	Compare register with Acc
1 0 1 1 1 1 1 0	1	7	CMP m	Compare memory with Acc
1 1 1 1 1 1 1 0	2	7	CPI	Compare with Acc, immediate
0 0 1 0 1 1 1 1	1	4	CMA	Compliment Acc
0 0 0 0 0 1 1 1	1	4	RLC	Rotate Acc left
0 0 0 0 1 1 1 1	1	4	RRC	Rotate Acc right
0 0 0 1 0 1 1 1	1	4	RAL	Rotate Acc left through carry
0 0 0 1 1 1 1 1	1	4	RAR	Rotate Acc right through carry

INCREMENT/DECREMENT

0 0 d d d 1 0 0	1	5	INR r	Increment register
0 0 1 1 0 1 0 0	1	10	INR m	Increment memory
0 0 0 0 0 0 1 1	1	5	INX B	Increment extended B & C
0 0 0 1 0 0 1 1	1	5	INX D	Increment extended D & E
0 0 1 0 0 0 1 1	1	5	INX H	Increment extended H & L
0 0 1 1 0 0 1 1	1	5	INX SP	Increment stack pointer
0 0 d d d 1 0 1	1	5	DCR r	Decrement register
0 0 1 1 0 1 0 1	1	10	DCR m	Decrement memory
0 0 0 0 1 0 1 1	1	5	DCX B	Decrement extended B & C
0 0 0 1 1 0 1 1	1	5	DCX D	Decrement extended D & E
0 0 1 0 1 0 1 1	1	5	DCX H	Decrement extended H & L
0 0 1 1 1 0 1 1	1	5	DCX SP	Decrement stack pointer

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
All Input or Output Voltages
With Respect to V_{BB} ..... −0.3V to +20V
V_{CC}, V_{DD} and V_{SS} With
Respect to V_{BB} ..... −0.3V to +20V
Power Dissipation 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5V ±5%
(V_{BB}) ..... −5V ±5%
(V_{DD}) ..... 12V ±5%
Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5V ±5%
(V_{BB}) ..... −5V ±5%
(V_{DD}) ..... 12V ±5%

Operating ranges define those limits between which the functionality and parameters of the device are guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

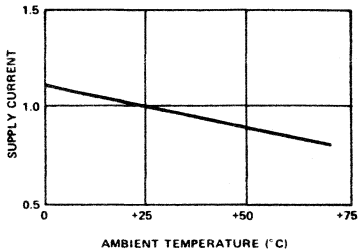
Parameter	Description	Test Conditions	COM'L/IND Min	COM'L/IND Typ	COM'L/IND Max	Units
V_{ILC}	Clock Input Low Voltage		$V_{SS}-1$		$V_{SS}+0.8A$	V
V_{IHC}	Clock Input High Voltage		9.0		$V_{DD}+1$	V
V_{IL}	Input Low Voltage		$V_{SS}-1$		$V_{SS}+0.8$	V
V_{IH}	Input High Voltage		3.3		$V_{CC}+1$	V
V_{OL}	Output Low Voltage	I_{OL} = 1.9mA on all outputs,			0.45	V
V_{OH}	Output High Voltage	I_{OH} = −150μA.	3.7			V
$I_{DD(AV)}$	Avg. Power Supply Current (V_{DD})	operation T_{CY} = .48μsec		40	70	mA
$I_{CC(AV)}$	Avg. Power Supply Current (V_{CC})			60	80	mA
$I_{BB(AV)}$	Avg. Power Supply Current (V_{BB})			0.01	1.0	mA
I_{IL}	Input Leakage	$V_{SS} \leqslant V_{IN} \leqslant V_{CC}$			±10	μA
I_{CL}	Clock Leakage	$V_{SS} \leqslant V_{CLOCK} \leqslant V_{DD}$			±10	μA
I_{DL}	Data Bus Leakage in Input Mode	$V_{SS} \leqslant V_{IN} \leqslant V_{SS}+0.8V$ $V_{SS}+0.8V \leqslant V_{IN} \leqslant V_{CC}$			−100 −2.0	μA mA
I_{FL}	Address and Data Bus Leakage During HOLD	$V_{ADDR/DATA} = V_{CC}$ $V_{ADDR/DATA} = V_{SS}+0.45V$			+10 −100	μA

See Section 6 for Thermal Characteristics Information.

CAPACITANCE (T_A = 25°C, V_{CC} = V_{DD} = V_{SS} = 0V, V_{BB} = −5V)

Parameters	Description	Test Conditions	Typ.	Max.	Units
C_ϕ	Clock Capacitance	f_c = 1 MHz	17	25	pf
C_{IN}	Input Capacitance	Unmeasured Pins	6	10	pf
C_{OUT}	Output Capacitance	Returned to V_{SS}	10	20	pf

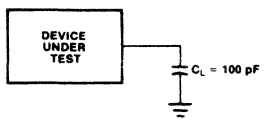
Notes: 1. The RESET signal must be active for a minimum of 3 clock cycles .
2. ΔI supply / ΔT_A = −0.45%/°C.



OP001690

Typical Supply Current vs. Temperature, Normalized [2]

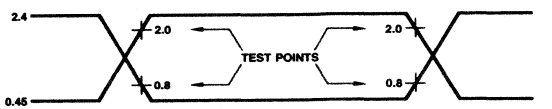
SWITCHING TEST LOAD CIRCUIT



TC001840

C_L = 100pF
C_L INCLUDES JIG CAPACITANCE

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF007450

3

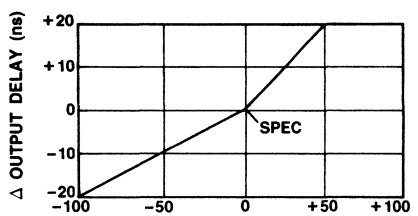
SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	-1 Min	-1 Max	-2 Min	-2 Max	Unit
t_{CY} [3]	Clock Period		0.48	2.0	0.32	2.0	0.38	2.0	μsec
t_r, t_f	Clock Rise and Fall Time		0	50	0	25	0	50	nsec
$t_{\phi 1}$	ϕ_1 Pulse Width		60		50		60		nsec
$t_{\phi 2}$	ϕ_2 Pulse Width		220		145		175		nsec
t_{D1}	Delay ϕ_1 to ϕ_2		0		0		0		nsec
t_{D2}	Delay ϕ_2 to ϕ_1		70		60		70		nsec
t_{D3}	Delay ϕ_1 to ϕ_2 Leading Edges		80		60		70		nsec
t_{DA}	Address Output Delay From ϕ_2	C_L = 100pF		200		150		175	nsec
t_{DD}	Data Output Delay From ϕ_2	C_L = 100pF		200		180		200	nsec
t_{DC}	Signal Output Delay From ϕ_1 or ϕ_2 (SYNC, WR, WAIT, HLDA)	C_L = 50pF		120		110		120	nsec
t_{DF}	DBIN Delay From ϕ_2	C_L = 50pF	25	140	25	130	25	140	nsec
t_{DI} [1]	Delay for Input Bus to Enter Input Mode			t_{DF}		t_{DF}		t_{DF}	nsec
t_{DS1}	Data Set-up Time During ϕ_1 and DBIN		30		10		20		nsec
t_{DS2}	Data Set-up Time to ϕ_2 During DBIN		150		120		130		nsec
t_{DH} [1]	Data Hold time From $\phi 2$ During DBIN		[1]		[1]		[1]		nsec
t_{IE}	INTE Output Delay From ϕ_2	C_L = 50 pF		200		200		200	nsec
t_{RS}	READY Set-up Time During ϕ_2		120		90		90		nsec
t_{HS}	HOLD Set-up Time to ϕ_2		140		120		120		nsec
t_{IS}	INT Set-up Time During ϕ_2		120		100		100		nsec
t_H	Hold Time From ϕ_2 (READY, INT, HOLD)		0		0		0		nsec
t_{FD}	Delay to Float During Hold (Address and Data Bus)			120		120		120	nsec
t_{AW}	Address Stable Prior to WR		[5]		[5]		[5]		nsec
t_{DW}	Output Data Stable Prior to WR		[6]		[6]		[6]		nsec
t_{WD}	Output Data Stable From WR		[7]		[7]		[7]		nsec
t_{WA}	Address Stable From WR	C_L = 100pF: Address, Data C_L = 50pF: $\overline{WR}$, HLDA, DBIN	[7]		[7]		[7]		nsec
t_{HF}	HLDA to Float Delay		[8]		[8]		[8]		nsec
t_{WF}	WR to Float Delay		[9]		[9]		[9]		nsec
t_{AH}	Address Hold Time After DBIN during HLDA		-20		-20		-20		nsec

Notes: (Parenthesis gives -1, -2 specifications, respectively)

1. Data input should be enabled with DBIN status. No bus conflict can then occur and data hold time is assured. t_{DH} = 50 ns or t_{DF}, whichever is less.
2. $t_{CY} = t_{D3} + t_{r\phi 2} + t_{\phi 2} + t_{f\phi 2} + t_{D2} + t_{r\phi 1} \geqslant 480$ ns (-1:320 ns, -2:380 ns).

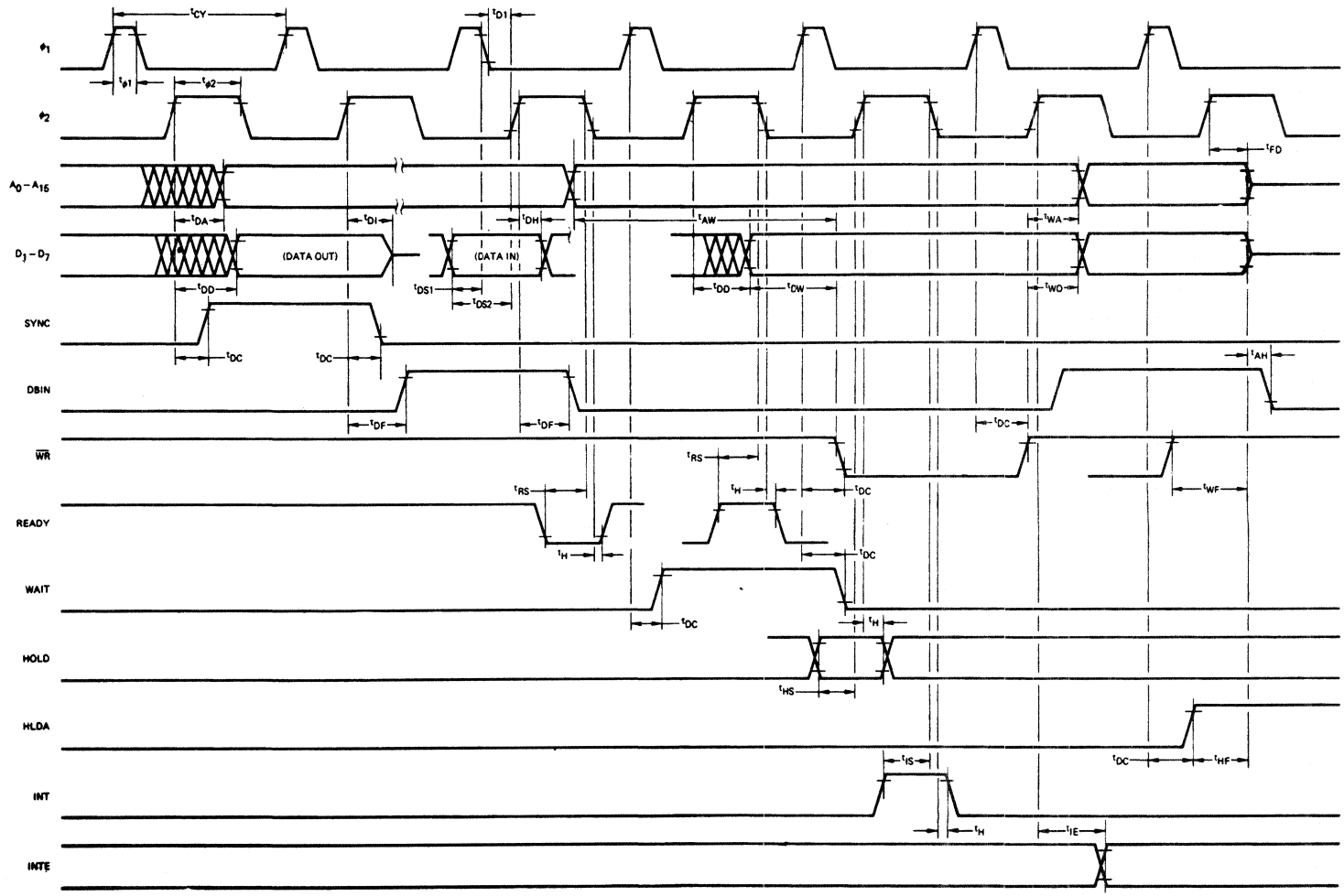
TYPICAL Δ OUTPUT DELAY VS. Δ CAPACITANCE



OP001810

3. The following are relevant when interfacing the 8080A to devices having V_{IH} = 3.3V:
   a) Maximum output rise time from .8V to 3.3V = 100ns @ C_L = SPEC.
   b) Output delay when measured to 3.0V = SPEC + 60ns @ C_L = SPEC.
   c) If C_L = SPEC, add .6ns/pF if $C_L > C_{SPEC}$, subtract .3ns/pF (from modified delay) if $C_L < C_{SPEC}$.
4. $t_{AW} = 2t_{CY} - t_{D3} - t_{r\phi 2} - 140$ ns (-1:110 ns, -2:130 ns).
5. $t_{DW} = t_{CY} - t_{D3} - t_{r\phi 2} - 170$ ns (-1:150 ns, -2:170 ns).
6. If not HLDA, $t_{WD} = t_{WA} = t_{D3} + t_{r\phi 2} + 10$ ns. If HLDA, $t_{WD} = t_{WA} = t_{WF}$.
7. $t_{HF} = t_{D3} + t_{r\phi 2} - 50$ ns).
8. $t_{WF} = t_{D3} + t_{r\phi 2} - 10$ ns.
9. Data in must be stable for this period during DBIN T_3. Both t_{DS1} and t_{DS2} must be satisfied.
10. Ready signal must be stable for this period during T_2 or T_W. (Must be externally synchronized.)
11. Hold signal must be stable for this period during T_2 or T_W when entering hold mode, and during T_3, T_4, T_5 and T_{WH} when in hold mode. (External synchronization is not required.)
12. Interrupt signal must be stable during this period of the last clock cycle of any instruction in order to be recognized on the following instruction. (External synchronization is not required.)
13. This timing diagram shows timing relationships only; it does not represent any specific machine cycle.

SWITCHING WAVEFORMS



WF007240

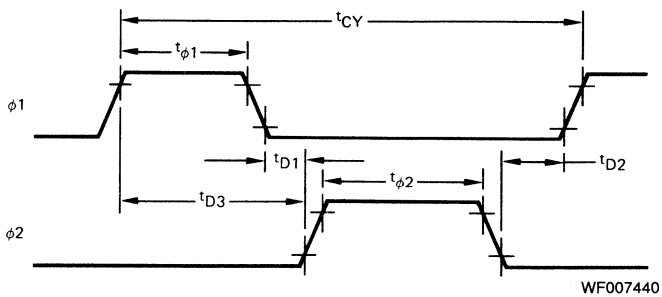
This chart presents relative timing waveform relationships and does not show actual processor operating cycles.

Note: Clock "1" = 8.0V, "0" = 1.0V; Inputs "1" = 3.3V, "0" = 0.8V; Outputs "1" = 2.0V, "0" = 0.8V.

CLOCK SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Am9080A-1, 8080A-1		Am9080A-2, 8080A-2		Am9080A, 8080A		Units
		Min	Max	Min	Max	Min	Max	
t_{CY}	Clock Period	320	2000	380	2000	480	2000	ns
t_r, t_f	Clock Transition Times	0	25	0	50	0	50	ns
$t_{\phi 1}$	Clock $\phi 1$ Pulse Width	50		60		60		ns
$t_{\phi 2}$	Clock $\phi 2$ Pulse Width	145		175		220		ns
t_{D1}	$\phi 1$ to $\phi 2$ Offset	0		0		0		ns
t_{D2}	$\phi 2$ to $\phi 1$ Offset	60		70		70		ns
t_{D3}	$\phi 1$ to $\phi 2$ Delay	60		70		80		ns

CLOCK WAVEFORM DETAIL



$$t_{CY} = t_{D3} + t_{r\phi 2} + t_{\phi 2} + t_{f\phi 2} + t_{D2} + t_{r\phi 1}$$

8085AH

8-Bit Microprocessor

DISTINCTIVE CHARACTERISTICS

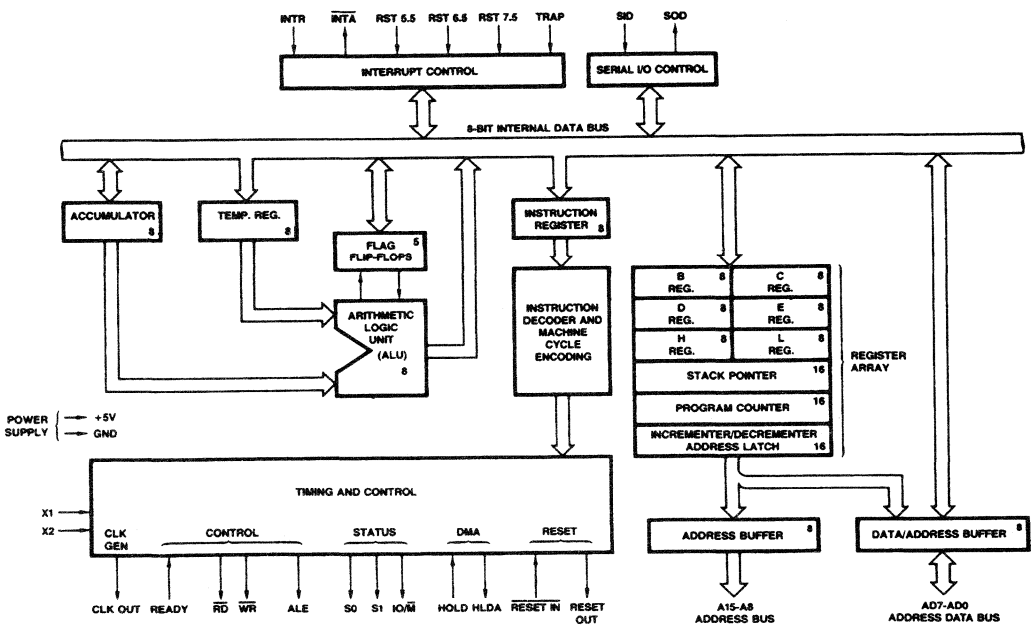
- 3 and 5 MHz selections available
- On-chip system controller; advanced cycle status information available for large system control
- Four vectored interrupts (one is non-maskable)
- On-chip clock generator (with external crystal, LC or R/C network)
- Serial in/serial out port
- Decimal, binary and double precision arithmetic
- Direct addressing capability to 64K bytes of memory
- 1.3μs instruction cycle (8085AH)
- 0.8μs instruction cycle (8085AH-2)
- 100% software compatible with 8080A
- Single +5V power supply

GENERAL DESCRIPTION

The 8085AH is a new generation, complete 8-bit parallel central processing unit (CPU). Its instruction set is 100% software compatible with the 8080A microprocessor. Specifically, the 8085AH incorporates all of the features that the 8224 (clock generator) and 8228 (system controller) provided for the 8080A. The 8085AH-2 is a faster version of the 8085AH. The 8085AH is a 3MHz CPU with 10% supply tolerances and lower power consumption.

The 8085AH uses a multiplexed Data Bus. The address is split between the 8-bit address bus and the 8-bit data bus. The on-chip address latches of 8155H/56H memory products allow a direct interface with 8085AH. The 8085AH components, including various timing compatible support chips, allow system speed optimization.

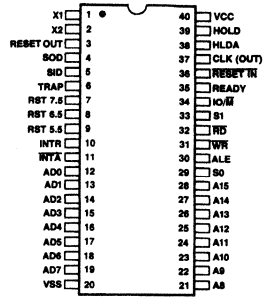
BLOCK DIAGRAM



BD003790

Publication # 04125 | Rev. B | Amendment /0
Issue Date: May 1987

CONNECTION DIAGRAM
Top View
DIPs



CD005564

Note: Pin 1 is marked for orientation.

Figure 1.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Temperature Range
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing

I D 8085AH — B

e. OPTIONAL PROCESSING
Blank = Standard processing
B = Burn-in

d. SPEED OPTION
Blank = 3MHz
-2 = 5MHz

c. DEVICE NUMBER/DESCRIPTION
8085A 8-Bit Microprocessor
8085AH Low-Power High V_{CC} Tolerance 8-Bit Microprocessor

b. PACKAGE TYPE
P = 40-Pin Plastic DIP (PD 040)
D = 40-Pin Ceramic DIP (CD 040)

a. TEMPERATURE RANGE*
Blank = Commercial (0 to +70°C)
I = Industrial (–40 to +85°C)

Valid Combinations	
P, D	8085A
	8085A-2
	8085AH
	8085AH-2
D	8085AB
	8085A-2B
	8085AHB
	8085AH-2B
ID	8085AB
	8085AHB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

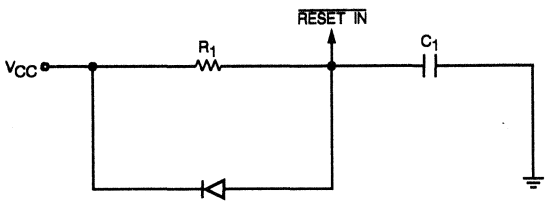
PIN DESCRIPTION

Pin No.	Name	I/O	Description
21-28	A8-A15	O	Address Bus. The most significant eight bits of the memory address or the eight bits of the I/O address, 3-stated during Hold and Halt modes and during RESET.
12-19	AD0-AD7	I/O	Multiplexed Address/Data Bus. Lower eight bits of the memory address (or I/O address), appears on the bus during the first clock cycle of a machine cycle. It then becomes the data bus during the second and third clock cycles. Three-stated during Hold and Halt modes.
30	ALE	O	Address Latch Enable. It occurs during the first clock cycle of a machine cycle and enables the address to get latched into the on-chip latch of peripherals. The falling edge of ALE is set to guarantee set-up and hold times for the address information. The falling edge ALE can also be used to strobe the status information. ALE is never 3-stated.
29, 33	S0, S1, IO/$\overline{M}$	O	Machine Cycle Status: **IO/$\overline{M}$ S1 S0 Status** 0 0 1 Memory write 0 1 0 Memory read 1 0 1 I/O write 1 1 0 I/O read 0 1 1 Opcode fetch 1 1 1 Interrupt Acknowledge * 0 0 Halt * X X Hold * X X Reset * = three-state (high impedance) X = unspecified S1 can be used as an advanced R/$\overline{W}$ status. IO/$\overline{M}$, S_0 and S_1 become valid at the beginning of a machine cycle and remain stable throughout the cycle. The falling edge of ALE may be used to latch the state of these lines.
32	$\overline{RD}$	O	READ. A low level on $\overline{RD}$ indicates the selected memory or I/O device is to be read and that the Data Bus is available for the data transfer. Three-stated during Hold and Halt and during RESET.
31	$\overline{WR}$	O	WRITE. A low level on $\overline{WR}$ indicates the data on the Data Bus is to be written into the selected memory or I/O location. Data is set up at the trailing edge of $\overline{WR}$. Three-stated during Hold and Halt modes.
35	READY	I	If READY is HIGH during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If READY is LOW, the CPU will wait an integral number of clock cycles for READY to go HIGH before completing the read or write cycle.
39	HOLD	I	HOLD. Indicates that another Master is requesting the use of the Address and Data Buses. The CPU, upon receiving the Hold request, will relinquish the use of buses as soon as the completion of the current machine cycle occurs. Internal processing can continue. The processor can regain the buses only after the Hold is removed. When the Hold is acknowledged, the Address, Data, $\overline{RD}$, $\overline{WR}$ and IO/$\overline{M}$ lines are three-stated.
38	HLDA	O	HOLD ACKNOWLEDGE. Indicates that the CPU has received the Hold request and that it will relinquish the buses in the next clock cycle. HLDA goes LOW after the Hold request is removed. The CPU takes the buses one half clock cycle after HLDA goes LOW.
10	INTR	I	INTERRUPT REQUEST. Is used as a general purpose interrupt. It is sampled only during the next to the last clock cycle of the instruction. If it is active, the Program Counter (PC) will be inhibited from incrementing and an $\overline{INTA}$ will be issued. During this cycle a RESTART or CALL instruction can be inserted to jump to the interrupt service routine. The INTR is enabled and disabled by software. It is disabled by Reset and immediately after an interrupt is accepted.
11	$\overline{INTA}$	O	INTERRUPT ACKNOWLEDGE. Is used instead of (and has the same timing as) $\overline{RD}$ during the Instruction cycle after an INTR is accepted. It can be used to activate the Am9519A Interrupt chip or some other interrupt port.
7-9	RST 7.5 RST 6.5 RST 5.5	I	RESTART INTERRUPTS. These three inputs have the same timing as INTR except they cause an internal RESTART to be automatically inserted. RST 7.5 → Highest Priority RST 6.5 RST 5.5 → Lowest Priority The priority of these interrupts is ordered as shown above. These interrupts have a higher priority than the INTR. However, they may be individually masked out using the SIM instructions.
6	TRAP	I	Trap interrupt is a non-maskable restart interrupt. It is recognized at the same time as INTR. It is unaffected by any mask or Interrupt Enable. It has the highest priority of any interrupt.
36	$\overline{RESET\ IN}$	I	Reset sets the Program Counter to zero and resets the Interrupt Enable and HLDA flip-flops. None of the other flags or registers (except the instruction register) is affected. The CPU is held in the reset condition as long as RESET is applied.
3	RESET OUT	O	Indicates CPU is being reset. Can be used as a system RESET. The signal is synchronized to the processor clock.
1, 2	X1, X2	I	Crystal, LC or R/C network connections to set the internal clock generator. X1 can also be an external clock input instead of a crystal. The input frequency is divided by 2 to give the internal operating frequency.
37	CLK	O	Clock Output for use as a system clock when a crystal or R/C network is used as an input to the CPU. The period of CLK is twice the X1, X2 input period.
5	SID	I	Serial input data line. The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed.
4	SOD	O	Serial output data line. The output SOD is set or reset as specified by the SIM instruction.
40	VCC		+5 volt supply.
20	VSS		Ground reference.

Interrupt Priority, Restart Address, and Sensitivity

Name	Priority	Address Branched To (1) When Interrupt Occurs	Type Trigger
TRAP	1	24H	Rising edge AND high level until sampled.
RST 7.5	2	3CH	Rising edge (latched).
RST 6.5	3	34H	High level until sampled.
RST 5.5	4	2CH	High level until sampled.
INTR	5	See Note (2)	High level until sampled.

Notes:
1. The processor pushes the PC on the stack before branching to the indicated address.
2. The address branched to depends on the instruction provided to the cpu when the interrupt is acknowledged.



TYPICAL POWER-ON RESET RC VALUES*
R_1 = 75 KΩ
C_1 = 1 μF

*VALUES MAY HAVE TO VARY DUE TO APPLIED POWER SUPPLY RAMP UP TIME.

TC004230

Power-On Reset Circuit

DETAILED DESCRIPTION

The 8085AH is a complete 8-bit parallel central processor. It is designed with N-channel depletion loads and requires a single +5 volt supply. Its basic clock speed is 3MHz (5MHz: 8085AH-2/6MHz: 8085AH-1), thus improving on the present Am9080's performance with higher system speed. Also, it is designed to fit into a minimum system of three ICs: the CPU, a RAM/IO, and a ROM or PROM/IO chip.

The 8085AH uses a multiplexed Data Bus. The address is split between the higher 8-bit Address Bus and the lower 8-bit Address/Data Bus. During the first cycle, the address is sent out. The lower eight bits are latched into the peripherals by the Address Latch Enable (ALE). During the rest of the machine cycle, the Data Bus is used for memory or I/O data.

The 8085AH provides $\overline{RD}$, $\overline{WR}$ and IO/$\overline{Memory}$ signals for bus control. An Interrupt Acknowledge signal ($\overline{INTA}$) is also provided. Hold, Ready and all Interrupts are synchronized. The 8085AH also provides serial input data (SID) and serial output data (SOD) lines for simple serial interface.

In addition to these features, the 8085AH has three maskable, restart interrupts and one non-maskable trap interrupt.

8085AH vs. 8080A/Am9080A

The 8085AH includes the following features on-chip in addition to all of the Am9080A functions:

a. Internal clock generator
b. Clock output
c. Fully synchronized Ready
d. Schmitt action on $\overline{RESET\ IN}$
e. $\overline{RESET}$ OUT pin
f. $\overline{RD}$, $\overline{WR}$ and IO/$\overline{M}$, S_0, S_1 Bus Control Signals
g. Encoded Status information
h. Multiplexed Address and Data
i. Direct Restarts and non-maskable interrupt
j. Serial Input/Output lines

The internal clock generator requires an external crystal or R/C network. It will oscillate at twice the basic CPU operating frequency. A 50% duty cycle, two-phase, non-overlapping clock is generated from this oscillator internally, and one phase of the clock ($\phi 2$) is available as an external clock. The 8085AH directly provides the external RDY synchronization previously provided by the 8224. The $\overline{RESET\ IN}$ input is provided with a Schmitt action input so that power-on reset only requires a resistor and capacitor. RESET OUT is provided for System RESET.

The 8085AH provides $\overline{RD}$, $\overline{WR}$, S_0, S_1 and IO/$\overline{M}$ signals for Bus control. An $\overline{INTA}$ which was previously provided by the 8228 in 8080A/Am9080A systems is also included in 8085AH.

Status Information

Status information is directly available from the 8085AH. ALE serves as a status strobe. The status is partially encoded and provides the user with advanced timing of the type of bus transfer being done. IO/$\overline{M}$ cycle status signal is provided directly also. Decoded S0, S1 carries the following status information:

MACHINE CYCLE STATUS

IO/$\overline{M}$	S1	S0	STATUS
0	0	1	Memory write
0	1	0	Memory read
1	0	1	I/O write
1	1	0	I/O read
0	1	1	Opcode fetch
1	1	1	Interrupt Acknowledge
*	0	0	Halt
*	X	X	Hold
*	X	X	Reset

* = 3-state (high-impedance)
X = unspecified

S1 can be interpreted as R/$\overline{W}$ in all bus transfers.

In the 8085AH the eight LSB of address are multiplexed with the data instead of status. The ALE line is used as a strobe to enter the lower half of the address into the memory or peripheral address latch. This also frees extra pins for expanded interrupt capability.

Interrupt and Serial I/O

The 8085AH/8085AH-2 has 5 interrupt inputs: INTR, RST 5.5, RST 6.5, RST 7.5 and TRAP. INTR is identical in function to the 8080A INT. Each of three RESTART inputs, 5.5, 6.5, 7.5, has a programmable mask. TRAP is also a RESTART interrupt except it is non-maskable.

The three RESTART interrupts cause the internal execution of RST (saving the program counter in the stack and branching to the RESTART address) if the interrupts are enabled and if the interrupt mask is not set. The nonmaskable TRAP causes the internal execution of a RST independent of the state of the interrupt enable or masks.

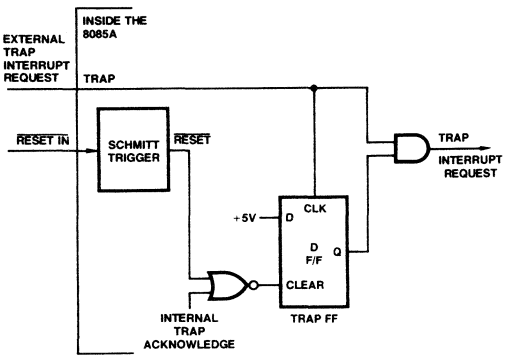
Name	RESTART Address (Hex)
TRAP	24_{16}
RST 5.5	$2C_{16}$
RST 6.5	34_{16}
RST 7.5	$3C_{16}$

There are two different types of inputs in the restart interrupts. RST 5.5 and RST 6.5 are high level-sensitive like INTR (and INT on the 8080A) and are recognized with the same timing as INTR. RST 7.5 is rising edge-sensitive. For RST 7.5, only a pulse is required to set an internal flip-flop which generates the internal interrupt request. The RST 7.5 request flip-flop remains set until the request is serviced. Then it is reset automatically. This flip-flop may also be reset by using the SIM instruction or by issuing a RESET IN to the 8085AH. The RST 7.5 internal flip-flop will be set by a pulse on the RST 7.5 pin even when the RST 7.5 interrupt is masked out.

The status of the three RST interrupt masks can only be affected by the SIM instruction and RESET IN.

The interrupts are arranged in a fixed priority (that determines which interrupt is to be recognized if more than one is pending) as follows: TRAP - highest priority, RST 7.5, RST 6.5, RST 5.5, INTR – lowest priority. This priority scheme does not take into account the priority of a routine that was started by a higher priority interrupt. RST 5.5 can interrupt a RST 7.5 routine if the interrupts were re-enabled before the end of the RST 7.5 routine.

The TRAP interrupt is useful for catastrophic errors, such as power failure or bus error. The TRAP input is recognized just as any other interrupt but has the highest priority. It is not affected by any flag or mask. The TRAP input is both edge- and level-sensitive. The TRAP input must go HIGH and remain HIGH to be acknowledged, but will not be recognized again until it goes LOW, then HIGH again. This avoids any false triggering due to noise or logic glitches. The following diagram illustrates the TRAP interrupt request circuitry within the 8085AH.



AF003070

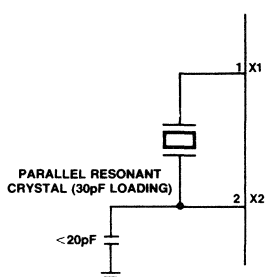
Note that the servicing of any interrupt (TRAP, RST 7.5, RST 6.5, RST 5.5, INTR) disables all future interrupts (except TRAPs) until an EI instruction is executed.

The TRAP interrupt is special in that it preserves the previous interrupt enable status. Performing the first RIM instruction following a TRAP interrupt allows you to determine whether interrupts were enabled or disabled prior to the TRAP. All subsequent RIM instructions provide current interrupt enable status.

The serial I/O system is also controlled by the RIM and SIM instructions. SID is read by RIM, and SIM sets the SOD data.

Driving the X1 and X2 Inputs

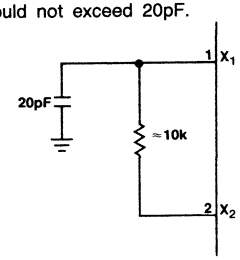
The user may drive the X1 and X2 inputs of the 8085AH or 8085AH-2 with a crystal, an external clock source or an R/C network as shown below. The driving frequency must be twice the desired internal operating frequency (the 8085AH would require a 6MHz crystal for 3MHz internal operation).



TC001960

1-6MHz Input Frequency

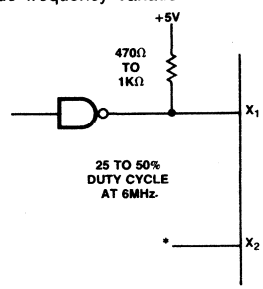
The 20pF capacitor is required to guarantee oscillation at the proper frequency during system start-up. Capacitance from X_2 to Ground should not exceed 20pF.



TC001980

≈3 MHz Input Frequency

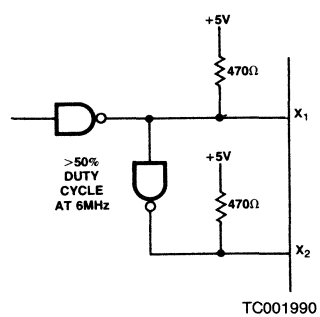
RC Mode causes a large drift in clock frequency because of the variation in on-chip timing generation parameters. Use of RC Mode should be limited to an application which can tolerate a wide frequency variation.



*X_2 Left Floating

TC001970

1-6 MHz Input Frequency



TC001990

≈6 MHz Input Frequency

Note: Duty cycle refers to the percentage of the clock input cycle when X_1 is high.

Figure 2. Driving the Clock Inputs (X1 and X2) of 8085AH

Generating 8085AH Wait State

The following circuit may be used to insert one WAIT state in each 8085AH machine cycle.

The D flip-flops should be chosen such that

- CLK is rising edge-triggered and
- CLEAR is low-level active.

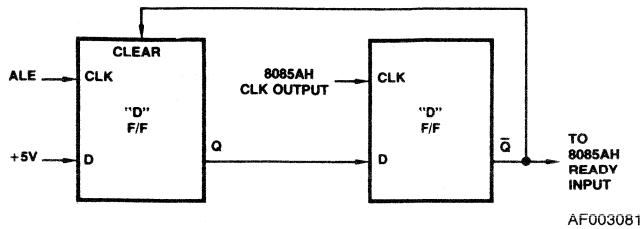


Figure 3. Generation of a Wait State for 8085AH CPU

Basic System Timing

The 8085AH has a multiplexed Data Bus. ALE is used as a strobe to sample the lower 8 bits of address on the Data Bus. Figure 4 shows an instruction fetch, memory read and I/O write cycle (OUT). Note that during the I/O write and read cycle that the I/O port address is copied on both the upper and lower half of the address.

As in the Am9080A, the READY line is used to extend the read and write pulse lengths so that the 8085AH can be used with slow memory. Hold causes the CPU to relinquish the bus when it is through with it by floating the Address and Data Buses.

Table 1. 8085AH Machine Cycle Chart

MACHINE CYCLE		STATUS			CONTROL		
		IO/$\overline{M}$	S1	S0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
OPCODE FETCH	(OF)	0	1	1	0	1	1
MEMORY READ	(MR)	0	1	0	0	1	1
MEMORY WRITE	(MW)	0	0	1	1	0	1
I/O READ	(IOR)	1	1	0	0	1	1
I/O WRITE	(IOW)	1	0	1	1	0	1
ACKNOWLEDGE OF INTR	(INA)	1	1	1	1	1	0
BUS IDLE	(BI): DAD	0	1	0	1	1	1
	ACK. OF RST, TRAP	1	1	1	1	1	1
	HALT	TS	0	0	TS	TS	1

Table 2. 8085AH Machine State Chart

Machine State	Status & Buses				Control		
	S1, S0	IO/$\overline{M}$	A_8–A_{15}	AD_0–AD_7	$\overline{RD}$, $\overline{WR}$	$\overline{INTA}$	ALE
T_1	X	X	X	X	1	1	1*
T_2	X	X	X	X	X	X	0
T_{WAIT}	X	X	X	X	X	X	0
T_3	X	X	X	X	X	X	0
T_4	1	0†	X	TS	1	1	0
T_5	1	0†	X	TS	1	1	0
T_6	1	0†	X	TS	1	1	0
T_{RESET}	X	TS	TS	TS	TS	1	0
T_{HALT}	0	TS	TS	TS	TS	1	0
T_{HOLD}	X	TS	TS	TS	TS	1	0

0 = Logic "0" TS = High Impedance
1 = Logic "1" X = Unspecified

*ALE not generated during 2nd and 3rd machine cycles of DAD instruction.
† IO/$\overline{M}$ = 1 during $T_4 - T_6$ of INA machine cycle.

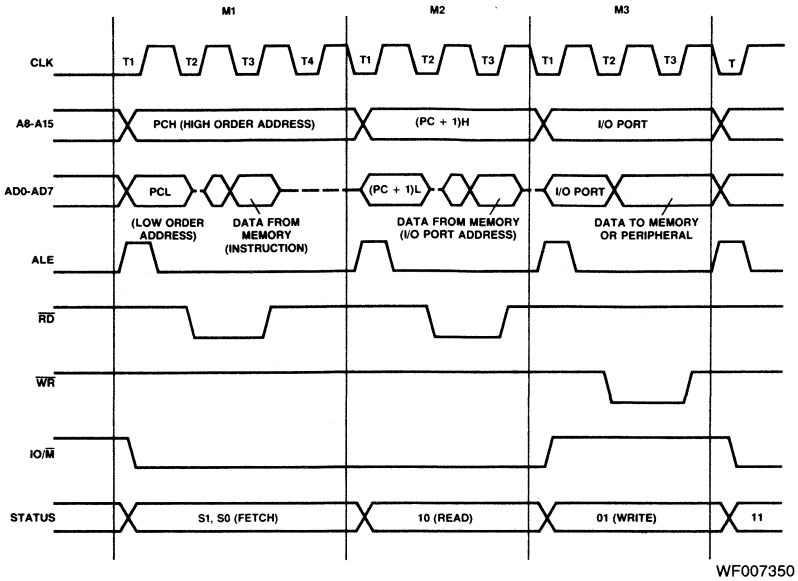


Figure 4. 8085AH Basic System Timing

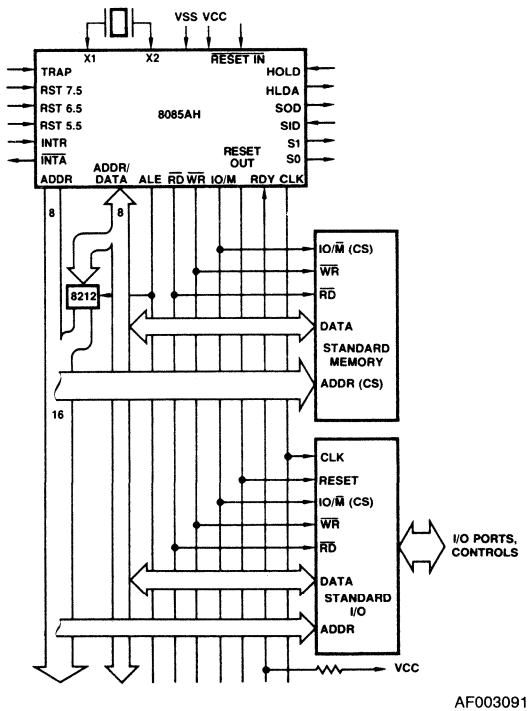


Figure 5. System Using Standard Memories

8085AH INSTRUCTION SET SUMMARY

Mnemonic*	Description	Instruction Code (Note 1) D7	D6	D5	D4	D3	D2	D1	D0	Clock Cycles (Note 2)
MOVE, LOAD AND STORE										
MOVr1r2	Move register to register	0	1	D	D	D	S	S	S	4
MOV Mr	Move register to memory	0	1	1	1	0	S	S	S	7
MOV rM	Move memory to register	0	1	D	D	D	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
STACK OPS										
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	12
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	12
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	12
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	12
POP B	Pop register Pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register Pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register Pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
XTHL	Exchange top of stack H & L	1	1	1	0	0	0	1	1	16
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	6
JUMP										
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	7/10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	7/10
JZ	Jump on zero	1	1	0	0	1	0	1	0	7/10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	7/10
JP	Jump on positive	1	1	1	1	0	0	1	0	7/10
JM	Jump on minus	1	1	1	1	1	0	1	0	7/10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	7/10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	7/10
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	6
CALL										
CALL	Call unconditional	1	1	0	0	1	1	0	1	18
CC	Call on carry	1	1	0	1	1	1	0	0	9/18
CNC	Call on no carry	1	1	0	1	0	1	0	0	9/18
CZ	Call on zero	1	1	0	0	1	1	0	0	9/18
CNZ	Call on no zero	1	1	0	0	0	1	0	0	9/18
CP	Call on positive	1	1	1	1	0	1	0	0	9/18
CM	Call on minus	1	1	1	1	1	1	0	0	9/18
CPE	Call on parity even	1	1	1	0	1	1	0	0	9/18
CPO	Call on parity odd	1	1	1	0	0	1	0	0	9/18
RETURN										
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	6/12
RNC	Return on no carry	1	1	0	1	0	0	0	0	6/12
RZ	Return on zero	1	1	0	0	1	0	0	0	6/12
RNZ	Return on no zero	1	1	0	0	0	0	0	0	6/12
RP	Return on positive	1	1	1	1	0	0	0	0	6/12
RM	Return on minus	1	1	1	1	1	0	0	0	6/12
RPE	Return on parity even	1	1	1	0	1	0	0	0	6/12
RPO	Return on parity odd	1	1	1	0	0	0	0	0	6/12
RESTART										
RST	Restart	1	1	A	A	A	1	1	1	12
INPUT/OUTPUT										
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10

8085AH INSTRUCTION SET SUMMARY (Cont'd.)

Mnemonic*	Description	Instruction Code (Note 1) D7	D6	D5	D4	D3	D2	D1	D0	Clock Cycles (Note 2)
INCREMENT AND DECREMENT										
INR r	Increment register	0	0	D	D	D	1	0	0	4
DCR r	Decrement register	0	0	D	D	D	1	0	1	4
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	6
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	6
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	6
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	6
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	6
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	6
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	6
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	6
ADD										
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
ADD M	Add memory to A	1	0	C	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
SUBTRACT										
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
LOGICAL										
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or Immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
ROTATE										
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
SPECIALS										
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
CONTROL										
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable Interrupts	1	1	1	1	0	0	1	1	4
NOP	No operation	0	0	0	0	0	0	0	0	4
HLT	Halt	0	1	1	1	0	1	1	0	5
NEW 8085AH INSTRUCTIONS										
RIM	Read Interrupt Mask	0	0	1	0	0	0	0	0	4
SIM	Set Interrupt Mask	0	0	1	1	0	0	0	0	4

Notes: 1. DDD or SSS: B = 000, C = 001, D = 010, E = 011, H = 100, L = 101, Memory = 110, A = 111.
2. Two possible cycle times (6/12) indicate instruction cycles dependent on condition flags.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin
With Respect to Ground ..... −0.5V to +7V
Power Dissipation 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC})
8085A, 8085A-2 ..... 5V ±5%
8085AH, 8085AH-2 ..... 5V ±10%
Supply Current (I_{CC})
8085A, 8085A-2 ..... 170 mA
8085AH, 8085AH-2 ..... 135 mA

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5V ±10%
Supply Current (I_{CC}) ..... 200 mA

Operating ranges define those limits between which the functionality of the device is guaranteed.

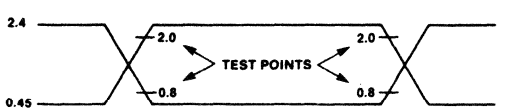
DC CHARACTERISTICS (8085A, 8085A-2) over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
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	I_{OL} = 2mA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400μA	2.4		V
I_{CC}	Power Supply Current			170	mA
I_{IL}	Input Leakage	$0 \leqslant V_{IN} \leqslant V_{CC}$		±10	μA
I_{LO}	Output Leakage	$0.45V \leqslant V_{out} \leqslant V_{CC}$		±10	μA
V_{ILR}	Input Low Level, RESET		−0.5	+0.8	V
V_{IHR}	Input High Level, RESET		2.4	V_{CC} +0.5	V
V_{HY}	Hysteresis, RESET		0.15		V

DC CHARACTERISTICS (8085AH, 8085AH-2) over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
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	I_{OL} = 2mA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400μA	2.4		V
I_{CC}	Power Supply Current	8085AH, 8085AH-2		135	mA
I_{IL}	Input Leakage	$0 \leqslant V_{IN} \leqslant V_{CC}$		±10	μA
I_{LO}	Output Leakage	$0.45V \leqslant V_{OUT} \leqslant V_{CC}$		±10	μA
V_{ILR}	Input Low Level, RESET		−0.5	+0.8	V
V_{IHR}	Input High Level, RESET		2.4	V_{CC} +0.5	V
V_{HY}	Hysteresis, RESET		0.15		V

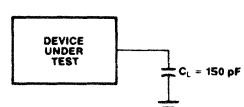
SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF009540

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."

SWITCHING TEST LOAD CIRCUIT



TC001841

C_L = 150pF
C_L INCLUDES JIG CAPACITANCE

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	8085A[2], AH		8085A-2[2], AH-2		Units
		Min	Max	Min	Max	
t_{CYC}	CLK Cycle Period	320	2000	200	2000	ns
t_1	CLK Low Time (Standard CLK Loading)	80		40		ns
t_2	CLK High Time (Standard CLK Loading)	120		70		ns
t_r, t_f	CLK Rise and Fall Time		30		30	ns
t_{XKR}	X_1 Rising to CLK Rising	20	120	20	100	ns
t_{XKF}	X_1 Rising to CLK Falling	20	150	20	110	ns
t_{AC}	A_{8-15} Valid to Leading Edge of Control[1]	270		115		ns
t_{ACL}	A_{0-7} Valid to Leading Edge of Control	240		115		ns
t_{AD}	A_{0-15} Valid to Valid Data In		575		350	ns
t_{AFR}	Address Float After Leading Edge of $\overline{READ}$ ($\overline{INTA}$)		0		0	ns
t_{AL}	A_{8-15} Valid Before Trailing Edge of ALE[1]	115		50		ns
t_{ALL}	A_{0-7} Valid Before Trailing Edge of ALE	90		50		ns
t_{ARY}	READY Valid from Address Valid		220		100	ns
t_{CA}	Address (A_{8-15}) Valid After Control	120		60		ns
t_{CC}	Width of Control Low ($\overline{RD}$, $\overline{WR}$, $\overline{INTA}$) Edge of ALE	400		230		ns
t_{CL}	Trailing Edge of Control to Leading Edge of ALE	50		25		ns
t_{DW}	Data Valid to Trailing Edge of $\overline{WRITE}$	420		230		ns
t_{HABE}	HLDA to Bus Enable		210		150	ns
t_{HABF}	Bus Float After HLDA		210		150	ns
t_{HACK}	HLDA Valid to Trailing Edge of CLK	110		40		ns
t_{HDH}	HOLD Hold Time	0		0		ns
t_{HDS}	HOLD Set-up Time to Trailing Edge of CLK	170		120		ns
t_{INH}	INTR Hold Time	0		0		ns
t_{INS}	INTR, RST, and TRAP Set-up Time to Falling Edge of CLK	160		150		ns
t_{LA}	Address Hold Time After ALE	100		50		ns
t_{LC}	Trailing Edge of ALE to Leading Edge of Control	130		60		ns
t_{LCK}	ALE Low During CLK High	100		50		ns
t_{LDR}	ALE to Valid Data During Read		460		270	ns
t_{LDW}	ALE to Valid Data During Write		200		140	ns
t_{LL}	ALE Width	140		80		ns
t_{LRY}	ALE to READY Stable		110		30	ns
t_{RAE}	Trailing Edge of $\overline{READ}$ to Re-Enabling of Address	150		90		ns
t_{RD}	$\overline{READ}$ (or $\overline{INTA}$) to Valid Data		300		150	ns
t_{RV}	Control Trailing Edge to Leading Edge of Next Control	400		220		ns
t_{RDH}	Data Hold Time After $\overline{READ}$ $\overline{INTA}$[7]	0		0		ns
t_{RYH}	READY Hold Time	0		0		ns
t_{RYS}	READY Set-up Time to Leading Edge of CLK	110		100		ns
t_{WD}	Data Valid After Trailing Edge of $\overline{WRITE}$	100		60		ns
t_{WDL}	LEADING Edge of WRITE to Data Valid		40		20	ns

Notes: 1. $A_8 - A_{15}$ address Specs apply to IO/$\overline{M}$, S_0, and S_1, except $A_8 - A_{15}$ are undefined during $T_4 - T_6$ of OF cycle; whereas, IO/$\overline{M}$, S_0, and S_1 are stable.

2. Test conditions: t_{CYC} = 320ns (8085A)/200ns (8085A-2); C_L = 150pF.

3. For all output timing where C_L = 150pF use the following correction factors:
   25pF $\leq C_L <$ 150pF: −0.10ns/pF
   150pF $< C_L \leq$ 300pF: +0.30ns/pF

4. Output timings are measured with purely capacitive load.

5. All timings are measured at output voltage V_L = 0.8V, V_H = 2.0V, and 1.5V with 20ns rise and fall time on inputs.

6. To calculate timing specifications at other values of t_{CYC} use Table 7.

7. Data hold time is guaranteed under all loading conditions.

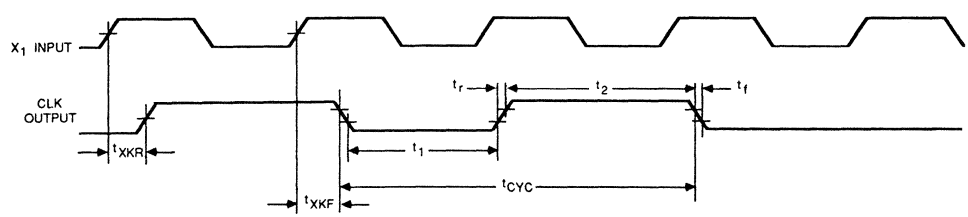
Table 3. Bus Timing Specification as a T_{CYC} Dependent

Symbol	8085AH, 8085A	8085AH-2, 8085A-2	
t_{AL}	(1/2) T – 45	(1/2) T – 50	Minimum
t_{LA}	(1/2) T – 60	(1/2) T – 50	Minimum
t_{LL}	(1/2) T – 20	(1/2) T – 20	Minimum
t_{LCK}	(1/2) T – 60	(1/2) T – 50	Minimum
t_{LC}	(1/2) T – 30	(1/2) T – 40	Minimum
t_{AD}	(5/2 + N) T – 225	(5/2 + N) T – 150	Maximum
t_{RD}	(3/2 + N) T – 180	(3/2 + N) T – 150	Maximum
t_{RAE}	(1/2) T – 10	(1/2) T – 10	Minimum
t_{CA}	(1/2) T – 40	(1/2) T – 40	Minimum
t_{DW}	(3/2 + N) T – 60	(3/2 + N) T – 70	Minimum
t_{WD}	(1/2) T – 60	(1/2) T – 40	Minimum
t_{CC}	(3/2 + N) T – 80	(3/2 + N) T – 70	Minimum
t_{CL}	(1/2) T – 110	(1/2) T – 75	Minimum
t_{ARY}	(3/2) T – 260	(3/2) T – 200	Maximum
t_{HACK}	(1/2) T – 50	(1/2) T – 60	Minimum
t_{HABF}	(1/2) T + 50	(1/2) T + 50	Maximum
t_{HABE}	(1/2) T + 50	(1/2) T + 50	Maximum
t_{AC}	(2/2) T – 50	(2/2) T – 85	Minimum
t_1	(1/2) T – 80	(1/2) T – 60	Minimum
t_2	(1/2) T – 40	(1/2) T – 30	Minimum
t_{RV}	(3/2) T – 80	(3/2) T – 80	Minimum
t_{LDR}	(4/2) T – 180	(4/2) T – 130	Maximum

Note: N is equal to the total WAIT states. $T = t_{CYC}$.

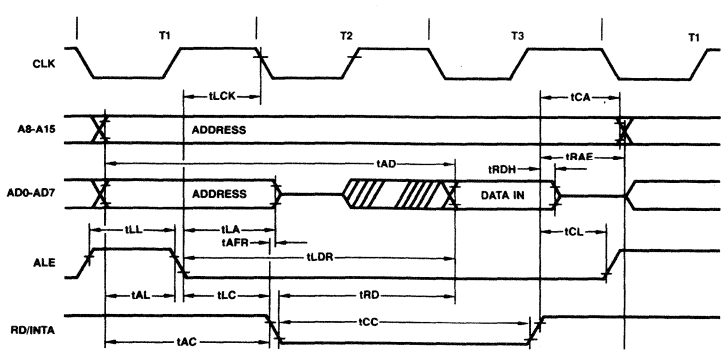
SWITCHING WAVEFORMS

CLOCK



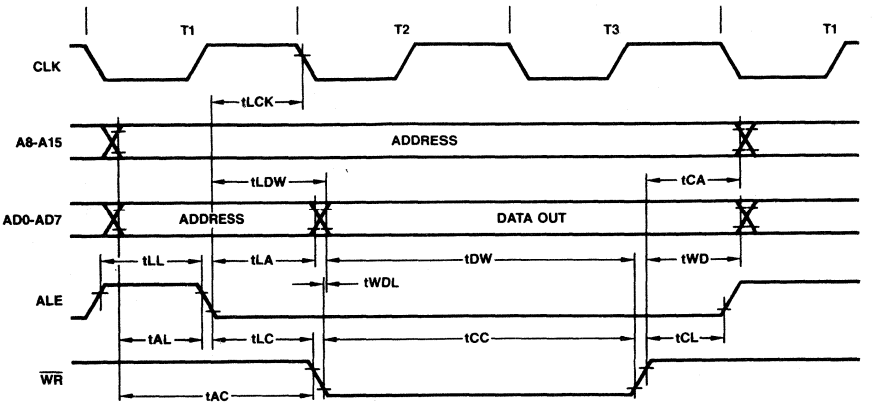
WF024450

READ OPERATION

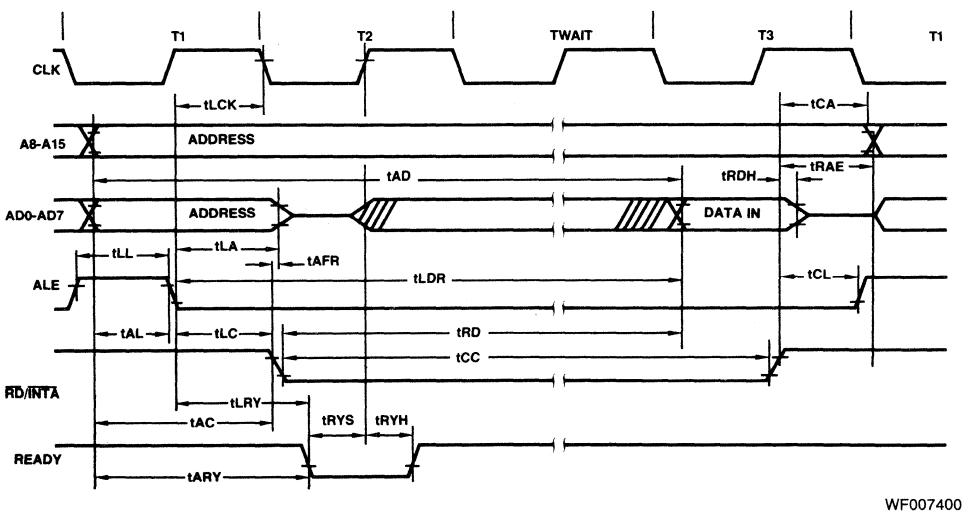


WF007380

WRITE OPERATION



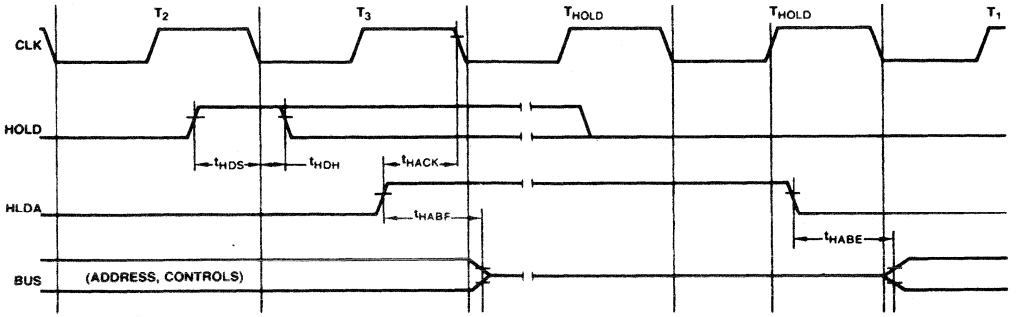
TYPICAL READ OPERATION WITH WAIT CYCLE



Same READY timing applies to WRITE operation.

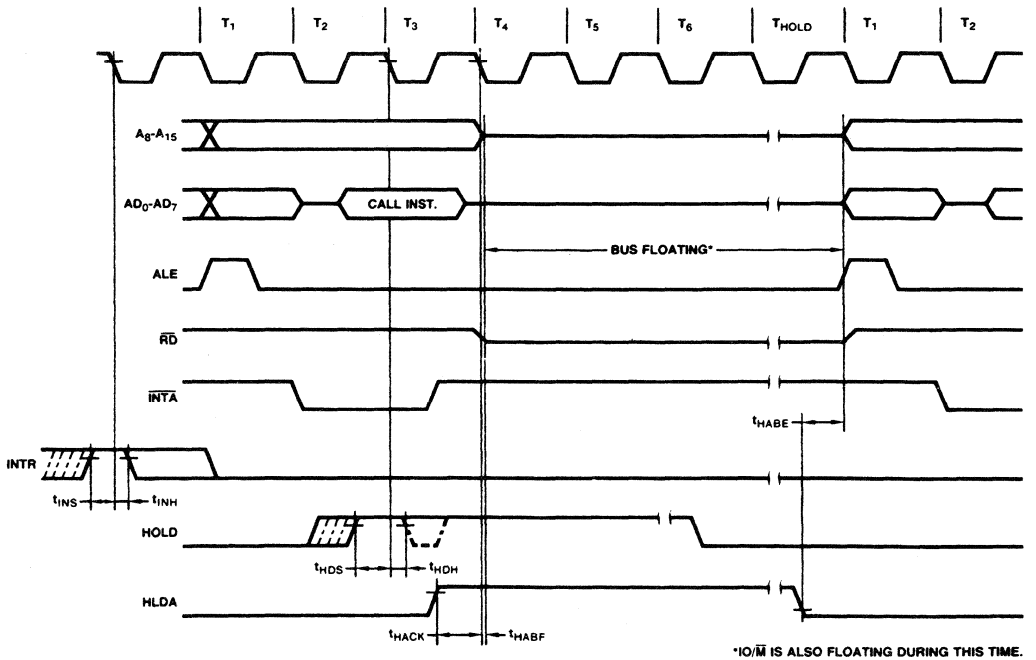
Figure 6. 8085AH/8085AH-2 Bus Timing

HOLD OPERATION



WF007410

Figure 7. 8085AH Hold Timing



WF007420

Figure 8. 8085AH Interrupt and Hold Timing

8086

16-Bit Microprocessor
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Directly addresses up to 1 Mbyte of memory
- 24 operand addressing modes
- Efficient implementation of high level languages
- Instruction set compatible with 8080 software
- Bit, byte, word, and block operations
- 8 and 16-bit signed and unsigned arithmetic in binary or decimal
- Multibus* system interface
- Three speed options
  - 5MHz for 8086
  - 8MHz for 8086-2
  - 10MHz for 8086-1

GENERAL DESCRIPTION

The 8086 is a general purpose 16-bit microprocessor CPU. Its architecture is built around thirteen 16-bit registers and nine 1-bit flags. The CPU operates on 16-bit address spaces and can directly address up to 1 megabyte using offset addresses within four distinct memory segments, designated as code, data, stack and extra code. The 8086 implements a powerful instruction set with 24 operand addressing modes. This instruction set is compatible with that of the 8080 and 8085. In addition, the 8086 is particularly effective in executing high level languages.

The 8086 can operate in minimum and maximum modes. Maximum mode offloads certain bus control functions to a peripheral device and allows the CPU to operate efficiently in a multi-processor system. The CPU and its high performance peripherals are MULTIBUS* compatible. The 8086 is implemented in N-channel, depletion load, silicon gate technology and is contained in a 40-pin CERDIP package, Molded DIP package, or Plastic Leaded Chip Carrier.

BLOCK DIAGRAM

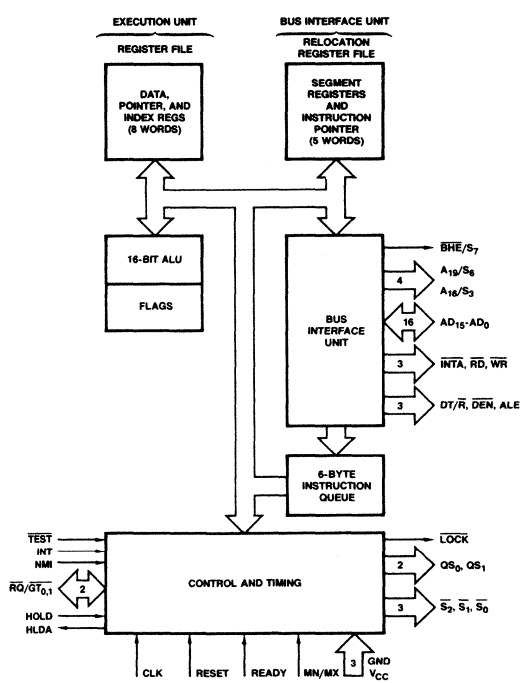


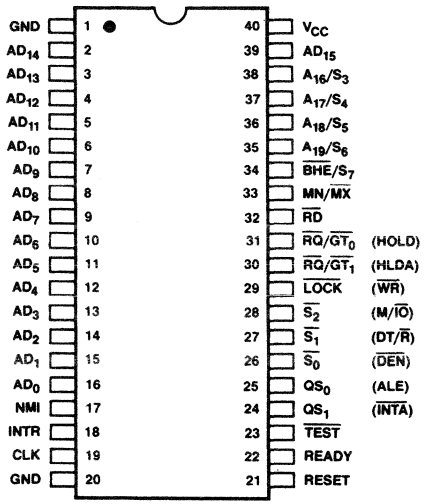
Figure 1.

MULTIBUS is a registered trademark of Intel Corp.

Publication # 01966 | Rev. C | Amendment /0
Issue Date: May 1987

CONNECTION DIAGRAMS
Top View

DIPs

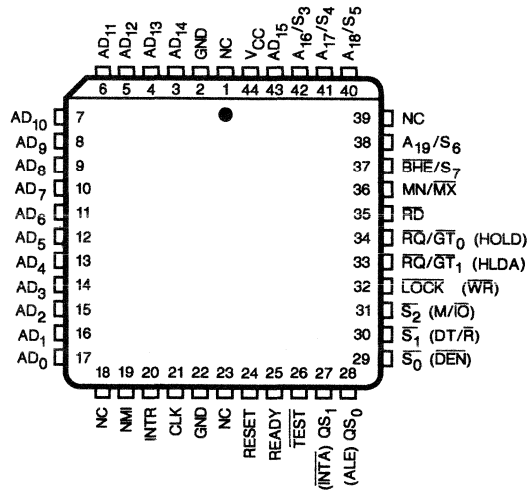


CD005511

Note: Pin 1 is marked for orientation.

Figure 2.1

PLCC



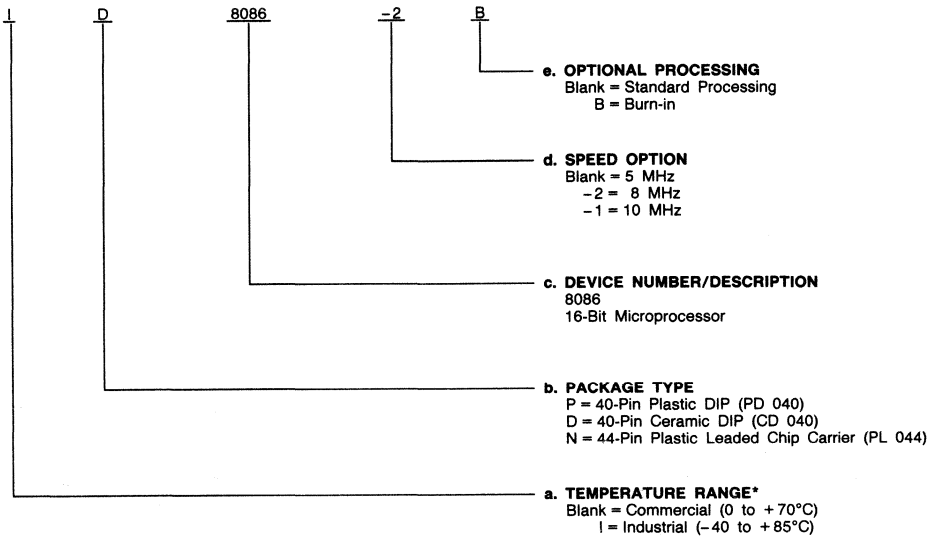
CD010701

Figure 2.2

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D, N	8086
	8086-2
	8086-1
D, ID	8086-2B
D	8086-1B

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

PIN DESCRIPTION

The following pin function descriptions are for 8086 systems in either minimum or maximum mode. The "Local Bus" in these descriptions is the direct multiplexed bus interface connection to the 8086 (without regard to additional bus buffers).

Pin No.*	Name	I/O	Description
39, 2-16	AD_{15}-AD_0	I/O	Address Data Bus. These lines constitute the time multiplexed memory/IO address (T_1) and data (T_2, T_3, T_W, T_4) bus. A_0 is analogous to $\overline{BHE}$ for the lower byte of the data bus, pins D_7-D_0. It is LOW during T_1 when a byte is to be transferred on the lower portion of the bus in memory or I/O operations. Eight-bit oriented devices tied to the lower half would normally use A_0 to condition chip select functions. (See $\overline{BHE}$.) These lines are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge."
35-38	A_{19}/S_6, A_{18}/S_5, A_{17}/S_4, A_{16}/S_3	O	Address/Status. During T_1 these are the four most significant address lines for memory operations. During I/O operations these lines are LOW. During memory and I/O operations, status information is available on these lines during T_2, T_3, T_W, and T_4. The status of the interrupt enable FLAG bit (S_5) is updated at the beginning of each CLK cycle. A_{17}/S_4 and A_{16}/S_3 are encoded as shown. This information indicates which relocation register is presently being used for data accessing. These lines float to 3-state OFF during local bus "hold acknowledge."
34	$\overline{BHE}/S_7$	O	Bus High Enable/Status. During T_1 the bus high enable signal ($\overline{BHE}$) should be used to enable data onto the most significant half of the data bus, pins D_{15}-D_8. Eight-bit oriented devices tied to the upper half of the bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is LOW during T_1 for read, write, and interrupt acknowledge cycles when a byte is to be transferred on the high portion of the bus. The S_7 status information is available during T_2, T_3, and T_4. The signal is active LOW and floats to 3-state OFF in "hold." It is LOW during T_1 for the first interrupt acknowledge cycle.
32	$\overline{RD}$	O	Read. Read strobe indicates that the processor is performing a memory of I/O read cycle, depending on the state of the S_2 pin. This signal is used to read devices which reside on the 8086 local bus. $\overline{RD}$ is active LOW during T_2, T_3 and T_W of any read cycle and is guaranteed to remain HIGH in T_2 until the 8086 local bus has floated. This signal floats to 3-state OFF in "hold acknowledge."
22	READY	I	READY. Is the acknowledgment from the addressed memory or I/O device that it will complete the data transfer. The READY signal from memory/IO is synchronized by the 8284A Clock Generator to form READY. This signal is active HIGH. The 8086 READY input is not synchronized. Correct operation is not guaranteed if the set-up and hold times are not met.
18	INTR	I	Interrupt Request. Is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.
23	$\overline{TEST}$	I	$\overline{TEST}$. Input is examined by the "Wait" instruction. If the $\overline{TEST}$ input is LOW, execution continues; otherwise, the processor waits in an "Idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.
17	NMI	I	Non-Maskable Interrupt. An edge-triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.
21	RESET	I	Reset. Causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution, as described in the Instruction Set description, when RESET returns LOW. RESET is internally synchronized.
19	CLK	I	Clock. Provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
40	V_{CC}		V_{CC}. The + 5V power supply pin.
1, 20	GND		Ground. The ground pin.
33	MN/$\overline{MX}$	I	Minimum/Maximum. Indicates what mode the processor is to operate in. The two modes are discussed in the following sections.

*Pin numbers correspond to DIPs only.

A_{17}/S_4	A_{16}-S_3	Characteristics
0 (LOW)	0	Alternate Data
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data
S_6 is 0 (LOW)		

$\overline{BHE}$	A_0	Characteristics
0	0	Whole word
0	1	Upper byte from/ to odd address
1	0	Lower byte from/ to even address
1	1	None

PIN DESCRIPTION (Cont'd.)

Pin No.*	Name	I/O	Description
28-26	$\overline{S}_2$, $\overline{S}_1$, $\overline{S}_0$	O	Status. Active during T_4, T_1, and T_2 and is returned to the passive state (1, 1, 1) during T_3 or during T_W when READY is HIGH. This status is used by the 8288 Bus Controller to generate all memory and I/O access control signals. Any change by S_2, S_1, or S_0 during T_4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T_3 or T_W is used to indicate the end of a bus cycle. These signals float to 3-state OFF in "hold acknowledge." These status lines are encoded as shown.

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	Characteristics
0 (LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1 (HIGH)	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

31, 30	$\overline{RQ}/\overline{GT}_0$, $\overline{RQ}/\overline{GT}_1$	I/O	Request/Grant. Pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with $\overline{RQ}/\overline{GT}_0$ having higher priority than $\overline{RQ}/\overline{GT}_1$. $\overline{RQ}/\overline{GT}$ has an internal pull-up resistor so it may be left unconnected. The request/grant sequence is as follows: 1. A pulse of 1 CLK wide from another local bus master indicates a local bus request ("hold") to the 8086 (pulse 1). 2. During a T_4 or T_1 clock cycle, a pulse 1 CLK wide from the 8086 to the requesting master (pulse 2), indicates that the 8086 has allowed the local bus to float and that it will enter the "hold acknowledge" state at the next CLK. The CPU's bus interface unit is disconnected logically from the local bus during "hold acknowledge." 3. A pulse 1 CLK wide from the requesting master indicates to the 8086 (pulse 3) that the "hold" request is about to end and that the 8086 can reclaim the local bus at the next CLK. Each master-master exchange of the local bus is a sequence of 3 pulses. There must be one dead CLK cycle after each bus exchange. Pulses are active LOW. If the request is made while the CPU is performing a memory cycle, it will release the local bus during T_4 of the cycle when all the following conditions are met: 1. Request occurs on or before T_2. 2. Current cycle is not the low byte of a word (on an odd address). 3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence. 4. A locked instruction is not currently executing. If the local bus is idle when the request is made, two possible events will follow: 1. Local bus will be released during the next clock. 2. A memory cycle will start within 3 clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.
29	$\overline{LOCK}$	O	$\overline{LOCK}$. Output indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active LOW. The $\overline{LOCK}$ signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW, and floats to 3-state OFF in "hold acknowledge."
24, 25	QS_1, QS_0	O	Queue Status. The queue status is valid during the CLK cycle after which the queue operation is performed. QS_1 and QS_0 provide status to allow external tracking of the internal 8086 instruction queue.
28	M/$\overline{IO}$	O	Status line. Logically equivalent to S_2 in the maximum mode. It is used to distinguish a memory access from an I/O access. M/IO becomes valid in the T_4 preceding a bus cycle and remains valid until the final T_4 of the cycle (M = HIGH, IO = LOW). M/$\overline{IO}$ floats to 3-state OFF in local bus ""hold acknowledge.""
29	$\overline{WR}$	O	Write. Indicates that the processor is performing a write memory or write I/O cycle, depending on the state of M/IO signal. WR is active for T_2, T_3 and T_W of any write cycle. It is active LOW, and floats to 3-state OFF in local bus "hold acknowledge."
24	$\overline{INTA}$	O	$\overline{INTA}$. Is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T_2, T_3 and T_W of each interrupt acknowlegde cycle.
25	ALE	O	Address Latch Enable. Provided by the processor to latch the address into 8282/8283 address latch. It is a HIGH pulse active during T_1 of any bus cycle. Note that ALE is never floated.
27	DT/$\overline{R}$	O	Data Transmit/Receive. Needed in minimum system that desires to use an 8286/8287 data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically DT/$\overline{R}$ is equivalent to $\overline{S}_1$ in the maximum mode, and its timing is the same as for M/$\overline{IO}$. (T = HIGH, R = LOW.) This signal floats to 3-state OFF in local bus "hold acknowledge."
26	$\overline{DEN}$	O	Data Enable. Provided as an output enable for the 8286/8287 in a minimum system which uses the transceiver. $\overline{DEN}$ is active LOW during each memory and I/O access and for INTA cycles. For a read or $\overline{INTA}$ cycle, it is active from the middle of T_2 until the middle of T_4, while for a write cycle, it is active from the beginning of T_2 until the middle of T_4. $\overline{DEN}$ floats to 3-state OFF in local bus "hold acknowledge."

*Pin numbers correspond to DIPs only.

PIN DESCRIPTION (Cont'd.)

Pin No.*	Name	I/O	Description
31, 30	HOLD, HLDA	I/O	HOLD. Indicates that another master is requesting a local bus "hold." To be acknowledged, HOLD must be active HIGH. The processor receiving the "hold" request will issue HLDA (HIGH) as an acknowledgement in the middle of a T_4 or T_1 clock cycle. Simultaneous with the issuance of HLDA, the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor will LOWer HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines. The same rules as for $\overline{RQ}/\overline{GT}$ apply, regarding when the local bus will be released. HOLD is not asynchroneous input. External synchronization should be provided if the system cannot otherwise guarantee the set-up time.

*Pin numbers correspond to DIPs only.

DETAILED DESCRIPTION

The 8086 CPU is internally organized into two processing units. These two units are the Bus Interface Unit (BIU) and the Execution Unit (EU). A block diagram of this organization is shown in Figure 1.

The BIU performs instruction fetch and queuing, operand fetch and store, address relocation, and basic bus control. The EU receives operands and instructions from the BIU and processes them on a 16-bit ALU. The EU accesses memory and peripheral devices through requests to the BIU. The BIU generates physical addresses in memory using the 4 segment registers and offset values.

The BIU and EU usually operate asynchronously. This permits the 8086 to overlap execution fetch and execution. Up to 6 instruction bytes can be queued. The instruction queue acts as a FIFO buffer for instructions, from which the EU extracts instruction bytes as required.

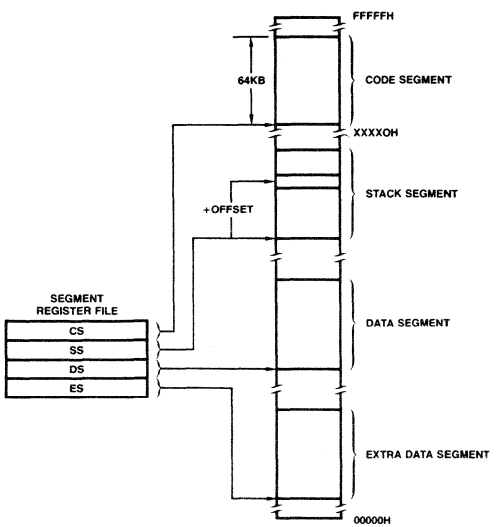
Memory Organization

The 8086 addresses up to 1 megabyte of memory. The address space is organized as a linear array, from 00000 to FFFFF in hexadecimal. Memory is subdivided into segments of 64K bytes each. There are 4 segments: code, stack, data, and extra (usually employed as an extra data segment). Each segment thus contains information of a similar type. Selection of a destination segment is automatically performed using the rules in the table below. This segmentation makes memory more easily relocatable and supports a more structured programming style.

Physical addresses in memory are generated by selecting the appropriate segment, obtaining the segment "base" address from the segment register, shifting the base address 4 digits to the left, and then adding this base to the "offset" address. For programming code, the offset address is obtained from the instruction pointer. For operands, the offset address is calculated in several ways, depending upon information contained in the addressing mode. Memory organization and address generation are shown in Figure 3a.

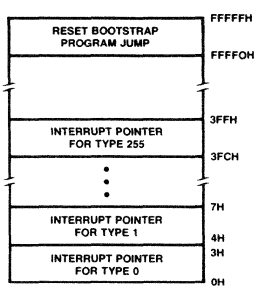
Certain memory locations are reserved for specific CPU operations. These are shown in Figure 3b. Addresses FFFFOH through FFFFFH are reserved for operations which include a jump to the initial program loading routine. After RESET, the CPU will always begin execution at location FFFFOH, where the jump must be located.

Addresses 00000H through 003FFH are reserved for interrupt operations. The service routine of each of the 256 possible interrupt types is signaled by a 4-byte pointer. The pointer elements must be stored in reserved memory addresses before the interrupts are invoked.



DF003310

Figure 3a. Memory Organization



DF003320

Figure 3b. Reserved Memory Locations

Memory Reference Need	Segment Register Used	Segment Selection Rule
Instructions	CODE (CS)	Automatic for all prefetching of instructions.
Stack	STACK (SS)	All stack pushes and pops, and all memory references relative to BP base register except data references.
Local Data	DATA (DS)	Data references which are relative to the stack, the destination of a string operation, or explicitly overriden.
External (Global) Data	EXTRA (ES)	Destination of string operations, when they are explicitly selected using a segment override.

Minimum and Maximum Modes

The 8086 has two system configurations, minimum and maximum mode. The CPU has a strap pin, MN/$\overline{MX}$, which defines the system configuration. The status of this strap pin defines the function of pin numbers 24 through 31.

When MN/$\overline{MX}$ is strapped to GND, the 8086 operates in maximum mode. The operations of pins 24 through 31 are redefined. In maximum mode, several bus timing and control functions are "off-loaded" to the 8288 bus controller, thus freeing up the CPU. The CPU communicates status information to the 8288 through pins S_0, S_1, and S_2. In maximum mode, the 8086 can operate in a multiprocessor system, using the LOCK signal within a Multibus format.

When MN/$\overline{MX}$ is strapped to V_{CC}, the 8086 operates in minimum mode. The CPU sends bus control signals itself through pins 24 through 31. This is shown in Figure 2 (in parentheses). Examples of minimum and maximum mode systems are shown in Figure 4.

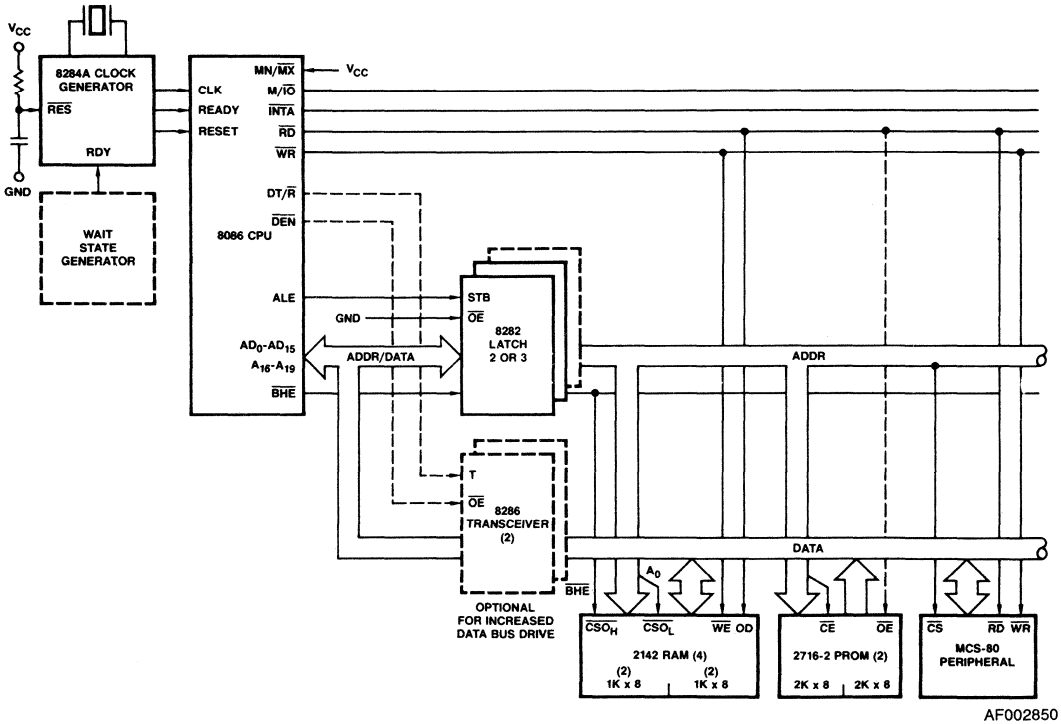


Figure 4a. Minimum Mode 8086 Typical Configuration

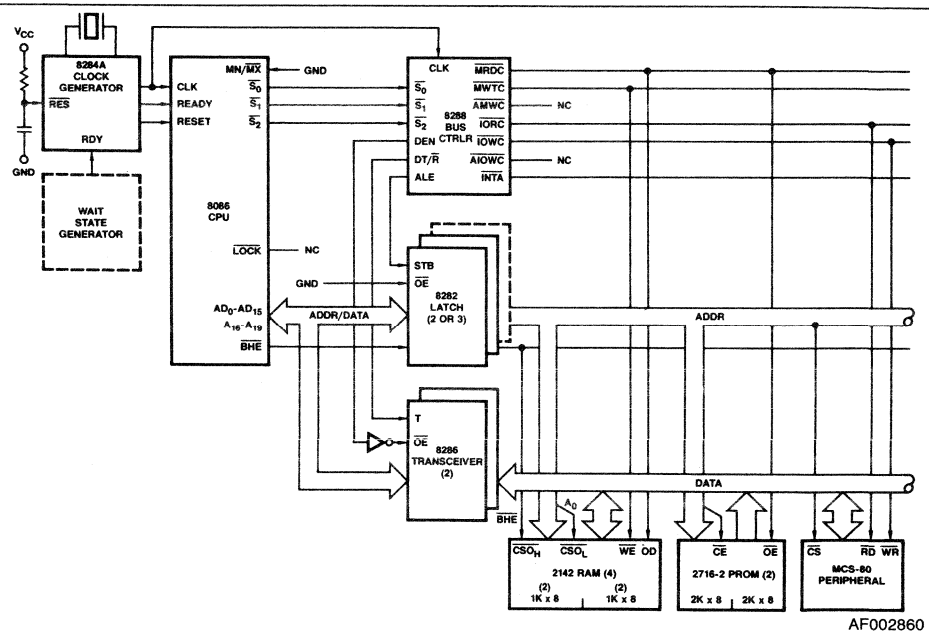


Figure 4b. Maximum Mode 8086 Typical Configuration

Bus Operation

The 8086 has a combined address and data bus, commonly referred to as "a time multiplexed bus." This technique provides the most efficient use of pins on the processor while permitting the use of a standard 40-lead package. This bus can be used throughout the system with address latching provided on memory and I/O modules. The bus can also be demultiplexed at the processor with a single set of address latches if a standard non-multiplexed bus is desired for the system.

Each bus cycle consists of at least four CLK cycles. These are referred to as T_1, T_2, T_3 and T_4 (see Figure 5). The address is sent from the processor during T_1. Data transfer occurs on the bus during T_3 and T_4. T_2 is used for changing the direction of the bus during read operations. In the event that a "NOT READY" indication is given by the addressed device, "Wait" states (T_W) are inserted between T_3 and T_4. Each inserted "Wait" state is of the same duration as a CLK cycle. "Idle" states (T_I) or inactive CLK cycles can occur between 8086 bus cycles. The processor uses these cycles for internal housekeeping.

During T_1 of any bus cycle, the ALE (Address Latch Enable) signal is emitted (by either the processor or the 8288 bus controller, depending on the MN/$\overline{MX}$ strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

Status bits $\overline{S_0}$, $\overline{S_1}$, and $\overline{S_2}$ are used, in maximum mode, by the bus controller to identify the type of bus transaction according to the following table:

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Characteristics
0(LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1(HIGH)	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

Status bits S_3 through S_7 are multiplexed with high-order address bits and the $\overline{BHE}$ signal, and are therefore valid during T_2 through T_4. S_3 and S_4 indicate which segment register (see Instruction Set description) was used for this bus cycle in forming the address, according to the following table:

S_4	S_3	Characteristics
0(LOW)	0	Alternate Data (extra segment)
0	1	Stack
1(HIGH)	0	Code or None
1	1	Data

S_5 is a reflection of the PSW interrupt enable bit. $S_6 = 0$ and S_7 is a spare status bit.

I/O Addressing

8086 I/O operations can address up to a maximum of 64K I/O byte registers or 32K I/O word registers. The I/O address appears in the same format as the memory address on bus lines $A_{15} - A_0$. The address lines $A_{19} - A_{16}$ are zero in I/O operations. I/O instructions which use register DX as a pointer have full address capability. Direct I/O instructions directly address one or two of the 256 I/O byte locations in page 0 of the I/O address space. I/O ports are addressed in the same manner as memory locations.

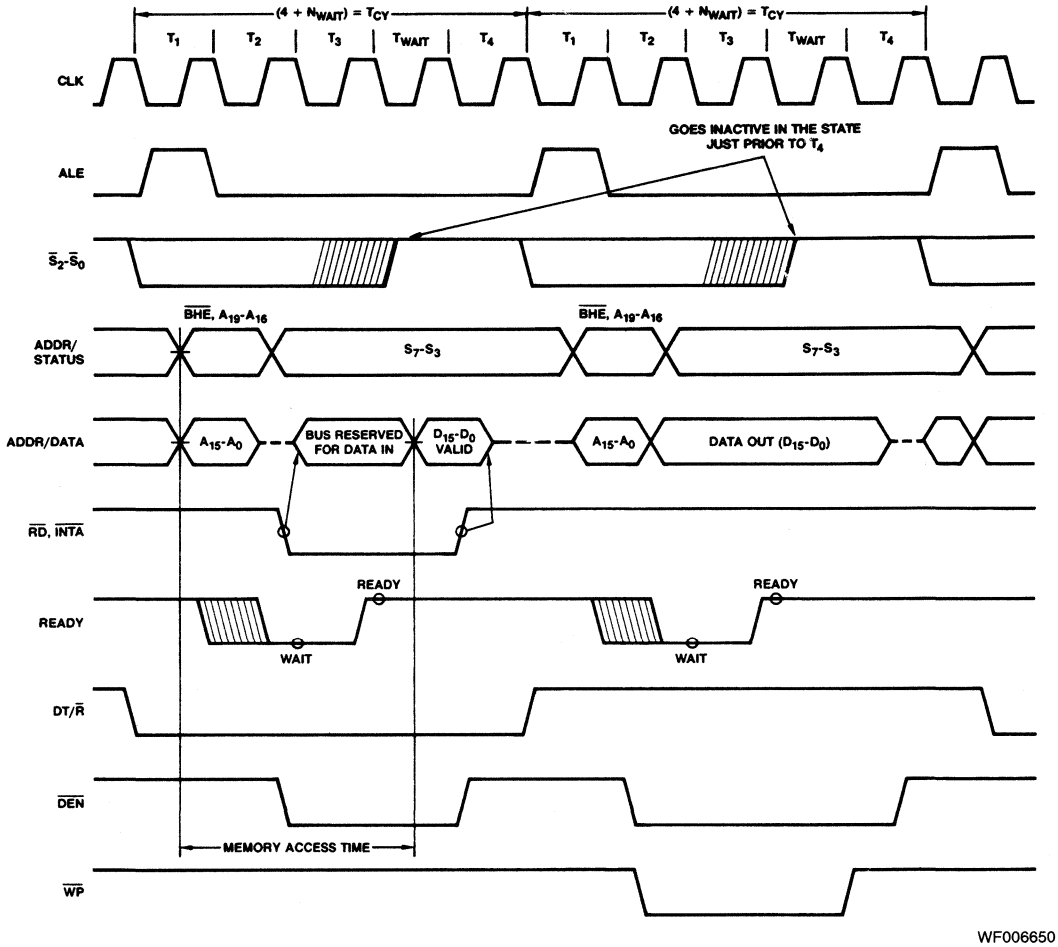


Figure 5. Basic System Timing

EXTERNAL INTERFACE

Processor Reset and Initialization

Processor initialization or start up is accomplished with activation (HIGH) of the RESET pin. The 8086 RESET is required to be HIGH for greater than 4 CLK cycles. The 8086 will terminate operations on the high-going edge of RESET and will remain dormant as long as RESET is HIGH. The low-going transition of RESET triggers an internal reset sequence for approximately 10 CLK cycles. After this interval the 8086 operates normally beginning with the instruction in absolute location FFFF0H (see Figure 3B). The details of this operation are explained in the Instruction Set description of the MCS-86 Family User's Manual. The RESET input is internally synchronized to the processor clock. At initialization the HIGH-to-LOW transition of RESET must occur no sooner than 50μs after power-up, to allow complete initialization of the 8086.

NMI may not be asserted prior to the 2nd CLK cycle following the end of RESET.

Interrupt Operations

Interrupt operations fall into two classes: software or hardware initiated. The software initiated interrupts and software aspects of hardware interrupts are described in the Instruction Set description. Hardware interrupts are either non-maskable or maskable.

Interrupts transfer control to a new program location. A 256-element table containing address pointers to the interrupt service program locations resides in absolute locations 0 through 3FFH (see Figure 3b), which are reserved for this purpose. Each element in the table is 4 bytes in size and corresponds to an interrupt "type." An interrupting device supplies an 8-bit type number during the interrupt acknowledge sequence, which is used to "vector" through the appropriate element to the new interrupt service program location.

Non-Maskable Interrupt (NMI)

The processor provides a single non-maskable interrupt pin (NMI) which has higher priority than the maskable interrupt request pin (INTR). A typical use would be to activate a power

failure routine. The NMI is edge-triggered on a LOW-to-HIGH transition. The activation of this pin causes a type 2 interrupt. (See Instruction Set description.)

NMI is required to have a duration in the HIGH state of greater than two CLK cycles, but is not required to be synchronized to the clock. Any high-going transition of NMI is latched on-chip and will be serviced at the end of the current instruction or between whole moves of a block-type instruction. Worst case response to NMI would be to multiply, divide, and variable shift instructions. There is no specification on the occurrence of the low-going edge; it may occur before, during, or after the servicing of NMI. Another high-going edge triggers another response if it occurs after the start of the NMI procedure. The signal must be free of logical spikes in general and be free of bounces on the low-going edge to avoid triggering extraneous responses.

Maskable Interrupt (INTR)

The 86/10 provides a single interrupt request input (INTR) which can be masked internally by software with the resetting of the interrupt enable FLAG status bit. The interrupt request signal is level-triggered. It is internally synchronized during each clock cycle on the high-going edge of CLK. To be responded to, INTR must be present (HIGH) during the clock period preceding the end of the current instruction or the end of a whole move for a block-type instruction. During the interrupt response sequence, further interrupts are disabled. The enable bit is reset as part of the response to any interrupt (INTR, NMI, software interrupt, or single-step), although the FLAGS register, which is automatically pushed onto the stack, reflects the state of the processor prior to the Interrupt. Until the old FLAGS register is restored, the enable bit will be zero unless specifically set by an instruction.

During the response sequence (Figure 6), the processor executes two successive (back-to-back) interrupt acknowledge cycles. The 8086 emits the LOCK signal from T_2 of the first bus cycle until T_2 of the second. A local bus "hold" request will not be honored until the end of the second bus cycle. In the second bus cycle, a byte is fetched from the external interupt system (e.g., 8259A PIC) which identifies the source (type) of the interrupt. This byte is multiplied by four and used as a pointer into the interrupt vector lookup table. An INTR signal left HIGH will be continually responded to within the limitations of the enable bit and sample period. The INTERRUPT RETURN instruction includes a FLAGS pop, which returns the status of the original interrupt enable bit when it restores the FLAGS.

HALT

When a software "HALT" instruction is executed, the processor indicates that it is entering the "HALT" state in one of two ways depending upon which mode is strapped. In minimum mode, the processor issues one ALE with no qualfying bus control signals. In Maximum Mode, the processor issues appropiate HALT status on $\overline{S}_2\overline{S}_1\overline{S}_0$, and the 8288 bus controller issues one ALE. The 8086 will not leave the "HALT" state when a local bus "hold" is entered while in "HALT." In this case, the processor reissues the HALT indicator. An interrupt request or RESET will force the 8086 out of the "HALT" state.

Read/Modify/Write (Semaphore) Operation Via Lock

The $\overline{LOCK}$ status information is provided by the processor when directly consecutive bus cycles are required during the execution of an instruction. This provides the processor with the capability of performing read/modify/write operations on memory (via the Exchange Register With Memory Instruction, for example) without the possibility of another system bus master receiving intervening memory cycles. This is useful in multiprocessor system configurations to accomplish "test and set lock" operations. The $\overline{LOCK}$ signal is activated (forced LOW) in the clock cycle following the one in which the software "LOCK" prefix instruction is decoded by the EU. It is deactivated at the end of the last bus cycle of the instruction following the "LOCK" prefix instruction. While $\overline{LOCK}$ is active, a request on a RQ/GT pin will be recorded and then honored at the end of the LOCK.

External Synchronization Via Test

As an alternative to the interrupts and general I/O capabilities, the 8086 provides a single software-testable input known as the $\overline{TEST}$ signal. At any time, the program may execute a WAIT instruction. If at that time the $\overline{TEST}$ signal is inactive (HIGH), program execution becomes suspended while the processor waits for $\overline{TEST}$ to become active. It must remain active for at least 5 CLK cycles. The WAIT instruction is re-executed repeatedly until that time. This activity does not consume bus cycles. The processor remains in an idle state while waiting. All 8086 drivers go to 3-state OFF if bus "HOLD" is entered. If interrupts are enabled, they may occur while the processor is waiting. When this occurs, the processor fetches the WAIT instruction one extra time, processes the interrupt, and then re-fetches and re-executes the WAIT instruction upon returning from the interrupt.

Basic System Timing

Typical system configurations for the processor operating in minimum mode and in maximum mode are shown in Figures 4a and 4b, respectively. In minimum mode, the processor emits bus control signals in a manner similar to the 8085. In maximum mode, the processor emits coded status information which the 8288 bus controller uses to generate MULTIBUS compatible bus control signals. Figure 5 illustrates the signal timing relationships.

System Timing – Minimum System

The read cycle begins in T_1 with the assertion of the Address Latch Enable (ALE) signal. The trailing (low-going) edge of this signal is used to latch the address information, which is valid on the local bus at this time, into the 8282/8283 latch. The $\overline{BHE}$ and A_0 signals address the low, high, or both bytes. From T_1 to T_4, the M/$\overline{IO}$ signal indicates a memory or I/O operation. At T_2 the address is removed from the local bus and the bus goes to a high impedance state. The read control signal is also asserted at T_2. The read ($\overline{RD}$) signal causes the addressed device to enable its data bus drivers to the local bus. Some time later valid data will be available on the bus and the addressed device will drive the READY line HIGH. When the processor returns the read signal to a HIGH level, the addressed device will again 3-state its bus drivers. If a transceiver (8286/8287) is required to buffer the 8086 local bus, signals DT/$\overline{R}$ and $\overline{DEN}$ are provided by the 8086.

A write cycle also begins with the assertion of ALE and the emission of the address. The M/$\overline{IO}$ signal is again asserted to indicate a memory or I/O write operation. In the T_2 immediately following the address emission, the processor emits the data to be written into the addressed location. This data remains valid until the middle of T_4. During T_2, T_3, and T_W, the processor asserts the write control signal. The write ($\overline{WR}$) signal becomes active at the beginning of T_2 as opposed to the read which is delayed somewhat into T_2 to provide time for the bus to float.

The $\overline{BHE}$ and A_0 signals are used to select the proper byte(s) of the memory/IO word to be read or written according to the following table.

$\overline{BHE}$	A_0	Characteristics
0	0	Whole word
0	1	Upper byte from/to odd address
1	0	Lower byte from/to even address
1	1	None

I/O ports are addressed in the same manner as memory location. Even addressed bytes are transferred on the D_7-D_0 bus lines and odd addressed bytes on D_{15}-D_8.

The basic difference between the interrupt acknowledge cycle and a read cycle is that the interrupt acknowledge signal ($\overline{INTA}$) is asserted in place of the read ($\overline{RD}$) signal and the address bus is floated. (See Figure 6.) In the second of two successive INTA cycles, a byte of information is read from bus lines D_7-D_0 as supplied by the interrupt system logic (i.e., 8259A Priority Interrupt Controller). This byte identifies the source (type) of the interrupt. It is multiplied by four and used as a pointer into a interrupt vector lookup table, as described earlier.

Bus Timing – Medium Size Systems

For medium size systems, the MN/$\overline{MX}$ pin is connected to V_{SS}, and the 8288 Bus Controller is added to the system as well as an 8282/8283 latch for latching the system address and a 8286/8287 transceiver to allow for bus loading greater than the 8086 is capable of handling. Signals ALE, DEN, and DT/$\overline{R}$ are generated by the 8288 instead of the processor in this configuration, although their timing remains relatively the same. The 8086 status ($\overline{S}_2$, $\overline{S}_1$, and $\overline{S}_0$) provide type-of-cycle information and become 8288 inputs. This bus cycle information specifies read (code, data, or I/O), write (data or I/O), interrupt acknowledge, or software halt. The 8288 thus issues control signals specifying memory read or write, I/O read or write, or interrupt acknowledge. The 8288 provides two types of write strobes, normal and advanced, to be applied as required. The normal write strobes have data valid at the leading edge of write. The advanced write strobes have the same timing as read strobes, and hence, data isn't valid at the leading edge of write. The 8286/8287 transceiver receives the usual T and OE inputs from the 8288's DT/$\overline{R}$ and DEN.

The pointer into the interrupt vector table, which is passed during the second INTA cycle, can derive from an 8259A located on either the local bus or the system bus. If the master 8259A Priority Interrupt Controller is positioned on the local bus, a TTL gate is required to disable the 8286/8287 transceiver when reading from the master 8259A during the interrupt acknowledge sequence and software "poll."

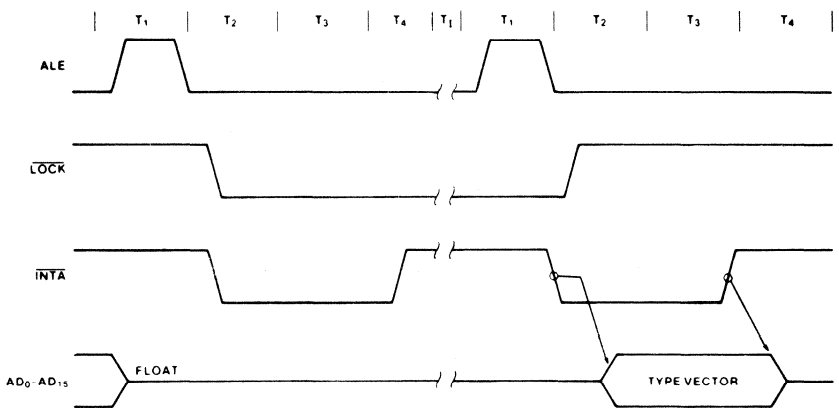


Figure 6. Interrupt Acknowledge Sequence

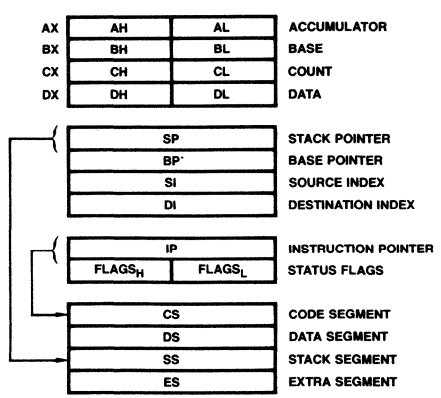


Figure 7. 8086 Register Model

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Ambient Temperature Under Bias ..... 0 to 70°C
Voltage on any Pin
with Respect to Ground ..... −1 to +7.0V
Power Dissipitation 2.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC})
8086 ..... 5 V ± 10%
8086-1, 8086-2 ..... 5 V ± 5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC})
8086 ..... 5 V ± 10%
8086-1, 8086-2 ..... 5 V ± 5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS

Parameters	Description	Test Conditions	Min	Max	Units
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	I_{OL} = 2.5mA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400μA	2.4		V
I_{CC}	Power Supply Current	All Speeds		340	mA
I_{LI}	Input Leakage Current	$0V \leqslant V_{IN} \leqslant V_{CC}$		±10	μA
I_{LO}	Output Leakage Current	$0.45V \leqslant V_{OUT} \leqslant V_{CC}$		±10	μA
V_{CL}	Clock Input Low Voltage		−0.5	+0.6	V
V_{CH}	Clock Input High Voltage		3.9	V_{CC} + 1.0	V
C_{IN}	Capacitance of Input Buffer (All input except AD_0-AD_{15}, $\overline{RQ}/\overline{GT}$)	fc = 1 MHz		15	pF
C_{IO}	Capacitance of I/O Buffer (AD_0-AD_{15}, $\overline{RQ}/\overline{GT}$)	fc = 1 MHz		15	pF

SWITCHING CHARACTERISTICS
MINIMUM COMPLEXITY SYSTEM TIMING REQUIREMENTS

Parameters	Description	Test Conditions	8086		8086-2		8086-1		Units
			Min	Max	Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	100	500	ns
TCLCH	CLK Low Time		118		68		53		ns
TCHCL	CLK High Time		69		44		39		ns
TCH1CH2	CLK Rise Time	From 1.0 to 3.5V		10		10		10	ns
TCL2CL1	CLK Fall Time	From 3.5 to 1.0V		10		10		10	ns
TDVCL	Data in Set-up Time		30		20		5		ns
TCLDX	Data in Hold Time		10		10		10		ns
TR1VCL	RDY Set-up Time into 8284A (See Notes 1, 2)		35		35		35		ns
TCLR1X	RDY Hold Time into 8284A (See Notes 1, 2)		0		0		0		ns
TRYHCH	READY Set-up Time into 8086		118		68		53		ns
TCHRYX	READY Hold Time into 8086		30		20		20		ns
TRYLCL	READY Inactive to CLK (See Note 3)		−8		−8		−10		ns
THVCH	HOLD Set-up Time		35		20		20		ns
TINVCH	INTR, NMI, $\overline{TEST}$ Set-up Time (See Note 2)		30		15		15		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0V		20		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8V		12		12		12	ns

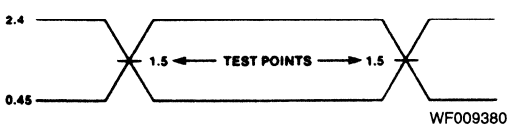
Notes: 1. Signal at 8284A shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T2 state (8ns into T3).

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS (Cont'd.)
TIMING RESPONSES

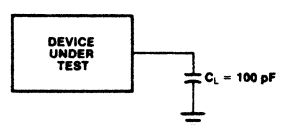
Parameters	Description	Test Conditions	8086		8086-2		8086-1		Units
			Min	Max	Min	Max	Min	Max	
TCLAV	Address Valid Delay	*C_L = 20-100 pF for all 8086 Outputs (in addition to 8086 self-load). Typical C_L = 100 pF.	10	110	10	60	10	50	ns
TCLAX	Address Hold Time		10		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	10	40	ns
TLHLL	ALE Width		TCLCH – 20		TCLCH – 10		TCLCH – 10		ns
TCLLH	ALE Active Delay			80		50		40	ns
TCHLL	ALE Inactive Delay			85		55		45	ns
TLLAX	Address Hold Time to ALE Inactive		TCHCL – 10		TCHCL – 10		TCHCL – 10		ns
TCLDV	Data Valid Delay		10	110	10	60	10	50	ns
TCHDX	Data Hold Time		10		10		10		ns
TWHDX	Data Hold Time After WR		TCLCH – 30		TCLCH – 30		TCLCH – 25		ns
TCVCTV	Control Active Delay 1		10	110	10	70	10	50	ns
TCHCTV	Control Active Delay 2		10	110	10	60	10	45	ns
TCVCTX	Control Inactive Delay		10	110	10	70	10	50	ns
TAZRL	Address Float to READ active		0		0		0		ns
TCLRL	$\overline{RD}$ Active Delay		10	165	10	100	10	70	ns
TCLRH	$\overline{RD}$ Inactive Delay		10	150	10	80	10	60	ns
TRHAV	$\overline{RD}$ Inactive to Next Address Active		TCLCL – 45		TCLCL – 40		TCLCL – 35		ns
TCLHAV	HLDA Valid Delay		10	160	10	100	10	60	ns
TRLRH	$\overline{RD}$ Width		2TCLCL – 75		2TCLCL – 50		2TCLCL – 40		ns
TWLWH	$\overline{WR}$ Width		2TCLCL – 60		2TCLCL – 40		2TCLCL – 35		ns
TAVAL	Address Valid to ALE Low		TCLCH – 60		TCLCH – 40		TCLCH – 35		ns
TOLOH	Output Rise Time	From 0.8 to 2.0V		20		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8V		12		12		12	ns

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF009380

SWITCHING TEST LOAD CIRCUIT



TC002190

AC TESTING INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 1.5V FOR BOTH A LOGIC "1" AND "0."

C_L INCLUDES JIG CAPACITANCE

SWITCHING CHARACTERISTICS (Cont'd.)
MAX MODE SYSTEM (USING 8288 BUS CONTROLLER)
TIMING REQUIREMENTS

Parameters	Description	Test Conditions	8086		8086-2		8086-1		Units
			Min	Max	Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	100	500	ns
TCLCH	CLK Low Time		118		68		53		ns
TCHCL	CLK High Time		69		44		39		ns
TCH1CH2	CLK Rise Time	From 1.0 to 3.5V		10		10		10	ns
TCL2CL1	CLK Fall Time	From 3.5 to 1.0V		10		10		10	ns
TDVCL	Data in Set-up Time		30		20		5		ns
TCLDX	Data in Hold Time		10		10		10		ns
TR1VCL	RDY Set-up Time into 8284A (See Notes 1, 2)		35		35		35		ns
TCLR1X	RDY Hold Time into 8284A (See Notes 1, 2)		0		0		0		ns
TRYHCH	READY Set-up Time into 8086		118		68		53		ns
TCHRYX	READY Hold Time into 8086		30		20		20		ns
TRYLCL	READY Inactive to CLK (See Note 4)		-8		-8		-10		ns
TINVCH	Set-up Time for Recognition (INTR, NMI, $\overline{\text{TEST}}$ (See Note 2)		30		15		15		ns
TGVCH	$\overline{\text{RQ}}/\overline{\text{GT}}$ Set-up Time		30		15		12		ns
TCHGX	$\overline{\text{RQ}}$ Hold Time into 8066		40		30		20		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0V		20		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8V		12		12		12	ns

Notes: 1. Signal at 8284A or 8288 shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8ns into T3).

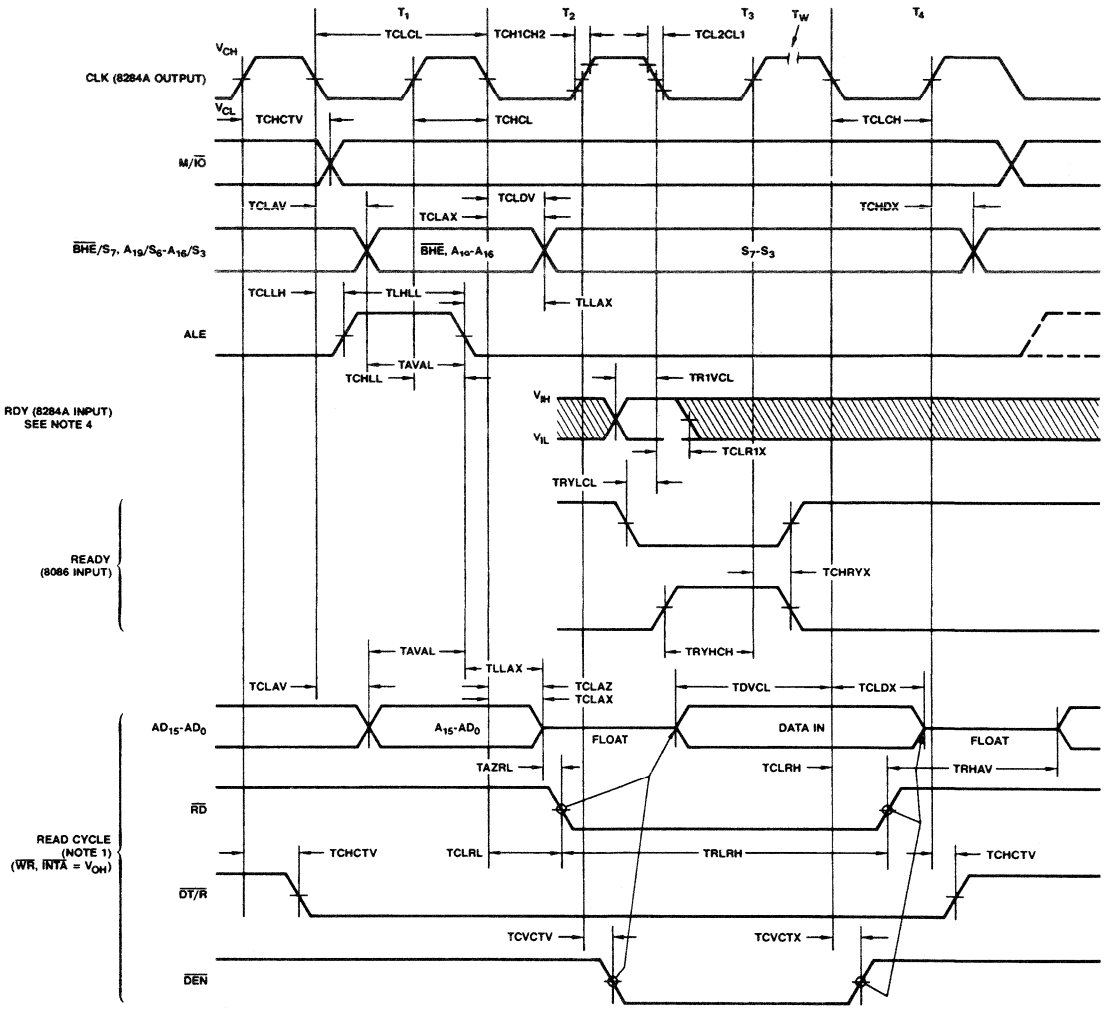
SWITCHING CHARACTERISTICS (Cont'd.)
TIMING RESPONSES

Parameters	Description	Test Conditions	8086		8086-2		8086-1		Units
			Min	Max	Min	Max	Min	Max	
TCLML	Command Active Delay (See Note 1)	C_L = 20-100 pF for all 8086 Outputs (In addition to 8086 self-load)	10	35	10	35	10	35	ns
TCLMH	Command Inactive Delay (See Note 1)		10	35	10	35	10	35	ns
TRYHSH	READY Active to Status Passive (See Note 3)			110		65		45	ns
TCHSV	Status Active Delay		10	110	10	60	10	45	ns
TCLSH	Status Inactive Delay		10	130	10	70	10	55	ns
TCLAV	Address Valid Delay		10	110	10	60	10	50	ns
TCLAX	Address Hold Time		10		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	10	40	ns
TSVLH	Status Valid to ALE High (See Note 1)			15		15		15	ns
TSVMCH	Status Valid to MCE High (See Note 1)			15		15		15	ns
TCLLH	CLK Low to ALE Valid (See Note 1)			15		15		15	ns
TCLMCH	CLK Low to MCE High (See Note 1)			15		15		15	ns
TCHLL	ALE Inactive Delay (See Note 1)			15		15		15	ns
TCLMCL	MCE Inactive Delay (See Note 1)			15		15		15	ns
TCLDV	Data Valid Delay		10	110	10	60	10	50	ns
TCHDX	Data Hold Time		10		10		10		ns
TCVNV	Control Active Delay (See Note 1)		5	45	5	45	5	45	ns
TCVNX	Control Inactive Delay (See Note 1)		10	45	10	45	10	45	ns
TAZRL	Address Float to Read Active		0		0		0		ns
TCLRL	$\overline{RD}$ Active Delay		10	165	10	100	10	70	ns
TCLRH	$\overline{RD}$ Inactive Delay		10	150	10	80	10	60	ns
TRHAV	$\overline{RD}$ Inactive to Next Address Active		TCLCL – 45		TCLCL – 40		TCLCL – 35		ns
TCHDTL	Direction Control Active Delay (See Note 1)			50		50		50	ns
TCHDTH	Direction Control Inactive Delay (See Note 1)			30		30		30	ns
TCLGL	GT Active Delay		0	85	0	50	0	38	ns
TCLGH	GT Inactive Delay		0	85	0	50	0	45	ns
TRLRH	$\overline{RD}$ Width		2TCLCL – 75		2TCLCL – 50		2TCLCL – 40		ns
TOLOH	Output Rise Time	From 0.8 to 2.0V		20		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8V		12		12		12	ns

Notes: 1. Signal at 8284A or 8288 shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8ns into T3).

SWITCHING WAVEFORMS

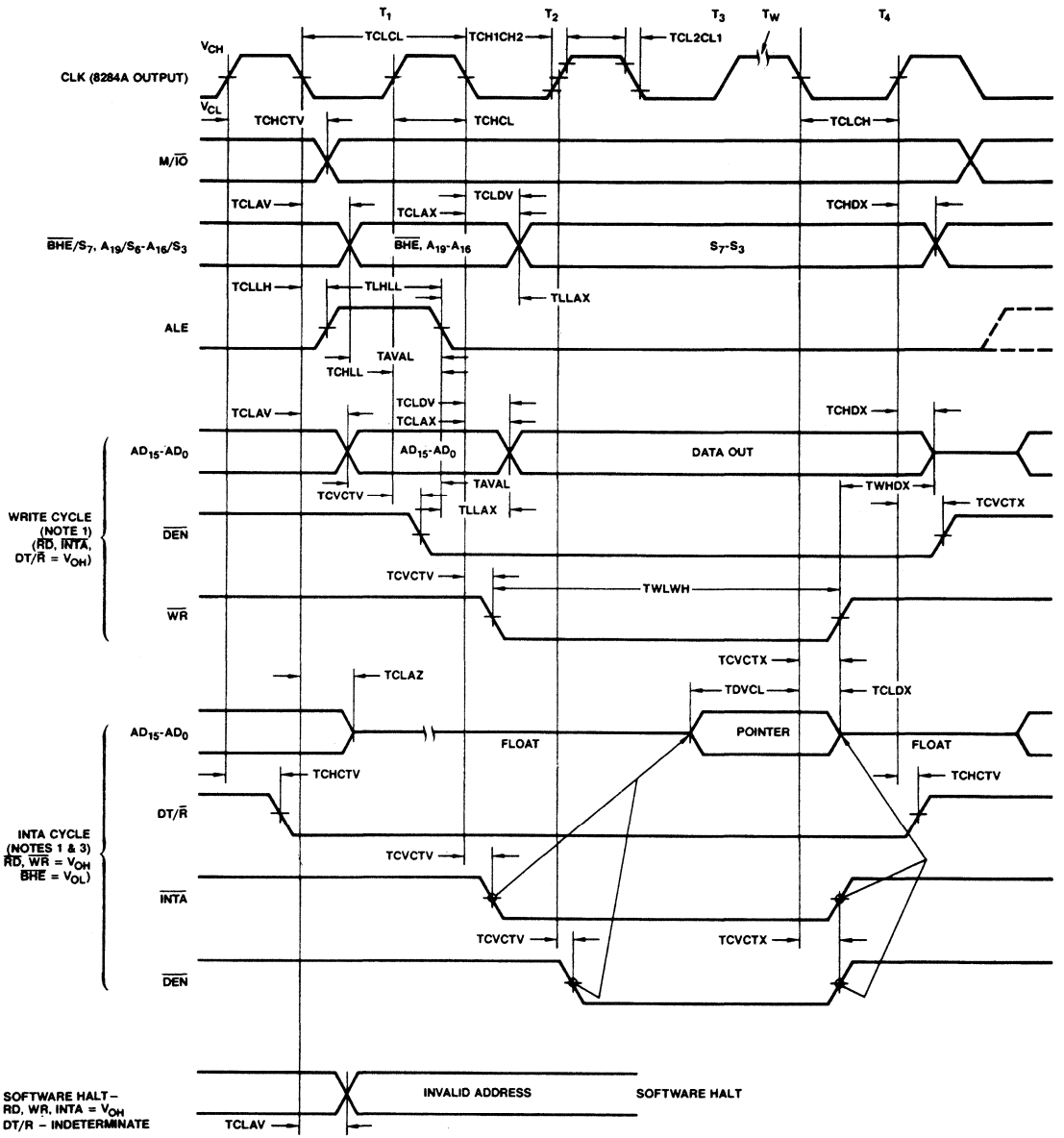
MINIMUM MODE



WF006660

SWITCHING WAVEFORMS (Cont'd.)

MINIMUM MODE

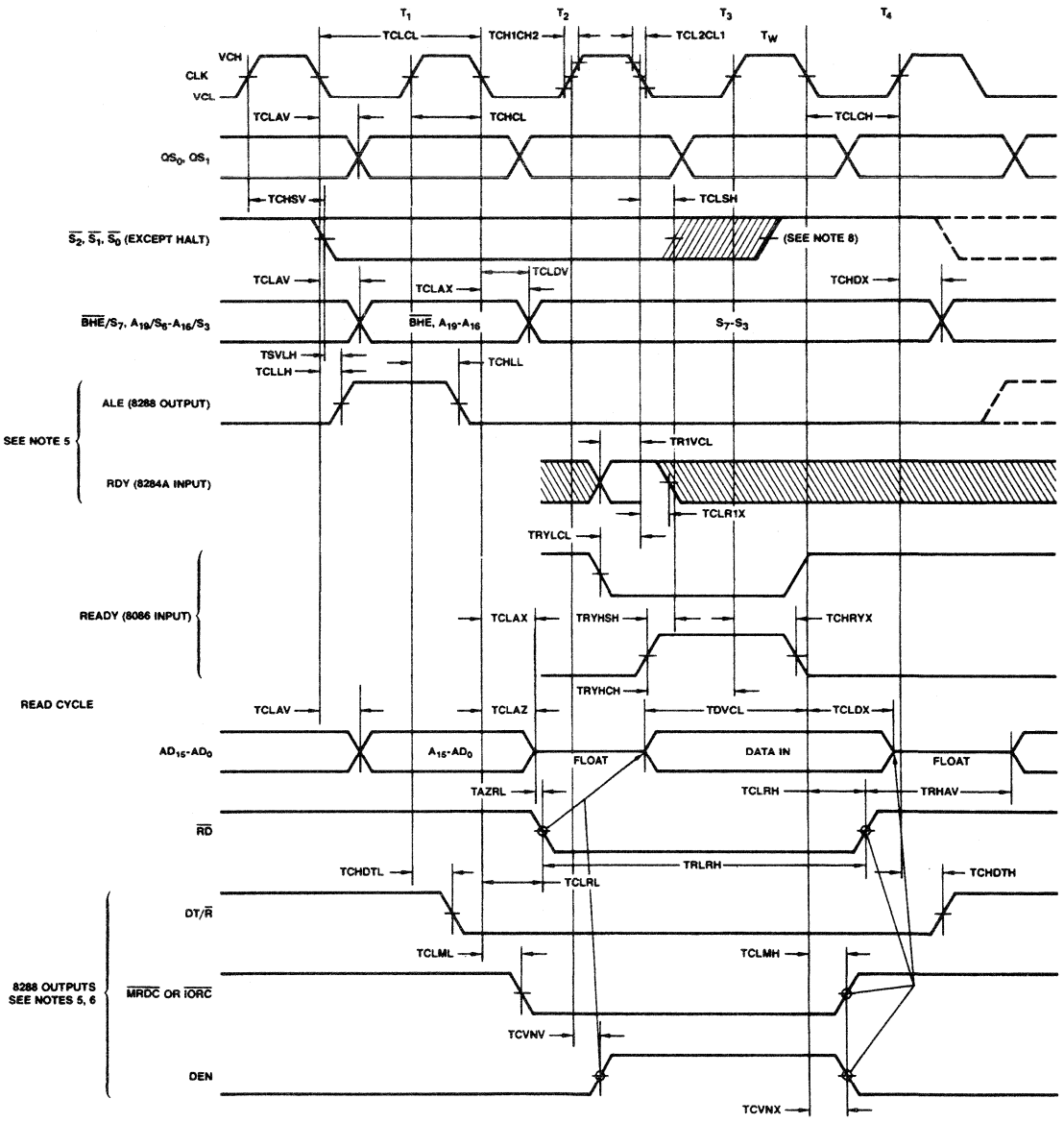


WF006670

Notes: 1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Two INTA cycles run back-to-back. The 8086 LOCAL ADDR/DATA BUS is floating during both INTA cycles. Control signals are shown for second INTA cycle.
4. Signals at 8284A are shown for reference only.
5. All timing measurements are made at 1.5V unless otherwise noted.

SWITCHING WAVEFORMS (Cont'd.)

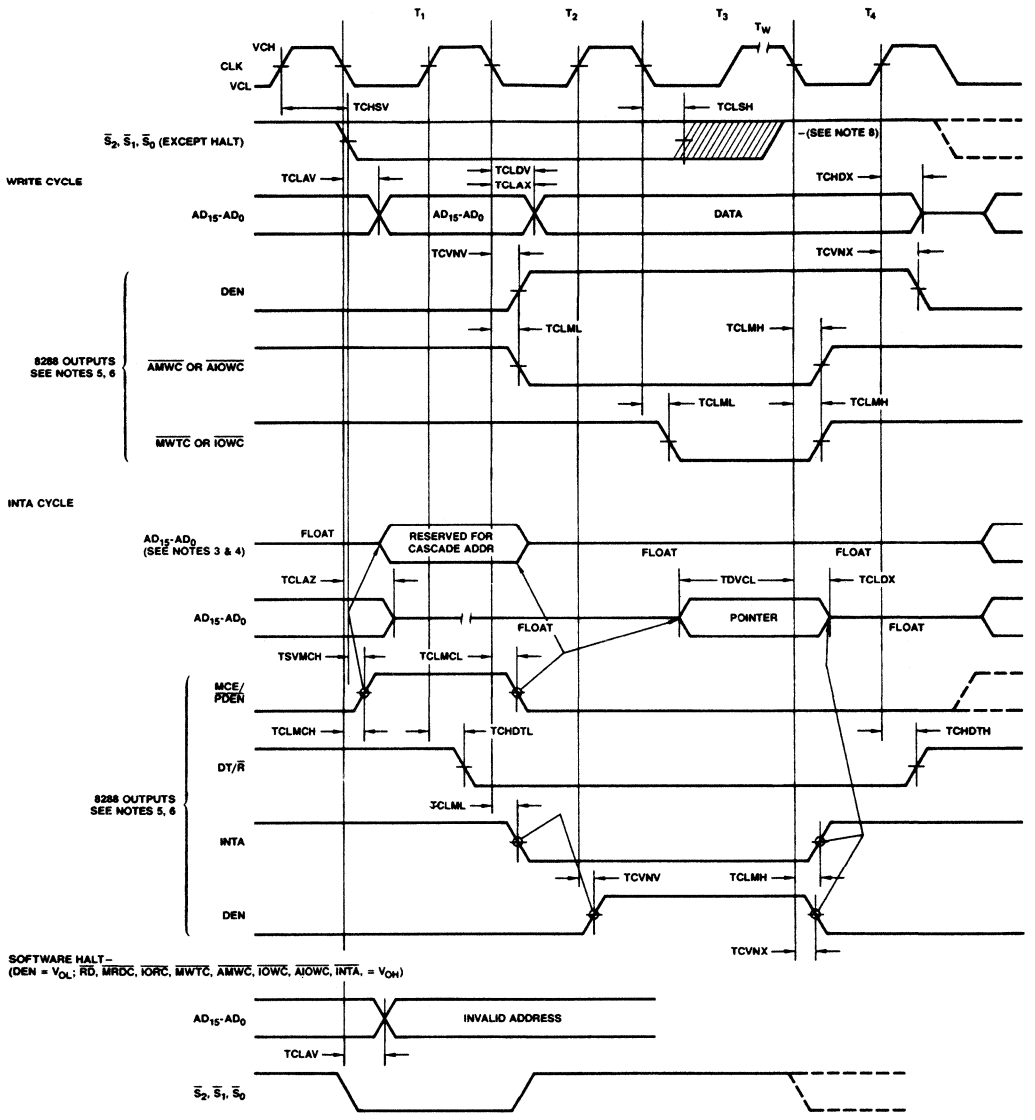
MAXIMUM MODE



WF006680

SWITCHING WAVEFORMS (Cont'd.)

MAXIMUM MODE (Cont'd.)

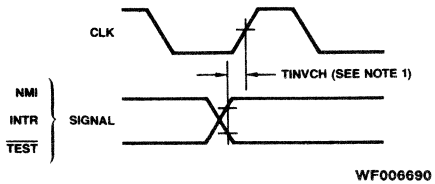


WF006730

Notes:
1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Cascade address is valid between first and second INTA cycle.
4. Two INTA cycles run back-to-back. The 8086 LOCAL ADDR/DATA BUS is floating during both INTA cycles. Control for pointer address is shown for second INTA cycle.
5. Signals at 8284A or 8288 are shown for reference only.
6. The issuance of the 8288 command and control signals ($\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMWC}$, $\overline{IORC}$, $\overline{IOWC}$, $\overline{AIOWC}$, $\overline{INTA}$ and DEN) lags the active high 8288 CEN.
7. All timing measurements are made at 1.5V unless otherwise noted.
8. Status inactive in state just prior to T_4.

SWITCHING WAVEFORMS (Cont'd.)

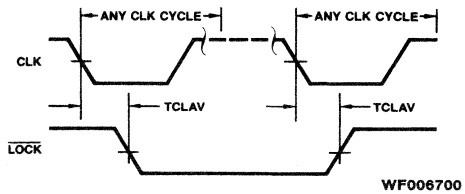
ASYNCHRONOUS SIGNAL RECOGNITION



WF006690

Note: 1. Set-up Requirements for Asynchronous signals only to guarantee recognition at next CLK.

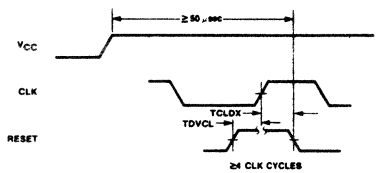
BUS LOCK SIGNAL TIMING



WF006700

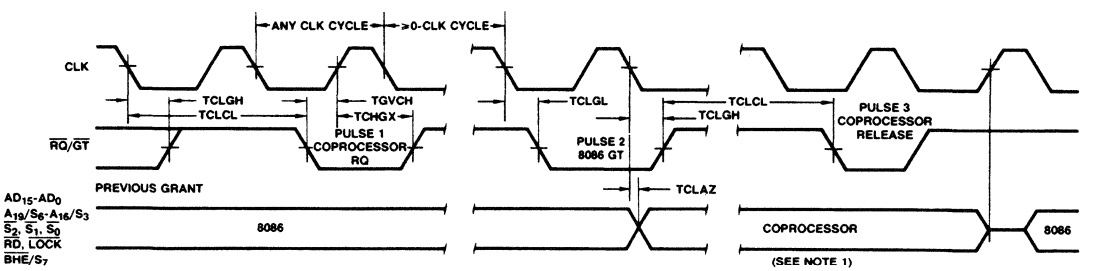
(MAXIMUM MODE ONLY)

RESET TIMING



WF009530

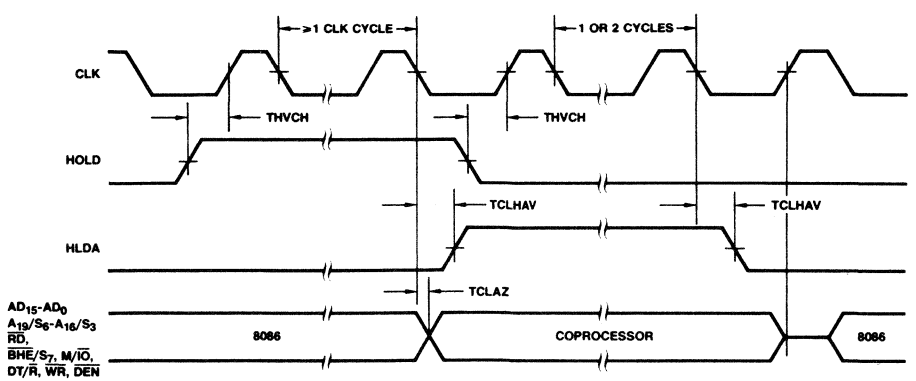
REQUEST/GRANT SEQUENCE TIMING (MAXIMUM MODE ONLY)



WF006710

Note: 1. The Coprocessor may not drive the buses outside the region shown without risking contention.

HOLD/HOLD ACKNOWLEDGE TIMING (MINIMUM MODE ONLY)



WF006720

8086/8088
INSTRUCTION SET SUMMARY

DATA TRANSFER

MOV = Move	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Register/memory to /from register	1 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 1 0 0 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate to register	1 0 1 1 w reg	data	data if w = 1	
Memory to accumulator	1 0 1 0 0 0 0 w	addr-low	addr-high	
Accumulator to memory	1 0 1 0 0 0 1 w	addr-low	addr-high	
Register/memory to segment register	1 0 0 0 1 1 1 0	mod 0 reg r/m		
Segment register to register/memory	1 0 0 0 1 1 0 0	mod 0 reg r/m		
PUSH = Push:				
Register/memory	1 1 1 1 1 1 1 1	mod 1 1 0 r/m		
Register	0 1 0 1 0 reg			
Segment register	0 0 0 reg 1 1 0			
POP = Pop:				
Register/memory	1 0 0 0 1 1 1 1	mod 0 0 0 r/m		
Register	0 1 0 1 1 reg			
Segment register	0 0 0 reg 1 1 1			
XCHG = Exchange:				
Register/memory with register	1 0 0 0 0 1 1 w	mod reg r/m		
Register with accumulator	1 0 0 1 0 reg			
IN = Input from:				
Fixed port	1 1 1 0 0 1 0 w	port		
Variable port	1 1 1 0 1 1 0 w			
OUT = Ouput to:				
Fixed port	1 1 1 0 0 1 1 w	port		
Variable port	1 1 1 0 1 1 1 w			
XLAT = Transtate byte to AL	1 1 0 1 0 1 1 1			
LEA = Load EA to register	1 0 0 0 1 1 0 1	mod reg r/m		
LDS = Load pointer to DS	1 1 0 0 0 1 0 1	mod reg r/m		
LES = Load pointer to ES	1 1 0 0 0 1 0 0	mod reg r/m		
LANF = Load AH with flags	1 0 0 1 1 1 1 1			
SANF = Store AH into flags	1 0 0 1 1 1 1 0			
PUSHF = Push flags	1 0 0 1 1 1 0 0			
POPF = Pop flags	1 0 0 1 1 1 0 1			

INSTRUCTION SET SUMMARY (Cont'd.)

ARITHMETIC

ADD = Add	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Reg/memory with register to either	0 0 0 0 0 0 d w	mod reg r/m		
Immediate to register / memory	1 0 0 0 0 0 s w	mod 0 0 0 r/m	data	data if s:w = 01
Immediate to accumulator	0 0 0 0 0 1 0 w	data	data if w = 1	
ADC = Add with carry:				
Reg/memory with register to either	0 0 0 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 s w	mod 0 1 0 r/m	data	data if s:w = 01
Immediate to accumulator	0 0 0 1 0 1 0 w	data	data if w = 1	
INC = Increment:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 0 r/m		
Register	0 1 0 0 0 reg			
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1			
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1			
SUB = Subtract:				
Reg/memory and register to either	0 0 1 0 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 1 0 1 r/m	data	data if s:w = 01
Immediate from accumulator	0 0 1 0 1 1 0 w	data	data if w = 1	
SBB = Subtract with borrow:				
Reg/memory and register to either	0 0 0 1 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 0 1 1 r/m	data	data if s:w = 01
Immediate from accumulator	0 0 0 1 1 1 0 w	data	data if w = 1	
DEC = Decrement:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 1 r/m		
Register	0 1 0 0 1 reg			
NEG Change sign	1 1 1 1 0 1 1 w	mod 0 1 1 r/m		
CMP = Compare:				
Register/memory with register	0 0 1 1 1 0 1 w	mod reg r/m		
Register with register/memory	0 0 1 1 1 0 0 w	mod reg r/m		
Immediate with register/memory	1 0 0 0 0 0 s w	mod 1 1 1 r/m	data	data if s:w = 01
Immediate with accumulator	0 0 1 1 1 1 0 w	data	data if w = 1	
AAS ASCII adjust for subtract	0 0 1 1 1 1 1 1			
DAS Decimal adjust for subtract	0 0 1 0 1 1 1 1			
MUL Mulitiply (unsigned)	1 1 1 1 0 1 1 w	mod 1 0 0 r/m		
IMUL Integer multiply (signed):	1 1 1 1 0 1 1 w	mod 1 0 1 r/m		
AAM ASCII adjust for multiply	1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0		
DIV Divide (unsigned):	1 1 1 1 0 1 1 w	mod 1 1 0 r/m		
IDIV Integer divide (signed)	1 1 1 1 0 1 1 w	mod 1 1 1 r/m		
AAD ASCH adjust for divide	1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0		
CBW Convert byte to word	1 0 0 1 1 0 0 0			
CWD Convert word to double word	1 0 0 1 1 0 0 1			

INSTRUCTION SET SUMMARY (Cont'd.)

LOGIC

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
NOT Invert	1 1 1 1 0 1 1 w	mod 0 1 0 r/m		
SHL/SAL Shift logical/arithmetic left	1 1 0 1 0 0 v w	mod 1 0 0 r/m		
SHR Shift logical right	1 1 0 1 0 0 v w	mod 1 0 1 r/m		
SAR Shift arithmetic right	1 1 0 1 0 0 v w	mod 1 1 1 r/m		
ROL Rotate left	1 1 0 1 0 0 v w	mod 0 0 0 r/m		
ROR Rotate right	1 1 0 1 0 0 v w	mod 0 0 1 r/m		
RCL Rotate through carry flag left	1 1 0 1 0 0 v w	mod 0 1 0 r/m		
RCR Rotate through carry right	1 1 0 1 0 0 v w	mod 0 1 1 r/m		
AND = And:				
Reg/memory and register to either	0 0 1 0 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w = 1	
TEST = And function to flags, no result:				
Register/memory and register	1 0 0 0 0 1 0 w	mod reg r/m		
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w = 1	
OR = Or:				
Reg/memory and register to either	0 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w = 1
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w = 1	
XOR = Exclusive or:				
Reg/memory and register to either	0 0 1 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w = 1	

STRING MANIPULATION:

REP = Repeat	1 1 1 1 0 0 1 z
MOVS = Move byte/word	1 0 1 0 0 1 0 w
CMPS = Compare byte/word	1 0 1 0 0 1 1 w
SCAS = Scan byte/word	1 0 1 0 1 1 1 w
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w
STOS = Stor byte/wd from AL/A	1 0 1 0 1 0 1 w

INSTRUCTION SET SUMMARY (Cont'd.)

CONTROL TRANSFER

CALL = Call	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Direct within segment	1 1 1 0 1 0 0 0	disp-low	disp-high	
indirect within segment	1 1 1 1 1 1 1 1	mod 0 1 0 r/m		
Direct intersegment	1 0 0 1 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect intersegment	1 1 1 1 1 1 1 1	mod 0 1 1 r/m		

JMP = Unconditional jump:				
Direct within segment	1 1 1 0 1 0 0 1	disp-low	disp-high	
Direct within segment-short	1 1 1 0 1 0 1 1	disp		
Indirect within segment	1 1 1 1 1 1 1 1	mod 1 0 0 r/m		
Direct intersegment	1 1 1 0 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect intersegment	1 1 1 1 1 1 1 1	mod 1 0 1 r/m		

RET = Return from CALL:			
Within segment	1 1 0 0 0 0 1 1		
Within seg adding immed to SP	1 1 0 0 0 0 1 0	data-low	data-high
Intersegment	1 1 0 0 1 0 1 1		
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0	data-low	data-high
JE/JZ = Jump on equal/zero	0 1 1 1 0 1 0 0	disp	
JL/JNGE = Jump on less/not greater or equal	0 1 1 1 1 1 0 0	disp	
JLE/JNG = Jump on less or equal/not greater	0 1 1 1 1 1 1 0	disp	
JB/JNAE = Jump on below/not above or equal	0 1 1 1 0 0 1 0	disp	
JBE/JNA = Jump on below or equal/not above	0 1 1 1 0 1 1 0	disp	
JP/JPE = Jump on parity/parity even	0 1 1 1 1 0 1 0	disp	
JO = Jump on overflow	0 1 1 1 0 0 0 0	disp	
JS = Jump on sign	0 1 1 1 1 0 0 0	disp	
JNE/JNZ = Jump on not equal/not zero	0 1 1 1 0 1 0 1	disp	
JNL/JGE = Jump on not less/greater or equal	0 1 1 1 1 1 0 1	disp	
JNLE/JG = Jump on not less or equal/greater	0 1 1 1 1 1 1 1	disp	
JNB/JAE = Jump on not below/above or equal	0 1 1 1 0 0 1 1	disp	
JNBE/JA = Jump on not below or equal/above	0 1 1 1 0 1 1 1	disp	
JNP/JPO = Jump on not par/par odd	0 1 1 1 1 0 1 1	disp	
JNO = Jump on not overflow	0 1 1 1 0 0 0 1	disp	
JNS = Jump on not sign	0 1 1 1 1 0 0 1	disp	
LOOP = Loop CX times	1 1 1 0 0 0 1 0	disp	
LOOPZ/LOOPE = Loop while zero/equal	1 1 1 0 0 0 0 1	disp	
LOOPNZ/LOOPNE = Loop while not zero/equal	1 1 1 0 0 0 0 0	disp	
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1	disp	

INSTRUCTION SET SUMMARY (Cont'd.)

CONTROL TRANSFER (Cont'd.)

INT = Interrupt	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Type specified	1 1 0 0 1 1 0 1	type		
Type 3	1 1 0 0 1 1 0 0			
INTO = Interrupt on overflow	1 1 0 0 1 1 1 0			
IRET = Interrupt return	1 1 0 0 1 1 1 1			

PROCESSOR CONTROL

CLC = Clear carry	1 1 1 1 1 0 0 0	
CMC = Complement carry	1 1 1 1 0 1 0 1	
STC = Set carry	1 1 1 1 1 0 0 1	
CLD = Clear direction	1 1 1 1 1 1 0 0	
STD = Set direction	1 1 1 1 1 1 0 1	
CLI = Clear interrupt	1 1 1 1 1 0 1 0	
STI = Set interrupt	1 1 1 1 1 0 1 1	
HLT = Halt	1 1 1 1 0 1 0 0	
WAIT = Wait	1 0 0 1 1 0 1 1	
ESC = Processor Extension Escape	1 1 0 1 1 x x x	mod x x x r/m
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0	

Footnotes:

AL = 8-bit accumulator
AX = 16-bit accumulator
CX = Count register
DS = Data segment
ES = Extra segment
Above/below refers to unsigned value.
Greater = more positive.
Less = less positive (more negative) signed values
if d = 1 then "to" reg; if d = 0 then "from" reg
w = 1 then word instruction; if w = 0 then byte instruction

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

if s:w = 01 then 16 bits of immediate data form the operand.
if s:w = 11 then an immediate data byte is sign extended to form the 16-bit operand.
if v = 0 then "count" = 1; if v = 1 then "count" in (CL)
x = don't care
z is used for string primitives for comparison with ZF Flag.

SEGMENT OVERRIDE PREFIX

0	0	1	reg	1	1	0

REG is assigned according to the following table:

16-Bit (w = 1)	8-Bit (w = 0)	Segment
000 AX	000 AL	00 ES
001 CX	001 CL	01 CS
010 DX	010 DL	10 SS
011 BX	011 BL	11 DS
100 SP	100 AH	
101 BP	101 CH	
110 SI	110 DH	
111 DI	111 BH	

Instructions which reference the flag register files as a 16-bit object use the symbol FLAGS to represent the file:

FLAGS = X:X:X:X:(OF):(DF):(TF):(SF):(ZF):X:(AF):X:(PF):X:(CF)

8088

8-Bit Microprocessor CPU
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

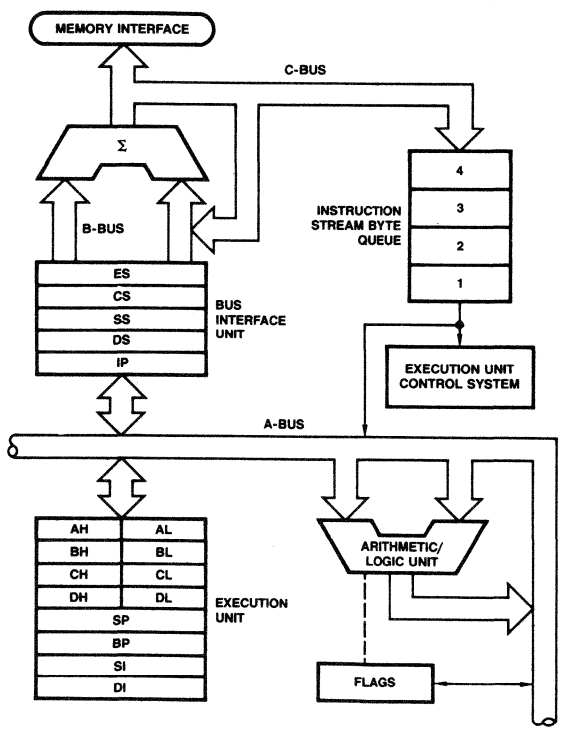
- 8-bit data bus, 16-bit internal architecture
- Directly addresses 1 Mbyte of memory
- Software compatible with 8086 CPU
- Byte, word, and block operations
- 24 operand addressing modes
- Powerful instruction set
- Efficient high level language implementation
- Three speed options: 5MHz 8088
  8MHz 8088-2
  10MHz 8088-1

GENERAL DESCRIPTION

The 8088 CPU is an 8-bit processor designed around the 8086 internal structure. Most functions of the 8088 are identical to the equivalent 8086 functions. The pinout is slightly different. The 8088 handles the external bus the same way the 8086 does, but it handles only 8 bits at a time. Sixteen-bit words are fetched or written in two consecutive bus cycles. Both processors will appear identical to the software engineer, with the exception of execution time.

The 8088 is made with N-channel silicon gate technology and is packaged in a 40-pin Plastic dip, CERDIP or Plastic Leaded Chip Carrier.

BLOCK DIAGRAM



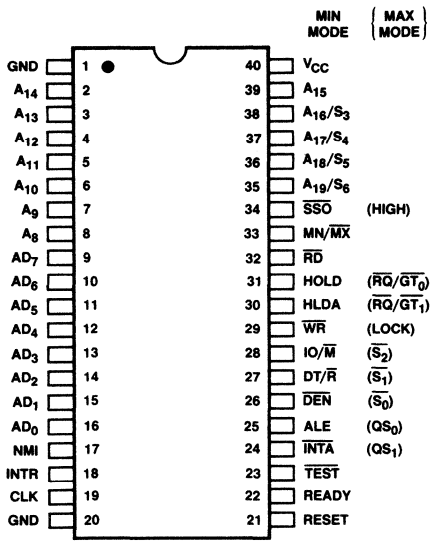
BD003750

Publication # 02338 | Rev. C | Amendment /0
Issue Date: May 1987

CONNECTION DIAGRAMS

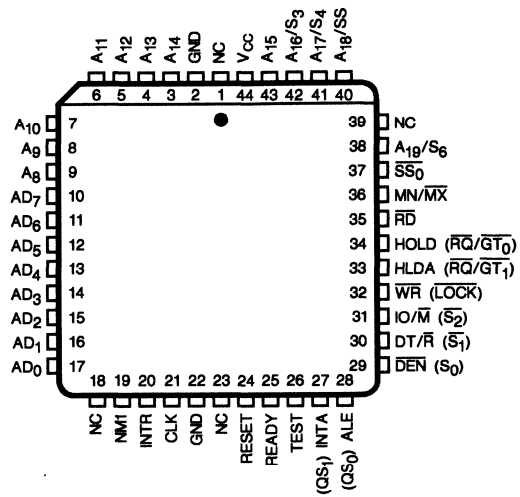
Top View

DIPs



CD005520

PLCC



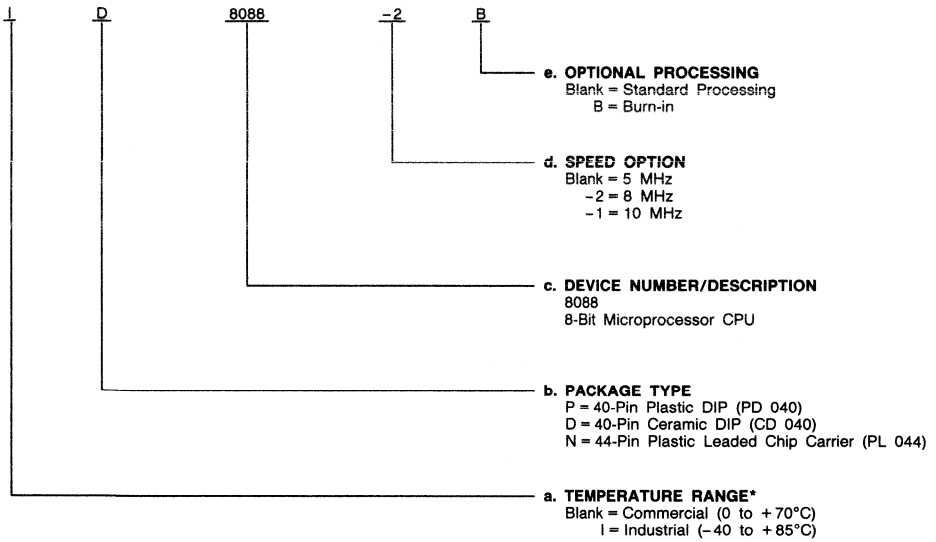
CD010680

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D, N	8088
	8088-2
	8088-1
D	8088B
	8088-2B
	8088-1B
ID	8088B
	8088-2B

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

PIN DESCRIPTION

The following pin function descriptions are for 8088 systems in either minimum or maximum mode. The "local bus" in these descriptions is the direct multiplexed bus interface connection to the 8088 (without regard to additional bus buffers).

Pin No.*	Name	I/O	Description
9-16	AD_7–AD_0	I/O	Address Data Bus. These lines constitute the time multiplexed memory/IO address (T_1) and data (T_2, T_3, T_W, and T_4) bus. These lines are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge."
39, 2-8	A_{15}–A_8	O	Address Bus. These lines provide address bits 8 through 15 for the entire bus cycle (T_1–T_4). These lines do not have to be latched by ALE to remain valid. A_{15}–A_8 are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge."
35-38	A_{19}/S_6, A_{18}/S_5, A_{17}/S_4, A_{16}/S_3	O	Address/Status. During T_1, these are the four most significant address lines for memory operations. During I/O operations, these lines are LOW. During memory and I/O operations, status information is available on these lines during T_2, T_3, T_W, and T_4. S_6 is always LOW. The status of the interrupt enable flat bit (S_5) is updated at the beginning of each clock cycle. S_4 and S_3 are encoded as shown. This information indicates which segment register is presently being used for data accessing. These lines float to 3-state OFF during local bus "hold acknowledge."
32	$\overline{RD}$	O	Read. Read strobe indicates that the processor is performing a memory or I/O read cycle, depending on the state of the IO/$\overline{M}$ pin or S_2. This signal is used to read devices which reside on the 8088 local bus. $\overline{RD}$ is active LOW during T_2, T_3 and T_W of any read cycle, and is guaranteed to remain HIGH in T_2 until the 8088 local bus has floated. This signal floats to 3-state OFF in "hold acknowledge."
22	READY	I	READY. The acknowledgement from the addressed memory or I/O device that it will complete the data transfer. The RDY signal from memory or I/O is synchronized by the 8284 clock generator to form READY. This signal is active HIGH. The 8088 READY input is not synchronized. Correct operation is not guaranteed if the set-up and hold times are not met.
18	INTR	I	Interrupt Request. A level-triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.
23	$\overline{TEST}$	I	$\overline{TEST}$. Input is examined by the "wait for test" instruction. If the $\overline{TEST}$ input is LOW, execution continues; otherwise, the processor waits in an "Idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.
17	NMI	I	Non-Maskable Interrupt. An edge-triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This Input is internally synchronized.
21	RESET	I	RESET. Causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution, as described in the instruction set description, when RESET returns LOW. RESET is internally synchronized.
19	CLK	I	Clock. Provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
40	V_{CC}		V_{CC}. The +5V ±10% power supply pin.
1, 20	GND		GND. The ground pins.
33	MIN/$\overline{MX}$	I	Minimum/Maximum. Indicates what mode the processor is to operate in. The two modes are discussed in the following sections.
28	IO/$\overline{M}$	O	Status Line. An inverted maximum mode $\overline{S_2}$. It is used to distinguish a memory access from an I/O access. IO/$\overline{M}$ becomes valid in the T_4 preceding a bus cycle and remains valid until the final T_4 of the cycle (I/O = HIGH, M = LOW). IO/$\overline{M}$ floats to 3-state OFF in local bus "hold acknowledge."
29	$\overline{WR}$	O	Write. Strobe indicates that the processor is performing a write memory or write I/O cycle, depending on the state of the IO/$\overline{M}$ signal. WR is active for T_2, T_3, and T_W of any write cycle. It is active LOW and floats to 3-state OFF in local bus "hold acknowledge."
24	$\overline{INTA}$	O	INTA. Used as a read strobe for interrupt acknowledge cycles. It is active LOW during T_2, T_3 and T_W of each interrupt acknowledge cycle.
25	ALE	O	Address Latch Enable. Provided by the processor to latch the address into 8282/8283 address latch. It is a HIGH pulse active during clock low of T_1 of any bus cycle. Note that ALE is never floated.
27	DT/$\overline{R}$	O	Data Transmit/Receive. Needed in a minimum system that desires to use an 8286/8287 data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically DT/$\overline{R}$ is equivalent to $\overline{S_1}$ in the maximum mode, and its timing is the same as for IO/$\overline{M}$ (T = HIGH, R = LOW.) This signal floats to 3-state OFF in local bus "hold acknowledge."
26	$\overline{DEN}$	O	Data Enable. Provided as an output enable for the 8286/8287 in a minimum system that uses the transceiver. $\overline{DEN}$ is active LOW during each memory and I/O access and for $\overline{INTA}$ cycles. For a read or $\overline{INTA}$ cycle, it is active from the middle of T_2 until the middle of T_4; while for a write cycle, it is active from the beginning of T_2 until the middle of T_4. $\overline{DEN}$ floats to 3-state OFF during local bus "hold acknowledge."

S_4	S_3	Characteristics
0 (LOW)	0	Alternate Data
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data
S_6 is 0 (LOW)		

*Pin numbers correspond to DIPs only.

PIN DESCRIPTION (Cont.)

Pin No.*	Name	I/O	Description
31, 30	HOLD, HLDA	I/O	HOLD. Indicates that another master is requesting a local bus "hold." To be acknowledged, HOLD must be active HIGH. The processor receiving the "hold" request will issue HLDA (HIGH) as an acknowledgement in the middle of a T_4 or T_1 clock cycle. Simultaneous with the issuance of HLDA, the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor lowers HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines. HOLD is not an asynchronous input. External synchronization should be provided if the system cannot otherwise guarantee the set-up time.
34	$\overline{SSO}$	O	Status Line. Logically equivalent to $\overline{SO}$ in the maximum mode. The combination of $\overline{SSO}$, IO/$\overline{M}$ and DT/$\overline{R}$ allows the system to completely decode the current bus cycle status.

IO/$\overline{M}$	DT/$\overline{R}$	$\overline{SSO}$	Characteristics
1 (HIGH)	0	0	Interrupt Acknowledge
1	0	1	Read I/O port
1	1	0	Write I/O port
1	1	1	Halt
0 (LOW)	0	0	Code Access
0	0	1	Read memory
0	1	0	Write memory
0	1	1	Passive

28-26	$\overline{S}_2$, $\overline{S}_1$, $\overline{S}_0$	O	Status. Active during clock high of T_4, T_1, and T_2 and is returned to the passive state (1, 1, 1) during T_3 or during T_W when READY is HIGH. This status is used by the 8288 bus controller to generate all memory and I/O access control signals. Any change by $\overline{S}_2$, $\overline{S}_1$, or $\overline{S}_0$ during T_4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T_3 or T_W is used to indicate the end of a bus cycle. These signals float to 3-state OFF during "hold acknowledge." During the first clock cycle after RESET becomes active, these signals are active HIGH. After this first clock, they float to 3-state OFF.

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	Characteristics
0 (LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1 (HIGH)	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

31, 30	$\overline{RQ}/\overline{GT}_0$, $\overline{RQ}/\overline{GT}_1$	I/O	Request/Grant. Pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with $\overline{RQ}/\overline{GT}_0$ having higher priority than $\overline{RQ}/\overline{GT}_1$. $\overline{RQ}/\overline{GT}$ has an internal pull-up resistor, so may be left unconnected. The request/grant sequence is as follows: 1. A pulse of one CLK wide from another local bus master indicates a local bus request ("hold") to the 8088 (pulse 1). 2. During a T_4 or T_1 clock cycle, a pulse one clock wide from the 8088 to the requesting master (pulse 2), indicates that the 8088 has allowed the local bus to float and that it will enter the "hold acknowledge" state at the next CLK. The CPU's bus interface unit is disconnected logically from the local bus during "hold acknowledge." The same rules as for HOLD/HLDA apply as for when the bus is released. 3. A pulse one CLK wide from the requesting master indicates to the 8088 (pulse 3) that the "hold" request is about to end and that the 8088 can reclaim the local bus at the next CLK. The CPU then enters T_4. Each master-master exchange of the local bus is a sequence of three pulses. There must be one idle CLK cycle after each bus exchange. Pulses are active LOW. If the request is made while the CPU is performing a memory cycle, it will release the local bus during T_4 of the cycle when all the following conditions are met: 1. Request occurs on or before T_2. 2. Current cycle is not the low bit of a word. 3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence. 4. A locked instruction is not currently executing. If the local bus is Idle when the request is made, two possible events will follow: 1. Local bus will be released during the next clock. 2. A memory cycle will start within 3 clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.
29	$\overline{LOCK}$	O	$\overline{LOCK}$. Indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active (LOW). The $\overline{LOCK}$ signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW and floats to 3-state off in "hold acknowledge."

*Pin numbers correspond to DIPs only.

PIN DESCRIPTION (Cont.)

Pin No.*	Name	I/O	Description
24, 25	QS_1, QS_0	O	Queue Status. Provides status to allow external tracking of the internal 8088 instruction queue. The queue status is valid during the CLK cycle after which the queue operation is performed.
34	–	O	Pin 34 is always HIGH in the maximum mode.

QS_1	QS_0	Characteristics
0 (LOW)	0	No Operation
0	1	First Byte of Opcode from Queue
1 (HIGH)	0	Empty the Queue
1	1	Subsequent Byte from Queue

*Pin numbers correspond to DIPs only.

DETAILED DESCRIPTION

The 8088 Compared to the 8086

- The queue length is 4 bytes in the 8088; whereas, the 8086 queue contains 6 bytes, or three words. The queue was shortened to prevent overuse of the bus by the BIU when prefetching instructions. This was required because of the additional time necessary to fetch instructions 8 bits at a time.
- To further optimize the queue, the prefetching algorithm was changed. The 8088 BIU will fetch a new instruction to load into the queue each time there is a 1 byte hole (space available) in the queue. The 8086 waits until a 2-byte space is available.
- The internal execution time of the instruction set is affected by the 8-bit interface. All 16-bit fetches and writes from/to memory take an additional four clock cycles. The CPU is also limited by the speed of instruction fetches. This latter problem only occurs when a series of simple operations occurs. When the more sophisticated instructions of the 8088 are being used, the queue has time to fill and the execution proceeds as fast as the execution unit will allow.

The 8088 and 8086 are completely software compatible by virtue of their identical execution units. Software that is system dependent may not be completely transferable, but software that is not system dependent will operate equally as well on an 8088 or an 8086.

The hardware interface of the 8088 contains the major differences between the two CPUs. The pin assignments are nearly identical, however, with the following functional changes:

- $A_8 - A_{15}$ – These pins are only address outputs on the 8088. These address lines are latched internally and remain valid throughout a bus cycle in a manner similar to the 8085 upper address lines.
- $\overline{BHE}$ has no meaning on the 8088 and has been eliminated.
- $\overline{SSO}$ provides the $\overline{S0}$ status information in the minimum mode. This output occurs on pin 34 in minimum mode only. DT/$\overline{R}$, IO/$\overline{M}$, and $\overline{SSO}$ provide the complete bus status in minimum mode.
- IO/$\overline{M}$ has been inverted to be compatible with the MCS-85 bus structure.
- ALE is delayed by one clock cycle in the minimum mode when entering HALT, to allow the status to be latched with ALE.

I/O Addressing

In the 8088, I/O operations can address up to a maximum of 64K I/O registers. The I/O address appears in the same format as the memory address on bus lines $A_{15} - A_0$. The address lines $A_{19} - A_{16}$ are zero in I/O operations. The variable I/O instructions, which use register DX as a pointer, have full address capability, while the direct I/O instructions directly address one or two of the 256 I/O byte locations in page 0 of the I/O address space. I/O ports are addressed in the same manner as memory locations.

Designers familiar with the 8085 or upgrading an 8085 design should note that the 8085 addresses I/O with an 8-bit address on both halves of the 16-bit address bus. The 8088 uses a full 16-bit address of its lower 16 address lines.

Bus Operation

The 8088 address/data bus is broken into three parts – the lower eight address/data bits ($AD_0 - AD_7$), the middle eight address bits ($A_8 - A_{15}$), and the upper four address bits (A_{16}-A_{19}). The address/data bits and the highest four address bits are time multiplexed. This technique provides the most efficient use of pins on the processor, permitting the use of a standard 40 lead package. The middle eight address bits are not multiplexed; i.e., they remain valid throughout each bus cycle. In addition, the bus can be demultiplexed at the processor with a single address latch if a standard, non-multiplexed bus is desired for the system.

Each processor bus cycle consists of at least four CLK cycles. These are referred to as T1, T2, T3, and T4. The address is emitted from the processor during T1 and data transfer occurs on the bus during T3 and T4. T2 is used primarily for changing the direction of the bus during read operations. In the event that a "NOT READY" indication is given by the addressed device, "wait" states (Tw) are inserted between T3 and T4. Each inserted "wait" state is of the same duration as a CLK cycle. Periods can occur between 8088 driven bus cycles. These are referred to as "idle" states (Ti), or inactive CLK cycles. The processor uses these cycles for internal housekeeping.

During T1 of any bus cycle, the ALE (address latch enable), signal is emitted (by either the processor or the 8288 bus controller, depending on the MN/$\overline{MX}$ strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

Status bits $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$ are used by the bus controller, in maximum mode, to identify the type of bus transaction according to the following table:

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	Characteristics
0 (LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1 (HIGH)	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

Status bits S3 through S6 are multiplexed with high order address bits and are therefore valid during T2 through T4. S3 and S4 indicate which segment register was used for this bus cycle in forming the address according to the following table:

S_4	S_3	Characteristics
0 (LOW)	0	Alternate Data (extra segment)
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data

S5 is a reflection of the PSW interrupt enable bit. S6 is always equal to 0.

External Interface

Processor Reset and Initialization

Processor initialization or start up is accomplished with activation (HIGH) of the RESET pin. The 8088 RESET is required to be HIGH for greater than four clock cycles. The 8088 will terminate operations on the high-going edge of RESET and will remain dormant as long as RESET is HIGH. The low-going transition of RESET triggers an internal reset sequence for approximately 7 clock cycles. After this interval the 8088 operates normally, beginning with the instruction in absolute location FFFF0H. (See Figure 3.) The RESET input is internally synchronized to the processor clock. At initialization, the HIGH to LOW transition of RESET must occur no sooner than 50 μs after power up, to allow complete initialization of the 8088.

If INTR is asserted sooner than nine clock cycles after the end of RESET, the processor may execute one instruction before responding to the interrupt.

All 3-state outputs float to 3-state OFF during RESET. Status is active in the idle state for the first clock after RESET becomes active and then floats to 3-state OFF.

Interrupt Operations

Interrupt operations fall into two classes: software or hardware initiated. The software initiated interrupts and software aspects of hardware interrupts are specified in the instruction set description in the iAPX 88 book or the iAPX 86, 88 User's Manual. Hardware interrupts can be classified as nonmaskable or maskable.

Interrupts result in a transfer of control to a new program location. A 256 element table containing address pointers to the interrupt service program locations resides in absolute locations 0 through 3FFH (see Figure 3), which are reserved for this purpose. Each element in the table is 4 bytes in size and corresponds to an interrupt "type." An interrupting device supplies an 8-bit type number, during the interrupt acknowledge sequence, which is used to vector through the appropriate element to the new interrupt service program location.

Non-Maskable Interrupt (NMI)

The processor provides a single non-maskable interrupt (NMI) pin which has higher priority than the maskable interrupt request (INTR) pin. A typical use would be to activate a power failure routine. The NMI is edge-triggered on a LOW to HIGH transition. The activation of this pin causes a type 2 interrupt.

NMI is required to have a duration in the HIGH state of greater than two clock cycles, but is not required to be synchronized to the clock. Any higher going transition of NMI is latched on-chip and will be serviced at the end of the current instruction or between whole moves (2 bytes in the case of word moves) of a block type instruction. Worst case response to NMI would be for multiply, divide, and variable shift instructions. There is no specification on the occurrence of the low-going edge; it may occur before, during, or after the servicing of NMI. Another high-going edge triggers another response if it occurs after the start of the NMI procedure. The signal must be free of logical spikes in general and be free of bounces on the low-going edge to avoid triggering extraneous responses.

Maskable Interrupt (INTR)

The 8088 provides a single interrupt request input (INTR) which can be masked internally by software with the resetting of the interrupt enable (IF) flag bit. The interrupt request signal is level triggered. It is internally synchronized during each clock cycle on the high-going edge of CLK. To be responded to, INTR must be present (HIGH) during the clock period preceding the end of the current instruction or the end of a whole move for a block type instruction. During interrupt response sequence, further interrupts are disabled. The enable bit is reset as part of the response to any interrupt (INTR, NMI, software interrupt, or single step), although the FLAGS register which is automatically pushed onto the stack reflects the state of the processor prior to the interrupt. Until the old FLAGS register is restored, the enable bit will be zero unless specifically set by an instruction.

During the response sequence (See Figure 1), the processor executes two successive (back to back) interrupt acknowledge cycles. The 8088 emits the LOCK signal (maximum mode only) from T2 of the first bus cycle until T2 of the second. A local bus "hold" request will not be honored until the end of the second bus cycle. In the second bus cycle, a byte is fetched from the external interrupt system (e.g., 8259A PIC) which identifies the source (type) of the interrupt. This byte is multiplied by four and used as a pointer into the interrupt vector lookup table. An INTR signal left HIGH will be continually responded to within the limitations of the enable bit and sample period. The interrupt return instruction includes a flags pop which returns the status of the original interrupt enable bit when it restores the flags.

HALT

When a software HALT instruction is executed, the processor indicates that it is entering the HALT state in one of two ways, depending upon which mode is strapped. In minimum mode, the processor issues ALE, delayed by one clock cycle, to allow the system to latch the halt status. Halt status is available on IO/$\overline{M}$, DT/$\overline{R}$, and $\overline{SSO}$. In maximum mode, the processor issues appropriate HALT status on $\overline{S2}$, $\overline{S1}$, and $\overline{S0}$, and the 8288 bus controller issues one ALE. The 8088 will not leave the HALT state when a local bus hold is entered while in HALT. In this case, the processor reissues the HALT indicator at the end of the local bus hold. An interrupt request or RESET will force the 8088 out of the HALT state.

Read/Modify/Write (Semaphore) Operations via LOCK

The LOCK status information is provided by the processor when consecutive bus cycles are required during the execution of an instruction. This allows the processor to perform read/modify/write operations on memory (via the "exchange register with memory" instruction), without another system bus master receiving intervening memory cycles. This is useful in multiprocessor system configurations to accomplish "test and set lock" operations. The $\overline{LOCK}$ signal is activated (LOW) in the clock cycle following decoding of the LOCK prefix instruction. It is deactivated at the end of the last bus cycle of the instruction following the LOCK prefix. While $\overline{LOCK}$ is active, a request on a $\overline{RQ}$/$\overline{GT}$ pin will be recorded, and then honored at the end of the LOCK.

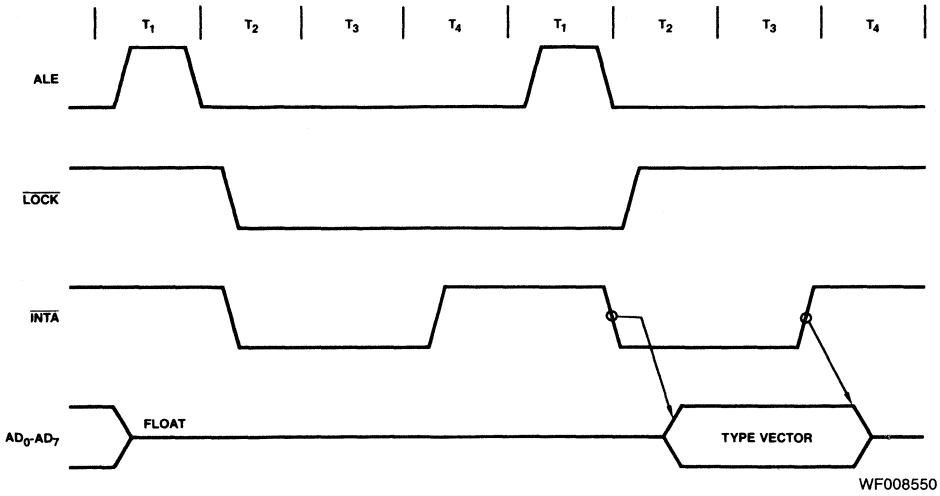


Figure 1. Interrupt Acknowledge Sequence

External Synchronization via TEST

As an alternative to interrupts, the 8088 provides a single software-testable input pin ($\overline{TEST}$). This input is utilized by executing a WAIT instruction. The single WAIT instruction is repeatedly executed until the $\overline{TEST}$ input goes active (LOW). The execution of WAIT does not consume bus cycles once the queue is full.

If a local bus request occurs during WAIT execution, the 8088 3-states all output drivers. If interrupts are enabled, the 8088 will recognize interrupts and process them. The WAIT instruction is then refetched, and reexecuted.

Basic System Timing

In minimum mode, the MN/$\overline{MX}$ pin is strapped to V_{CC} and the processor emits bus control signals compatible with the 8085 bus structure. In maximum mode, the MN/$\overline{MX}$ pin is strapped to GND and the processor emits coded status information which the 8288 bus controller uses to generate MULTIBUS compatible bus control signals.

System Timing — Minimum System

The read cycle begins in T1 with the assertion of the address latch enable (ALE) signal. The trailing (low going) edge of this signal is used to latch the address information, which is valid on the address/data bus (AD0 – AD7) at this time, into the 8282/8283 latch. Address lines A8 through A15 do not need to be latched because they remain valid throughout the bus cycle. From T1 to T4 the IO/$\overline{M}$ signal indicates a memory or I/O operation. At T2 the address is removed from the address/data bus and the bus goes to a high impedance state. The read control signal is also asserted at T2. The read ($\overline{RD}$) signal causes the addressed device to enable its data bus drivers to the local bus. Some time later, valid data will be available on the bus and the addressed device will drive the READY line HIGH. When the processor returns the read signal to a HIGH level, the addressed device will again 3-state its bus drivers. If a transceiver (8286/8287) is required to buffer the 8088 local bus, signals DT/$\overline{R}$ and $\overline{DEN}$ are provided by the 8088.

A write cycle also begins with the assertion of ALE and the emission of the address. The IO/$\overline{M}$ signal is again asserted to indicate a memory or I/O write operation. In T2, immediately following the address emission, the processor emits the data to be written into the addressed location. This data remains valid until at least the middle of T4. During T2, T3, and T_W, the processor asserts the write control signal. The write ($\overline{WR}$) signal becomes active at the beginning of T2, as opposed to the read, which is delayed somewhat into T2 to provide time for the bus to float.

The basic difference between the interrupt acknowledge cycle and a read cycle is that the interrupt acknowledge ($\overline{INTA}$) signal is asserted in place of the read ($\overline{RD}$) signal and the address bus is floated. (See Figure 1.) In the second of two successive $\overline{INTA}$ cycles, a byte of information is read from the data bus, as supplied by the interrupt system logic (i.e. 8259A priority interrupt controller). This byte identifies the source (type) of the interrupt. It is multiplied by four and used as a pointer into the interrupt vector lookup table, as described earlier.

Bus Timing — Medium Complexity Systems

For medium complexity systems, the MN/$\overline{MX}$ pin is connected to GND and the 8288 bus controller is added to the system, as well as an 8282/8283 latch for latching the system address, and an 8286/8287 transceiver to allow for bus loading greater than the 8088 is capable of handling. Signals ALE, $\overline{DEN}$, and DT/$\overline{R}$ are generated by the 8288 instead of the processor in this configuration, although their timing remains relatively the same. The 8088 status outputs ($\overline{S2}$, $\overline{S1}$, and $\overline{S0}$) provide type of cycle information and become 8288 inputs. This bus cycle information specifies read (code, data, or I/O), write (data or I/O), interrupt acknowledge, or software halt. The 8288 thus issues control signals specifying memory read or write, I/O read or write, or interrupt acknowledge. The 8288 provides two types of write strobes, normal and advanced, to be applied as required. The normal write strobes have data valid at the leading edge of write. The advanced write strobes have the same timing as read strobes, and hence, data is not valid at the leading edge of write. The 8286/8287 transceiver receives the usual T and $\overline{OE}$ inputs from the 8288's DT/$\overline{R}$ and $\overline{DEN}$ outputs.

The pointer into the interrupt vector table, which is passed during the second $\overline{INTA}$ cycle, can derive from an 8259A located on either the local bus or the system bus. If the master 8289A priority interrupt controller is positioned on the local bus, a TTL gate is required to disable the 8286/8287 transceiver when reading from the master 8259A during the interrupt acknowledge sequence and software "poll".

Memory Organization

The processor provides a 20-bit address to memory which locates the byte being referenced. The memory is organized as a linear array of up to 1 million bytes, addressed as 00000(H) to FFFFF(H). The memory is logically divided into code, data, extra data, and stack segments of up to 64K bytes each, with each segment falling on 16-byte boundaries. (See Figure 2.)

All memory references are made relative to base addresses contained in high speed segment registers. The segment types were chosen based on the addressing needs of programs. The segment register to be selected is automatically chosen according to the rules of the following table. All information in one segment type share the same logical attributes (e.g. code or data). By structuring memory into relocatable areas of similar characteristics and by automatically selecting segment registers, programs are shorter, faster, and more structured.

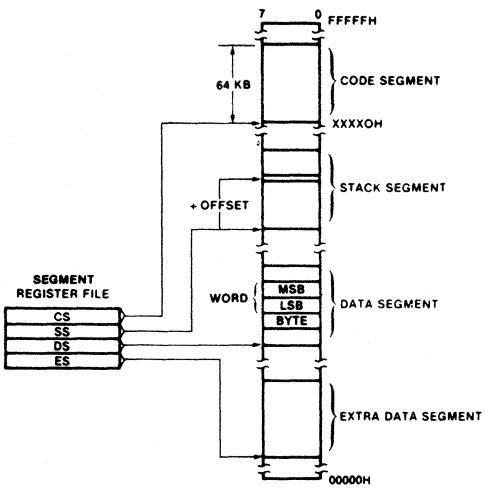


Figure 2. Memory Organization

Word (16-bit) operands can be located on even or odd address boundaries. For address and data operands, the least significant byte of the word is stored in the lower valued address location and the most significant byte in the next higher address location. The BIU will automatically execute two fetch or write cycles for 16-bit operands.

Certain locations in memory are reserved for specific CPU operations. (See Figure 3.) Locations from addresses FFFF0H through FFFFFH are reserved for operations including a jump to the initial system initialization routine. Following RESET, the CPU will always begin execution at location FFFF0H where the jump must be located. Locations 00000H through 003FFH are reserved for interrupt operations. Four-byte pointers consisting of a 16-bit segment address and a 16-bit offset address direct program flow to one of the 256 possible interrupt service routines. The pointer elements are assumed to have been stored at their respective places in reserved memory prior to the occurrence of interrupts.

Minimum and Maximum Modes

The requirements for supporting minimum and maximum 8088 systems are sufficiently different that they cannot be done efficiently with 40 uniquely defined pins. Consequently, the 8088 is equipped with a strap pin (MN/$\overline{MX}$) which defines the system configuration. The definition of a certain subset of the pins changes, dependent on the condition of the strap pin. When the MN/$\overline{MX}$ pin is strapped to GND, the 8088 defines pins 24 through 31 and 34 in maximum mode. When the MN/$\overline{MX}$ pin is strapped to V_{CC}, the 8088 generates bus control signals itself on pins 24 through 31 and 34.

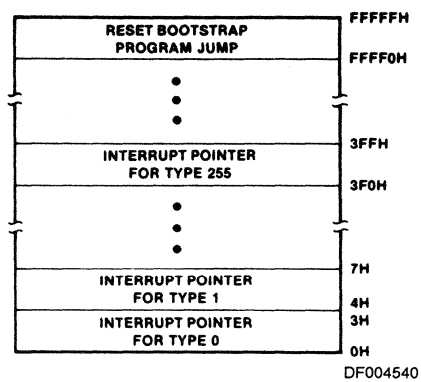


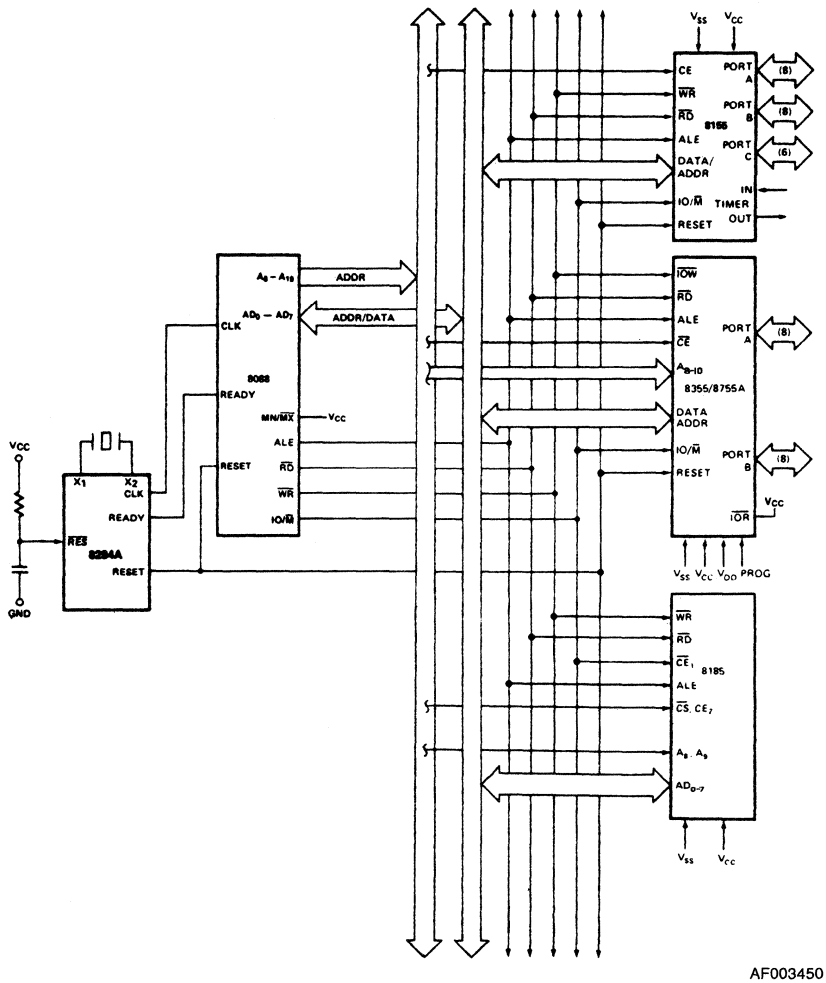
Figure 3. Reserved Memory Locations

Memory Reference Need	Segment Register Used	Segment Selection Rule
Instructions	CODE (CS)	Automatic with all instruction prefetch.
Stack	STACK (SS)	All stack pushes and pops. Memory references relative to BP base register except data references.
Local Data	DATA (DS)	Data references when: relative to stack, destination of string operation, or explicitly overridden.
External (Global) Data	EXTRA (ES)	Destination of string operations: Explicitly selected using a segment override.

The minimum mode 8088 can be used with either a multiplexed or demultiplexed bus. The multiplexed bus configuration is compatible with the MCS-85™ multiplexed bus peripherals (8155, 8156, 8355, 8755A, and 8185). This configuration (see Figure 4) provides the user with a minimum chip count system. This architecture provides the 8088 processing power in a highly integrated form.

The demultiplexed mode requires one latch (for 64K addressability) or two latches (for a full megabyte of addressing). A third latch can be used for buffering if the address bus loading requires it. An 8286 or 8287 transceiver can also be used if data bus buffering is required. (See Figure 5.) The 8088 provides $\overline{DEN}$ and DT/$\overline{R}$ to control the transceiver, and ALE to latch the addresses. This configuration of the minimum mode provides the standard demultiplexed bus structure with heavy bus buffering and relaxed bus timing requirements.

The maximum mode employs the 8288 bus controller. (See Figure 6.) The 8288 decodes status lines $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$, and provides the system with all bus control signals. Moving the bus control to the 8288 provides better source and sink current capability to the control lines, and frees the 8088 pins for extended large system features. Hardware lock, queue status, and two request/grant interfaces are provided by the 8088 in maximum mode. These features allow co-processors in local bus and remote bus configurations.



AF003450

Figure 4. Multiplexed Bus Configuration

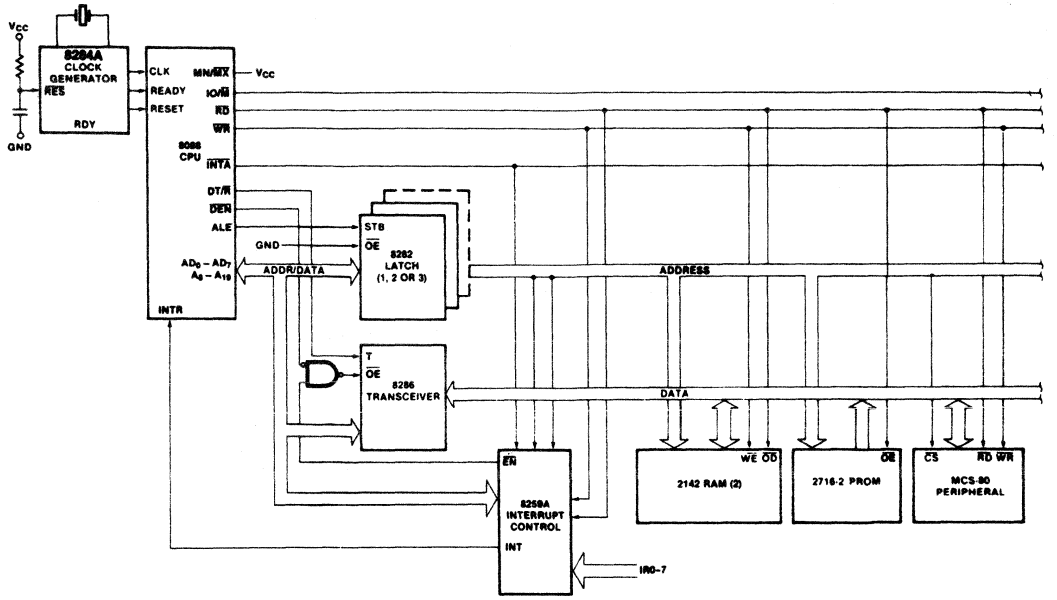


Figure 5. Demultiplexed Bus Configuration

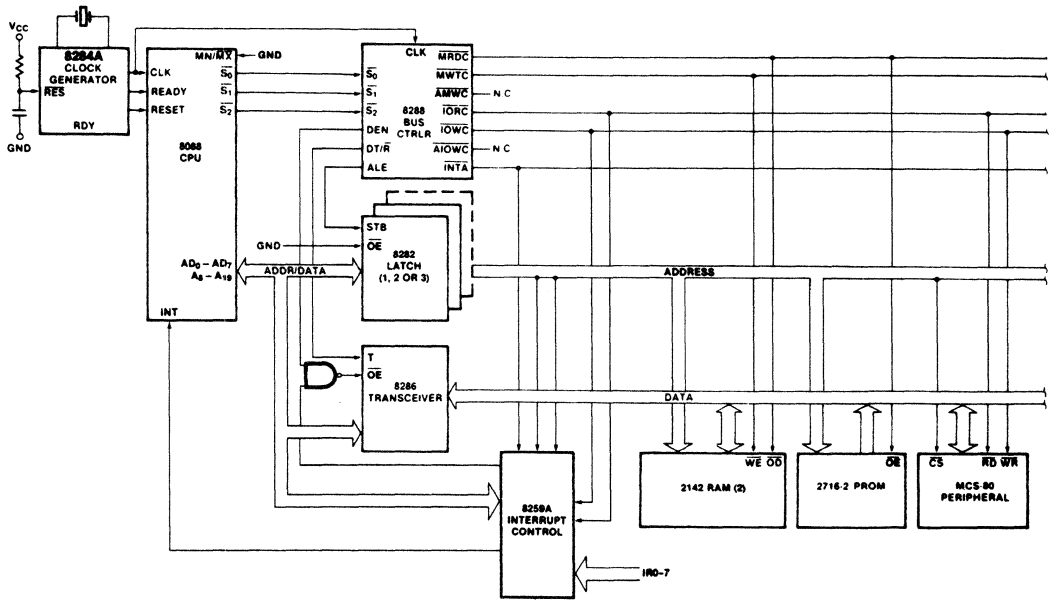


Figure 6. Fully Buffered System Using Bus Controller

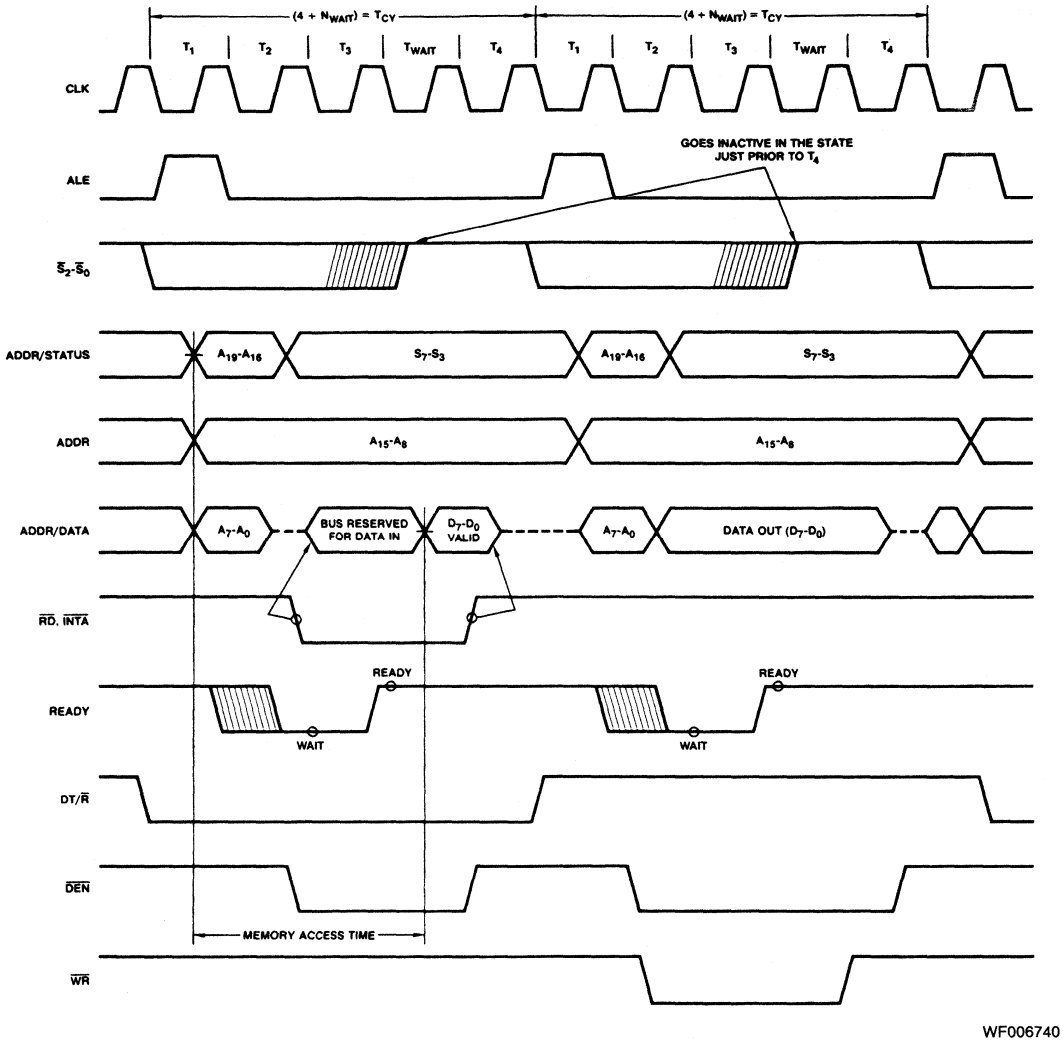
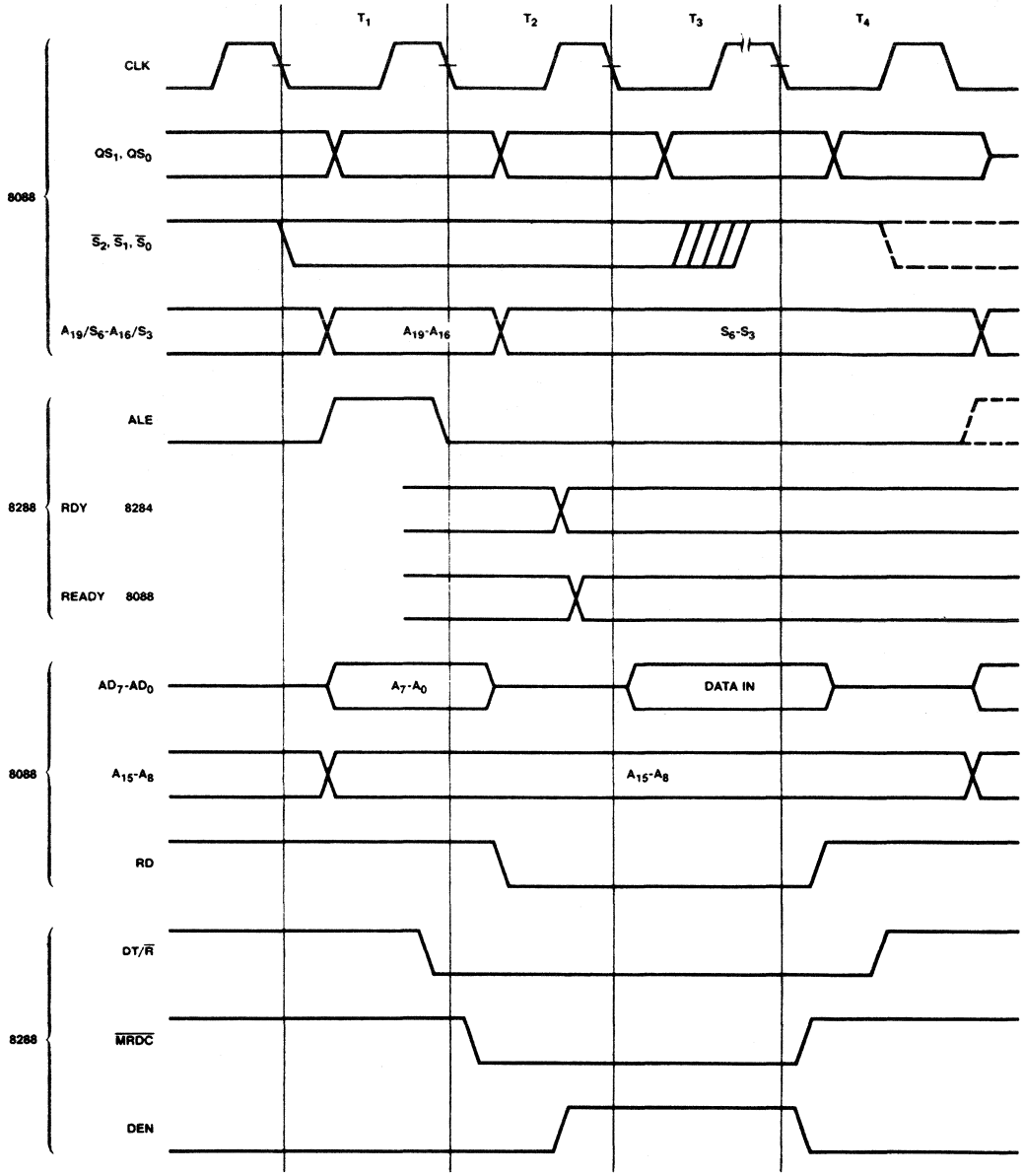


Figure 7. Basic System Timing



WF006750

Figure 8. Medium Complexity System Timing

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on any Pin
with Respect to Ground ..... −1.0 to +7.0V
Power Dissipation 2.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC})
8088 ..... 5 V ± 10%
8088-1, 8088-2 ..... 5 V ± 5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC})
8088 ..... 5 V ± 10%
8088-1, 8088-2 ..... 5 V ± 5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS

Parameters	Description	Test Conditions		Min	Max	Units
V_{IL}	Input Low Voltage	(Note 1)		−0.5	+0.8	V
V_{IH}	Input High Voltage	(Notes 1, 2)		2.0	V_{CC} +0.5	V
V_{OL}	Output Low Voltage	I_{OL} = 2.0mA			0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400μA		2.4		V
I_{CC}	Power Supply Current				340	mA
I_{LI}	Input Leakage Current	$0V \leq V_{IN} \leq V_{CC}$			±10	μA
I_{LO}	Output Leakage Current	$0.45V \leq V_{OUT} \leq V_{CC}$			±10	μA
V_{CL}	Clock Input Low Voltage			−0.5	+0.6	V
V_{CH}	Clock Input High Voltage			3.9	V_{CC} +1.0	V
C_{IN}	Capacitance of Input Buffer (All input except AD_0–AD_7, RQ/GT)	fc = 1 MHz			15	pF
C_{IO}	Capacitance of I/O Buffer (AD_0–AD_7, RQ/GT)	fc = 1 MHz			15	pF
I_{CC}	Power Supply Current	T_A = 25°C	8088		340	mA
			8088-1, -2		350	
			P8088		250	

SWITCHING CHARACTERISTICS

MINIMUM COMPLEXITY SYSTEM TIMING REQUIREMENTS

Parameters	Description	Test Conditions	8088		8088-2		8088-1		Units
			Min	Max	Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	100	500	ns
TCLCH	CLK Low Time		118		68		53		ns
TCHCL	CLK High Time		69		44		39		ns
TCH1CH2	CLK Rise Time	From 1.0 to 3.5V		10		10		10	ns
TCL2CL1	CLK Fall Time	From 3.5 to 1.0V		10		10		10	ns
TDVCL	Data in Set-up Time		30		20		5		ns
TCLDX	Data in Hold Time		10		10		10		ns
TR1VCL	RDY Set-up Time into 8284 (See Notes 3, 4)		35		35		35		ns
TCLR1X	RDY Hold Time into 8284 (See Notes 3, 4)		0		0		0		ns
TRYHCH	READY Set-up Time into 8088		118		68		53		ns
TCHRYX	READY Hold Time into 8088		30		20		20		ns
TRYLCL	READY Inactive to CLK (See Note 5)		−8		−8		−10		ns
THVCH	HOLD Set-up Time		35		20		20		ns
TINVCH	INTR, NMI, $\overline{TEST}$ Set-up Time (See Note 4)		30		15		15		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0V		20		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8V		12		12		12	ns

Notes: 1. V_{IL} tested with MN/$\overline{MX}$ pin = 0 V; V_{IH} tested with MN/$\overline{MX}$ pin = 5 V; MN/$\overline{MX}$ is a strap pin.
2. Not applicable to $\overline{RQ}/\overline{GT0}$ and $\overline{RQ}/\overline{GT1}$ pins (pins 30 and 31).
3. Signal at 8284 or 8288 shown for reference only.
4. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
5. Applies only to T_3 and Wait states.
6. Applies only to T_2 state (8 ns into T_3 state).

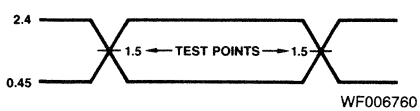
See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS (Cont.)

TIMING RESPONSES

Parameters	Description	Test Conditions	8088		8088-2		8088-1		Units
			Min	Max	Min	Max	Min	Max	
TCLAV	Address Valid Delay	C_L = 20-100 pF for all 8088 Outputs (in addition to internal loads)	10	110	10	60	10	50	ns
TCLAX	Address Hold Time		10		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	10	40	ns
TLHLL	ALE Width		TCLCH -20		TCLCH -10		TCLCH -10		ns
TCLLH	ALE Active Delay			80		50		40	ns
TCHLL	ALE Inactive Delay			85		55		45	ns
TLLAX	Address Hold Time to ALE Inactive		TCHCL -10		TCHCL -10		TCHCL -10		ns
TCLDV	Data Valid Delay		10	110	10	60	10	50	ns
TCHDX	Data Hold Time		10		10		10		ns
TWHDX	Data Hold Time After $\overline{WR}$		TCLCH -30		TCLCH -30		TCLCH -25		ns
TCVCTV	Control Active Delay 1		10	110	10	70	10	50	ns
TCHCTV	Control Active Delay 2		10	110	10	60	10	45	ns
TCVCTX	Control Inactive Delay		10	110	10	70	10	50	ns
TAZRL	Address Float to READ Active		0		0		0		ns
TCLRL	$\overline{RD}$ Active Delay		10	165	10	100	10	70	ns
TCLRH	$\overline{RD}$ Inactive Delay		10	150	10	80	10	60	ns
TRHAV	$\overline{RD}$ Inactive to Next Address Active		TCLCL -45		TCLCL -40		TCLCL -35		ns
TCLHAV	HLDA Valid Delay		10	160	10	100	10	60	ns
TRLRH	$\overline{RD}$ Width		2TCLCL -75		2TCLCL -50		2TCLCL -40		ns
TWLWH	$\overline{WR}$ Width		2TCLCL -60		2TCLCL -40		2TCLCL -35		ns
TAVAL	Address Valid to ALE Low		TCLCH -60		TCLCH -40		TCLCH -35		ns
TOLOH	Output Rise Time	From 0.8 to 2.0 V		20		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8 V		12		12		12	ns

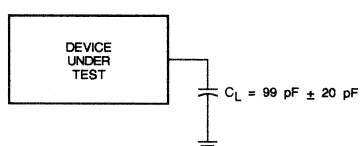
SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF006760

AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". The clock is driven at 4.3V and 0.25V. Timing measurements are made at 1.5V for both a logic "1" and "0."

SWITCHING TEST LOAD CIRCUIT



WF006771

C_L Includes JIG Capacitance.

SWITCHING CHARACTERISTICS (Cont.)

MAX MODE SYSTEM (USING 8288 BUS CONTROLLER) TIMING REQUIREMENTS

Parameters	Description	Test Conditions	8088		8088-2		8088-1		Units
			Min	Max	Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	100	500	ns
TCLCH	CLK Low Time		118		68		53		ns
TCHCL	CLK High Time		69		44		39		ns
TCH1CH2	CLK Rise Time	From 1.0 to 3.5V		10		10		10	ns
TCL2CL1	CLK Fall Time	From 3.5 to 1.0V		10		10		10	ns
TDVCL	Data in Set-up Time		30		20		5		ns
TCLDX	Data in Hold Time		10		10		10		ns
TR1VCL	RDY Set-up Time into 8284 (See Notes 1, 2)		35		35		35		ns
TCLR1X	RDY Hold Time into 8284 (See Notes 1,2)		0		0		0		ns
TRYHCH	READY Set-up Time into 8088		118		68		53		ns
TCHRYX	READY Hold Time into 8088		30		20		20		ns
TRYLCL	READY Inactive to CLK (See Note 4)		-8		-8		-10		ns
TINVCH	Set-up Time for Recognition (INTR, NMI, TEST) (See Note 2)		30		15		15		ns
TGVCH	RQ/GT Set-up Time		30		15		12		ns
TCHGX	RQ Hold Time into 8086		40		30		20		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0V		20		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8V		12		12		12	ns

Notes: 1. Signal at 8284 or 8288 shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T_3 and Wait states.
4. Applies only to T_2 state (8ns into T_3 state).

SWITCHING CHARACTERISTICS (Cont.)

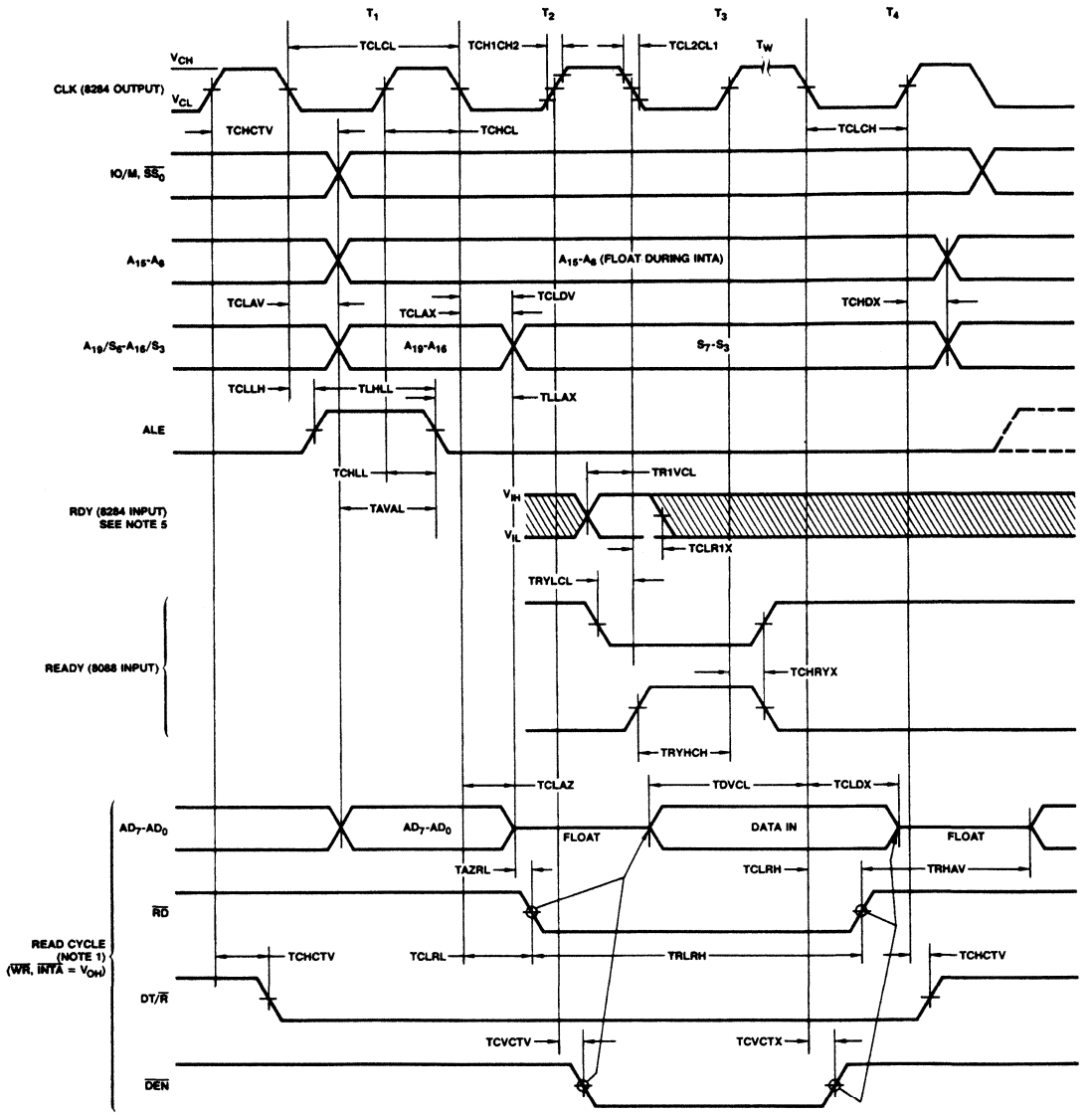
TIMING RESPONSES

Parameters	Description	Test Conditions	8088		8088-2		8088-1		Units
			Min	Max	Min	Max	Min	Max	
TCLML	Command Active Delay (See Note 1)	C_L = 20-100 pF for all 8088 outputs (in addition to internal loads)	10	35	10	35	10	35	ns
TCLMH	Command Inactive Delay (See Note 1)		10	35	10	35	10	35	ns
TRYHSH	READY Active to Status Passive (See Note 3)			110		65		45	ns
TCHSV	Status Active Delay		10	110	10	60	10	45	ns
TCLSH	Status Inactive Delay		10	130	10	70	10	55	ns
TCLAV	Address Valid Delay		10	110	10	60	10	50	ns
TCLAX	Address Hold Time		10		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	10	40	ns
TSVLH	Status Valid to ALE High (See Note 1)			15		15		15	ns
TSVMCH	Status Valid to MCE High (See Note 1)			15		15		15	ns
TCLLH	CLK Low to ALE Valid (See Note 1)			15		15		15	ns
TCLMCH	CLK Low to MCE High (See Note 1)			15		15		15	ns
TCHLL	ALE Inactive Delay (See Note 1)			15		15		15	ns
TCLMCL	MCE Inactive Delay (See Note 1)			15		15		15	ns
TCLDV	Data Valid Delay		10	110	10	60	10	50	ns
TCHDX	Data Hold Time		10		10		10		ns
TCVNV	Control Active Delay (See Note 1)		5	45	5	45	5	45	ns
TCVNX	Control Inactive Delay (See Note 1)		10	45	10	45	10	45	ns
TAZRL	Address Float to Read Active		0		0		0		ns
TCLRL	RD Active Delay		10	165	10	100	10	70	ns
TCLRH	RD Inactive Delay		10	150	10	80	10	60	ns
TRHAV	RD Inactive to Next Address Active		TCLCL −45		TCLCL −40		TCLCL −35		ns
TCHDTL	Direction Control Active Delay (See Note 1)			50		50		50	ns
TCHDTH	Direction Control Inactive Delay (See Note 1)			30		30		30	ns
TCLGL	GT Active Delay			85		50	0	45	ns
TCLGH	GT Inactive Delay			85		50	0	45	ns
TRLRH	RD Width		2TCLCL −75		2TCLCL −50		2TCLCL −40		ns
TOLOH	Output Rise Time	From 0.8 to 2.0V		20		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8V		12		12		12	ns

Notes: 1. Signal at 8284 or 8288 shown for reference only.
2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T_3 state and Wait states.
4. Applies only to T_2 state (8 ns into T_3 state).

SWITCHING WAVEFORMS

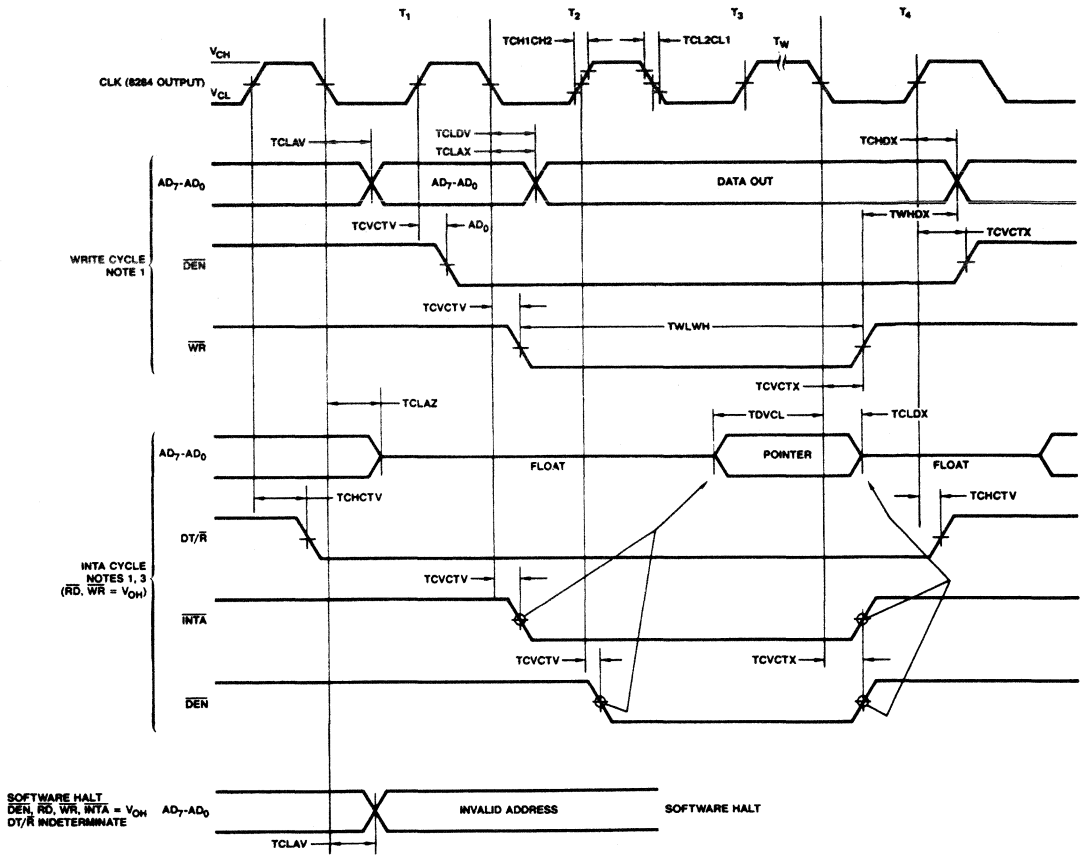
BUS TIMING – MINIMUM MODE SYSTEM



WF006790

SWITCHING WAVEFORMS (Cont.)

BUS TIMING – MINIMUM MODE SYSTEM (Cont.)

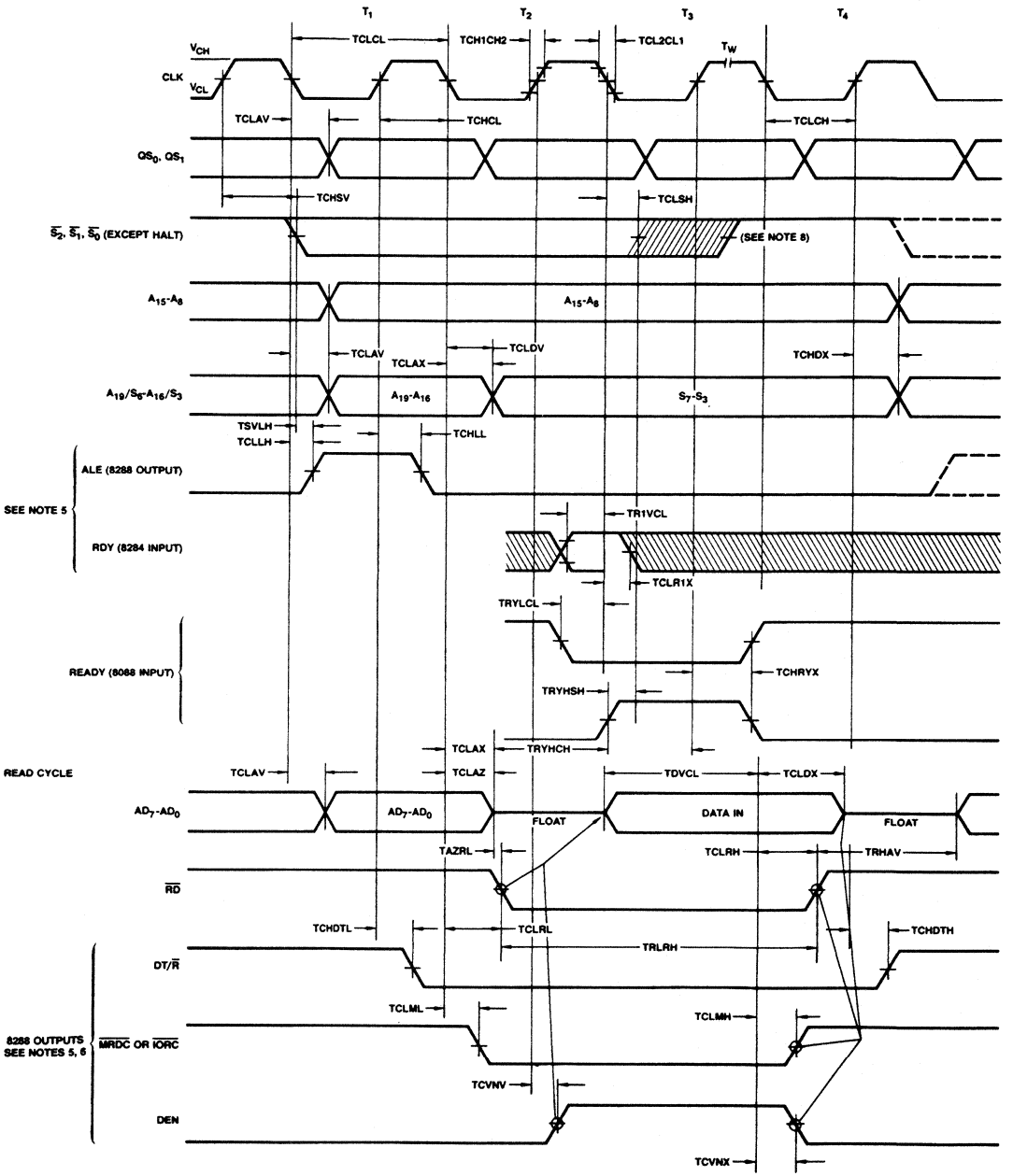


WF006780

Notes: 1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Two INTA cycles run back-to-back. The 8088 local ADDR/DATA bus is floating during both INTA cycles. Control signals are shown for the second INTA cycle.
4. Signals at 8284 are shown for reference only.
5. All timing measurements are made at 1.5V unless otherwise noted.

SWITCHING WAVEFORMS (Cont.)

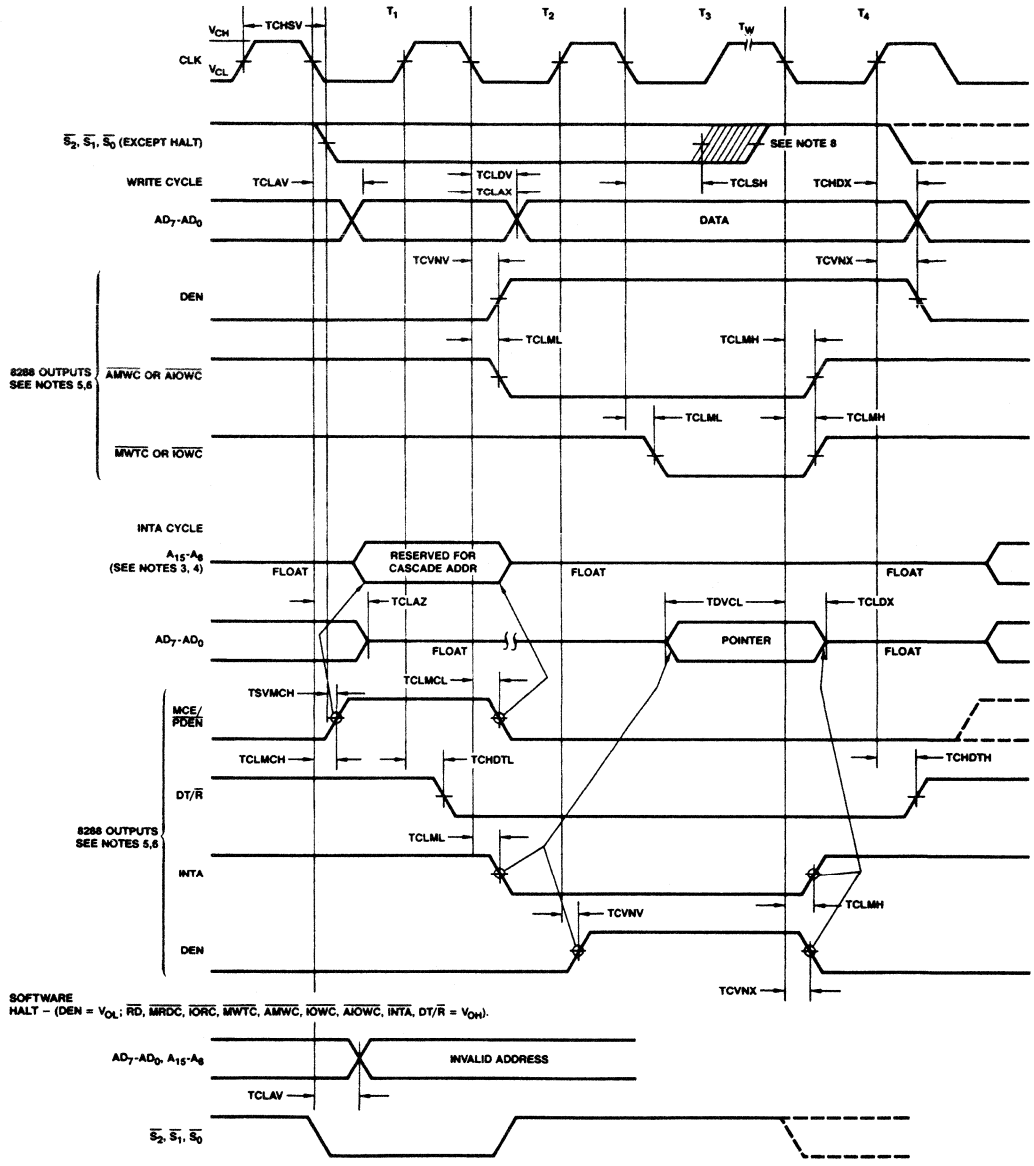
BUS TIMING – MAXIMUM MODE



WF006810

SWITCHING WAVEFORMS (Cont.)

BUS TIMING – MAXIMUM MODE SYSTEM (USING 8288)

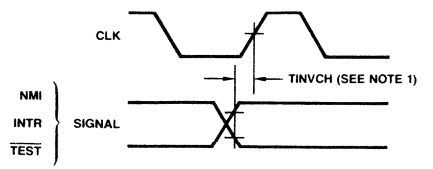


WF006800

Notes: 1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Cascade address is valid between first and second INTA cycles.
4. Two INTA cycles run back-to-back. The 8088 local ADDR/DATA bus is floating during both INTA cycles. Control for pointer address is shown for second INTA cycle.
5. Signals at 8284 or 8288 are shown for reference only.
6. The issuance of the 8288 command and control signals ($\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMWC}$, $\overline{IORC}$, $\overline{IOWC}$, $\overline{AIOWC}$, $\overline{INTA}$, and DEN) lags the active high 8288 CEN.
7. All timing measurements are made at 1.5V unless otherwise noted.
8. Status inactive in state just prior to T_4.

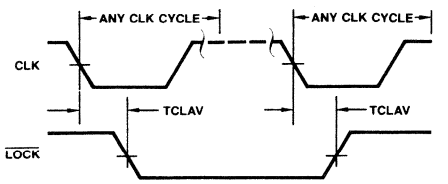
SWITCHING WAVEFORMS (Cont.)

ASYNCHRONOUS SIGNAL RECOGNITION



WF006820

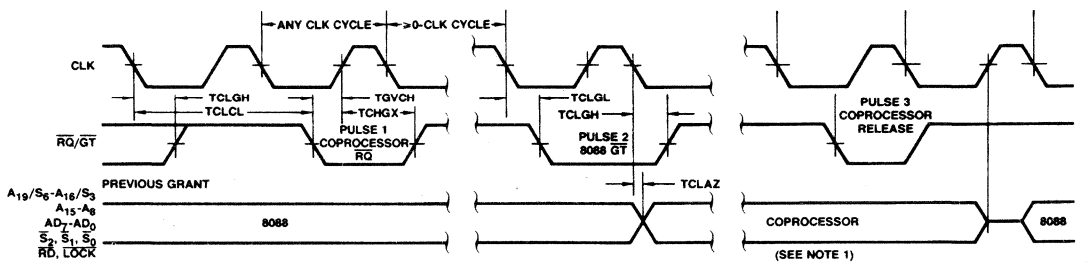
BUS LOCK SIGNAL TIMING (MAXIMUM MODE ONLY)



WF006830

Note 1: Set-up requirements for asynchronous signals only to guarantee recognition at next CLK.

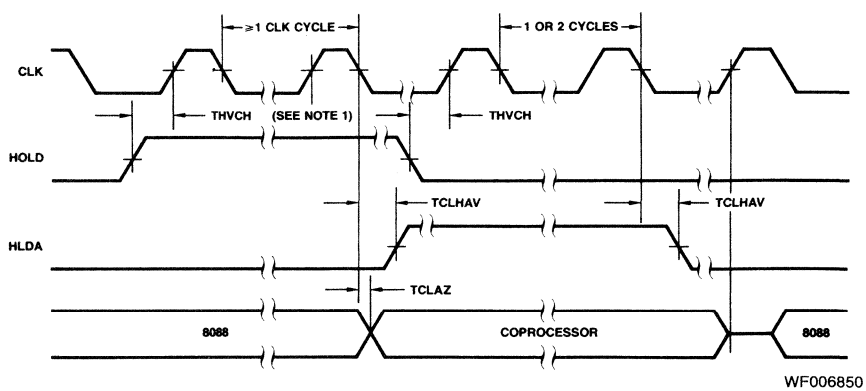
REQUEST/GRANT SEQUENCE TIMING (MAXIMUM MODE ONLY)



WF006840

Note 1: The coprocessor may not drive the buses outside the region shown without rising contention.

HOLD/HOLD ACKNOWLEDGE TIMING (MIMIMUM MODE ONLY)



WF006850

Note 1: All signals switch between V_{OH} and V_{OL} unless otherwise specified.

8086/8088
INSTRUCTION SET SUMMARY

DATA TRANSFER

MOV = Move	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Register/memory to/from register	1 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 1 0 0 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate to register	1 0 1 1 w reg	data	data if w = 1	
Memory to accumulator	1 0 1 0 0 0 0 w	addr-low	addr-high	
Accumulator to memory	1 0 1 0 0 0 1 w	addr-low	addr-high	
Register/memory to segment register	1 0 0 0 1 1 1 0	mod 0 reg r/m		
Segment register to register/memory	1 0 0 0 1 1 0 0	mod 0 reg r/m		
PUSH = Push:				
Register/memory	1 1 1 1 1 1 1 1	mod 1 1 0 r/m		
Register	0 1 0 1 0 reg			
Segment register	0 0 0 reg 1 1 0			
POP = Pop:				
Register/memory	1 0 0 0 1 1 1 1	mod 0 0 0 r/m		
Register	0 1 0 1 1 reg			
Segment register	0 0 0 reg 1 1 1			
XCHG = Exchange:				
Register/memory with register	1 0 0 0 0 1 1 w	mod reg r/m		
Register with accumulator	1 0 0 1 0 reg			
IN = Input from:				
Fixed port	1 1 1 0 0 1 0 w	port		
Variable port	1 1 1 0 1 1 0 w			
OUT = Ouput to:				
Fixed port	1 1 1 0 0 1 1 w	port		
Variable port	1 1 1 0 1 1 1 w			
XLAT = Transtate byte to AL	1 1 0 1 0 1 1 1			
LEA = Load EA to register	1 0 0 0 1 1 0 1	mod reg r/m		
LDS = Load pointer to DS	1 1 0 0 0 1 0 1	mod reg r/m		
LES = Load pointer to ES	1 1 0 0 0 1 0 0	mod reg r/m		
LANF = Load AH with flags	1 0 0 1 1 1 1 1			
SANF = Store AH into flags	1 0 0 1 1 1 1 0			
PUSHF = Push flags	1 0 0 1 1 1 0 0			
POPF = Pop flags	1 0 0 1 1 1 0 1			

INSTRUCTION SET SUMMARY (Cont.)

ARITHMETIC

ADD = Add	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Reg/memory with register to either	0 0 0 0 0 0 d w	mod reg r/m		
Immediate to register / memory	1 0 0 0 0 0 s w	mod 0 0 0 r/m	data	data if s:w = 01
Immediate to accumulator	0 0 0 0 0 1 0 w	data	data if w = 1	

ADC = Add with carry:				
Reg/memory with register to either	0 0 0 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 s w	mod 0 1 0 r/m	data	data if s:w = 01
Immediate to accumulator	0 0 0 1 0 1 0 w	data	data if w = 1	

INC = Increment:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 0 r/m		
Register	0 1 0 0 0 reg			
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1			
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1			

SUB = Subtract:				
Reg/memory and register to either	0 0 1 0 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 1 0 1 r/m	data	data if s:w = 01
Immediate from accumulator	0 0 1 0 1 1 0 w	data	data if w = 1	

SBB = Subtract with borrow:				
Reg/memory and register to either	0 0 0 1 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 0 1 1 r/m	data	data if s:w = 01
Immediate from accumulator	0 0 0 1 1 1 0 w	data	data if w = 1	

DEC = Decrement:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 1 r/m		
Register	0 1 0 0 1 reg			
NEG Change sign	1 1 1 1 0 1 1 w	mod 0 1 1 r/m		

CMP = Compare:				
Register/memory with register	0 0 1 1 1 0 1 w	mod reg r/m		
Register with register/memory	0 0 1 1 1 0 0 w	mod reg r/m		
Immediate with register/memory	1 0 0 0 0 0 s w	mod 1 1 1 r/m	data	data if s:w = 01
Immediate with accumulator	0 0 1 1 1 1 0 w	data	data if w = 1	
AAS ASCII adjust for subtract	0 0 1 1 1 1 1 1			
DAS Decimal adjust for subtract	0 0 1 0 1 1 1 1			
MUL Mulitiply (unsigned)	1 1 1 1 0 1 1 w	mod 1 0 0 r/m		
IMUL Integer multiply (signed):	1 1 1 1 0 1 1 w	mod 1 0 1 r/m		
AAM ASCII adjust for multiply	1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0		
DIV Divide (unsigned):	1 1 1 1 0 1 1 w	mod 1 1 0 r/m		
IDIV Integer divide (signed)	1 1 1 1 0 1 1 w	mod 1 1 1 r/m		
AAD ASCH adjust for divide	1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0		
CBW Convert byte to word	1 0 0 1 1 0 0 0			
CWD Convert word to double word	1 0 0 1 1 0 0 1			

INSTRUCTION SET SUMMARY (Cont.)

LOGIC

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
NOT Invert	1 1 1 1 0 1 1 w	mod 0 1 0 r/m		
SHL/SAL Shift logical/arithmetic left	1 1 0 1 0 0 v w	mod 1 0 0 r/m		
SHR Shift logical right	1 1 0 1 0 0 v w	mod 1 0 1 r/m		
SAR Shift arithmetic right	1 1 0 1 0 0 v w	mod 1 1 1 r/m		
ROL Rotate left	1 1 0 1 0 0 v w	mod 0 0 0 r/m		
ROR Rotate right	1 1 0 1 0 0 v w	mod 0 0 1 r/m		
RCL Rotate through carry flag left	1 1 0 1 0 0 v w	mod 0 1 0 r/m		
RCR Rotate through carry right	1 1 0 1 0 0 v w	mod 0 1 1 r/m		
AND = And:				
Reg/memory and register to either	0 0 1 0 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w = 1	
TEST = And function to flags, no result:				
Register/memory and register	1 0 0 0 0 1 0 w	mod reg r/m		
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w = 1	
OR = Or:				
Reg/memory and register to either	0 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w = 1
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w = 1	
XOR = Exclusive or:				
Reg/memory and register to either	0 0 1 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w = 1	

STRING MANIPULATION:

REP = Repeat	1 1 1 1 0 0 1 z
MOVS = Move byte/word	1 0 1 0 0 1 0 w
CMPS = Compare byte/word	1 0 1 0 0 1 1 w
SCAS = Scan byte/word	1 0 1 0 1 1 1 w
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w
STOS = Stor byte/wd from AL/A	1 0 1 0 1 0 1 w

INSTRUCTION SET SUMMARY (Cont.)

CONTROL TRANSFER

CALL = Call	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Direct within segment	1 1 1 0 1 0 0 0	disp-low	disp-high	
indirect within segment	1 1 1 1 1 1 1 1	mod 0 1 0 r/m		
Direct intersegment	1 0 0 1 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect intersegment	1 1 1 1 1 1 1 1	mod 0 1 1 r/m		
JMP = Unconditional jump:				
Direct within segment	1 1 1 0 1 0 0 1	disp-low	disp-high	
Direct within segment-short	1 1 1 0 1 0 1 1	disp		
Indirect within segment	1 1 1 1 1 1 1 1	mod 1 0 0 r/m		
Direct intersegment	1 1 1 0 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect intersegment	1 1 1 1 1 1 1 1	mod 1 0 1 r/m		
RET = Return from CALL:				
Within segment	1 1 0 0 0 0 1 1			
Within seg adding immed to SP	1 1 0 0 0 0 1 0	data-low	data-high	
Intersegment	1 1 0 0 1 0 1 1			
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0	data-low	data-high	
JE/JZ = Jump on equal/zero	0 1 1 1 0 1 0 0	disp		
JL/JNGE = Jump on less/not greater or equal	0 1 1 1 1 1 0 0	disp		
JLE/JNG = Jump on less or equal/not greater	0 1 1 1 1 1 1 0	disp		
JB/JNAE = Jump on below/not above or equal	0 1 1 1 0 0 1 0	disp		
JBE/JNA = Jump on below or equal/not above	0 1 1 1 0 1 1 0	disp		
JP/JPE = Jump on parity/parity even	0 1 1 1 1 0 1 0	disp		
JO = Jump on overflow	0 1 1 1 0 0 0 0	disp		
JS = Jump on sign	0 1 1 1 1 0 0 0	disp		
JNE/JNZ = Jump on not equal/not zero	0 1 1 1 0 1 0 1	disp		
JNL/JGE = Jump on not less/greater or equal	0 1 1 1 1 1 0 1	disp		
JNLE/JG = Jump on not less or equal/greater	0 1 1 1 1 1 1 1	disp		
JNB/JAE = Jump on not below/above or equal	0 1 1 1 0 0 1 1	disp		
JNBE/JA = Jump on not below or equal/above	0 1 1 1 0 1 1 1	disp		
JNP/JPO = Jump on not par/par odd	0 1 1 1 1 0 1 1	d isp		
JNO = Jump on not overflow	0 1 1 1 0 0 0 1	disp		
JNS = Jump on not sign	0 1 1 1 1 0 0 1	disp		
LOOP = Loop CX times	1 1 1 0 0 0 1 0	disp		
LOOPZ/LOOPE = Loop while zero/equal	1 1 1 0 0 0 0 1	disp		
LOOPNZ/LOOPNE = Loop while not zero/equal	1 1 1 0 0 0 0 0	disp		
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1	disp		

INSTRUCTION SET SUMMARY (Cont.)

CONTROL TRANSFER (Cont.)

INT = Interrupt	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Type specified	1 1 0 0 1 1 0 1	type		
Type 3	1 1 0 0 1 1 0 0			
INTO = Interrupt on overflow	1 1 0 0 1 1 1 0			
IRET = Interrupt return	1 1 0 0 1 1 1 1			

PROCESSOR CONTROL

CLC = Clear carry	1 1 1 1 1 0 0 0	
CMC = Complement carry	1 1 1 1 0 1 0 1	
STC = Set carry	1 1 1 1 1 0 0 1	
CLD = Clear direction	1 1 1 1 1 1 0 0	
STD = Set direction	1 1 1 1 1 1 0 1	
CLI = Clear interrupt	1 1 1 1 1 0 1 0	
STI = Set interrupt	1 1 1 1 1 0 1 1	
HLT = Halt	1 1 1 1 0 1 0 0	
WAIT = Wait	1 0 0 1 1 0 1 1	
ESC = Processor Extension Escape	1 1 0 1 1 x x x	mod x x x r/m
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0	

Footnotes:

AL = 8-bit accumulator
AX = 16-bit accumulator
CX = Count register
DS = Data segment
ES = Extra segment
Above/below refers to unsigned value.
Greater = more positive.
Less = less positive (more negative) signed values
if d = 1 then "to" reg; if d = 0 then "from" reg
w = 1 then word instruction; if w = 0 then byte instruction

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

if s:w = 01 then 16 bits of immediate data form the operand.
if s:w = 11 then an immediate data byte is sign extended to form the 16-bit operand.
if v = 0 then "count" = 1; if v = 1 then "count" in (CL)
x = don't care
z is used for string primitives for comparison with Z.F Flag.

SEGMENT OVERRIDE PREFIX

0	0	1	reg	1	1	0

REG is assigned according to the following table:

16-Bit (w = 1)	8-Bit (w = 0)	Segment
000 AX	000 AL	00 ES
001 CX	001 CL	01 CS
010 DX	010 DL	10 SS
011 BX	011 BL	11 DS
100 SP	100 AH	
101 BP	101 CH	
110 SI	110 DH	
111 DI	111 BH	

Instructions which reference the flag register files as a 16-bit object use the symbol FLAGS to represent the file:

FLAGS = X:X:X:X:(OF):(DF):(TF):(SF):(ZF):X:(AF):X:(PF):X:(CF)

8155(H)/8156(H)

2048-Bit Static MOS RAM with I/O Ports and Timer

DISTINCTIVE CHARACTERISTICS

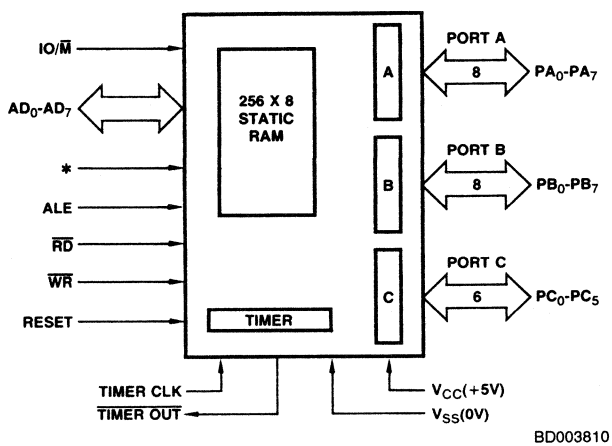
- 256 word x 8-bits
- Single +5 V power supply
- Completely static operation
- Internal address latch
- 2 programmable 8-bit I/O ports
- 1 programmable 6-bit I/O port
- Programmable 14-bit binary counter/timer
- Multiplexed address and data bus

GENERAL DESCRIPTION

The 8155(H) and 8156(H) are RAM and I/O chips to be used in the 8085AH MPU system. The RAM portion is designed with 2K bit static cells organized as 256 x 8. They have a maximum access time of 400ns to permit use with no wait states in 8085AH CPU. The 8155H-2 and 8156H-2 have maximum access times of 330ns for use with the 8085AH. The I/O portion consists of three general purpose I/O ports. One of the three ports can be programmed to be status pins, thus allowing the other two ports to operate in handshake mode.

A 14-bit programmable counter/timer is also included on chip to provide either a square wave or terminal count pulse for the CPU system depending on timer mode.

BLOCK DIAGRAM



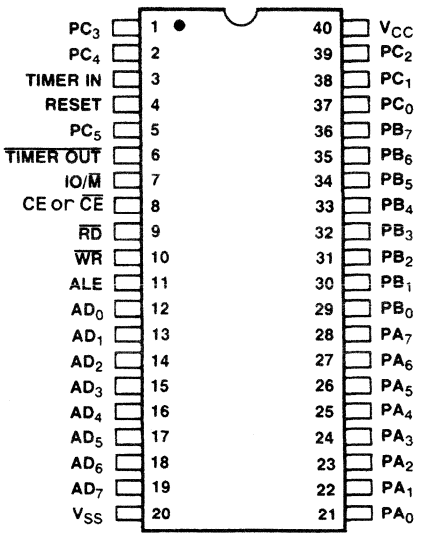
BD003810

*8155H = $\overline{CE}$, 8156H = CE

Publication # 00934 | Rev. C | Amendment /0
Issue Date: April 1987

CONNECTION DIAGRAM
Top View

DIPs



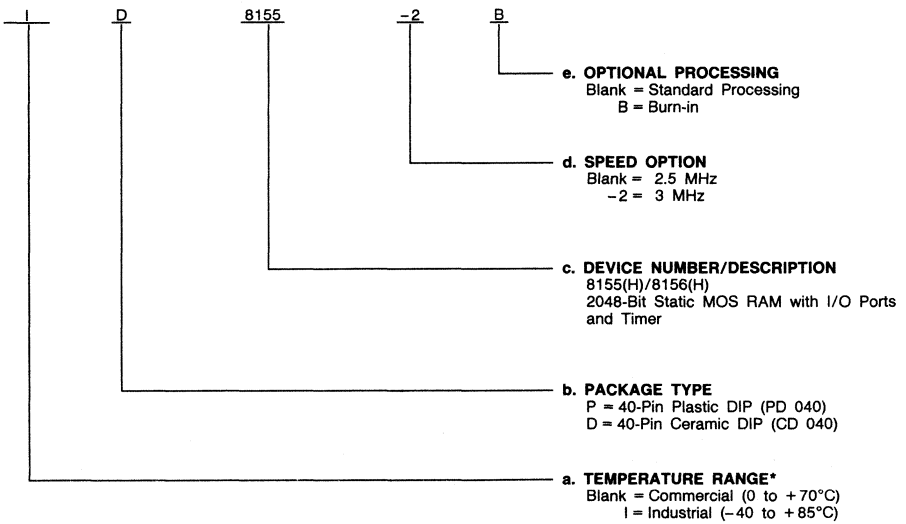
CD005584

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D	8155
	8155H
	8155-2
	8155H-2
	8156
	8156H
	8156-2
	8156H-2
D, ID	8155B
	8155HB
	8155-2B
	8155H-2B
	8156B
	8156HB
	8156-2B
	8156H-2B

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
4	RESET	I	The Reset signal is a pulse provided by the 8085AH to initialize the system. Input high on this line resets the chip and initializes the three I/O ports to input mode. The width of RESET pulse should typically be 600ns. (Two 8085AH clock cycle times).
12-19	AD_0-AD_7	I/O	These are 3-state Address/Data lines that interface with the CPU lower 8-bit Address/Data Bus. The 8-bit address is latched into the address latch on the falling edge of the ALE. The address can be either for the memory section or the I/O section depending on the polarity of the IO/$\overline{M}$ input signal. The 8-bit data is either written into the chip or read from the chip depending on the status of $\overline{WRITE}$ or $\overline{READ}$ input signal.
8	CE OR $\overline{CE}$	I	Chip Enable: On the 8155(H) this pin is $\overline{CE}$ and is active low. On the 8156(H) this pin is CE and is active high.
9	$\overline{RD}$	I	Input low on this line with the Chip Enable active enables the AD_{0-7} buffers. If IO/$\overline{M}$ pin is LOW, the RAM content will be read out to the AD bus. Otherwise, the content of the selected I/O port will be read to the AD bus.
10	$\overline{WR}$	I	Input low on this line with the Chip Enable active causes the data on the AD lines to be written to the RAM or I/O ports, depending on the polarity of IO/$\overline{M}$.
11	ALE	I	Address Latch Enable: This control signal latches the address on the AD_{0-7} lines and the state of the Chip Enable and IO/$\overline{M}$ into the chip at the falling edge of ALE.
7	IO/$\overline{M}$	I	IO/$\overline{MEMORY}$ Select: This line selects the memory if LOW and selects the IO if HIGH.
21-28	PA_0-PA_7	I/O	These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the Command/Status Register.
29-36	PB_0-PB_7	I/O	These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the Command/Status Register.
37-39, 1, 2, 5	PC_0-PC_5	I/O	These 6 pins can function as either input port, output port, or as control signals for PA and PB. Programming is done through the C/S Register. When PC_{0-5} are used as control signals, they will provide the following: PC_0-A INTR (Port A Interrupt) PC_1-A BF (Port A Buffer Full) PC_2-$\overline{A\ STB}$ (Port A Strobe) PC_3-B INTR (Port B Interrupt) PC_4-B BF (Port B Buffer Full) PC_5-$\overline{B\ STB}$ (Port B Strobe)
3	TIMER IN	I	This is the timer input to the counter timer.
6	$\overline{TIMER\ OUT}$	O	This pin is the timer output. This output can be either a square wave or a pulse depending on the timer mode.
40	V_{CC}		+5 volt supply.
20	V_{SS}		Ground reference.

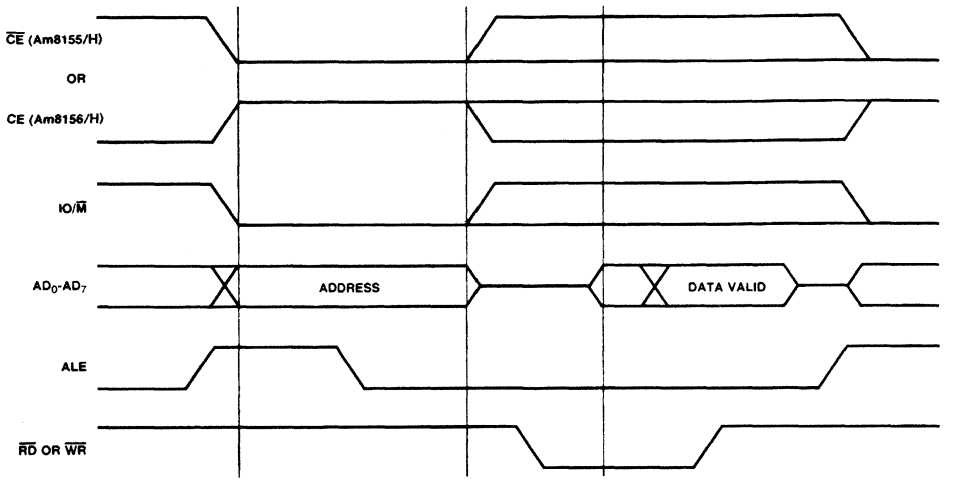
DETAILED DESCRIPTION

The 8155(H)/8156(H) includes the following operational features:

- 2K Bit Static RAM organized as 256 x 8
- Two 8-bit I/O ports (PA and PB) and one 6-bit I/O port (PC)
- 14-bit down counter

The I/O portion contains four registers (Command/Status, PA_{0-7}, PB_{0-7}, PC_{0-5}). The IO/$\overline{M}$ (IO/$\overline{\text{Memory}}$ Select) pin selects the I/O or the memory (RAM) portion. Detailed descriptions of memory, I/O ports and timer functions will follow.

The 8-bit address on the AD lines, the Chip Enable input, and IO/$\overline{M}$ are all latched on chip at the falling edge of ALE. A LOW on the IO/$\overline{M}$ must be provided to select the memory section.



WF008872

Note: For detailed timing diagram information, see Read/Write Cycle Timing Diagrams and Switching Characteristics.

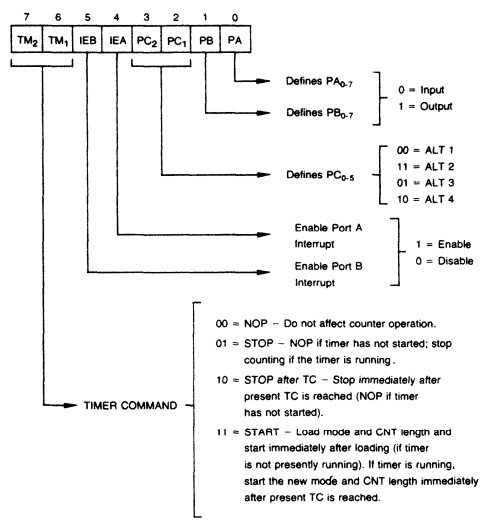
Figure 1. Memory Read/Write Cycle

PROGRAMMING INFORMATION

The Command/Status Register

The command register consists of eight latches, one for each bit. Four bits (0-3) define the mode of the ports. Two bits (4-5) enable or disable the interrupt from Port C when it acts as control port, and the last two bits (6-7) are for the timer.

The C/S register contents can be altered at any time by using the I/O address XXXXX000 during a WRITE operation. The meaning of each bit of the command byte is defined as follows:



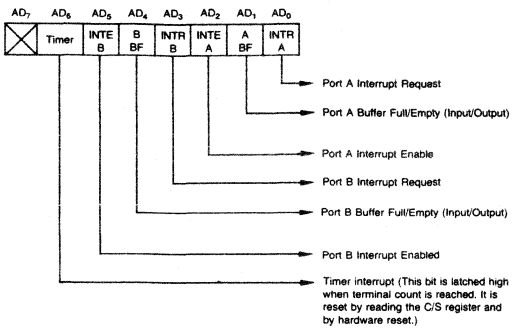
DF003361

Figure 2. Command/Status Register Bit Assignment

Reading the Command/Status Register

The status register consists of seven latches, one for each bit: six (0-5) for the status of the ports and one (6) for the status of the timer.

The status of the timer and the I/O section can be polled by reading the C/S Register (Address XXXXX000). Status word format is shown below:



DF003370

Figure 3. Command/Status Register Status Word Format

Input/Output Section

The I/O section of the 8155(H)/8156(H) consists of four registers as described below.

- Command/Status Register (C/S) — This register is assigned the address XXXXXOOO. The C/S address serves a dual purpose.
  When the C/S register is selected during WRITE operation, a command is written into the command register. The contents of this register are not accessible through the pins.
  When the C/S (XXXXX000) is selected during a READ operation, the status information of the I/O ports and the timer becomes available on the AD_{0-7} lines.
- PA Register — This register can be programmed to be either input or output ports, depending on the status of the contents of the C/S Register. Also, depending on the command, this port can operate in either the basic mode or the strobed mode (see timing diagram). The I/O pins assigned in relation to this register are PA_{0-7}. The address of this register is XXXXX001.
- PB Register — This register functions the same as PA Register. The I/O pins assigned are PB_{0-7}. The address of this register is XXXXX010.
- PC Register — This register has the address XXXXX011 and contains only 6 bits. The 6 bits can be programmed to be either input ports, output ports or as control signals for PA and PB properly programming the AD_2 and AD_3 bits of the C/S register.
  When PC_{0-5} is used as a control port, 3 bits are assigned for Port A and 3 for Port B. The first bit is an interrupt that the 8155(H) sends out. The second is an output signal indicating whether the buffer is full or empty, and the third is an input pin to accept a strobe for the strobed input mode. See Table 1.

When the "C" port is programmed to either ALT3 or ALT4, the control signals for PA and PB are initialized as follows:

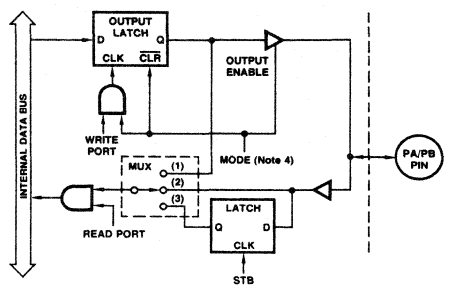
Control	Input Mode	Output Mode
BF	LOW	LOW
INTR	LOW	HIGH
$\overline{STB}$	Input Control	Input Control

The set and reset of INTR and BF with respect to $\overline{STB}$, $\overline{WR}$ and $\overline{RD}$ timing are shown in Strobed I/O Timing Diagrams.

To summarize, the register's assignments are:

Address	Pinouts	Functions	No. of Bits
XXXXX000	Internal	Command/Status Register	8
XXXXX001	PA_{0-7}	General Purpose I/O Port	8
XXXXX010	PB_{0-7}	General Purpose I/O Port	8
XXXXX011	PC_{0-5}	General Purpose I/O Port or Control Lines	6

The following diagram shows how I/O Ports A and B are structured within the 8155(H) and 8156(H):



AF003060

Figure 4. 8155(H)/8156(H) One Bit of Port A or Port B

Notes: 1. Output Mode
2. Simple Input
3. Strobed Input

} Multiplexer Control

4. = 1 for output mode.
   = 0 for input mode.

Read Port = (IO/$\overline{M}$ = 1) • ($\overline{RD}$ = 0) • (CE active) • (Port address selected)

Write Port = (IO/$\overline{M}$ = 1) • ($\overline{WR}$ = 0) • (CE active) • (Port address selected)

Note in the diagram that when the I/O ports are programmed to be output ports, the contents of the output ports can still be read by a READ operation when appropriately addressed.

Note also that the output latch is cleared when the port enters the input mode. The output latch cannot be loaded by writing to the port if the port is in the input mode. The result is that each time a port mode is changed from input to output, the output pins will go LOW. When the 8155(H)/8156(H) is RESET, the output latches are all cleared and all 3 ports enter the input mode.

When in the ALT 1 or ALT 2 modes, the bits of Port C are structured like the diagram above in the simple input or output mode, respectively.

Reading from an input port with nothing connected to the pins will provide unpredictable results.

Table 1. Table of Port Control Assignment

Pin	ALT 1	ALT 2	ALT 3	ALT 4
PC0	Input Port	Output Port	A INTR (Port A Interrupt)	A INTR (Port A Interrupt)
PC1	Input Port	Output Port	A BF (Port A Buffer Full)	A BF (Port A Buffer Full)
PC2	Input Port	Output Port	A $\overline{STB}$ (Port A Strobe)	A $\overline{STB}$ (Port A Strobe)
PC3	Input Port	Output Port	Output Port	B $\overline{INTR}$ (Port B Interrupt)
PC4	Input Port	Output Port	Output Port	B $\overline{BF}$ (Port B Buffer Full)
PC5	Input Port	Output Port	Output Port	B $\overline{STB}$ (Port B Strobe)

Timer Section

The timer is a 14-bit down counter that counts the "timer input" pulses and provides either a square wave or pulse when terminal count (TC) is reached.

The timer has the I/O address XXXXX100 for the low order byte of the register and the I/O address XXXXX101 for the high order byte of the register.

To program the timer, the COUNT LENGTH REG is loaded first, one byte at a time, by selecting the timer addresses. Bits 0 – 13 will specify the length of the next count, and bits 14 – 15 will specify the timer output mode. The value loaded into the count length register can have any value from 2_H through $3FFF_H$ in bits 0 – 13.

There are four modes to choose from:

0 – Puts out LOW during second half of count

1 – Square wave

2 – Single pulse upon TC being reached

3 – Repetitive single pulse every time TC is readied and automatic reload of counter upon TC being reached until instructed to stop by a new command loaded into C/S.

Bits 6 – 7 of the Command/Status Register Contents are used to start and stop the counter. There are four commands to choose from. (See the further description on Command/Status Register.)

C/S7	C/S6	
0	0	NOP – Do not affect counter operation.
0	1	STOP – NOP if timer has not started; stop counting if the timer is running.
1	0	STOP AFTER TC – Stop immediately after present TC is reached (NOP if timer has not started).
1	1	START – Load mode and CNT length and start immediately after loading (if timer is not presently running). If timer is running, start the new mode and CNT length immediately after present TC is reached.

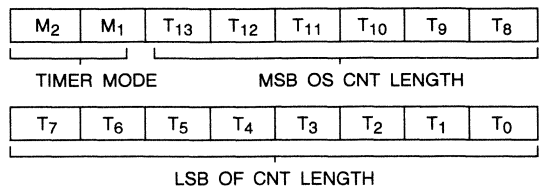
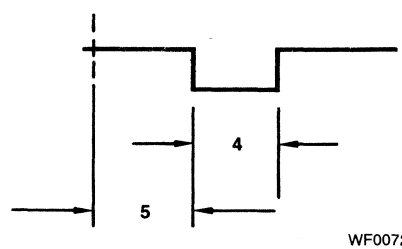


Figure 5. Timer Format

M2 and M1 define the timer mode as follows:

M2	M1	
0	0	Puts out LOW during second half of count.
0	1	Square wave, i.e., the period of the square wave equals the count length programmed with automatic reload at terminal count.
1	0	Single pulse upon TC being reached.
1	1	Automatic reload, i.e., single pulse every time TC is reached.

Note: In case of an asymmetric count, i.e., 9, larger half of the count will be HIGH, the larger count will stay active as shown in Figure 5.



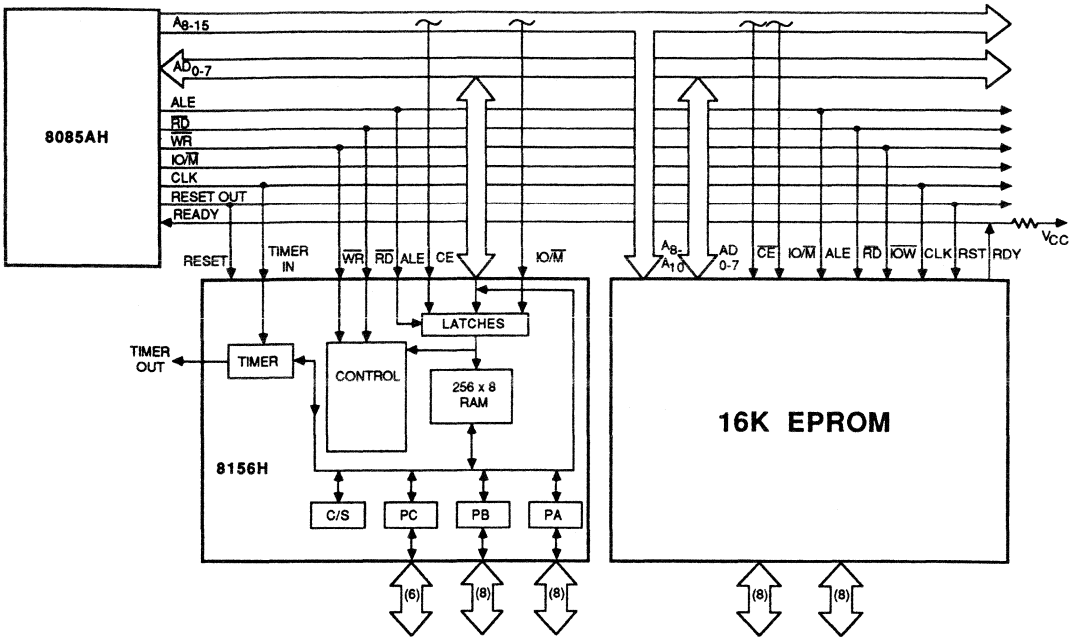
Note: 5 and 4 refer to the number of clock cycles in that time period.

Figure 6. Asymmetric Count

The counter in the 8155(H) is not initialized to any particular mode or count when hardware RESET occurs, but RESET does stop the counting. Therefore, counting cannot begin following RESET until a START command is issued via the C/S register.

8185A Minimum System Configuration

Figure 7 shows a minimum system using three chips, containing 256 Bytes RAM, 2K Bytes EPROM, 38 I/O Pins, 1 Interval Timer, and 4 Interrupt Levels.



AF004690

Figure 7. 8085AH Minimum System Configuration (Memory Mapped I/O)

8088 Five-Chip System

Figure 8 shows a five-chip system containing 1.25K Bytes RAM, 2K Bytes EPROM, 38 I/O Pins, 1 Interval Timer, and 2 Interrupt Levels.

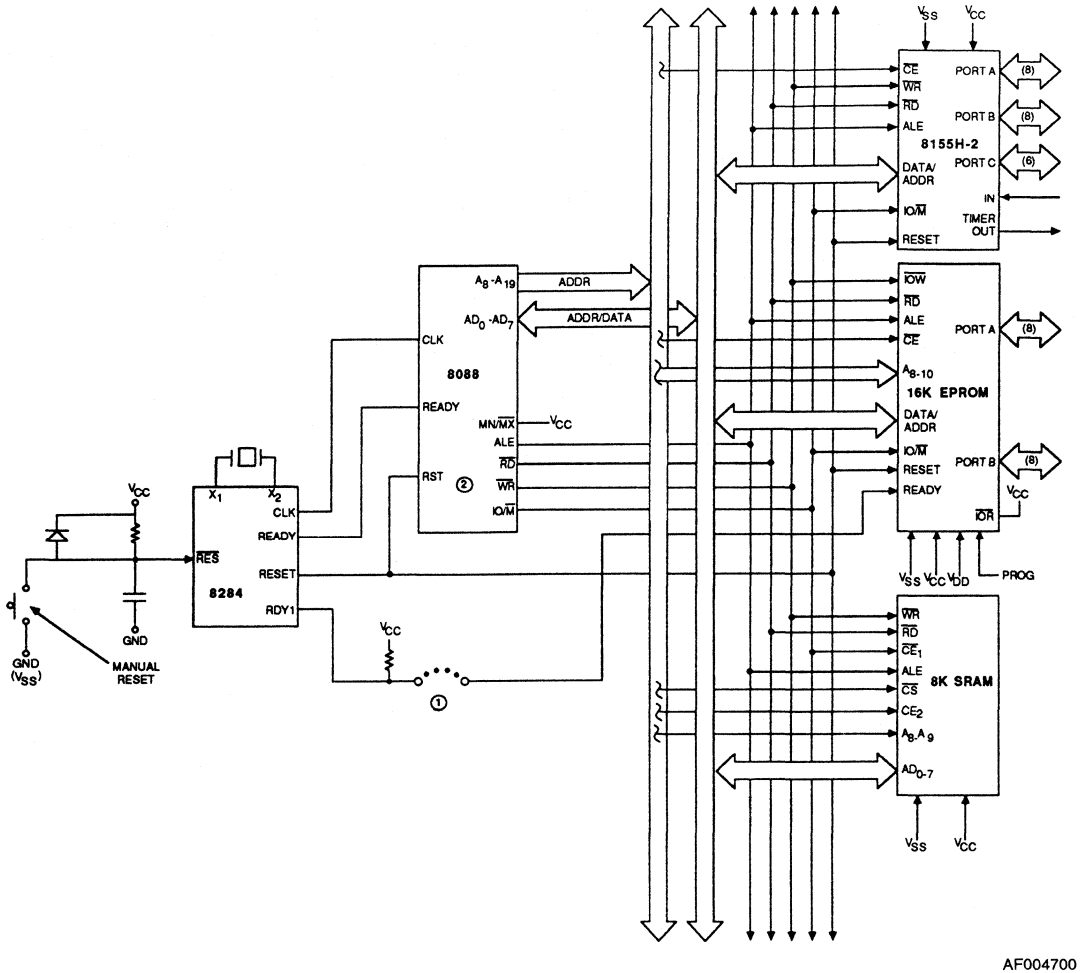


Figure 8. 8088 Five-Chip System Configuration

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to +70 V
All Signal Voltages With
Respect to V_{SS} ..... −0.5 V to +7.0 V
Power Dissipation ..... 1.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC})
8155/8156 ..... 5 V ±5%
8155H/8156H ..... 5 V ±10%
Supply Current (I_{CC})
8155/8156 ..... 180 mA
8155H/8156H ..... 125 mA

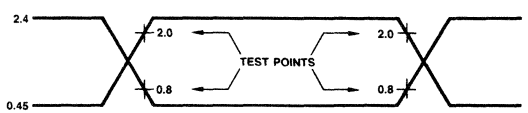
Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ±10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameters	Description		Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage			−0.5	0.8	Volts
V_{IH}	Input High Voltage			2.0	V_{CC}+0.5	Volts
V_{OL}	Output Low Voltage		I_{OL} = 2 mA		0.45	Volts
V_{OH}	Output High Voltage		I_{OH} = −400 µA	2.4		Volts
I_{IL}	Input Leakage		V_{IN} = V_{CC} to 0 V		±10	µA
I_{LO}	Output Leakage Current		0.45 V ≤ V_{OUT} ≤ V_{CC}		±10	µA
I_{CC}	V_{CC} Supply Current	8155, 8156			180	mA
		8155H, 8156H			125	mA
I_{IL} (CE)	Chip Enable Leakage	8155H, 8155	V_{IN} = V_{CC} to 0 V		+100	µA
		8156H, 8156			−100	µA

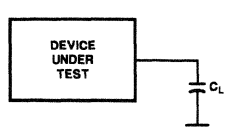
SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF007340

Inputs are driven at 2.4 V for a Logic "1" and 0.45 V for a Logic "0". Timing measurements are made at 2.0 V for a Logic "1" and 0.8 V for a Logic "0".

SWITCHING TEST CIRCUIT



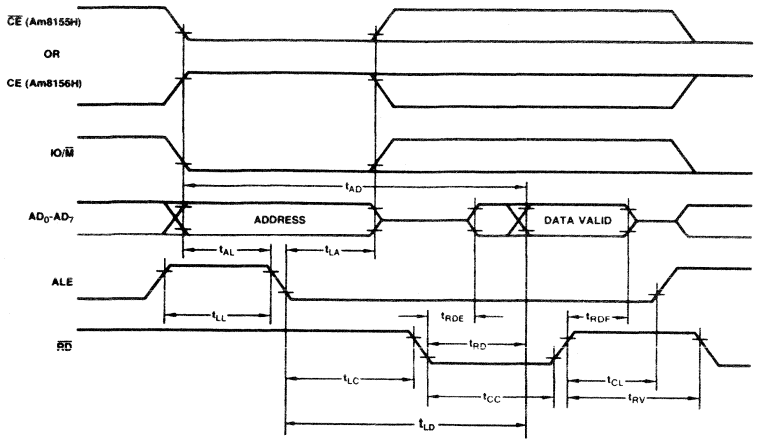
TC002191

C_L = 150pF
C_L Includes Jig Capacitance

SWITCHING CHARACTERISRICS over operating ranges unless otherwise specified

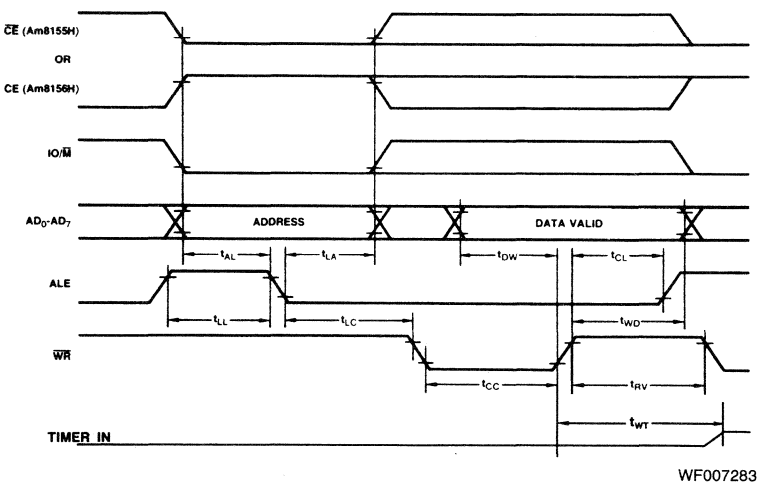
Parameters	Description	8155, 8156 8155H, 8156H		8155-2, 8156-2 8155H-2, 8156H-2		Units
		Min	Max	Min	Max	
t_{AL}	Address to Latch Setup Time	50		30		ns
t_{LA}	Address Hold Time after Latch	80		30		ns
t_{LC}	Latch to READ/WRITE Control	100		40		ns
t_{RD}	Valid Data Out Delay from READ Control		170		140	ns
t_{AD}	Address Stable to Data Out Valid		400		330	ns
t_{LD}	Latch to Data Out Valed		350		270	ns
t_{WT}	WRITE to TIMER-IN (For Writes Which Start Counting)	360		200		ns
t_{LL}	Latch Enable Width	100		70		ns
t_{RDF}	Data Bus Float After READ	0	100	0	80	ns
t_{CL}	READ/WRITE Control to Latch Enable	20		10		ns
t_{CLL}	WRITE Control to Latch Enable for C/S Register	125		125		ns
t_{CC}	READ/WRITE Control Width	250		200		ns
t_{DW}	Data in to WRITE Setup Time	150		100		ns
t_{WD}	Data in Hold Time After WRITE	25		25		ns
t_{RV}	Recovery Time Between Controls	300		200		ns
t_{WP}	WRITE to Port Output		400		300	ns
t_{PR}	Port Input Setup Time	70		50		ns
t_{RP}	Port Input Hold Time	50		10		ns
t_{SBF}	Strobe to Buffer Full		400		300	ns
t_{SS}	Strobe Width	200		150		ns
t_{RBE}	READ to Buffer Empty		400		300	ns
t_{SI}	Strobe to INTR On		400		300	ns
t_{RDI}	READ to INTR Off		400		300	ns
t_{PSS}	Port Setup Time to Strobe	50		0		ns
t_{PHS}	Port Hold Time After Strobe	120		100		ns
t_{SBE}	Strobe to Buffer Empty		400		300	ns
t_{WBF}	WRITE to Buffer Full		400		300	ns
t_{WI}	WRITE to INTR Off		400		300	ns
t_{TL}	TIMER-IN to TIMER-OUT Low		400		300	ns
t_{TH}	TIMER-IN to TIMER-OUT High		400		300	ns
t_{RDE}	Data Bus Enable from READ Control	10		10		ns
t_1	TIMER-IN Low Time	80		40		ns
t_2	TIMER-IN High Time	120		70		ns

SWITCHING WAVEFORMS



WF007273

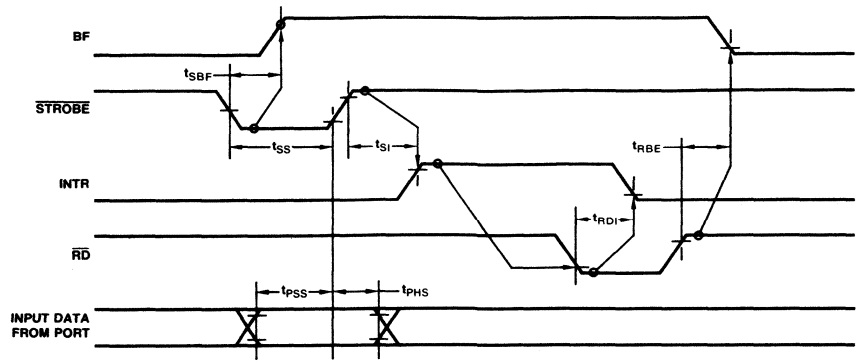
8155(H)/8156(H) Read Cycle



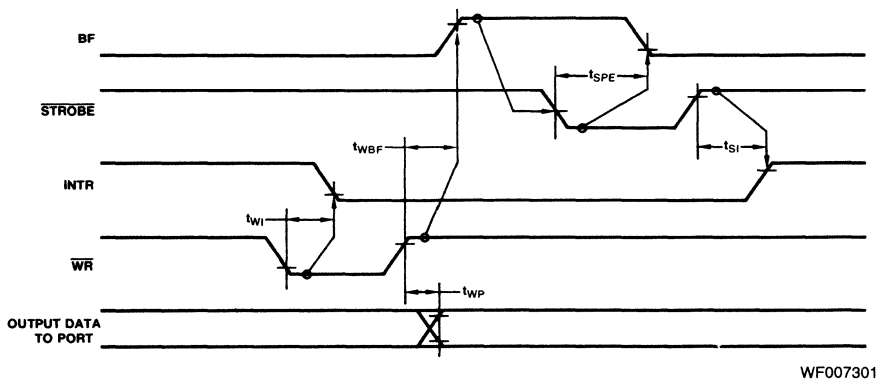
WF007283

8155(H)/8156(H) Write Cycle

SWITCHING WAVEFORMS (Cont'd.)

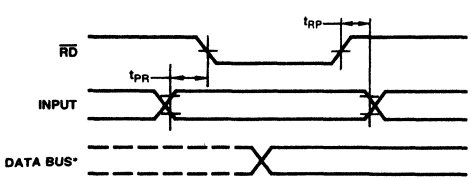


Strobed Input Mode

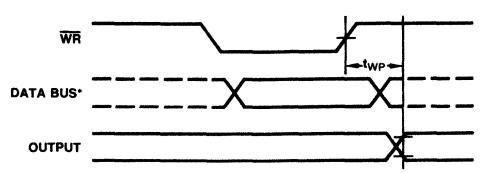


Strobed Output Mode

Input



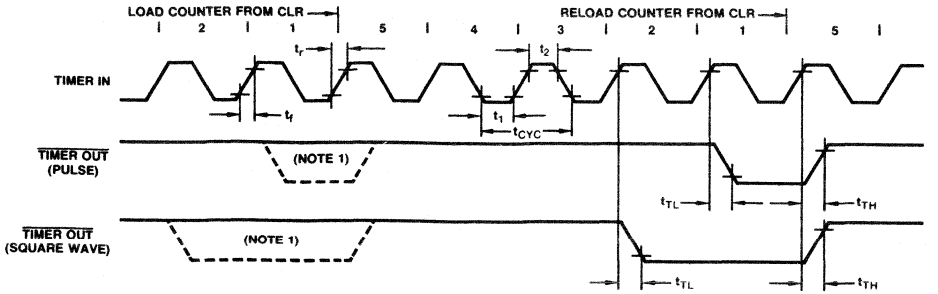
Output



*Data bus timing is shown in Read/Write Cycle diagrams.

Basic I/O Timing Waveform

SWITCHING WAVEFORMS (Cont'd.)



WF007330

Note 1: The timer output is periodic if in an automatic reload mode (M_1 mode bit = 1).

Timer Output Waveform Countdown from 5 to 1

82284

Clock Driver and Ready Interface
for iAPX 286 Processors
PRELIMINARY

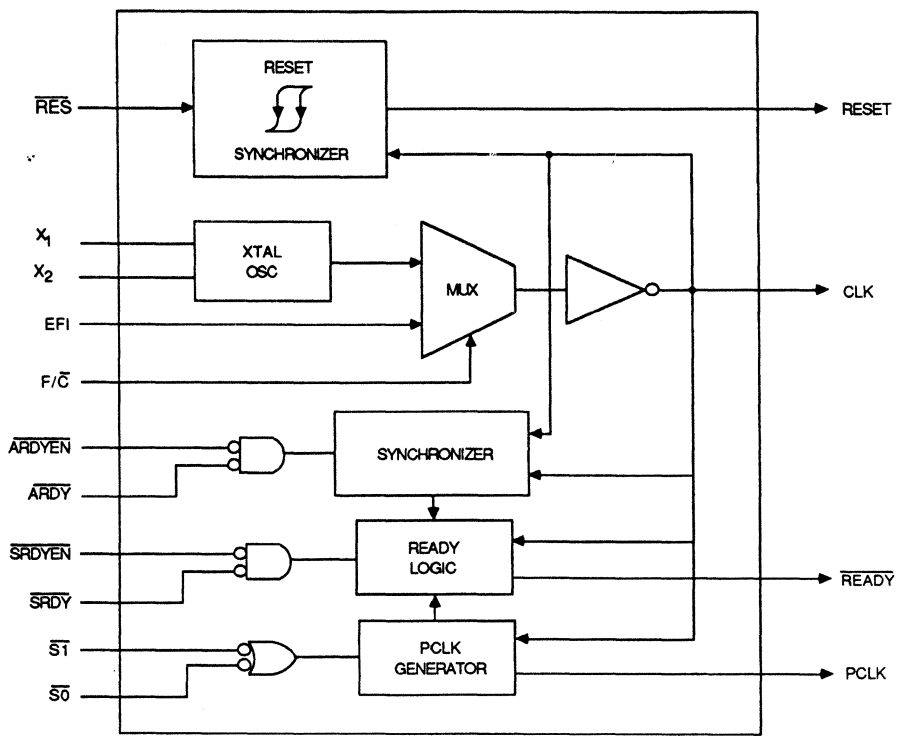
DISTINCTIVE CHARACTERISTICS

- Generates system clock for iAPX 286 processors
- Uses crystal or TTL signal for frequency source
- Provides local $\overline{\text{READY}}$ and MULTIBUS* $\overline{\text{READY}}$ synchronization
- Generates system reset output from Schmitt Trigger input
- 18-pin package
- Single +5 V power supply

GENERAL DESCRIPTION

The 82284 is a clock generator/driver which provides clock signals for iAPX 286 processors and support components. The device contains logic to supply $\overline{\text{READY}}$ to the CPU from either asynchronous or synchronous sources. It also generates a synchronous reset signal from an asynchronous input with hysteresis.

BLOCK DIAGRAM



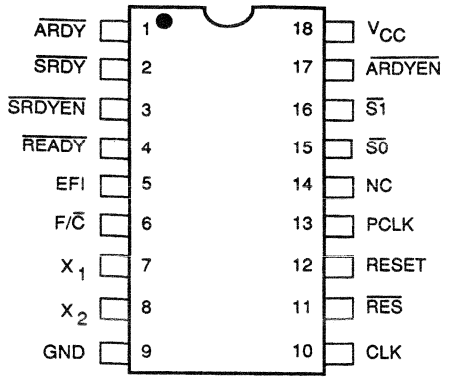
BD007270

*MULTIBUS is a registered trademark of Intel Corporation.

Publication # 05917 | Rev. C | Amendment /0
Issue Date: May 1987

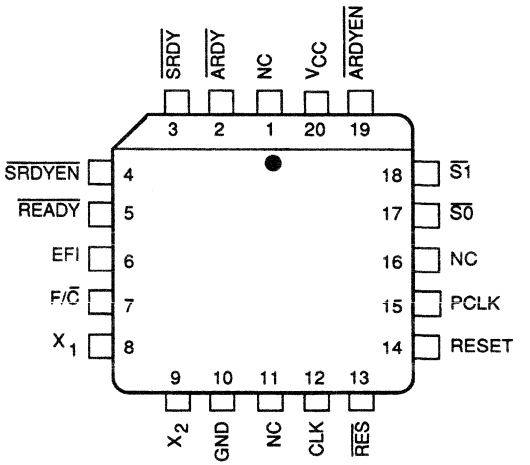
CONNECTION DIAGRAMS
Top View

DIPs



CD010750

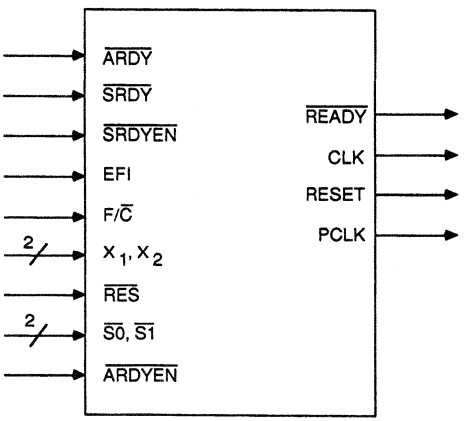
PLCC



CD010760

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL



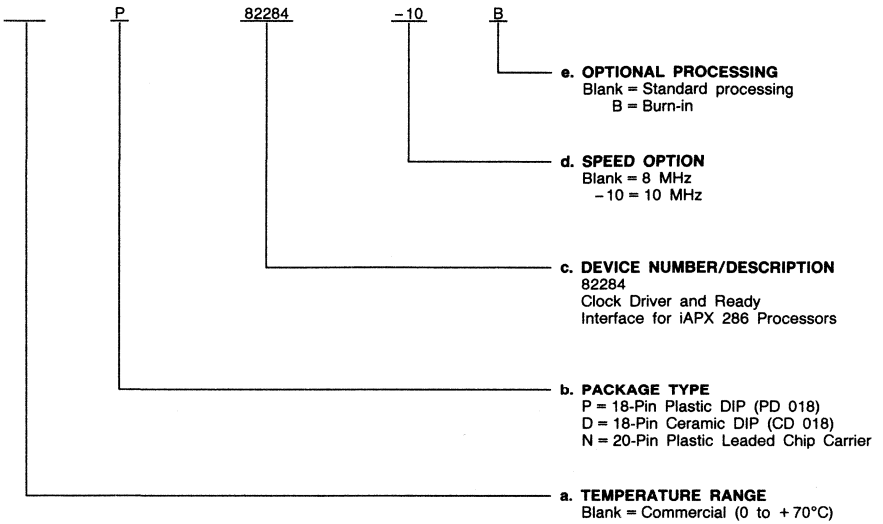
LS003040

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Temperature Range
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D, N	82284
	82284-10
P, D	82284B
	82284-10B
Unpackaged Die	AM82284XC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

$\overline{ARDY}$ Asynchronous Ready (Input; Active LOW)
$\overline{ARDY}$ is an active-LOW input used to terminate the current bus cycle. The $\overline{ARDY}$ input is qualified by $\overline{ARDYEN}$. Inputs to $\overline{ARDY}$ may be applied asynchronously to CLK. Setup and hold times are given to assure a guaranteed response to synchronous inputs.

$\overline{ARDYEN}$ Asynchronous Ready Enable (Input; Active LOW)
$\overline{ARDYEN}$ is an active-LOW input which qualifies the $\overline{ARDY}$ input. $\overline{ARDYEN}$ selects $\overline{ARDY}$ as the source of $\overline{READY}$ for the current bus cycle. Inputs to $\overline{ARDYEN}$ may be applied asynchronously to CLK. Setup and hold times are given to assure a guaranteed response to synchronous inputs.

CLK System Clock (Output)
CLK output is used by the processor and any support devices which must be synchronized with the processor. The frequency of the CLK output is twice the processor's internal clock frequency. CLK can drive both TTL and MOS level inputs.

EFI External Frequency In (Input)
The EFI input drives CLK when F/$\overline{C}$ is strapped HIGH. The EFI input frequency must be twice the processor's internal clock frequency.

F/$\overline{C}$ Frequency/$\overline{Crystal}$ Select (Input)
F/$\overline{C}$ is a strapping option used to select the source for the CLK output. When F/$\overline{C}$ is strapped LOW, the internal crystal drives CLK. When F/$\overline{C}$ is strapped HIGH, the EFI input drives the CLK output.

GND System Ground: 0 V

PCLK Peripheral Clock (Output)
PCLK is an output which provides a 50% duty cycle clock with one half the frequency of CLK. PCLK will be in phase with the processor's internal clock following the first bus cycle after the processor has been reset.

$\overline{READY}$ Ready (Output; Active LOW)
$\overline{READY}$ is an active-LOW output which signals the current bus cycle is to be completed. The $\overline{SRDY}$, $\overline{SRDYEN}$, $\overline{ARDY}$, $\overline{ARDYEN}$, $\overline{S1}$, $\overline{S0}$ and $\overline{RES}$ inputs control $\overline{READY}$ as explained later in the $\overline{READY}$ generator section. $\overline{READY}$ is an open collector output requiring an external 910 ohm pull-up resistor.

$\overline{RES}$ Reset In (Input; Active LOW)
$\overline{RES}$ is an active-LOW input which generates the system reset signal RESET. Signals to $\overline{RES}$ may be applied asynchronously to CLK. A Schmitt Trigger input is provided on $\overline{RES}$, so that an RC circuit can be used to provide a time delay. Setup and hold times are given to assure a guaranteed response to synchronous inputs.

RESET Reset (Output; Active HIGH)
RESET is an active-HIGH output which is derived from the $\overline{RES}$ input. RESET is used to force the system into an initial state. When RESET is active, $\overline{READY}$ will be active (LOW).

$\overline{S0}$, $\overline{S1}$ Status (Input)
These inputs prepare the 82284 for a subsequent bus cycle. $\overline{S0}$ and $\overline{S1}$ synchronize PCLK to the internal processor clock and control $\overline{READY}$. These inputs have pullup resistors to keep them HIGH if nothing is driving them. Setup and hold times must be satisfied for proper operation.

$\overline{SRDY}$ Synchronous Ready (Input; Active LOW)
$\overline{SRDY}$ is an active-LOW input used to terminate the current bus cycle. The $\overline{SRDY}$ input is qualified by the $\overline{SRDYEN}$ input. Setup and hold times must be satisfied for proper operation.

$\overline{SRDYEN}$ Synchronous Ready Enable (Input; Active LOW)
$\overline{SRDYEN}$ is an active-LOW input which qualifies $\overline{SRDY}$. $\overline{SRDYEN}$ selects $\overline{SRDY}$ as the source for $\overline{READY}$ to the CPU for the current bus cycle. Setup and hold times must be satisfied for proper operation.

V_{CC} +5-V Power Supply (Input)

X_1, X_2 Crystal In (Input)
These are the pins to which a parallel resonant fundamental mode crystal is attached for the internal oscillator. When F/$\overline{C}$ is strapped LOW, the oscillator will drive the CLK output at the crystal frequency. The crystal frequency must be twice the processor's internal clock frequency.

FUNCTIONAL DESCRIPTION

Introduction

The 82284 generates the clock, ready, and reset signals required for iAPX 286 processors and support components. The 82284 is packaged in an 18-pin DIP and contains a crystal-controlled oscillator, MOS clock generator, peripheral clock generator, MULTIBUS-ready synchronization logic and system reset generation logic.

Clock Generator

The CLK output provides the basic timing control for an iAPX 286 system. CLK has output characteristics sufficient to drive MOS devices. CLK is generated by either an internal crystal oscillator or an external source as selected by the F/$\overline{C}$ strapping option. When F/$\overline{C}$ is LOW, the crystal oscillator drives the CLK output. When F/$\overline{C}$ is HIGH, the EFI input drives the CLK output.

The 82284 provides a second clock output (PCLK) for peripheral devices. PCLK is CLK divided by two. PCLK has a duty cycle of 50% and TTL output drive characteristics. PCLK is normally synchronized to the internal processor clock.

After reset, the PCLK signal may be out of phase with the internal processor clock. The $\overline{S1}$ and $\overline{S0}$ signals of the first bus cycle are used to synchronize PCLK to the internal processor clock. The phase of the PCLK output changes by extending its HIGH time beyond one system clock (see waveforms). PCLK is forced HIGH when either $\overline{S0}$ or $\overline{S1}$ was active (LOW) for the two previous CLK cycles. PCLK continues to oscillate when both $\overline{S0}$ and $\overline{S1}$ are HIGH.

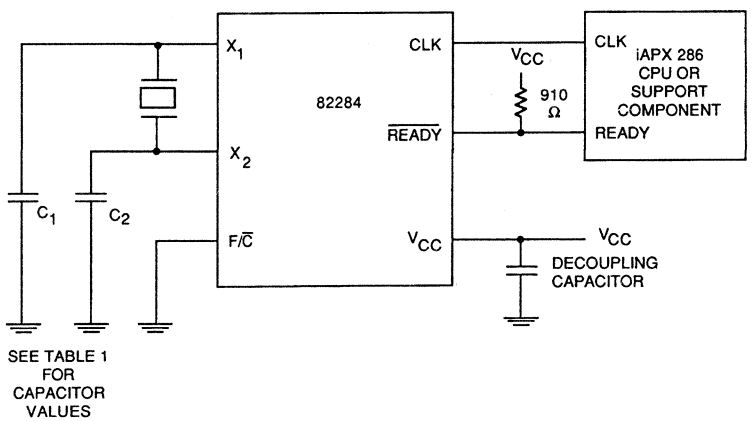
Since the phase of the internal processor clock will not change except during reset, the phase of PCLK will not change except during the first bus cycle after reset.

Oscillator

The oscillator circuit of the 82284 is a linear Pierce oscillator which requires an external parallel resonant fundamental mode crystal. The output of the oscillator is internally buffered. The crystal frequency chosen should be twice the processor's internal clock frequency. The crystal should have a typical load capacitance of 32 pF.

X_1 and X_2 are the oscillator crystal connections. For stable operation of the oscillator, two loading capacitors are recommended, as shown in Figure 1. The sum of the board capacitance and loading capacitance should equal the values shown. It is advisable to limit stray board capacitances (not including the effect of the loading capacitors or crystal capacitance) to less than 10 pF between the X_1 and X_2 pins. V_{CC} and GND pins should be decoupled as close to the 82284 as possible.

Figure 1. Recommended Crystal and READY Connections



TC004240

TABLE 1. 82284 CRYSTAL LOADING CAPACITANCE VALUES

Crystal Frequency	C_1 Capacitance (pin 7)	C_2 Capacitance (pin 8)
1 to 8 MHz	60 pF	40 pF
8 to 16 MHz	25 pF	15 pF

Note: Capacitance values must include stray board capacitance.

Reset Operation

The reset logic provides the RESET output to force the system into a known, initial state. When the $\overline{RES}$ input is active (LOW), the RESET output becomes active (HIGH). $\overline{RES}$ is synchronized internally at the falling edge of CLK before generating the RESET output (see waveforms). Synchronization of the $\overline{RES}$ input introduces a one or two CLK delay before affecting the RESET output.

At power up, a system does not have a stable V_{CC} and CLK. To prevent spurious activity, $\overline{RES}$ should be asserted until V_{CC} and CLK stabilize at their operating values. iAPX 286 processors and support components also require their RESET inputs be HIGH a minimum of 16 CLK cycles. An RC network, as shown in Figure 2, will keep $\overline{RES}$ LOW long enough to satisfy both needs.

A Schmitt Trigger input with hysteresis on $\overline{RES}$ assures a single transition of RESET with an RC circuit on $\overline{RES}$. The hysteresis separates the input voltage level at which the circuit output switches from HIGH to LOW from the input voltage level at which the circuit output switches from LOW to HIGH. The $\overline{RES}$ HIGH to LOW input transition voltage is lower than the $\overline{RES}$ LOW to HIGH input transition voltage. As long as the slope of the $\overline{RES}$ input voltage remains in the same direction (increasing or decreasing) around the $\overline{RES}$ input transition voltage, the RESET output will make a single transition.

Ready Operation

The 82284 accepts two ready sources for the system ready signal which terminates the current bus cycle. Either a synchronous ($\overline{SRDY}$) or asynchronous ready ($\overline{ARDY}$) source may be used. Each ready input has an enable ($\overline{SRDYEN}$ and $\overline{ARDYEN}$) for selecting the type of ready source required to terminate the current bus cycle. An address decoder would normally select one of the enable inputs.

$\overline{READY}$ is enabled (LOW) if either $\overline{SRDY}$ + $\overline{SRDYEN}$ = 0 or $\overline{ARDY}$ + $\overline{ARDYEN}$ = 0 when sampled by the 82284 $\overline{READY}$ generation logic. $\overline{READY}$ will remain active for at least two CLK cycles, except when RESET overrides it.

The $\overline{READY}$ output has an open-collector driver allowing other ready circuits to be wire OR'ed with it, as shown in Figure 1. The $\overline{READY}$ signal of an iAPX 286 system requires an external 910 ohm ± 5% pull-up resistor. To force the $\overline{READY}$ signal inactive (HIGH) at the start of a bus cycle, the $\overline{READY}$ output floats when either $\overline{S1}$ or $\overline{S0}$ are sampled LOW at the falling edge of CLK. Two system clock periods are allowed for the pull-up resistor to pull the $\overline{READY}$ signal to V_{IH}. When RESET is active, $\overline{READY}$ is forced active one CLK later (see waveforms).

Figure 3 illustrates the operation of $\overline{SRDY}$ and $\overline{SRDYEN}$. These inputs are sampled on the falling edge of CLK when $\overline{S1}$ and $\overline{S0}$ are inactive and PCLK is HIGH. $\overline{READY}$ is forced active when both $\overline{SRDY}$ and $\overline{SRDYEN}$ are sampled as LOW.

Figure 4 shows the operation of $\overline{ARDY}$ and $\overline{ARDYEN}$. These inputs are sampled by an internal synchronizer at each falling edge of CLK. The output of the synchronizer is then sampled when PCLK is HIGH. If the synchronizer resolved both the $\overline{ARDY}$ and $\overline{ARDYEN}$ inputs to have been LOW, $\overline{READY}$, becomes LOW. When both $\overline{ARDY}$ and $\overline{ARDYEN}$ have been resolved as active, the $\overline{SRDY}$ and $\overline{SRDYEN}$ inputs are ignored. Either $\overline{ARDY}$ or $\overline{ARDYEN}$ must be HIGH at the end of T_S (see Figure 4).

$\overline{READY}$ remains active until either $\overline{S1}$ or $\overline{S0}$ are sampled LOW, or the ready inputs are sampled as inactive.

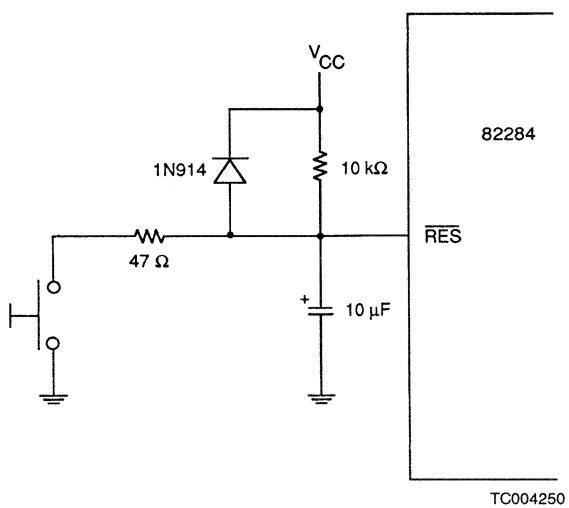


Figure 2. Typical RC $\overline{RES}$ Timing Circuit

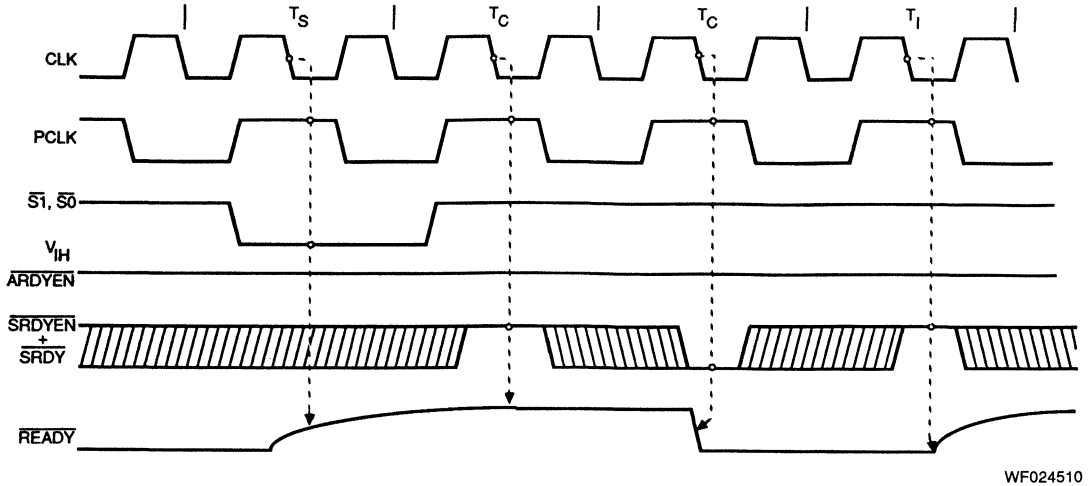


Figure 3. Synchronous Ready Operation

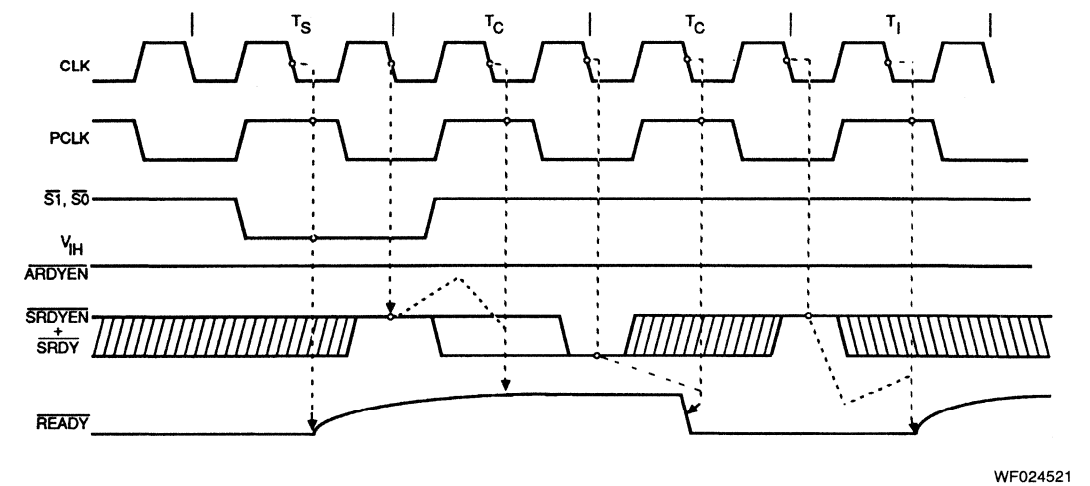


Figure 4. Asynchronous Ready Operation

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Temperature Under Bias ..... 0 to +70°C
All Output and Supply Voltages ..... −0.5 V to +7 V
All Input Voltages ..... −1.0 V to +5.5 V
Power Dissipation ..... 1 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.5 to +5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
V_{OH}	RESET, PCLK Output HIGH Voltage	$I_{OH} = -1$ mA	2.4		V
V_{OL}	RESET, PCLK Output LOW Voltage	$I_{OL} = 5$ mA		.45	V
V_{IH}	Input HIGH Voltage		2.0		V
V_{IL}	Input LOW Voltage			0.8	V
V_C	Input Forward Clamp Voltage	$I_C = -5$ mA		−1.0	V
I_{IL}	Input LOW Current	$V_F = .45$ V		−0.5	mA
I_{IH}	Input HIGH Current	$V_R = V_{CC}$ Max.		50	μA
V_{IHR}	$\overline{RES}$ Input HIGH Voltage		2.6		V
V_{HYS}	$\overline{RES}$ Input Hysteresis		0.25		V
V_{OLR}	$\overline{READY}$ Output LOW Voltage	$I_{OL} = 7$ mA		.45	V
V_{OLC}	CLK Output LOW Voltage	$I_{OL} = 5$ mA		.45	V
V_{OHC}	CLK Output HIGH Voltage	$I_{OH} = -800$ μA	4.0		V
I_{CC}	Power Supply Current			145	mA

CAPACITANCE ($T_A = +25°C$, $V_{CC} = GND = 0$ V, $V_{IN} = +5$ V or GND)

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
C_I	Input Capacitance (Note 1)	$f_C = 1$ MHz		10	pF

Notes: 1. This specification is provided for reference only.

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

No.	Parameter Description	Test Conditions	8 MHz		10 MHz		Units
			Min.	Max.	Min.	Max.	
1	EFI-to-CLK Delay	At 1.5 V (Note 1)		30		30	ns
2	EFI LOW Time	At 1.5 V (Notes 1 & 7)	25		25		ns
3	EFI HIGH Time	At 1.5 V (Notes 1 & 7)	30		25		ns
4	CLK Period		62	500	50	500	ns
5	CLK LOW Time	At 1.0 V (Notes 1, 2, & 8)	15		12		ns
6	CLK HIGH Time	At 3.6 V (Notes 1, 2, & 8)	25		16		ns
7	CLK Rise Time	1.0 to 3.6 V (Note 1)		10		8	ns
8	CLK Fall Time	3.6 to 1.0 V (Note 1)		10		8	ns
9	Status Setup Time	(Note 1)	22		20		ns
10	Status Hold Time	(Note 1)	1		1		ns
11	$\overline{SRDY}$ or $\overline{SRDYEN}$ Setup Time	(Note 1)	17		15		ns
12	$\overline{SRDY}$ or $\overline{SRDYEN}$ Hold Time	(Note 1)	0		0		ns
13	$\overline{ARDY}$ or $\overline{ARDYEN}$ Setup Time	(Notes 1 & 3)	0		0		ns
14	$\overline{ARDY}$ or $\overline{ARDYEN}$ Hold Time	(Note 1 & 3)	30		30		ns
15	$\overline{RES}$ Setup Time	(Notes 1 & 3)	20		20		ns
16	$\overline{RES}$ Hold Time	(Notes 1 & 3)	10		10		ns
17	$\overline{READY}$ Inactive Delay	At 0.8 V (Note 4)	5		5		ns
18	$\overline{READY}$ Active Delay	At 0.8 V (Note 4)	0	24	0	24	ns
19	PCLK Delay	(Note 5)	0	45	0	35	ns
20	RESET Delay	(Note 5)	5	34	5	27	ns
21	PCLK LOW Time	(Notes 5 & 6)	$t_4 - 20$		$t_4 - 20$		ns
22	PCLK HIGH Time	(Notes 5 & 6)	$t_4 - 20$		$t_4 - 20$		ns

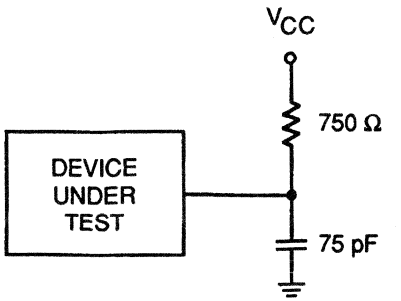
Notes: 1. CLK loading: $C_L = 150$ pF.
2. With the internal oscillator crystal using recommmended crystal and capacitive loading, or with the EFI input meeting specifications t_2 and t_3, use a parallel-resonant, fundamental mode crystal. The recommended crystal loading for CLK frequencies of 8-16 MHz are 25 pF from pin X_1 to ground, and 15 pF from pin X_2 to ground. These recommended values are ±5 pF and include all stray capacitance. Decouple V_{CC} and GND as close to the 82284 as possible.
3. This is an asynchronous input. This specification is given for testing purposes only, to assure recognition at specific CLK edge.
4. $\overline{READY}$ loading: $I_{OL} = 7$ mA, $C_L = 150$ pF. In system application, use 910 ohm ±5% pull-up resistor to meet 80286 timing requirements. For systems which operate faster than 10 MHz, care should be taken to minimize capacitive loading on $\overline{READY}$. The user must ensure the RC time constant allows the pin to be pulled HIGH in two clock cycles.
5. PCLK and RESET loading: $C_L = 75$ pF. PCLK also has 750 ohm pull-up resistor.
6. t_4 refers to any allowable CLK period.
7. When driving the 82284 with EFI, provide minimum EFI HIGH and LOW times as follows:

CLK Output Frequency	16-MHz CLK	20-MHz CLK*
Min. Required EFI HIGH Time	30 ns	25 ns
Min. Required EFI LOW Time	25 ns	25 ns

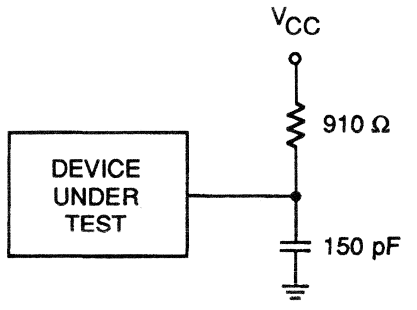
*At CLK frequencies above 16 MHz, CLK output HIGH and LOW times are guaranteed only when using a crystal with recommended capacitive loading per Table 1, not when driving component from EFI. All features of the 82284 remain functional whether EFI or a crystal is used to drive the 82284.

8. When using a crystal (with recommended loading capacitance per Table 1) appropriate for the speed of the 80286, CLK output HIGH and LOW times are guaranteed to meet 80286 requirements.

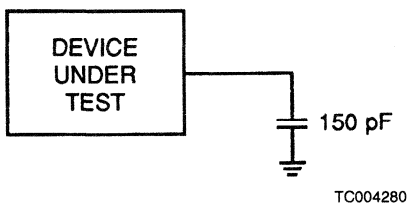
SWITCHING TEST CIRCUITS



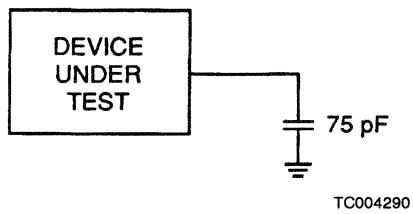
A. PCLK Output



B. $\overline{\text{READY}}$ Output

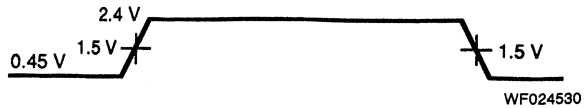


C. CLK Outputs

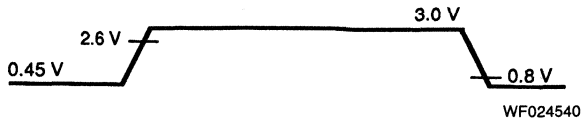


D. RESET Output

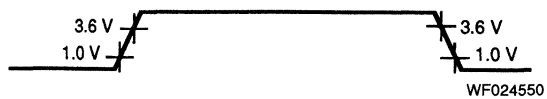
SWITCHING TEST WAVEFORMS



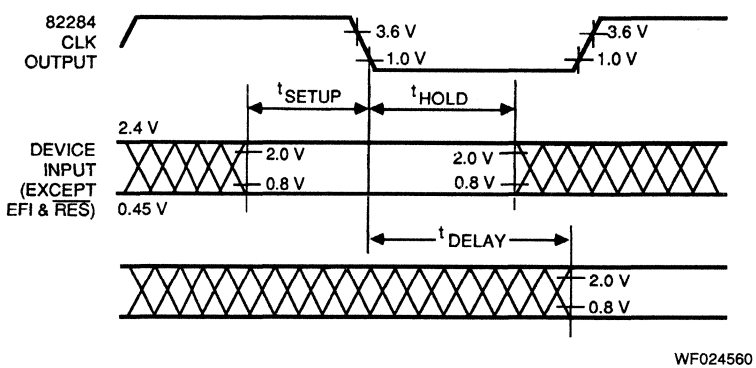
A. EFI Drive and Measurement Points



B. $\overline{RES}$ Drive and Measurement Points

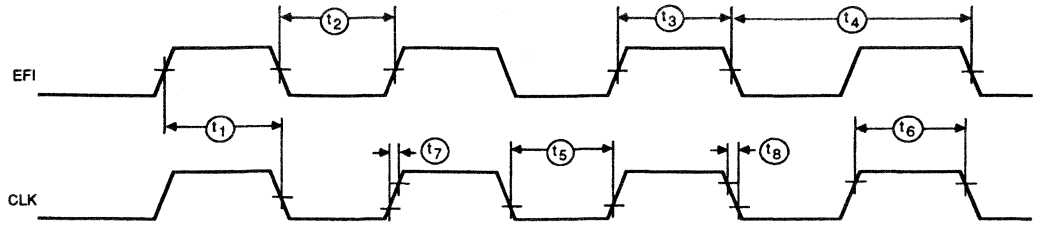


C. CLK Output Measurement Points



D. AC Setup, Hold and Delay Time Measurement – General

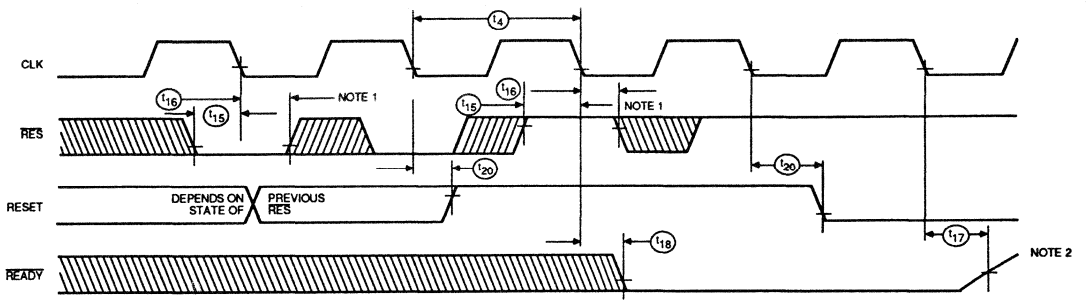
SWITCHING WAVEFORMS



WF024570

Note: The EFI input LOW and HIGH times as shown are required to guarantee the CLK LOW and HIGH times shown.

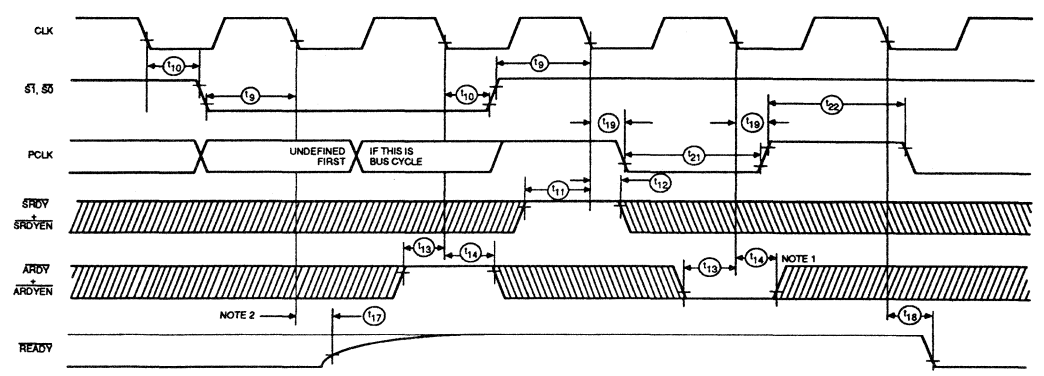
CLK as a Function of EFI



WF024580

Notes: 1. This is an asynchronous input. The setup and hold times shown are required to guarantee the response shown.
2. Tie 910 ohm ±5% pull-up resistor to the $\overline{\text{READY}}$ output. This LOW-to-HIGH transition depends on the state of $\overline{\text{ARDY}}$, $\overline{\text{ARDYEN}}$, $\overline{\text{SRDY}}$, and $\overline{\text{SRDYEN}}$.

RESET and $\overline{\text{READY}}$ Timing as a Function of $\overline{\text{RES}}$ with $\overline{\text{S1}}$ and $\overline{\text{S0}}$ HIGH



WF024590

Notes: 1. This is an asynchronous input. The setup and hold times shown are required to guarantee the response shown.
2. Tie 910 ohm ±5% pull-up resistor to the $\overline{\text{READY}}$ output.

$\overline{\text{READY}}$ and PCLK Timing with $\overline{\text{RES}}$ HIGH

82C288

Bus Controller
for iAPX 286 Processors

DISTINCTIVE CHARACTERISTICS

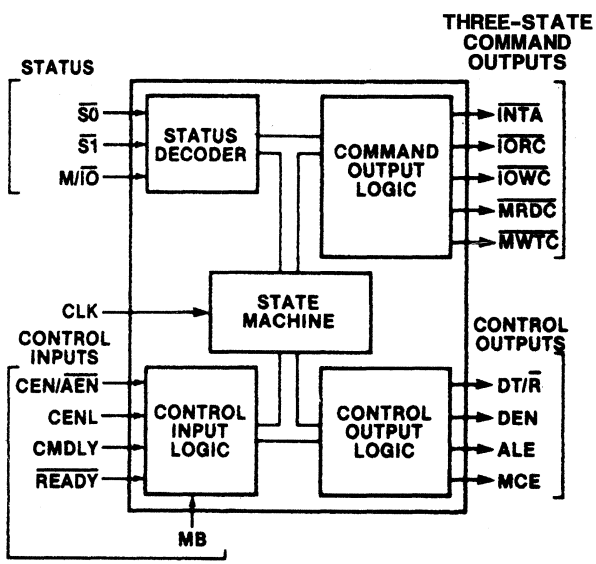
- Provides commands and control for local and system bus
- Offers wide flexibility in system configurations
- Flexible command timing
- Optional Multibus* compatible timing
- Control drivers with 16 mA I_{OL} and three-state command drivers with 32 mA I_{OL}
- Single +5 V supply
- Low power CMOS operation:
  — I_{CCOP} = 24 mA Maximum

GENERAL DESCRIPTION

The 82C288 Bus Controller is a 20-pin CMOS component for use in iAPX 286 microsystems. The bus controller provides command and control outputs with flexible timing options. Separate command outputs are used for memory and I/O devices. The data bus is controlled with separate data enable and direction control signals.

Two modes of operation are possible via a strapping option: Multibus-compatible bus cycles and high-speed bus cycles.

BLOCK DIAGRAM

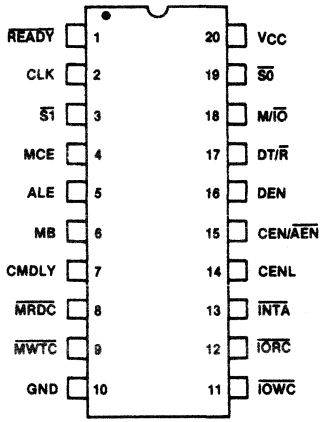


BD004001

*Multibus is a registered trademark of Intel Corporation.

Publication # 06100 | Rev. C | Amendment /0
Issue Date: April 1987

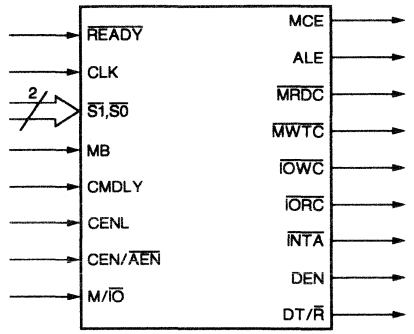
CONNECTION DIAGRAM
Top View
DIPs



CD005623

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL



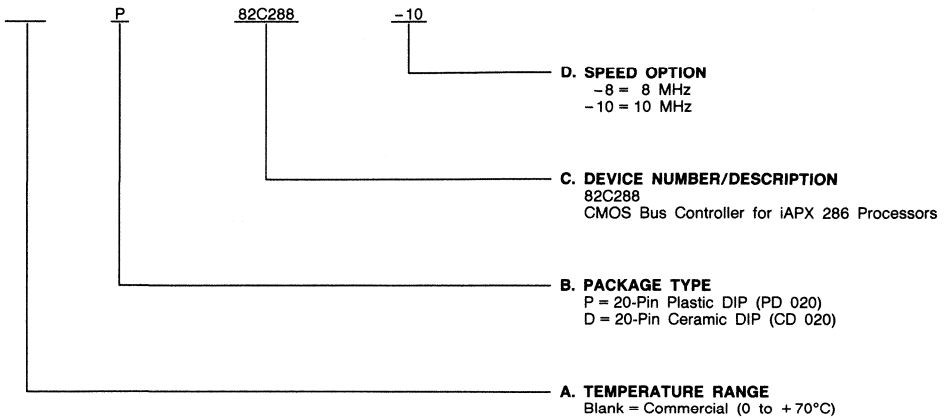
LS002690

GND = System Ground: 0 V
V_{CC} = Supply Power: +5 V

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **A. Temperature Range**
B. Package Type
C. Device Number
D. Speed Option



Valid Combinations	
P, D	82C288-8
	82C288-10

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

$\overline{\text{READY}}$ Ready (Input, Active LOW)
$\overline{\text{READY}}$ is an active-LOW input that indicates the end of the current bus cycle. The 82284 drives $\overline{\text{READY}}$ LOW during RESET to force the 82C288 into the Idle state. Multibus mode requires at least one wait state to allow the command outputs to become active. Setup and hold times must be met for proper operation.

CLK System Clock (Input)
CLK provides the basic timing control for the 82C288. Its frequency is twice the processor's internal clock frequency. The falling edge of this input signal establishes when inputs are sampled and command and control outputs change.

$\overline{\text{S1}}$, $\overline{\text{S0}}$ Bus Cycle Status (Input, Active LOW)
$\overline{\text{S0}}$ and $\overline{\text{S1}}$ are active-LOW inputs that start a bus cycle and, along with M/$\overline{\text{IO}}$, define the type of bus cycle. (See Table 1 for iAPX 286 bus cycle status definitions.) A bus cycle is started when either $\overline{\text{S1}}$ or $\overline{\text{S0}}$ is sampled LOW at the falling edge of CLK. These inputs have internal pull-up resistors to hold them HIGH when not being driven. Setup and hold times must be met for proper operation.

MCE Master Cascade Enable (Output, Active HIGH)
MCE signals that a cascade address from a master 8259A Interrupt Controller may be placed onto the CPU address bus for latching by the address latches under ALE control. The CPU's address bus may then be used to broadcast the cascade address to slave interrupt controllers so only one of them will respond to the interrupt acknowledge cycle. This control output is active HIGH. MCE is only active during interrupt acknowledge cycles and is not affected by any control input. Using MCE to enable cascade address drivers requires latches which save the cascade address on the falling edge of ALE.

ALE Address Latch Enable (Output, Active HIGH)
ALE is an active-HIGH output that controls the address latches used to hold an address stable during a bus cycle. ALE is not issued for the halt bus cycle and is not affected by any control inputs.

MB Multibus Mode Select (Input, Active HIGH)
MB determines the timing of the command and control outputs. When HIGH, the bus controller operates with Multibus-compatible timings. When LOW, the bus controller optimizes the command and control output timing for short bus cycles. The function of the CEN/$\overline{\text{AEN}}$ input pin is selected by this pin. MB is intended to be a strapping option and not dynamically changed; it may be connected to V_{CC} or GND.

CMDLY Command Delay (Input, Active HIGH)
CMDLY is an active-HIGH input that allows the delaying of a command start. If sampled HIGH, the command output is not activated and CMDLY is again sampled at the next CLK cycle. When sampled LOW, the selected command is enabled. If $\overline{\text{READY}}$ is detected LOW before the command output is activated, the 82C288 will terminate the bus cycle, even if no command was issued. Setup and hold times must be satisfied for proper operation. This input may be connected to GND if no delays are required before starting a command. This input has no effect on control outputs.

$\overline{\text{MRDC}}$ Memory Read Command (Output, Active LOW)
$\overline{\text{MRDC}}$ is an active-LOW control output that instructs the memory device to place data onto the data bus. The MB and CMDLY inputs control when this output becomes active. $\overline{\text{READY}}$ controls when it becomes inactive.

$\overline{\text{MWTC}}$ Memory Write Command (Output, Active LOW)
$\overline{\text{MWTC}}$ is an active LOW-command output that instructs a memory device to read the data on the data bus. The MB and CMDLY inputs control when this output becomes active. $\overline{\text{READY}}$ controls when it becomes inactive.

GND System Ground
System Ground: 0 V.

$\overline{\text{IOWC}}$ I/O Write Command (Output, Active LOW)
$\overline{\text{IOWC}}$ is an active-LOW command output that instructs an I/O device to read the data on the data bus. The MB and CMDLY inputs control when this output becomes active. $\overline{\text{READY}}$ controls when it becomes inactive.

$\overline{\text{IORC}}$ I/O Read Command (Output, Active LOW)
$\overline{\text{IORC}}$ is an active-LOW command output that instructs an I/O device to place data onto the data bus. The MB and CMDLY inputs control when this output becomes active. $\overline{\text{READY}}$ controls when it becomes inactive.

$\overline{\text{INTA}}$ Interrupt Acknowledge (Output, Active LOW)
$\overline{\text{INTA}}$ is an active-LOW control output that tells an interrupting device that its interrupt request is being acknowledged. The MB and CMDLY inputs control when this output becomes active. $\overline{\text{READY}}$ controls when it becomes inactive.

CENL Common Enable Latched (Input, Active HIGH)
CENL is a select signal which enables the bus controller to respond to the current bus cycle being initiated. CENL is an active-HIGH input latched internally at the end of each t_S cycle. CENL is used to select the appropriate bus controller for each bus cycle in a system where the CPU has more than one bus it can use. This input may be connected to V_{CC} to select this 82C288 for all transfers. No control inputs affect CENL. Setup and hold times must be met for proper operation.

CEN/$\overline{\text{AEN}}$ Common Enable/$\overline{\text{Address Enable}}$ (Input, Active HIGH/Active LOW)
CEN/$\overline{\text{AEN}}$ controls the command and DEN outputs of the bus controller. This input may be asynchronous to CLK. Setup and hold times are given to assure a guaranteed response to synchronous inputs. This input may be connected to V_{CC} or GND.

When MB is HIGH, this pin has the $\overline{\text{AEN}}$ function. $\overline{\text{AEN}}$ is an active-LOW input which indicates that the CPU has been granted use of a shared bus and the bus controller command outputs may exit three-state OFF and become inactive (HIGH). $\overline{\text{AEN}}$ HIGH indicates that the CPU does not have control of the shared bus and forces the command outputs into three-state OFF and DEN inactive (LOW). $\overline{\text{AEN}}$ would normally be controlled by an 82289 bus arbiter which activates $\overline{\text{AEN}}$ when that arbiter owns the bus to which the bus controller is attached.

When MB is LOW, this pin has the CEN function. CEN is an unlatched active-HIGH input which allows the bus controller to activate its command and DEN outputs. With MB LOW, CEN LOW forces the command and DEN outputs inactive but does not tristate them.

DEN Data Enable (Output, Active HIGH)
DEN determines when data transceivers connected to the local data bus should be enabled. DEN is an active-HIGH control. DEN is delayed for write cycles in the Multibus mode.

DT/$\overline{R}$ Data Transmit/$\overline{Receive}$ (Output, Active HIGH/ Active LOW)
DT/$\overline{R}$ establishes the direction of data flow to or from the local data bus. When HIGH, this control output indicates that a write bus cycle is being performed. A LOW indicates a read bus cycle. DEN is always inactive when DT/$\overline{R}$ changes states. This output is HIGH when no bus cycle is active. DT/$\overline{R}$ is not affected by any of the control inputs.

M/$\overline{IO}$ Memory or $\overline{I/O}$ Select (Input, Active HIGH/ Active LOW)
M/$\overline{IO}$ determines whether the current bus cycle is in the memory space or I/O space. When LOW, the current bus cycle is in the I/O space. Setup and hold times must be met for proper operation.

V_{CC} Supply Power
Supply Power: +5 V.

TABLE 1. iAPX 286 BUS CYCLE STATUS DEFINITIONS

M/$\overline{IO}$	$\overline{S1}$	$\overline{S0}$	Type of Bus Cycle
0	0	0	Interrupt acknowledge
0	0	1	I/O Read
0	1	0	I/O Write
0	1	1	None; idle
1	0	0	Halt or shutdown
1	0	1	Memory read
1	1	0	Memory write
1	1	1	None; idle

FUNCTIONAL DESCRIPTION

Introduction

The 82C288 Bus Controller is used in iAPX 286 systems to provide address latch control, data transceiver control, and standard level-type command outputs. The command outputs are timed and have sufficient drive capabilities for large TTL buses and meet all IEEE-796 requirements for Multibus. A special Multibus mode is provided to satisfy all address/data setup and hold time requirements. Command timing may be tailored to special needs via a CMDLY input to determine the start of a command, and $\overline{READY}$ to determine the end of a command.

Connection to multiple buses are supported with a latched enable input (CENL). An address decoder can determine which, if any, bus controller should be enabled for the bus cycle. This input is latched to allow an address decoder to take full advantage of the pipelined timing on the iAPX 286 local bus.

Buses shared by several bus controllers are supported. An $\overline{AEN}$ input prevents the bus controller from driving the shared bus command and data signals except when enabled by an external bus arbiter such as the 82289.

Separate DEN and DT/$\overline{R}$ outputs control the data transceivers for all buses. Bus contention is eliminated by disabling DEN before changing DT/$\overline{R}$. The DEN timing allows sufficient time for tristate bus drivers to enter three-state OFF before enabling other drivers onto the same bus.

The term CPU refers to any iAPX 286 processor or support component which may become an iAPX 286 local bus master and thereby drive the 82C288 status inputs.

Processor Cycle Definition

Any CPU which drives the local bus uses an internal clock which is one-half the frequency of the system clock (CLK) (see Figure 1). Knowledge of the phase of the local bus master internal clock is required for proper operation of the iAPX 286 local bus. The local bus master informs the bus controller of its internal clock phase when it asserts the status signals. Status signals are always asserted beginning in Phase 1 of the local bus master's internal clock.

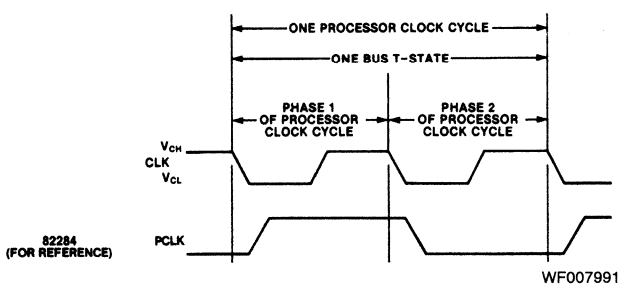


Figure 1. CLK Relationship to the Processor Clock and Bus T-States

Bus State Definition

The 82C288 bus controller has three bus states (see Figure 2): Idle (T_I), Status (T_S), and Command (T_C). Each bus state is two CLK cycles long. Bus state phases correspond to the internal CPU processor clock phases.

The T_I bus state occurs when no bus cycle is currently active on the iAPX 286 local bus. This state may be repeated indefinitely. When control of the local bus is being passed between masters, the bus remains in the T_I state.

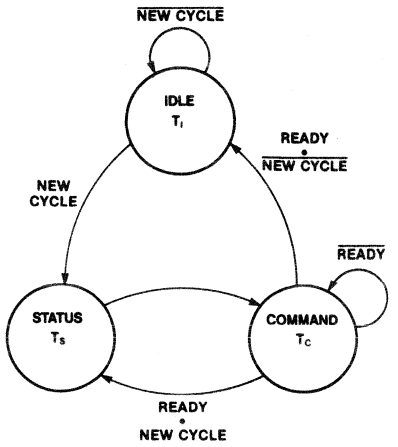


Figure 2. 82C288 Bus States

Bus Cycle Definition

The $\overline{S1}$ and $\overline{S0}$ inputs signal the start of a bus cycle. When either input becomes LOW, a bus cycle is started. The T_S bus state is defined to be the two CLK cycles during which either $\overline{S1}$ or $\overline{S0}$ are active (see Figure 3). These inputs are sampled by the 82C288 at every falling edge of CLK. When either $\overline{S1}$ or $\overline{S0}$ are sampled LOW, the next CLK cycle is considered the second phase of the internal CPU clock cycle.

The local bus enters the T_C bus state after the T_S state. The shortest bus cycle may have one T_S state and one T_C state. Longer bus cycles are formed by repeating T_C states. A repeated T_C bus state is called a wait state.

The $\overline{READY}$ input determines whether the current T_C bus state is to be repeated. The $\overline{READY}$ input has the same timing and effect for all bus cycles. $\overline{READY}$ is sampled at the end of each T_C bus state to see if it is active. If sampled HIGH, the T_C bus state is repeated. This is called inserting a wait state. The control and command outputs do not change during wait states.

When $\overline{READY}$ is sampled LOW, the current bus cycle is terminated. Note that the bus controller may enter the T_S bus state directly from T_C if the status lines are sampled active at the next falling edge of CLK.

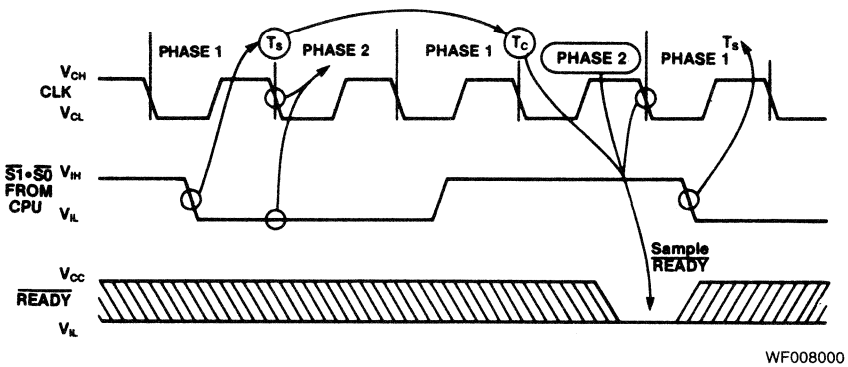


Figure 3. Bus Cycle Definition

TABLE 2. COMMAND AND CONTROL OUTPUTS FOR EACH TYPE OF BUS CYCLE

Type of Bus Cycle	M/$\overline{IO}$	$\overline{S1}$	$\overline{S0}$	Command Activated	DT/$\overline{R}$ State	ALE, DEN Issued?	MCE Issued?
Interrupt Acknowledge	0	0	0	$\overline{INTA}$	LOW	YES	YES
I/O Read	0	0	1	$\overline{IORC}$	LOW	YES	NO
I/O Write	0	1	0	$\overline{IOWC}$	HIGH	YES	NO
None; Idle	0	1	1	None	HIGH	NO	NO
Halt/Shutdown	1	0	0	None	HIGH	NO	NO
Memory Read	1	0	1	$\overline{MRDC}$	LOW	YES	NO
Memory Write	1	1	0	$\overline{MWTC}$	HIGH	YES	NO
None; Idle	1	1	1	None	HIGH	NO	NO

Operating Modes

Two types of buses are supported by the 82C288-Multibus and non-Multibus. When the MB input is strapped HIGH, Multibus timing is used. In Multibus mode, the 82C288 delays command and data activation to meet IEEE-796 requirements on address to command active and write data to command active setup timing. Multibus mode requires at least one wait state in the bus cycle since the command outputs are delayed. The non-Multibus mode does not delay any outputs and does not require wait states. The MB input affects the timing of the command and DEN outputs.

Command and Control Outputs

The type of bus cycle performed by the local bus master is encoded in the M/$\overline{IO}$, $\overline{S1}$, and $\overline{S0}$ inputs. Different command and control outputs are activated depending on the type of bus cycle. Table 2 indicates the cycle decode done by the 82C288 and the effect on command, DT/$\overline{R}$, ALE, DEN, and MCE outputs.

Bus cycles come in three forms: read, write, and halt. Read bus cycles include memory read, I/O read, and interrupt acknowledge. The timing of the associated read command outputs ($\overline{MRDC}$, $\overline{IORC}$ and $\overline{INTA}$), control outputs (ALE, DEN, DT/$\overline{R}$) and control inputs (CEN/$\overline{AEN}$, CENL, CMDLY, MB, and $\overline{READY}$) are identical for all read bus cycles. Read cycles differ only in which command output is activated. The MCE control output is only asserted during interrupt acknowledge cycles.

Write bus cycles activate different control and command outputs with different timing than read bus cycles. Memory write and I/O write are write bus cycles whose timing for command outputs ($\overline{MWTC}$ and $\overline{IOWC}$), control outputs (ALE, DEN, DT/$\overline{R}$) and control inputs (CEN/$\overline{AEN}$, CENL, CMDLY, MB, and $\overline{READY}$) are identical. They differ only in which command output is activated.

Halt bus cycles are different because no command or control output is activated. All control inputs are ignored until the next bus cycle is started via $\overline{S1}$ and $\overline{S0}$.

Figures 4 – 8 show the basic command and control output timing for read and write bus cycles. Halt bus cycles are not shown since they activate no outputs. The basic idle-read-idle and idle-write-idle bus cycles are shown. The signal label CMD represents the appropriate command output for the bus cycle. For Figures 4 – 8, the CMDLY input is connected to GND and CENL to V_{CC}. The effects of CENL and CMDLY are described later in the section on control inputs.

Figures 4, 5 and 6 show non-Multibus cycles. MB is connected to GND while CEN is connected to V_{CC}. Figure 4 shows a read cycle with no wait states while Figure 5 shows a write cycle with one wait state. The $\overline{READY}$ input is shown to illustrate how wait states are added.

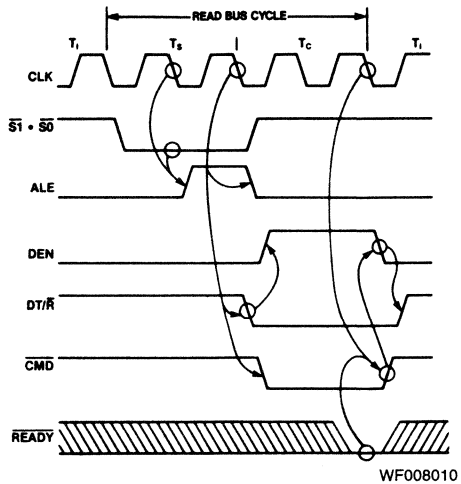


Figure 4. Idle-Read-Idle Bus Cycles with MB = 0

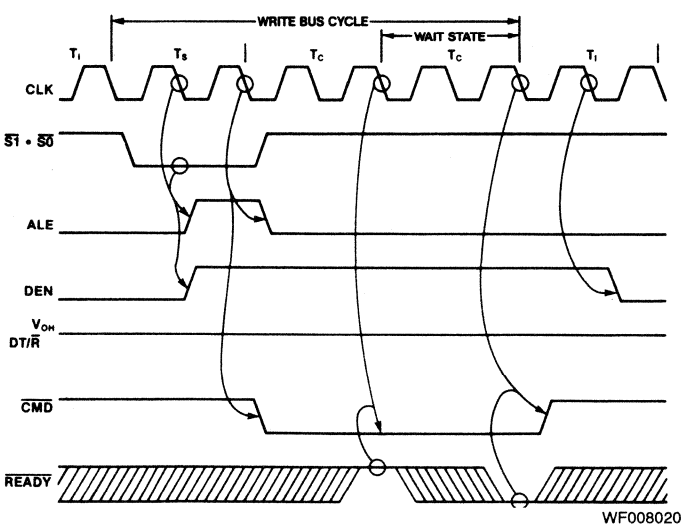


Figure 5. Idle-Write-Idle Bus Cycles with MB = 0

Bus cycles can occur back-to-back with no T_I bus states between T_C and T_S. Back-to-back cycles do not affect the timing of the command and control outputs. Command and control outputs always reach the states shown for the same clock edge (within T_S, T_C, or following bus state) of a bus cycle.

A special case in control timing occurs for back-to-back write cycles with MB = 0. In this case, DT/$\overline{R}$ and DEN remain HIGH between the bus cycles (see Figure 6). The command and ALE output timing does not change.

Figures 7 and 8 show a Multibus cycle with MB = 1. $\overline{AEN}$ and CMDLY are connected to GND. The effects of CMDLY and $\overline{AEN}$ are described later in the section on control inputs. Figure 7 shows a read cycle with one wait state and Figure 8 shows a write cycle with two wait states. The second wait state of the write cycle is shown only for example purposes and is not required. The $\overline{READY}$ input is shown to illustrate how wait states are added.

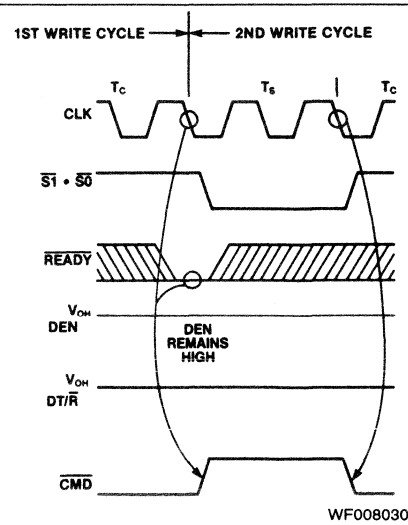


Figure 6. Write-Write Bus Cycles with MB = 0

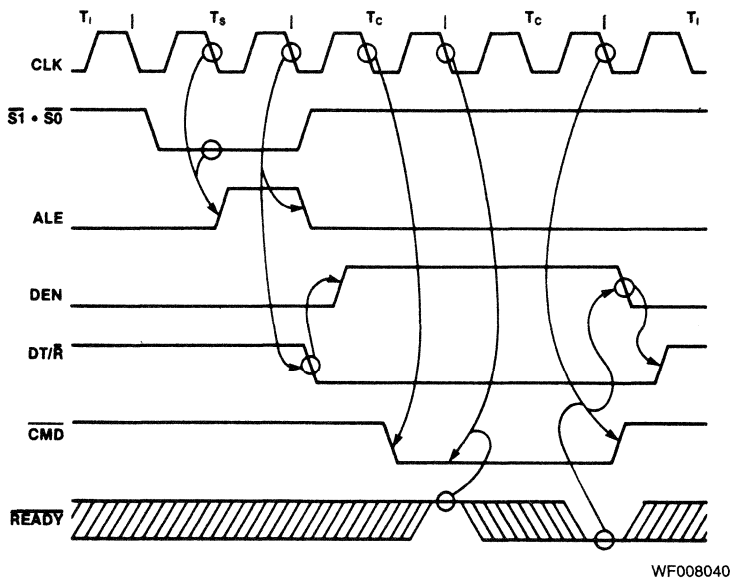


Figure 7. Idle-Read-Idle Bus Cycles with MB = 1

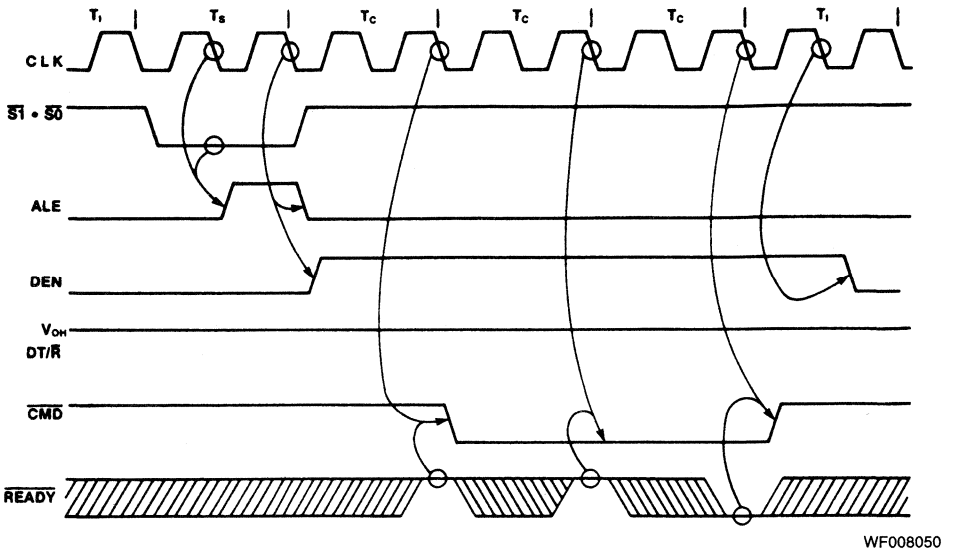


Figure 8. Idle-Write-Idle Bus Cycles with MB = 1

The MB control input affects the timing of the command and DEN outputs. These outputs are automatically delayed in Multibus mode to satisfy three requirements:

1) 50 ns minimum setup time for valid address before any command output becomes active.

2) 50 ns minimum setup time for valid write data before any write command output becomes active.

3) 65 ns maximum time from when any read command becomes inactive until the slave's read data drivers reach three-state OFF.

Three signal transitions are delayed by MB = 1 as compared to MB = 0:

1) The HIGH-to-LOW transition of the read command outputs ($\overline{IORC}$, $\overline{MRDC}$, and $\overline{INTA}$) are delayed one CLK cycle.

2) The HIGH-to-LOW transition of the write command outputs ($\overline{IOWC}$ and $\overline{MWTC}$) are delayed two CLK cycles.

3) The LOW-to-HIGH transition of DEN for write cycles is delayed one CLK cycle.

Back-to-back bus cycles with MB = 1 do not change the timing of any of the command or control outputs. DEN always becomes inactive between bus cycles with MB = 1.

Except for a halt or shutdown bus cycle, ALE will be issued during the second half of T_S for any bus cycle. ALE becomes inactive at the end of the T_S to allow latching the address to keep it stable during the entire bus cycle. The address outputs may change during Phase 2 of any T_C bus state. ALE is not affected by any control input.

Figure 9 shows how MCE is timed during interrupt acknowledge ($\overline{INTA}$) bus cycles. MCE is one CLK cycle longer than ALE to hold the cascade address from a master 8259A valid after the falling edge of ALE. With the exception of the MCE control output, an $\overline{INTA}$ bus cycle is identical in timing to a read bus cycle. MCE is not affected by any control input.

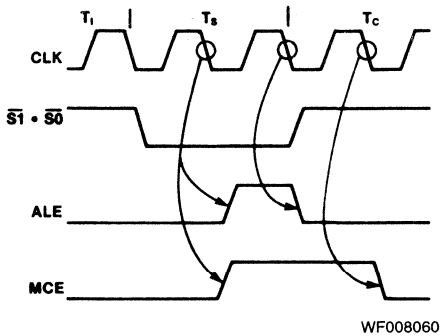


Figure 9. MCE Operation for an $\overline{INTA}$ Bus Cycle

Control Inputs

The control inputs can alter the basic timing of command outputs, allow interfacing to multiple buses, and share a bus between different masters. For many iAPX 286 systems, each CPU will have more than one bus which may be used to perform a bus cycle. Normally, a CPU will only have one bus controller active for each bus cycle. Some buses may be shared by more than one CPU (i.e. Multibus) requiring only one of them use the bus at a time.

Systems with multiple and shared buses use two control input signals of the 82C288 Bus Controller, CENL and $\overline{AEN}$ (see Figure 10). CENL enables the bus controller to control the current bus cycle. The $\overline{AEN}$ input prevents a bus controller from driving its command outputs. $\overline{AEN}$ HIGH means that another bus controller may be driving the shared bus.

In Figure 10, two buses are shown: a local bus and a Multibus. Only one bus is used for each CPU bus cycle. The CENL

inputs of the bus controllers select which bus controller is to perform the bus cycle. An address decoder determines which bus to use for each bus cycle. The 82C288 connected to the shared Multibus must be selected by CENL and be given access to the Multibus by $\overline{AEN}$ before it will begin a Multibus operation.

CENL must be sampled HIGH at the end of the T_S bus state (see Switching Waveforms) to enable the bus controller to activate its command and control outputs. If sampled LOW, the commands and DEN will not go active and DT/$\overline{R}$ will remain HIGH. The bus controller will ignore the CMDLY, CEN, and $\overline{READY}$ inputs until another bus cycle is started via $\overline{S1}$ and $\overline{S0}$. Since an address decoder is commonly used to identify which bus is required for each bus cycle, CENL is latched to avoid the need for latching its input.

The CENL input can affect the DEN control output. When MB = 0, DEN normally becomes active during Phase 2 of T_S in write bus cycles. This transition occurs before CENL is sampled. If CENL is sampled LOW, the DEN output will be forced LOW during T_C as shown in the timing waveforms.

When MB = 1, CEN/$\overline{AEN}$ becomes $\overline{AEN}$. $\overline{AEN}$ controls when the bus controller command outputs enter and exit three-state OFF. $\overline{AEN}$ is intended to be driven by a bus arbiter, like the 82289, which assures only one bus controller is driving the shared bus at any time. When $\overline{AEN}$ makes a LOW-to-HIGH transition, the command outputs immediately enter three-state OFF and DEN is forced inactive. An inactive DEN should force the local data transceivers connected to the shared data bus into three-state OFF (see Figure 10). The LOW-to-HIGH transition of $\overline{AEN}$ should only occur during T_I or T_S bus states.

The HIGH-to-LOW transition of $\overline{AEN}$ signals that the bus controller may now drive the shared bus command signals. Since a bus cycle may be active or be in the process of starting, $\overline{AEN}$ can become active during any T-state. $\overline{AEN}$ LOW immediately allows DEN to go to the appropriate state. Three CLK edges later, the command outputs will go active (see Switching Waveforms). The Multibus requires this delay for the address and data to be valid on the bus before the commands become active.

When MB = 0, CEN/$\overline{AEN}$ becomes CEN. CEN is an asynchronous input which immediately affects the command and DEN outputs. When CEN makes a HIGH-to-LOW transition, the commands and DEN are immediately forced inactive. When CEN makes a LOW-to-HIGH transition, the commands and DEN outputs immediately go to the appropriate state (see Switching Waveforms). $\overline{READY}$ must still become active to terminate a bus cycle if CEN remains LOW for a selected bus controller (CENL was latched HIGH).

Some memory or I/O systems may require more address or write data set-up time to command active than provided by the basic command output timing. To provide flexible command timing, the CMDLY input can delay the activation of command outputs. The CMDLY input must be sampled LOW to activate the command outputs. CMDLY does not affect the control outputs ALE, MCE, DEN, and DT/$\overline{R}$.

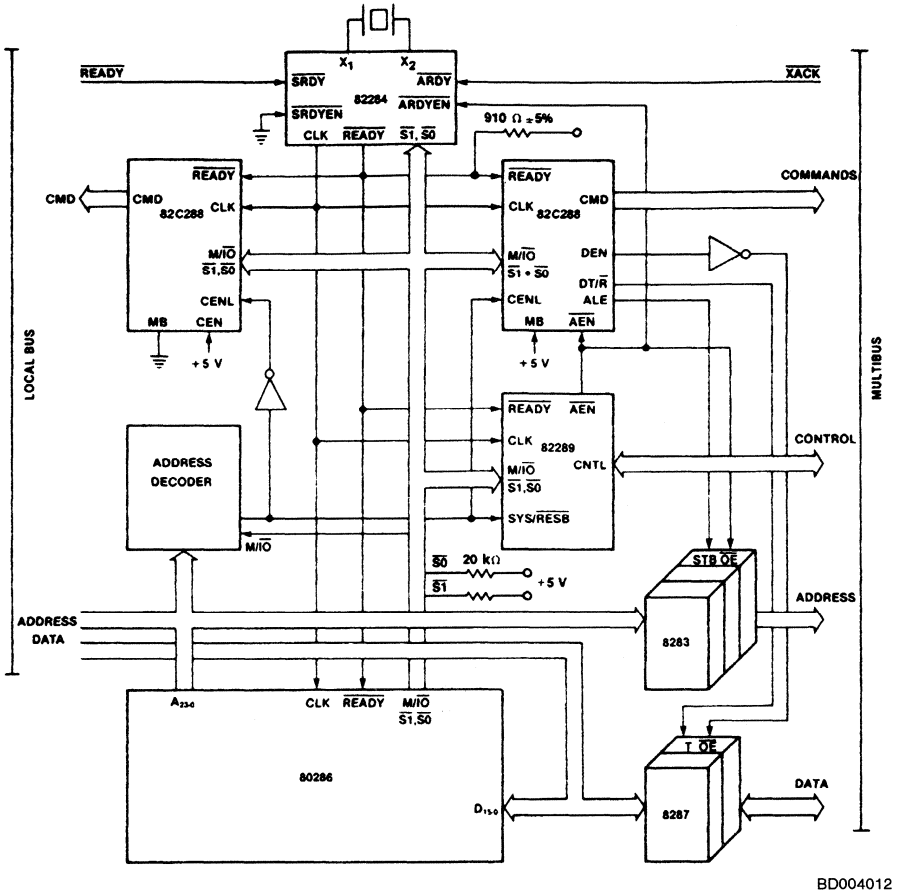


Figure 10. System Use of $\overline{AEN}$ and CENL

CMDLY is first sampled on the falling edge of the CLK ending T_S. If sampled HIGH, the command output is not activated, and CMDLY is again sampled on the next falling edge of CLK. Once sampled LOW, the proper command output becomes active immediately if MB = 0. If MB = 1, the proper command goes active no earlier than shown in Figures 7 and 8.

$\overline{READY}$ can terminate a bus cycle before CMDLY allows a command to be issued. In this case no commands are issued and the bus controller will deactivate DEN and DT/$\overline{R}$ in the same manner as if a command had been issued.

Waveforms Discussion

The waveforms show the timing relationships of inputs and outputs and do not show all possible transitions of all signals in all modes. Instead, all signal timing relationships are shown via the general cases. Special cases are shown when needed. The waveforms provide some functional descriptions of the 82C288; however, most functional descriptions are provided in Figures 3 through 9.

To find the timing specification for a signal transition in a particular mode, first look for a special case in the waveforms. If no special case applies, then use a timing specification for the same or related function in another mode.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin
with Respect to GND ..... −0.5 V to +7.0 V
Power Dissipation ..... 135 mW

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.75 to +5.25 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

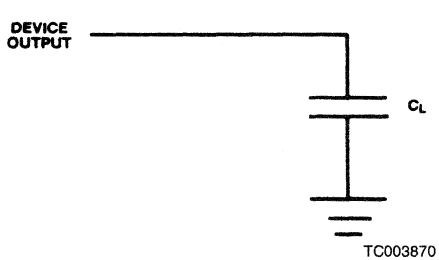
Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
V_{IL}	Input LOW Voltage		−.5	.8	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC} + .5	V
V_{ILC}	CLK Input LOW Voltage		−.5	.6	V
V_{IHC}	CLK Input HIGH Voltage		3.8	V_{CC} + .5	V
V_{OL}	Output LOW Voltage Command Outputs Control Outputs	I_{OL} = 32 mA (Note 1) I_{OL} = 16 mA (Note 2)		.45 .45	V V
V_{OH}	Output HIGH Voltage Command Outputs Control Outputs	I_{OH} = −5 mA (Note 1) I_{OH} = −1 mA (Note 2)	2.4 2.4		V V
I_F	Input Current ($\overline{S0}$, $\overline{S1}$ and M/$\overline{IO}$ Inputs)	Vf = .45 V		−0.5	mA
I_{IL}	Input Leakage Current (All Other Inputs)	$0V \leq V_{IN} \leq V_{CC}$		±10	μA
I_{LO}	Output Leakage Current	$.45\ V \leq V_{OUT} \leq V_{CC}$		±10	μA
I_{CCOP}	Operating Power Supply Current	V_{CC} = 5.5 V, Outputs Open, 8 MHz		22	mA
		V_{CC} = 5.5 V, Outputs Open, 10 MHz		24	mA

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0 V, V_{IN} = +5 V or GND)

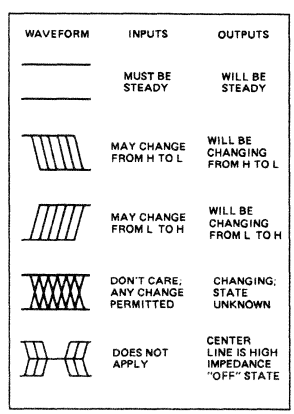
C_{CLK}	CLK Input Capacitance	F_c = 1 MHz		12	pF
C_I	Input Capacitance	F_c = 1 MHz		10	pF
C_O	Output Capacitance	F_c = 1 MHz		20	pF

Notes: 1. Command Outputs are $\overline{INTA}$, $\overline{IORC}$, $\overline{IOWC}$, $\overline{MRDC}$, $\overline{MWRC}$.
2. Control Outputs are DT/$\overline{R}$, DEN, ALE and MCE.

SWITCHING TEST CIRCUIT



KEY TO SWITCHING WAVEFORMS



KS000010

See Section 6 for Thermal Characteristics Information.

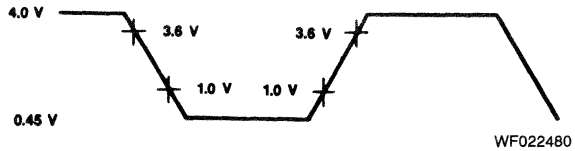
SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified

Switching timings are referenced to 0.8 V and 2.0 V points of signals as illustrated in data sheet waveforms, unless otherwise noted.

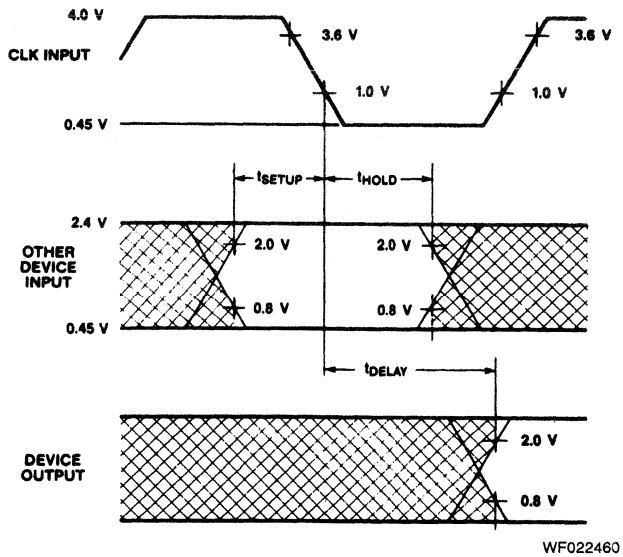
No.	Parameter Description	Test Conditions	82C288-8 (8 MHz)		82C288-10 (10 MHz)		Units
			Min.	Max.	Min.	Max.	
1	CLK Period		62	250	50	250	ns
2	CLK HIGH Time	at 3.6 V	20	235	16	238	ns
3	CLK LOW Time	at 1.0 V	15	230	12	234	ns
4	CLK Rise Time	1.0V to 3.6 V		10		8	ns
5	CLK Fall Time	3.6 V to 1.0V		10		8	ns
6	M/$\overline{IO}$ and Status Setup Time		22		18		ns
7	M/$\overline{IO}$ and Status Hold Time		1		1		ns
8	CENL Setup Time		20		15		ns
9	CENL Hold Time		1		1		ns
10	$\overline{READY}$ Setup Time		38		26		ns
11	$\overline{READY}$ Hold Time		25		25		ns
12	CMDLY Setup Time		20		15		ns
13	CMDLY Hold Time		1		1		ns
14	$\overline{AEN}$ Setup Time	(Note 3)	20		15		ns
15	$\overline{AEN}$ Hold Time	(Note 3)	0		0		ns
16	ALE, MCE Active Delay from CLK	(Note 4)	3	20	3	16	ns
17	ALE, MCE Inactive Delay from CLK	(Note 4)		25		19	ns
18	DEN (Write) Inactive from CENL	(Note 4)		35		23	ns
19	DT/$\overline{R}$ LOW from CLK	(Note 4)		25		23	ns
20	DEN (Read) Active from DT/$\overline{R}$	(Note 4)	5	35	5	21	ns
21	DEN (Read) Inactive Delay from CLK	(Note 4)	3	35	3	21	ns
22	DT/$\overline{R}$ HIGH from DEN Inactive	(Note 4)	5	35	5	20	ns
23	DEN (Write) Active Delay from CLK	(Note 4)		30		23	ns
24	DEN (Write) Inactive Delay from CLK	(Note 4)	3	30	3	19	ns
25	DEN Inactive from CEN	(Note 4)		30		25	ns
26	DEN Active from CEN	(Note 4)		30		24	ns
27	DT/$\overline{R}$ HIGH from CLK (when CEN = LOW)	(Note 4)		35		25	ns
28	DEN Active from $\overline{AEN}$	(Note 4)		30		26	ns
29	$\overline{CMD}$ Active Delay from CLK	(Note 5)	3	25	3	21	ns
30	$\overline{CMD}$ Inactive Delay from CLK	(Note 5)	5	25	5	20	ns
31	$\overline{CMD}$ Inactive from CEN	(Note 5)		25		25	ns
32	$\overline{CMD}$ Active from CEN	(Note 5)		25		25	ns
33	$\overline{CMD}$ Inactive Enable from $\overline{AEN}$	(Note 5)		40		40	ns
34	$\overline{CMD}$ Float Delay from $\overline{AEN}$	(Note 6)		40		40	ns
35	MB Setup Time		20		20		ns
36	MB Hold Time		0		0		ns
37	Command Inactive Enable from MB ↑	(Note 5)		40		40	ns
38	Command Float Time from MB ↓	(Note 6)		40		40	ns
39	DEN Inactive from MB ↓	(Note 4)		30		26	ns
40	DEN Inactive from MB ↑	(Note 4)		30		30	ns

Notes: 3. $\overline{AEN}$ is an asynchronous input. This specification is for testing purposes only, to assure recognition at a specific CLK edge.
4. Control output load: C_L = 150 pF.
5. Command output load: C_L = 300 pF.
6. Float condition occurs when output current is less than I_{LO} in magnitude.

SWITCHING WAVEFORMS

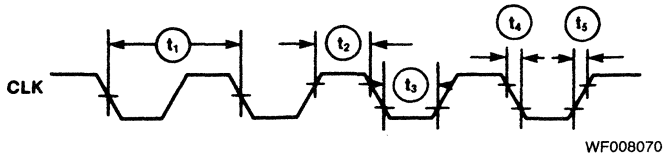


AC Drive and Measurement Points — CLK input

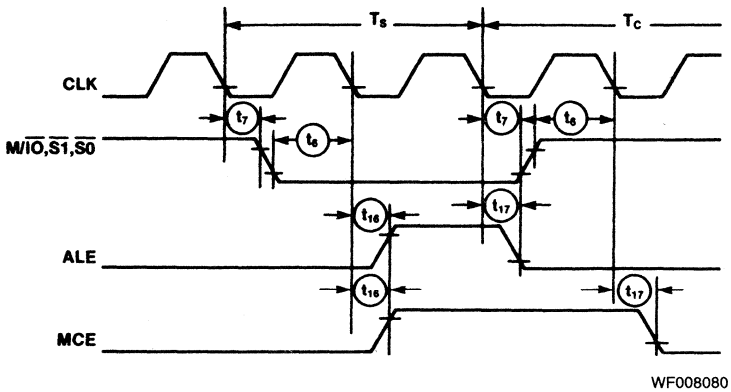


AC Setup, Hold, and Delay Time Measurement — General

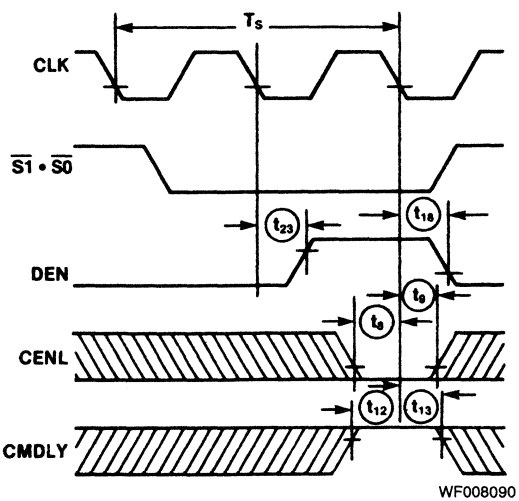
SWITCHING WAVEFORMS (Cont'd.)



CLK Characteristics

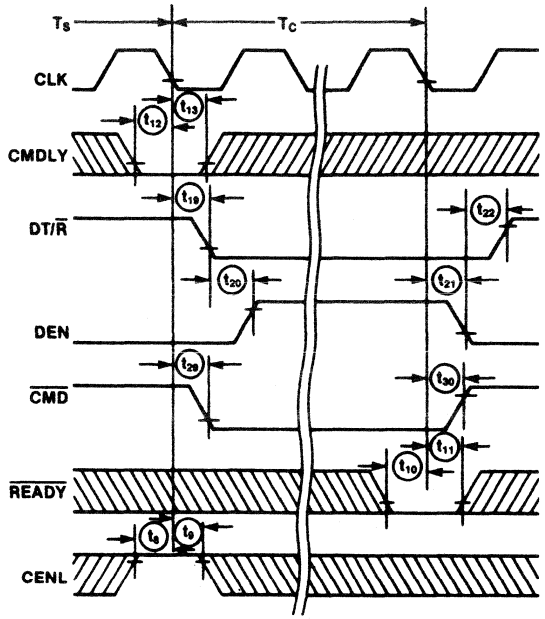


Status, ALE, MCE Characteristics



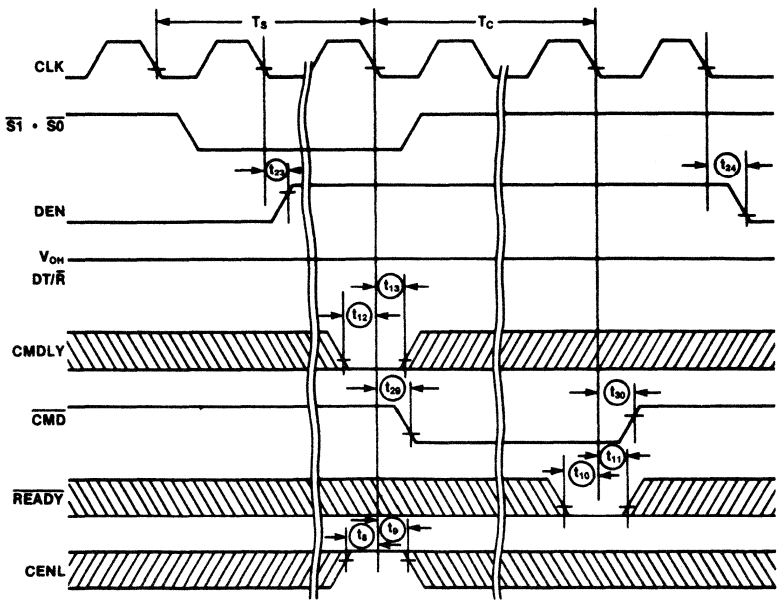
CENL, CMDLY, DEN Characteristics with MB = 0 and CEN = 1 During Write Cycle

SWITCHING WAVEFORMS (Cont'd.)



WF008100

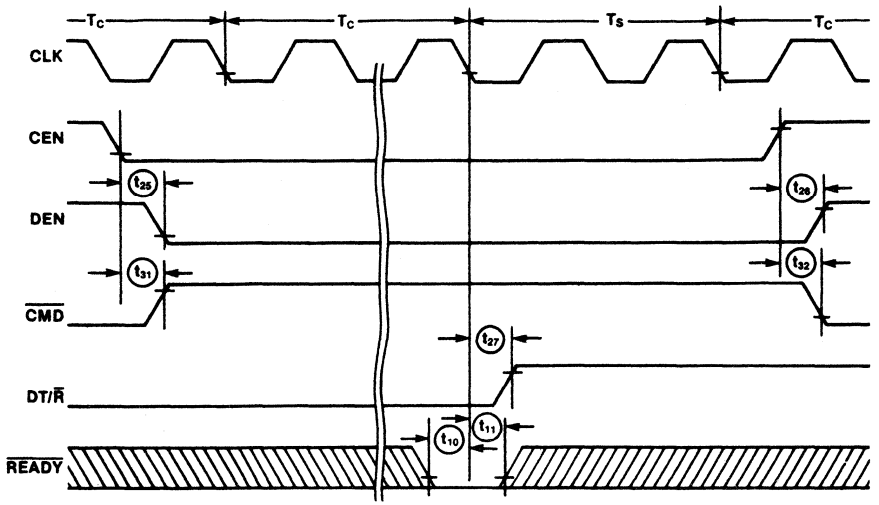
Read Cycle Characteristics with MB = 0 and CEN = 1



WF008110

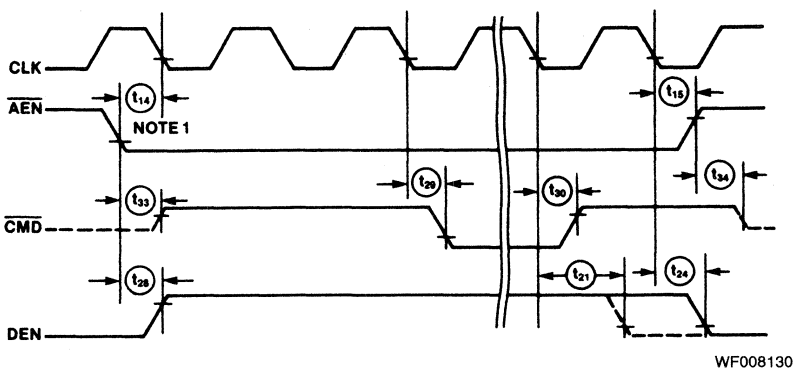
Write Cycle Characteristics With MB = 0 AND CEN = 1

SWITCHING WAVEFORMS (Cont'd.)



WF008120

CEN Characteristics with MB = 0

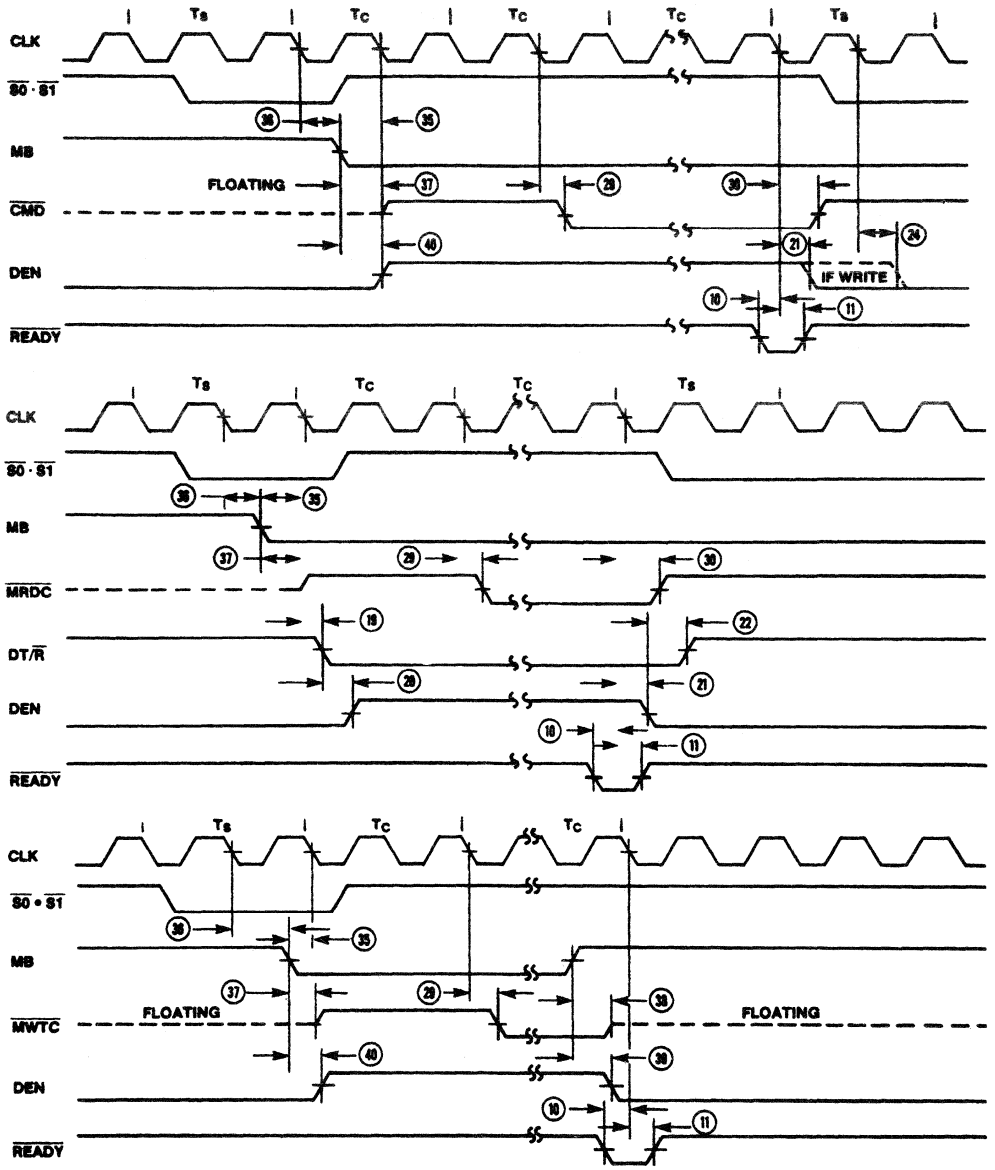


WF008130

$\overline{\text{AEN}}$ Characteristics with MB = 1

Note 1: $\overline{\text{AEN}}$ is an asynchronous input. $\overline{\text{AEN}}$ setup and hold time is specified to guarantee the response shown in the waveforms.

SWITCHING WAVEFORMS (Cont'd.)



WF022470

MB Characteristics with $\overline{AEN}$/CEN = HIGH

Notes: 1. MB is an asynchronous input. MB setup and hold times specified to guarantee the response shown in the waveforms.
2. If the setup time, t_{35}, is met two clock cycles will occur before $\overline{CMD}$ becomes active after the falling edge of MB.

8251/Am9551

Programmable Communication Interface
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Separate control and transmit register input buffers
- Synchronous or asynchronous serial data transfer
- Parity, overrun and framing errors detected
- Half or full duplex signalling
- Character length of 5, 6, 7 or 8 bits
- Internal or external synchronization
- Odd parity, even parity or no parity bit
- Modem interface controlled by processor
  - Programmable Sync pattern
  - Fully TTL compatible logic levels

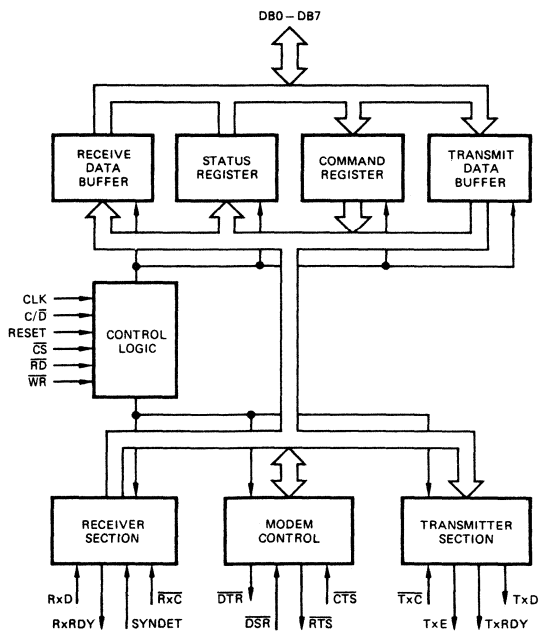
GENERAL DESCRIPTION

The 8251/Am9551 is a programmable serial data communication interface that provides a Universal Synchronous/Asynchronous Receiver/Transmitter (USART) function. It is normally used as a peripheral device for an associated processor and may be programmed by the processor to operate in a variety of standard serial communication formats.

The device accepts parallel data from the CPU, formats and serializes the information based on its current operating mode, and then transmits the data as a serial bit stream. Simultaneously, serial data can be received, converted into parallel form, deformatted, and then presented to the CPU. The USART can operate in an independent full duplex mode.

Data, Control, operation and format options are all selected by commands from an associated processor. This provides an unusual degree of flexibility and allows the 8251/Am9551 to service a wide range of communication disciplines and applications.

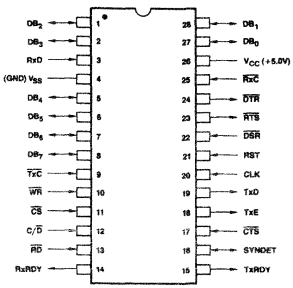
BLOCK DIAGRAM



Publication # 02334 | Rev. C | Amendment /0
Issue Date: April 1987

CONNECTION DIAGRAM
Top View

DIPs



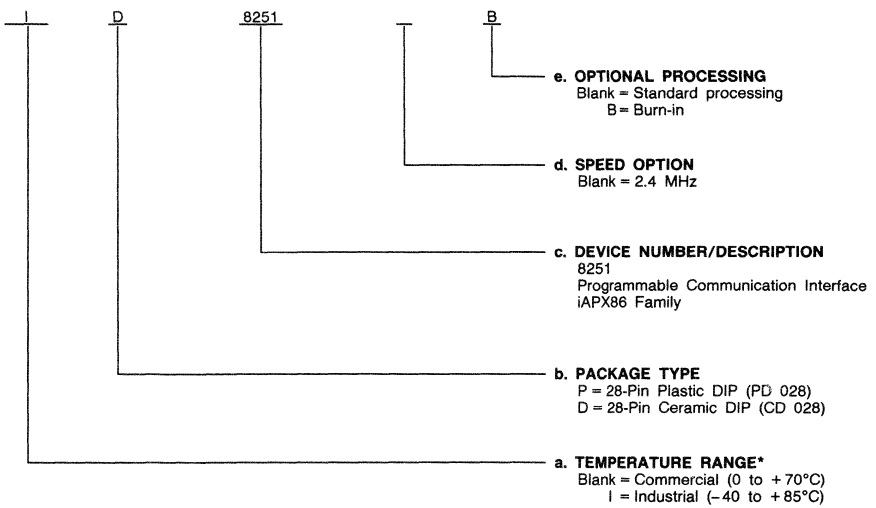
CD005482

Note: Pin 1 is marked for orientation.

8251 ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D	8251
D, ID	8251B

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

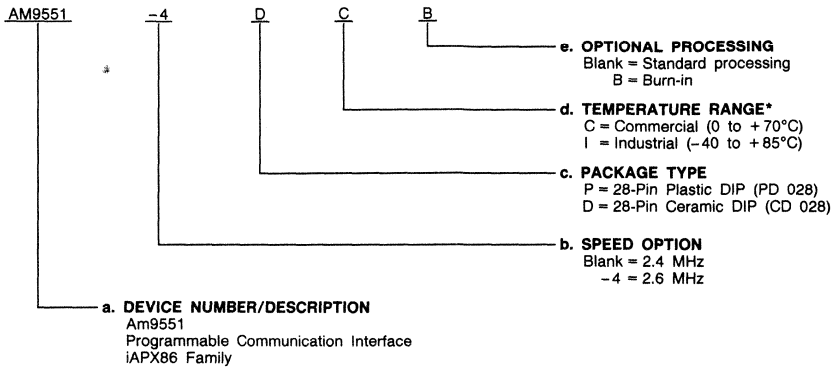
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

Am9551 ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM9551	PC, DC, DCB, DIB
AM9551-4	

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
27, 28, 1, 2, 5-8	Data Bus (DB_0-DB_7)	I/O	The Am9551 uses an 8-bit bidirectional data bus to exchange information with an associated processor. Internally, data is routed between the data bus buffers and the transmitter section or receiver section as selected by the Read ($\overline{RD}$) or Write ($\overline{WR}$) control inputs.
11	Chip Select ($\overline{CS}$)	I	The active low Chip Select input allows the Am9551 to be individually selected from other devices within its address range. When Chip Select is HIGH, reading or writing is inhibited, and the data bus output is in its high-impedance state.
21	Reset (RST)	I	The Am9551 will assume an Idle state when a high level is applied to the Reset input. When the Reset is returned LOW, the Am9551 will remain in the Idle state until it receives a new mode control instruction.
13	Read ($\overline{RD}$)	I	The active low Read input enables data to be transferred from the Am9551 to the processor.
10	Write ($\overline{WR}$)	I	The active low Write input enables data to be transferred from the processor to the Am9551.
12	Control/Data ($C\overline{D}$)	I	During a Read operation, if this input is at a high level, the status byte will be read, and if it is at a low level, the receive data will be read by the processor. When a Write operation is being performed, this input will indicate to the Am9551 that the bus information being written is a command if $C/\overline{D}$ is HIGH and data if $C/\overline{D}$ is LOW.

$C/\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	1	0	Am9551 DATA → DATA BUS
0	1	0	0	DATA BUS → Am9551 DATA
1	0	1	0	Am9551 STATUS → DATA BUS
1	1	0	0	DATA BUS → Am9551 COMMAND
X	X	X	1	DATA BUS → THREE-STATE

20	Clock (CLK)	I	This input is used for internal timing within the Am9551. It does not control the transmit or receive rate. However, it should be at least 30 times the receive or transmit rate in the synchronous mode and 4.5 times the receive or transmit rate in the asynchronous mode. The CLK frequency is also restricted by both an upper and a lower bound. This input is often connected to a clock from the associated processor.
3	Receiver Data (RxD)	I	Serial data is received from the communication line on this input.
25	Receiver Clock ($\overline{RxC}$)	I	The serial data on input RxD is clocked into the Am9551 by the $\overline{RxC}$ clock signal. In the synchronous mode, $\overline{RxC}$ is determined by the baud rate and supplied by the modem. In the asynchronous mode, $\overline{RxC}$ is 1, 16, or 64 times the baud rate as selected in the mode control instruction. Data is sampled by the Am9551 on the rising edge or $\overline{RxC}$.
14	Receiver Ready (RxRDY)	O	The RxRDY output signal indicates to the processor that data has been shifted into the receiver buffer from the receiver section and may be read. The signal is active high and will be reset when the buffer is read by the processor. RxRDY can be activated only if the receiver enable (RxE) has been set in the command register, even though the receiver may be running. If the processor does not read the receiver buffer before the next character is shifted from the receiver section, then an overrun error will be indicated in the status buffer.
16	Sync Detect (SYNDET)	I/O	This signal is used only in the synchronous mode. It can be either an output or input depending on whether the program is set for internal or external synchronization. As an output, a high level indicates when the sync character has been detected in the received data stream after the Internal Synchronization mode has been programmed. If the Am9551 is programmed to utilize two sync characters, then SYNDET will go to a high level when the last bit of the second sync character is received. SYNDET is reset when the status buffer is read or when a Reset signal is activated. SYNDET will perform as an input when the External Synchronization mode is programmed. External logic can supply a positive-going signal to indicate to the Am9551 that synchronization has been attained. This will cause it to initialize the assembly of characters on the next falling edge of $\overline{RxC}$. To successfully achieve synchronization, the SYNDET signal should be maintained in a high condition for at least one full period of $\overline{RxC}$.
19	Transmit Data (TxD)	O	Serial data is transmitted to the communication line on this output.
9	Transmitter Clock ($\overline{TxC}$)	O	The serial data on TxD is clocked out with the $\overline{TxC}$ signal. The relationship between clock rate and baud rate is similar to that for RxC. Data is shifted out of the Am9551 on the falling edge of $\overline{TxC}$.
15	Transmitter Ready (TxRDY)	O	The TxRDY output signal goes HIGH when data in the Transmit Data Buffer has been shifted into the transmitter section allowing the Transmit Data Buffer to accept the next byte from the processor. TxRDY will be reset when information is written into the Transmit Data Buffer. Loading command register also resets TxRDY. TxRDY will be available on this output pin only when the Am9551 is enabled to transmit (CTS = 0, TxEN = 1). However, the TxRDY bit in the status Buffer will always be set when the Transmit Data Buffer is empty regardless of the state of TxEN and CTS.
18	Transmitter Empty (TxE)	O	The TxE output signal goes HIGH when the Transmitter section has transmitted its data and is empty. The signal will remain HIGH until a new data byte is shifted from the Transmit Data Buffer to the Transmitter section. In the synchronous mode if the processor does not load a new byte into the buffer in time, TxE will, independent of the status of the TxEN bit in the command register, momentarily go to a high level as SYNC characters are loaded into the Transmitter Section.
24	Data Terminal Ready ($\overline{DTR}$)	O	This signal is a general purpose output which reflects the state of bit 1 in the Command instruction. It is commonly connected to an associated modem to indicate that the Am9551 is ready.
22	Data Set Ready ($\overline{DSR}$)	I	This is a general purpose input signal and forms part of the status byte that may be read by the processor. DSR is generally used as a response to DTR, by the Modem, to indicate that it is ready. The signal acts only as a flag and does not control any internal logic.
23	Request to Send ($\overline{RTS}$)	O	This is a general purpose output, similar to $\overline{DTR}$, and reflects the state of bit 5 in Command Instruction. It is normally used to initiate a data transmission by requesting the modem to prepare to send.
17	Clear to Send ($\overline{CTS}$)	I	This is a general purpose input signal used to enable the 8251/Am9551 to transmit data if the TxEN bit in the Command byte is a one. $\overline{CTS}$ is generally used as a response to RTS by a modem to indicate that transmission may begin. Designers not using $\overline{CTS}$ in their systems should remember to tie it LOW so that 8251/Am9551 data transmission will not be disabled.

PROGRAMMING INFORMATION

The microcomputer program controlling the Am9551 performs these tasks:

- Outputs control codes
- Inputs status
- Outputs data to be transmitted
- Inputs data which have been received

Control codes determine the mode in which the Am9551 will operate and are used to set or reset control signals output by the Am9551.

The Status register contents will be read by the program monitoring this device's operation to determine error conditions and when and how to read data, write data or output control codes. Program logic may be based on reading status bit levels, or control signals may be used to request interrupts.

Initializing the Am9551

The Am9551 may be initialized following a system reset or prior to starting a new serial I/O sequence. The USART must be reset following power up and subsequently may be reset at any time following completion of one activity and preceding a new set of operations. Following a reset, the Am9551 enters an Idle state in which it can neither transmit nor receive data.

The Am9551 is initialized with two, three or four control words from the processor. Figure 1 shows the sequence of control words needed to initialize the Am9551, for synchronous or for asynchronous operation. Note that in asynchronous operation a mode control is output to the device followed by a command. For synchronous operation, the mode control is followed by one or two SYNC characters and then a command.

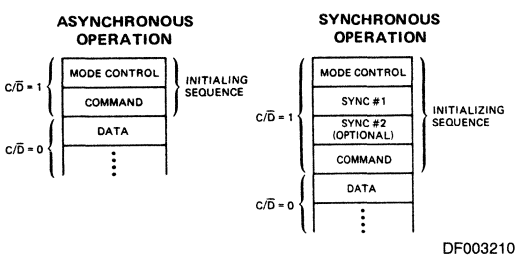


Figure 1. Control Word Sequence for Initialization

Only a single address is set aside for mode control bytes, command bytes and SYNC character bytes. For this to be possible, logic internal to the chip directs control information to its proper destination based on the sequence in which it is received. Following a reset, the first control code output is interpreted as a mode control. If the mode control specifies synchronous operation, then the next one or two bytes (as determined by the mode byte) output as control codes will be interpreted as SYNC characters. For either asynchronous or synchronous operation, the next byte output as a control code is interpreted as command. All subsequent bytes output as control codes are interpreted as commands. There are two ways in which control logic may return to anticipating a mode control input: either following an external Reset signal or following an internal Reset command.

MODE CONTROL CODES

The Am9551 interprets mode control codes as illustrated in Figures 2 and 3.

Control code bits 0 and 1 determine whether synchronous or asynchronous operation is specified. A non-zero value in bits 0 and 1 specifies asynchronous operation and defines the relationship between data transfer baud rate and receiver or transmitter clock rate. Asynchronous serial data may be received or transmitted on every clock pulse, on every 16th clock pulse, or on every 64th clock pulse. A zero in both bits 0 and 1 defines the mode of operation as synchronous.

For synchronous and asynchronous modes, control bits 2 and 3 determine the number of data bits which will be present in each data character.

For synchronous and asynchronous modes, bits 4 and 5 determine whether there will be a parity bit in each character, and if so, whether odd or even parity will be adopted. Thus in synchronous mode a character will consist of five, six, seven or eight data bits, plus an optional parity bit. In asynchronous mode, the data unit will consist of five, six, seven, or eight data bits, an optional parity bit, a preceding start bit, plus 1, 1½, or 2 trailing stop bits. Interpretation of subsequent bits differs for synchronous or asynchronous modes.

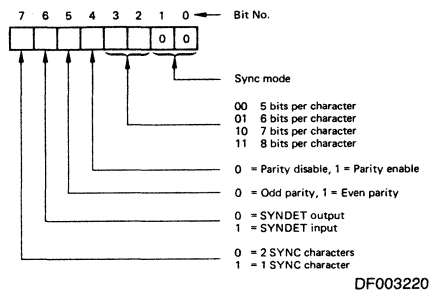


Figure 2. Synchronous Mode Control Code

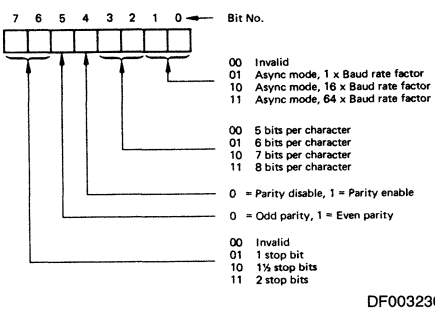


Figure 3. Asynchronous Mode Control Code

Control code bits 6 and 7 in asynchronous mode determine how many stop bits will trail each data unit. 1½ stop bits can only be specified with a 16x or 64x baud rate factor. In these two cases, the half stop bit will be equivalent to 8 or 32 clock pulses, respectively.

In synchronous mode, control bits 6 and 7 determine how character synchronization will be achieved. When SYNDET is an output, internal synchronization is specified; one or two SYNC characters, as specified by control bit 7, must be detected at the head of a data stream to establish synchronization.

Command Words

Command words are used to initiate specific functions within the Am9551, such as "reset all error flags" or "start searching for sync." Consequently, Command Words may be issued by

the micro-processor to the Am9551 at any time during the execution of a program in which specific functions are to be initiated within the communication circuit.

Figure 4 shows the format for the Command Word.

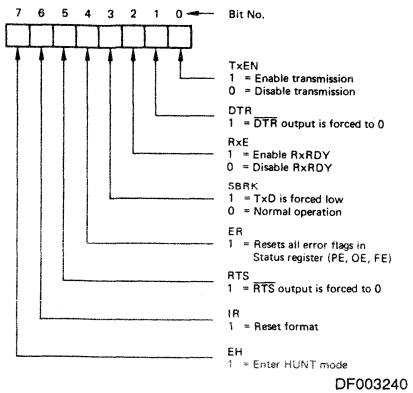


Figure 4. Am9551 Control Command

Bit 0 of the Command Word is the Transmit Enable bit (TxEN). Data transmission from the Am9551 cannot take place unless TxEN is set in the command register. Figure 5 defines the way in which TxEN, TxE and TxRDY combine to control transmitter operations.

Bit 1 is the Data Terminal Ready (DTR) bit. When the DTR command bit is set, the $\overline{DTR}$ output connection is active (low). DTR is used to advise a modem that the data terminal is prepared to accept or transmit data.

Bit 2 is the Receiver Enable Command bit (RxE). RxE is used to enable the RxRDY output signal. RxE prevents the RxRDY signal from being generated to notify the processor that a complete character is framed in the Receive Character Buffer. It does not inhibit the assembly of data characters at the input, however. Consequently, if communication circuits are active, characters will be assembled by the receiver and transferred to the Receiver Character Buffer. If RxE is disabled, the overrun error (OE) will probably be set; to insure proper operation, the overrun error is usually reset with the same command that enables RxE.

Bit 3 is the Send Break Command bit (SBRK). When SBRK is set, the transmitter output (TxD) is interrupted and a continuous binary "0" level (spacing) is applied to the TxD output signal. The break will continue until a subsequent Command Word is sent to the Am9551 to remove SBRK.

Bit 4 is the Error Reset bit (ER). When a Command Word is transmitted with the ER bit set, all three error flags in the Status Register are reset. Error Reset occurs when the Command Word is loaded into the Am9551. No latch is provided in the Command Register to save the ER command bit.

Bit 5, the Request To Send Command bit (RTS), sets a latch to reflect the RTS signal level. The output of this latch is created independently of other signals in the Am9551. As a result, data transfers may be made by the microprocessor to the Transmit Register, and data may be actively transmitted to the communication line through TxD regardless of the status of RTS.

Bit 6, the Internal Reset (IR), causes the Am9551 to return to the Idle mode. All functions within the Am9551 cease and no new operation can be resumed until the circuit is reinitialized. If the operating mode is to be altered during the execution of a microprocessor program, the Am9551 must first be reset. Either the external reset connection can be activated, or the Internal Reset Command can be sent to the Am9551. Internal Reset is a momentary function performed only when the command is issued.

Bit 7 is the Enter Hunt command bit (EH). The Enter Hunt mode command is only effective for the Am9551 when it is operating in the Synchronous mode. EH causes the receiver to stop assembling characters at the RxD input and start searching for the prescribed sync pattern. Once the "Enter Hunt" mode has been initiated, the search for the sync pattern will continue indefinitely until EH is reset when a subsequent Command Word is sent, when the IR command is sent to the Am9551, or when SYNC characters are recognized.

TxEN	Txe	TxRDY	
1	1	1	Transmit Output Register and Transmit Character Buffer empty. TxD continues to mark if Am9551 is in the asynchronous mode. TxD will send Sync pattern if Am9551 is in the Synchronous Mode. Data can be entered into Buffer.
1	0	1	Transmit Output Register is shifting a character. Transmit Character Buffer is available to receive a new byte from the processor.
1	1	0	Transmit Register has finished sending. A new character is waiting for transmission. This is a transient condition.
1	0	0	Transmit Register is currently sending and an additional character is stored in the Transmit Character Buffer for transmission.
0	0/1	0/1	Transmitter is disabled.

Figure 5. Operation of the Transmitter Section as a Function of TxE, TxRDY and TxEN

Status Register

The Status Register maintains information about the current operational status of the Am9551. Figure 6 shows the fomat of the Status Register.

TxRDY signals the processor that the Transmit Character Buffer is empty and that the Am9551 can accept a new character for transmission.

RxRDY signals the processor that a completed character is holding in the Receive Character Buffer Register for transfer to the processor.

TxE signals the processor that the Transmit Register is empty.

PE is the Parity Error signal indicating to the CPU that the character stored in the Receive Character Buffer was received with an incorrect number of binary "1" bits.

OE is the receiver Overrun Error. OE is set whenever a byte stored in the Receiver Character Register is overwritten with a new byte before being transferred to the processor.

FE is the character framing error which indicates that the asynchronous mode byte stored in the Receiver Character Buffer was received with incorrect character bit format as specified by the current mode.

SYNDET is the synchronous mode status bit associated with internal sync detection.

DSR is the status bit set by the external Data Set Ready signal to indicate that the communication Data Set is operational. All status bits are set by the functions described for them. SYNDET is reset whenever the processor reads the Status Register. OE, FE, PE are reset only by command.

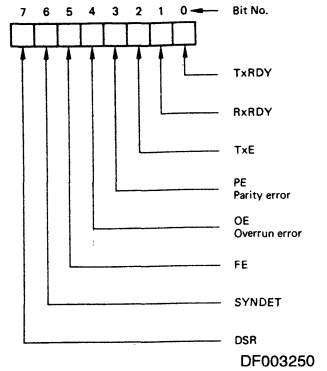


Figure 6. The Am9551 Status Register

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to +7.0V
All Signal Voltages
with Respect to V_{SS} ..... −0.5V to +7.0V
Power Dissipation 1.0W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5V ±5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5V ±10%

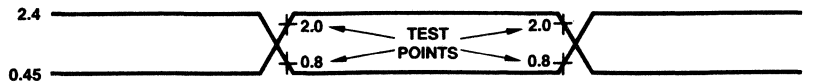
Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified.

Parameters	Description	Test Conditions	8251		Am9551		Units
			Min	Max	Min	Max	
V_{OH}	Output HIGH Voltage	I_{OH} = −200μA			2.4		Volts
		I_{OH} = −100μA	2.4				
V_{OL}	Output LOW Voltage	I_{OL} = 3.2mA				0.45	Volts
		I_{OL} = 1.6mA		0.45			
V_{IH}	Input HIGH Voltage		2.2	V_{CC}	2.2	V_{CC}	Volts
V_{IL}	Input LOW Voltage		−0.5	0.8	−0.5	0.8	Volts
I_{LI}	Input Load Current	$V_{SS} \leq V_{IN} \leq V_{CC}$		±10		±10	μA
I_{DL}	Data Bus Leakage	V_{OUT} = 0.45V		−50		−50	μA
		V_{OUT} = V_{CC}		10		10	
I_{CC}	V_{CC} Supply Current	T_A = 0°C		80		80	mA
C_O	Output Capacitance	fc = 1.0MHz, Inputs = 0V		15		15	pF
C_I	Input Capacitance			10		10	pF
$C_{I/O}$	I/O Capacitance			20		20	pF

See Section 6 for Thermal Characteristics Information.

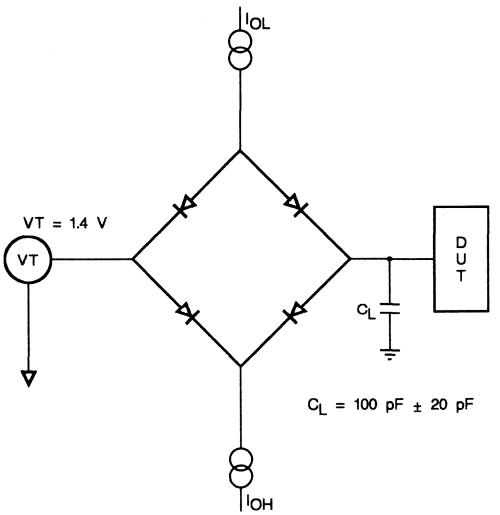
SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF006490

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."

SWITCHING TEST CIRCUIT



TC003851

This test circuit is the dynamic load of a Teradyne J941.

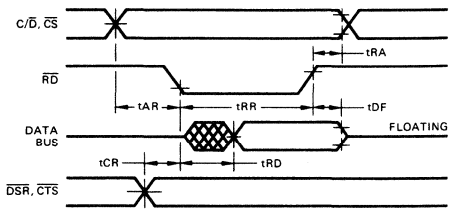
SWITCHING CHARACTERISTICS over Operating Range unless otherwise specified.

Parameters	Description		8251 Min	8251 Max	Am9551 Min	Am9551 Max	Am9551-4 Min	Am9551-4 Max	Units
t_{AR}	$\overline{CS}$, C/$\overline{D}$ Stable to $\overline{READ}$ Low Set-up Time		50		50		50		ns
t_{AW}	$\overline{CS}$, C/$\overline{D}$ Stable to $\overline{WRITE}$ Low Set-up Time		20		20		20		ns
t_{CR}	$\overline{DSR}$, $\overline{CTS}$ to $\overline{READ}$ Low Set-up Time			16		16		16	tCY
t_{CY}	Clock Period		.420	1.35	.380	1.35	.380	1.35	μs
t_{DF}	$\overline{READ}$ High to Data Bus Off Delay		25	200	25	200	25	200	ns
t_{DTx}	TxC Low to TxD Delay			1.0		1.0		1.0	μs
t_{DW}	Data to $\overline{WRITE}$ High Set-up Time		200		150		100		ns
t_{ES}	External SYNDET to RxC Low Set-up Time		16		16		16		tCY
t_{HRx}	Sampling Pulse to Rx Data Hold Time		2.0		2.0		2.0		μs
t_{IS}	Data Bit (Center) to Internal SYNDET Delay			25		25		25	tCY
$t_{\phi W}$	Clock Puise Width		220	0.6tCY	175	0.6tCY	175	0.6tCY	ns
t_R, t_F	Clock Rise & Fall Time		0	50	0	50	0	50	ns
t_{RA}	$\overline{READ}$ High to $\overline{CS}$, C/D Hold Time		5.0		5.0		5.0		ns
t_{RD}	$\overline{READ}$ Low to Data Bus On Delay			350		250		180	ns
t_{RPD}	Receiver Clock High Time	1xBaud Rate	15		15		15		tCY
		16x & 64x Baud Rate	3.0		3.0		3.0		
t_{RPW}	Receiver Clock High Time	1xBaud Rate	12		12		12		tCY
		16x & 64x Baud Rate	3.0		3.0		3.0		
t_{RR}	$\overline{READ}$ Pulse Width		430		380		250		ns
t_{RV}	Time Between $\overline{WRITE}$ Pulses During Initialization (Note 1)		6.0		6.0		6.0		tCY
t_{Rx}	Data Bit (Center) to RxRDY Delay			20		20		20	tCY
t_{SRx}	Rx Data to Sampling Pulse Set-up Time		2.0		2.0		2.0		μS
t_{TPD}	Transmitter Clock High Time	1xBaud Rate	15		15		15		tCY
		16x & 64x Baud Rate	3.0		3.0		3.0		
t_{TPW}	Transmitter Clock Low Time	1xBaud Rate	12		12		12		tCY
		16x & 64x Baud Rate	1.0		1.0		1.0		
t_{TX}	Data Bit (Center) to TxRDY Delay			16		16		16	tCY
t_{TxE}	Data Bit (Center) to Tx EMPTY Delay			16		16		16	tCY
t_{WA}	$\overline{WRITE}$ High to $\overline{CS}$, C/D Hold Time		20		20		20		ns
t_{WC}	$\overline{WRITE}$ High to TxE, $\overline{DTR}$, $\overline{RTS}$ Delay			16		16		16	tCY
t_{WD}	$\overline{WRITE}$ High to Data Hold Time		40		40		40		ns
t_{WW}	$\overline{WRITE}$ Pulse Width		400		380		250		ns
f_{Rx}	Receiver Clock Frequency	1x Baud Rate	DC	56	DC	56	DC	56	kHz
		16x & 64x Baud Rate	DC	520	DC	520	DC	520	
f_{Tx}	Transmitter Clock Frequency	1x Baud Rate	DC	56	DC	56	DC	56	kHz
		16x & 64x Baud Rate	DC	520	DC	520	DC	520	

Notes: 1. This time period between write pulses is specified for initialization purposes only when MODE, SYNC 1, SYNC 2, COMMAND and first DATA BYTE are written into the Am9551. Subsequent writing of both COMMAND and DATA are only allowed when TxRDY = 1. t_{RV} after internal Reset = 8 * t_{CY}.
2. Reset Pulse Width = 6tCY min.
3. Switching Characteristics parameters are listed in alphabetical order.

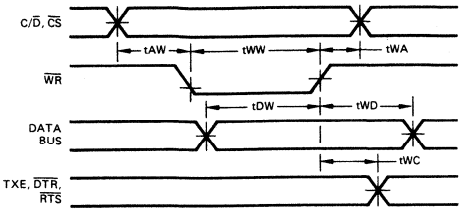
SWITCHING WAVEFORMS

READ OPERATION



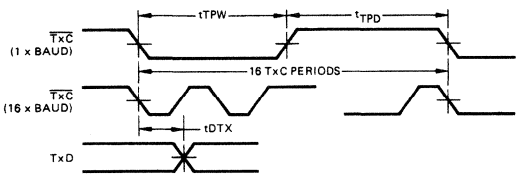
WF006500

WRITE OPERATION



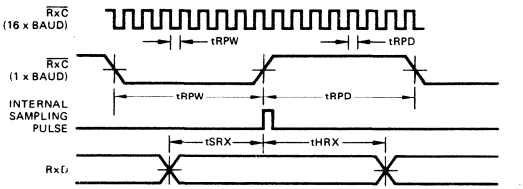
WF006510

TRANSMITTER CLOCK AND DATA



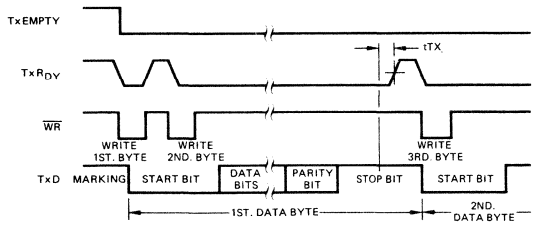
WF006521

RECEIVER CLOCK AND DATA



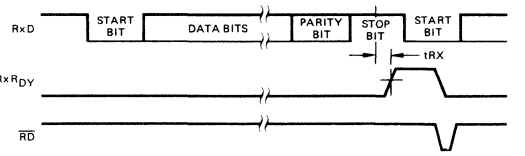
WF006530

TxRD TIMING (ASYNC MODE)



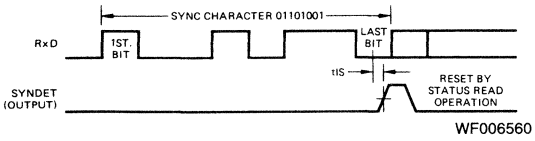
WF006540

RxRDY TIMING (ASYNC MODE)



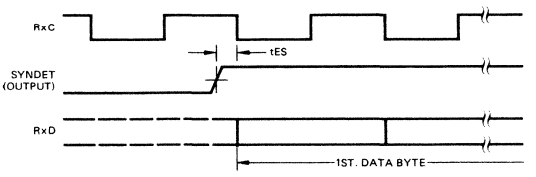
WF006550

INTERNAL SYNC DETECT (SYNC MODE ONLY)



WF006560

EXTERNAL SYNC DETECT (SYNC MODE ONLY)



WF006570

8251A

Programmable Communication Interface
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Synchronous and Asynchronous Operation
- Synchronous 5 – 8 Bit Characters; Internal or External Character Synchronization; Automatic Sync Insertion
- Asynchronous 5 – 8 Bit Characters; Clock Rate – 1, 16 or 64 Times Baud Rate; Break Character Generation; 1, 1 1/2, or 2 Stop Bits; False Start Bit Detection; Automatic Break Detect and Handling
- Synchronous Baud Rate – DC to 64K Baud
- Asynchronous Baud Rate – DC to 19.2K Baud
- Full-Duplex, Double-Buffered Transmitter and Receiver
- Error Detection – Parity, Overrun and Framing
- Compatible with an Extended Range of Microprocessors
- 28-Pin DIP Package and PLCC
- All Inputs and Outputs are TTL Compatible

GENERAL DESCRIPTION

The 8251A is the enhanced version of the industry standard, 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART), designed for data communications with microprocessor families, such as the iAPX86, 88. The 8251A is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM "bi-sync"). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals, such as SYNDET, TxEMPTY. The chip is fabricated using N-channel silicon gate technology.

BLOCK DIAGRAM

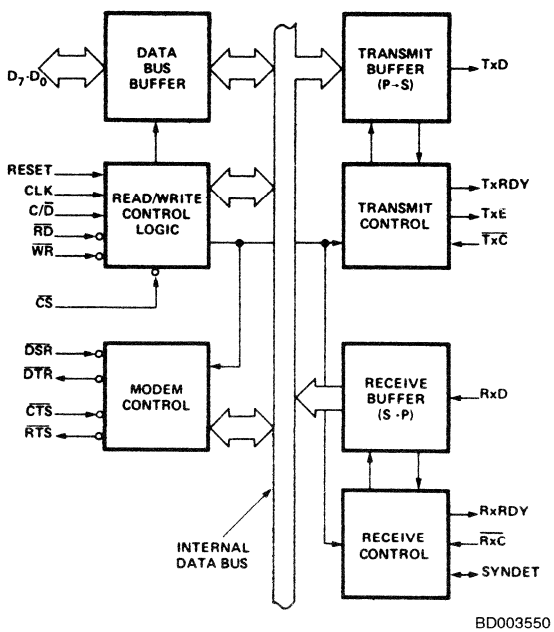


Figure 1.

Publication # 04133 | Rev. B | Amendment /0
Issue Date: May 1987

CONNECTION DIAGRAMS
Top View

DIPs

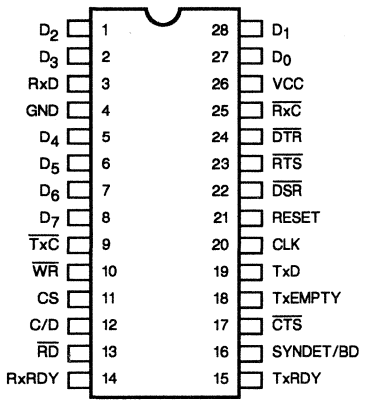


Figure 2.1

PLCC

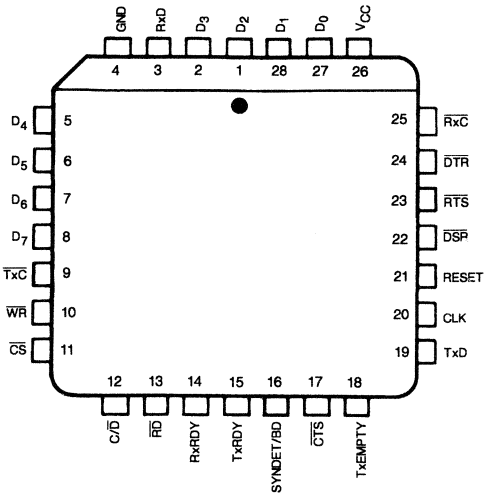


Figure 2.2

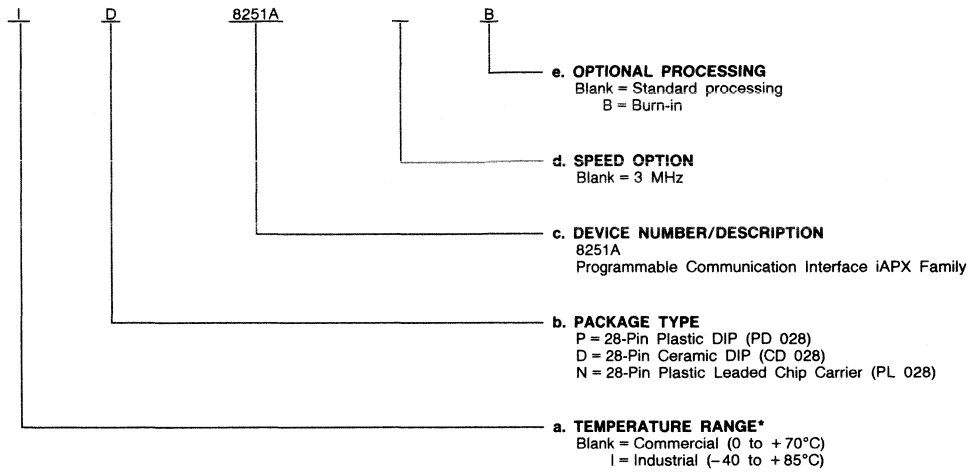
Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Temperature Range
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D, N	8251A
D, ID	8251AB

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION and RELATED INFORMATION

CLK (Clock)

The CLK input is used to generate internal device timing and is normally connected to the Phase 2 (TTL) output of the Clock Generator. No external inputs or outputs are referenced to CLK but the frequency of CLK must be greater than 30 times the receiver or Transmitter data bit rates.

$\overline{WR}$ (Write)

A "LOW" on this input informs the 8251A that the CPU is writing data or control words to the 8251A.

$\overline{RD}$ (Read)

A "LOW" on this input informs the 8251A that the CPU is reading data or status information from the 8251A.

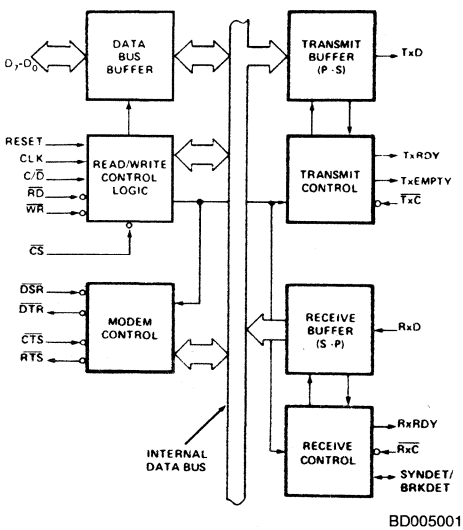


Figure 3. 8251A Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

C/$\overline{D}$	$\overline{RD}$	$\overline{W}$	$\overline{CS}$	
0	0	1	0	8251A DATA → DATA BUS
0	1	0	0	DATA BUS → 8251A DATA
1	0	1	0	STATUS → DATA BUS
1	1	0	0	DATA BUS → CONTROL
X	1	1	0	DATA BUS → 3 STATE
X	X	X	1	DATA BUS → 3 STATE

C/$\overline{D}$ (Control/Data)

This input, in conjunction with the $\overline{WR}$ and $\overline{RD}$ inputs, informs the 8251A that the word on the Data Bus is either a data character, control word or status information.

1 = CONTROL/STATUS; 0 = DATA.

$\overline{CS}$ (Chip Select)

A "LOW" on this input selects the 8251A. No reading or writing will occur unless the device is selected. When $\overline{CS}$ is high, the Data Bus is in the float state and $\overline{RD}$ and $\overline{WR}$ have no effect on the chip.

Modem Control

The 8251A has a set of control inputs and outputs that can be used to simplify the interface to almost any modem. The modem control signals are general purpose in nature and can be used for functions other than modem control, if necessary.

$\overline{DSR}$ (Data Set Ready)

The $\overline{DSR}$ input signal is a general-purpose, 1-bit inverting input port. Its condition can be tested by the CPU using a Status Read operation. The $\overline{DSR}$ input is normally used to test modem conditions such as Data Set Ready.

$\overline{DTR}$ (Data Terminal Ready)

The $\overline{DTR}$ output signal is a general-purpose, 1-bit inverting output port. It can be set "LOW" by programming the appropriate bit in the Command Instruction word. The $\overline{DTR}$ output signal is normally used for modem control such as Data Terminal Ready.

$\overline{RTS}$ (Request to Send)

The $\overline{RTS}$ output signal is a general-purpose, 1-bit inverting output port. It can be set "LOW" by programming the appropriate bit in the Command Instruction word. The $\overline{RTS}$ output signal is normally used for modem control such as Request to Send.

$\overline{CTS}$ (Clear to Send)

A "LOW" on this input enables the 8251A to transmit serial data if the Tx Enable bit in the Command byte is set to a "one." If either a Tx Enable off or $\overline{CTS}$ off condition occurs while the Tx is in operation, the Tx will transmit all the data in the USART, written prior to Tx Disable command before shutting down.

Transmitter Buffer

The Transmitter Buffer accepts parallel data from the Data Bus Buffer, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin on the falling edge of $\overline{TxC}$. The transmitter will begin transmission upon being enabled if $\overline{CTS}$ = 0. The TxD line will be held in the marking state immediately upon a master Reset or when Tx Enable or $\overline{CTS}$ is off or the transmitter is empty.

Transmitter Control

The Transmitter Control manages all activities associated with the transmission of serial data. It accepts and issues signals both externally and internally to accomplish this function.

TRDY (Transmitter Ready)

This output signals the CPU that the transmitter is ready to accept a data character. The TxRDY output pin can be used as an interrupt to the system, since it is masked by TxEnable; or for Polled operation, the CPU can check TxRDY using a Status Read operation. TxRDY is automatically reset by the leading edge of $\overline{WR}$ when a data character is loaded from the CPU.

Note that when using the Polled operation, the TxRDY status bit is not masked by TxEnable, but will only indicate the Empty/Full Status of the Tx Data Input Register.

TxE (Transmitter Empty)

When the 8251A has no characters to send, the TxEMPTY output will go "HIGH." It resets upon receiving a character from CPU if the transmitter is enabled. TxEMPTY remains high

when the transmitter is disabled. TxEMPTY can be used to indicate the end of a transmission mode, so that the CPU "knows" when to "turn the line around" in the half-duplex operational mode.

In the Synchronous mode, a "high" on this output indicates that a character has not been loaded and the SYNC character or characters are about to be or are being transmitted automatically as "fillers." TxEMPTY does not go LOW when the SYNC characters are being shifted out.

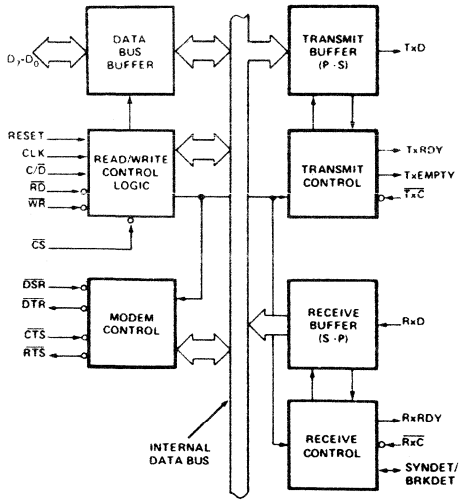


Figure 4. 8251A Block Diagram Showing Modem and Transmitter Buffer and Control Functions

$\overline{\text{TxC}}$ (Transmitter Clock)

The Transmitter Clock controls the rate at which the character is to be transmitted. In the Synchronous transmission mode, the Baud Rate (1x) is equal to the $\overline{\text{TxC}}$ frequency. In Asynchronous transmission mode, the baud rate is a fraction of the actual $\overline{\text{TxC}}$ frequency. A portion of the mode instruction selects this factor; it can be 1, 1/16 or 1/64 the $\overline{\text{TxC}}$.

For Example:

If Baud Rate equals 110 Baud,
$\overline{\text{TxC}}$ equals 110 Hz in the 1x mode.
$\overline{\text{TxC}}$ equals 1.72 kHz in the 16x mode.
$\overline{\text{TxC}}$ equals 7.04 kHz in the 64x mode.

The falling edge of $\overline{\text{TxC}}$ shifts the serial data out of the 8251A.

Receiver Buffer

The Receiver accepts serial data, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU. Serial data is input to RxD pin and is clocked in on the rising edge of $\overline{\text{RxC}}$.

Receiver Control

This functional block manages all receiver-related activities which consist of the following features.

The RxD initialization circuit prevents the 8251A from mistaking an unused input line for an active low data line in the "break condition." Before starting to receive serial characters on the RxD line, a valid "1" must first be detected after a chip master Reset. Once this has been determined, a search for a valid LOW (Start bit) is enabled. This feature is only active in the asynchronous mode and is only done once for each master Reset.

The False Start bit detection circuit prevents false starts due to a transient noise spike by first detecting the falling edge and then strobing the nominal center of the Start bit (RxD = LOW).

Parity error detection sets the corresponding status bit.

The Framing Error status bit is set if the Stop bit is absent at the end of the data byte (asynchronous mode).

RxRDY (Receiver Ready)

This output indicates that the 8251A contains a character that is ready to be input to the CPU. RxRDY can be connected to the interrupt structure of the CPU or, for polled operation, the CPU can check the condition of RxRDY using a Status Read operation.

RxEnable, when off, holds RxRDY in the Reset Condition. For Asynchronous mode, to set RxRDY, the Receiver must be enable to sense a Start Bit, and a complete character must be assembled and transferred to the Data Output Register. For Synchronous mode, to set RxRDY, the Receiver must be enabled, and a character must finish assembly and be transferred to the Data Output Register.

Failure to read the received character from the Rx Data Output Register prior to the assembly of the next Rx Data character will set overrun condition error, and the previous character will be written over and lost. If the Rx Data is being read by the CPU when the internal transfer is occurring, overrun error will be set, and the old character will be lost.

$\overline{\text{RxC}}$ (Receiver Clock)

The Receiver Clock controls the rate at which the character is to be received. In Synchronous Mode, the Baud Rate (1x) is equal to the actual frequency of $\overline{\text{RxC}}$. In Asynchronous Mode, the Baud Rate is a fraction of the actual $\overline{\text{RxC}}$ frequency. A portion of the mode instruction selects this factor: 1, 1/16 or 1/64 the $\overline{\text{RxC}}$.

For example:

Baud Rate equals 300 Baud, if
$\overline{\text{RxC}}$ equals 300 Hz in the 1x mode;
$\overline{\text{RxC}}$ equals 4800 Hz in the 16x mode;
$\overline{\text{RxC}}$ equals 19.2 kHz in the 64x mode.

Baud Rate equals 2400 Baud, if
$\overline{\text{RxC}}$ equals 2400 Hz in the 1x mode;
$\overline{\text{RxC}}$ equals 38.4 kHz in the 16x mode;
$\overline{\text{RxC}}$ equals 153.6 kHz in the 64x mode.

Data is sampled into the 8251A on the rising edge of $\overline{\text{RxC}}$.

NOTE: In most communications systems, the 8251A will be handling both the transmission and reception operations of a single link. Consequently, the Receive and Transmit Baud Rates will be the same. Both $\overline{\text{TxC}}$ and $\overline{\text{RxC}}$ will require identical frequencies for this operation and can be tied together and connected to a single frequency source (Baud Rate Generator) to simplify the interface.

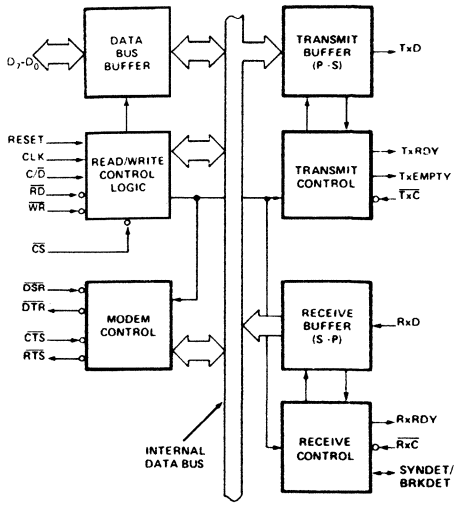


Figure 5. 8251A Block Diagram Showing Receiver Buffer and Control Functions

SYNDET (SYNC Detect/ BRKDET Break Detect)

This pin is used in Synchronous Mode for SYNDET and may be used as either input or output, programmable through the Control Word. It is reset to output mode LOW upon RESET. When used as an output (internal Sync mode), the SYNDET pin will go "HIGH" to indicate that the 8251A has located the SYNC character in the Receive mode. If the 8251A is programmed to use double Sync characters (bisync), then SYNDET will go "HIGH" in the middle of the last bit of the second Sync character. SYNDET is automatically reset upon a Status Read operation.

When used as an input (external SYNC detect mode), a positive going signal will cause the 8251A to start assembling data characters on the rising edge of the next $\overline{RxC}$. Once in SYNC, the "HIGH" input signal can be removed. When External SYNC Detect is programmed, Internal SYNC Detect is disabled.

BREAK (Async Mode Only)

This output will go HIGH whenever the receiver remains LOW through two consecutive stop bit sequences (including the start bits, data bits, and parity bits). Break Detect may also be read as a Status bit. It is reset only upon a master chip Reset or Rx Data returning to a "one" state.

FEATURES AND ENHANCEMENTS

The 8251A is an advanced design of the industry standard USART, the 8251. The 8251A operates with an extended range of microprocessors and maintains compatibility with the 8251. Familiarization time is minimal because of compatibility and involves only knowing the additional features and enhancements, and reviewing the AC and DC specifications of the 8251A.

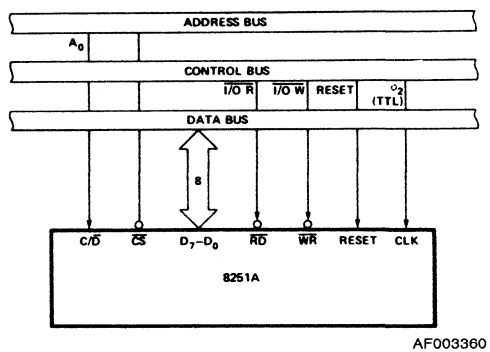


Figure 6. 8251A Interface to 8080A Standard System Bus

The 8251A incorporates all the key features of the 8251 and has the following additional features and enhancements:

- 8251A has double-buffered data paths with separate I/O registers for control, status, Data In, and Data Out, which considerably simplifies control programming and minimizes CPU overhead.
- In asynchronous operations, the Receiver detects and handles "break" automatically, relieving the CPU of this task.
- A refined Rx initialization prevents the Receiver from starting when in "break" state, preventing unwanted interrupts from a disconnected USART.
- At the conclusion of a transmission, TxD line will always return to the marking state unless SBRK is programmed.
- Tx Enable logic enhancement prevents a Tx Disable command from halting transmission until all data previously written has been transmitted. The logic also prevents the transmitter from turning off in the middle of a word.

- When External Sync Detect is programmed, Internal Sync Detect is disabled, and an External Sync Detect status is provided via a flip-flop which clears itself upon a status read.
- Possibility of false sync detect is minimized by ensuring that, if double character sync is programmed, the characters be contiguously detected and the Rx register cleared to all ones whenever the Enter Hunt command is issued in Sync mode.
- As long as the 8251A is not selected, the $\overline{RD}$ and $\overline{WR}$ do not affect the internal operation of the device.
- The 8251A Status can be read at any time, but the status update will be inhibited during status read.
- The 8251A is free from extraneous glitches and has enhanced AC and DC characteristics, providing higher speed and better operating margins.
- Synchronous Baud rate from DC to 64K.

PRODUCT OVERVIEW

General

The 8251A is a Universal Synchronous/Asynchronous Receiver/Transmitter designed for a wide range of microcomputers, such as the 8080A, 8085, 8086 and 8088. Like other I/O devices in a microcomputer system, its functional configuration is programmed by the system's software for maximum flexibility. The 8251A can support most serial data techniques in use, including IBM "bi-sync."

In a communication environment, an interface device must convert parallel format system data into serial format for transmission and convert incoming serial format data into parallel system data for reception. The interface device must also delete or insert bits or characters that are functionally unique to the communication technique. In essence, the interface should appear "transparent" to the CPU, a simple input or output of byte-oriented system data.

Data Bus Buffer

This 3-state, bidirectional, 8-bit buffer is used to interface the 8251A to the system Data Bus. Data is transmitted or received by the buffer upon execution of INput or OUTput instructions of the CPU. Control words, Command words and Status information are also transferred through the Data Bus Buffer. The Command Status, Data-In and Data-Out registers are separate, 8-bit registers communicating with the system bus through the Data Bus Buffer.

This functional block accepts inputs from the system Control bus and generates control signals for overall device operation. It contains the Control Word Register and Command Word Register that store the various control formats for the device functional definition.

RESET (Reset)

A "HIGH" on this input forces the 8251A into an "Idle" mode. The device will remain at "Idle" until a new set of control words is written into the 8251A to program its functional definition. Minimum RESET pulse width is 6 t_{cy} (clock must be running).

A command reset operation also puts the device into the "Idle" state.

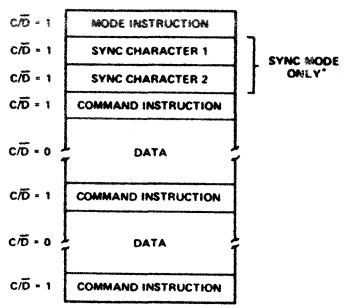
DETAILED DESCRIPTION

General

The complete functional definition of the 8251A is programmed by the system's software. A set of control words must be sent out by the CPU to initialize the 8251A to support the desired communications format. These control words will program the: BAUD RATE, CHARACTER LENGTH, NUMBER OF STOP BITS, SYNCHRONOUS or ASYNCHRONOUS OPERATION, EVEN/ODD/OFF PARITY, etc. In the Synchronous Mode, options are also provided to select either internal or external character synchronization.

Once programmed, the 8251A is ready to perform its communication functions. The TxRDY output is raised "HIGH" to signal the CPU that the 8251A is ready to receive a data character from the CPU. This output (TxRDY) is reset automatically when the CPU writes a character into the 8251A. On the other hand, the 8251A receives serial data from the MODEM or I/O device. Upon receiving an entire character, the RxRDY output is raised "HIGH" to signal the CPU that the 8251A has a complete character ready for the CPU to fetch. RxRDY is reset automatically upon the CPU data read operation.

The 8251A cannot begin transmission until the Tx Enable (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send ($\overline{CTS}$) input. The TxD output will be held in the marking state upon Reset.



DF003920

*THE SECOND SYNC CHARACTER IS SKIPPED IF MODE INSTRUCTION HAS PROGRAMMED THE 8251A TO SINGLE CHARACTER SYNC MODE. BOTH SYNC CHARACTERS ARE SKIPPED IF MODE INSTRUCTION HAS PROGRAMMED THE 8251A TO ASYNC MODE.

Figure 7. Typical Data Block

Programming the 8251A

Prior to starting data transmission or reception, the 8251A must be loaded with a set of control words penetrated by the CPU. These control signals define the complete functional definition of the 8251A and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instruction

This instruction defines the general operational characteristics of the 8251A. It must follow a Reset operation (internal or external). Once the Mode Instruction has been written into the 8251A by the CPU, SYNC characters or Command Instructions may be written.

Command Instruction

This instruction defines a word that is used to control the actual operation of the 8251A.

Both the Mode and Command Instructions must conform to a specified sequence for proper device operation (see Figure 7). The Mode Instruction must be written immediately following a Reset operation, prior to using the 8251A for data communication.

All control words written into the 8251A after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251A at any time in the data block during the operation of the 8251A. To return to the Mode Instruction format, the master Reset bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251A back into the Mode Instruction format. Command Instructions must follow the Mode Instructions or Sync characters.

Mode Instruction Definition

The 8251A can be used for either Asynchronous or Synchronous data communication. To understand how the Mode Instruction defines the functional operation of the 8251A, the designer can best view the device as two separate components, one Asynchronous and the other Synchronous, sharing the same package. The format definition can be changed only after a master chip Reset. For explanation purposes the two formats will be isolated.

NOTE: When parity is enabled, it is not considered as one of the data bits for the purpose of programming the word length. The actual parity bit received on the Rx Data line cannot be read on the Data Bus. In the case of a programmed character length of less than 8 bits, the least significant Data Bus bits will hold the data; unused bits are "don't care" when writing data to the 8251A and will be "zeros" when reading the data from the 8251A.

Asynchronous Mode (Transmission)

Whenever a data character is sent by the CPU, the 8251A automatically adds a Start bit (low level) followed by the data bits (least significant bit first) and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TxD output. The serial data is shifted out on the falling edge of $\overline{TxC}$ at a rate equal to 1, 1/16, or 1/64 that of the $\overline{TxC}$, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TxD if commanded to do so.

When no data characters have been loaded into the 8251A, the TxD output remains "HIGH" (marking) unless a Break (continuously LOW) has been programmed.

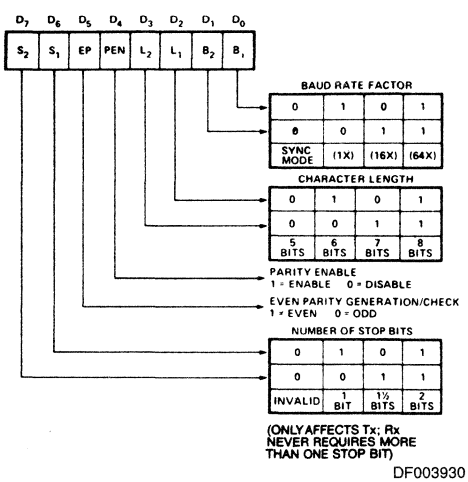
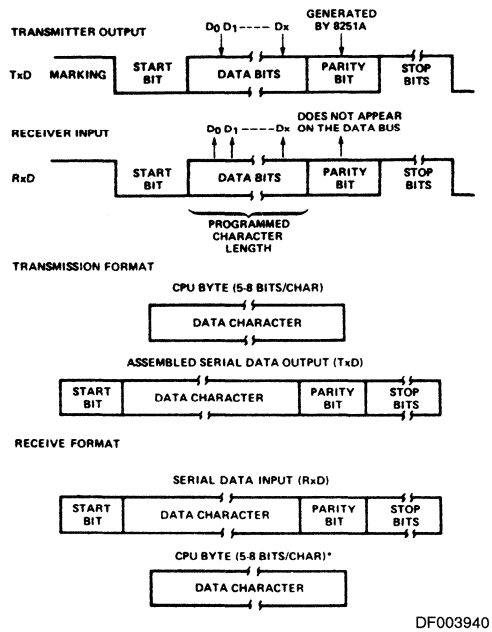


Figure 8. Mode Instruction Format, Asynchronous Mode

Asynchronous Mode (Receive)

The RxD line is normally HIGH. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center (16X or 64X mode only). If a LOW is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter thus locates the center of the data bits, the parity bit (if it exists) and the STOP bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RxD pin with the rising edge of $\overline{RxC}$. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. Note that the receiver requires only one STOP bit, regardless of the number of STOP bits programmed. This character is then loaded into the parallel I/O buffer of the 8251A. The RxRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN Error flag is raised (thus the previous character is lost). All of the error flags can be reset by an Error Reset Instruction. The occurrence of any of these errors will not affect the operation of the 8251A.



*Note: IF CHARACTER LENGTH IS DEFINED AS 5, 6 OR 7 BITS, THE UNUSED BITS ARE SET TO "ZERO."

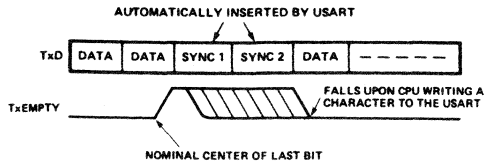
Figure 9. Asynchronous Mode

Synchronous Mode (Transmission)

The TxD output is continuously HIGH until the CPU sends its first character to the 8251A which usually is a SYNC character. When the $\overline{CTS}$ line goes LOW, the first character is serially transmitted out. All characters are shifted out on the falling edge of $\overline{TxC}$. Data is shifted out at the same rate as the $\overline{TxC}$.

Once transmission has started, the data stream at the TxD output must continue at the $\overline{TxC}$ rate. If the CPU does not provide the 8251A with a data character before the 8251A Transmitter Buffers become empty, the SYNC characters (or character if in single SYNC character mode) will be automatically inserted in the TxD data stream. In this case, the

TxEMPTY pin is raised HIGH to signal that the 8251A is empty and SYNC characters are being sent out. TxEMPTY does not go LOW when the SYNC is being shifted out (see figure below). The TxEMPTY pin is internally reset by a data character being written into the 8251A.



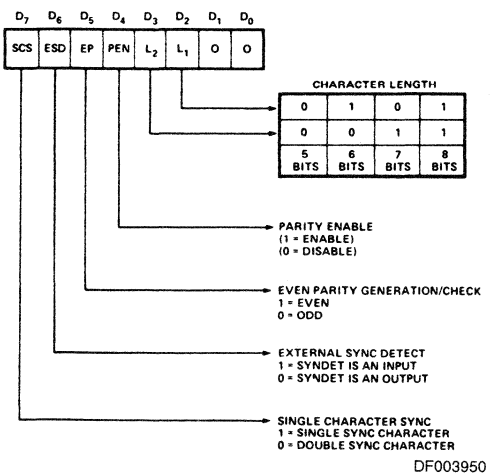
WF008860

Synchronous Mode (Receive)

In this mode, character synchronization can be internally or externally achieved. If the SYNC mode has been programmed, ENTER HUNT command should be included in the first comand instruction word written. Data on the RxD pin is then sampled on the rising edge of $\overline{RxC}$. The content of the Rx buffer is compared at every bit boundary with the first SYNC character until a match occurs. If the 8251A has been programmed for two SYNC characters, the subsequent received character is also compared. When both SYNC characters have been detected, the USART ends the HUNT mode and is in character synchronization. The SYNDET pin is then set HIGH and is reset automatically by a STATUS READ. If parity is programmed, SYNDET will not be set until the middle of the parity bit instead of the middle of the last data bit.

In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET pin, thus forcing the 8251A out of the HUNT mode. The high level can be removed after one $\overline{RxC}$ cycle. An ENTER HUNT command has no effect in the asynchronous mode of operation.

Parity error and overrun error are both checked in the same way as in the Asynchronous Rx mode. Parity is checked when not in Hunt, regardless of whether the Receiver is enabled or not.

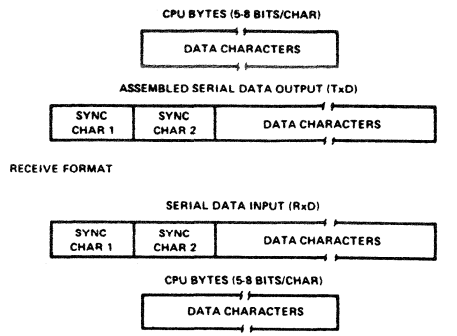


DF003950

Note: IN EXTERNAL SYNC MODE, PROGRAMMING DOUBLE CHARACTER SYNC WILL AFFECT ONLY THE T_X.

Figure 10. Mode Instruction Format, Synchronous Mode

The CPU can command the receiver to enter the HUNT mode if synchronization is lost. This will also set all the used character bits in the buffer to a "one," thus preventing a possible false SYNDET caused by data that happens to be in the Rx Buffer at ENTER HUNT time. Note that the SYNDET F/F is reset at each Status Read, regardless of whether internal or external SYNC has been programmed. This does not cause the 8251A to return to the HUNT mode. When in SYNC mode, but not in HUNT, Sync Detection is still functional, but only occurs at the "known" word boundaries. Thus, if one Status Read indicates SYNDET and a second Status Read also indicates SYNDET, then the programmed SYNDET characters have been received since the previous Status Read. (If double character sync has been programmed, then both sync characters have been contiguously received to gate a SYNDET indication.) When external SYNDET mode is selected, internal Sync Detect is disabled, and the SYNDET F/F may be set at any bit boundary.



DF003960

Figure 11. Data Format, Synchronous Mode

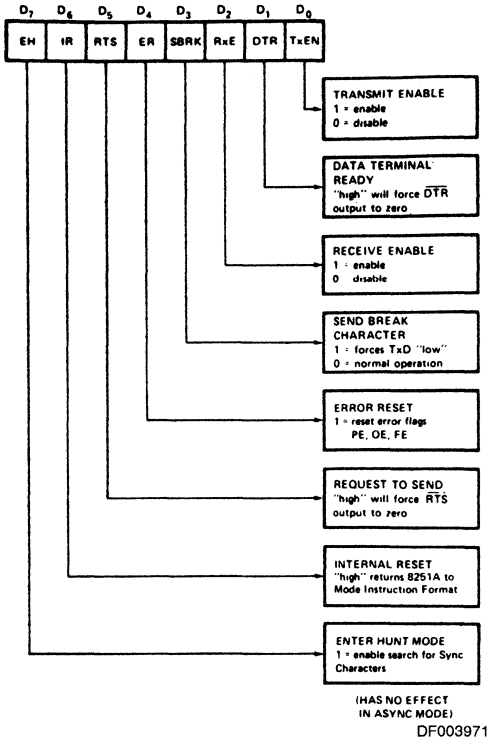
COMMAND INSTRUCTION DEFINITION

Once the functional definition of the 8251A has been programmed by the Mode Instruction and the sync characters are loaded (if in Sync Mode), then the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format. Functions, such as Enable Transmit/Receive, Error Reset and Modem Controls, are provided by the Command Instruction.

Once the Mode Instruction has been written into the 8251A and Sync characters inserted, if necessary, then all further "control writes" ($C/\overline{D} = 1$) will load a Command Instruction. A Reset Operation (internal or external) will return the 8251A to the Mode Instruction format.

Note: Internal Reset on Power-up

When power is first applied, the 8251A may come up in the Mode, Sync character or Command format. To guarantee that the device is in the Command Instruction format before the Reset command is issued, it is safest to execute the worst-case initialization sequence (sync mode with two sync characters). Loading three 00Hs consecutively into the device with $C/\overline{D} = 1$ configures sync operation and writes two dummy 00H sync characters. An Internal Reset command (40H) may then be issued to return the device to the "Idle" state.



Note: Error Reset must be performed whenever RxEnable and Enter Hunt are programmed.

Figure 12. Command Instruction Format

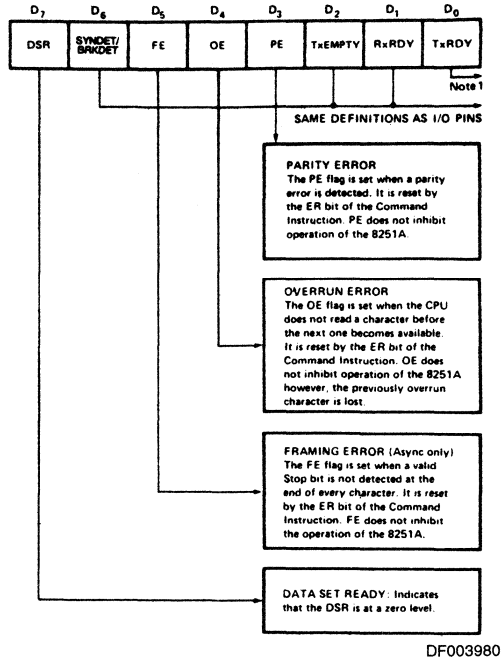
STATUS READ DEFINITION

In data communication systems, it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251A has facilities that allow the programmer to "read" the status of the device at any time during the functional operation. (Status update is inhibited during status read.)

A normal "read" command is issued by the CPU with C/$\overline{D}$ = 1 to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins, so that the 8251A can be used in a completely polled or interrupt-driven environment. TxRDY is an exception.

Note that status update can have a maximum delay of 28 clock periods from the actual event affecting the status.



Note 1: TxRDY status bit has different meanings from the TxRDY output pin. The former is not conditioned by $\overline{CTS}$ and TxEN; the latter is conditioned by both $\overline{CTS}$ and TxEN. i.e. TxRDY status bit = DB Buffer Empty; TxRDY pin out = DB Buffer Empty • ($\overline{CTS}$ 0) • (TxEN = 1)

Figure 13. Status Read Format

APPLICATIONS INFORMATION

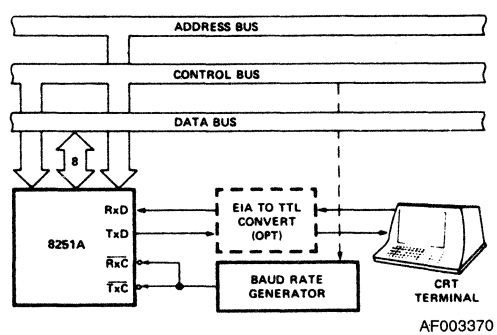


Figure 14. Asynchronous Serial Interface to CRT Terminal, DC – 9600 Baud

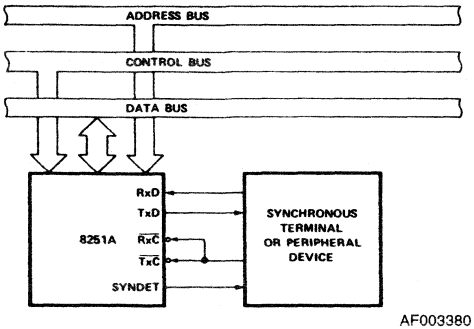


Figure 15. Synchronous Interface to Terminal or Peripheral Device

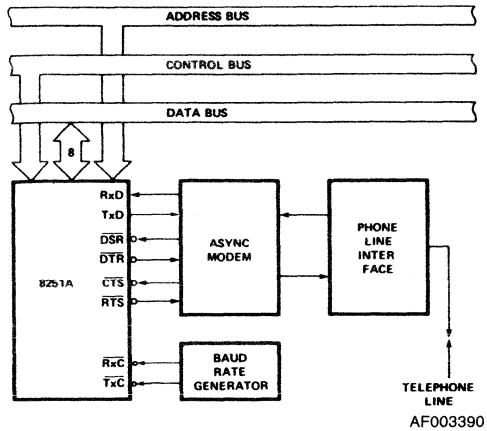


Figure 16. Asynchronous Interface to Telephone Lines

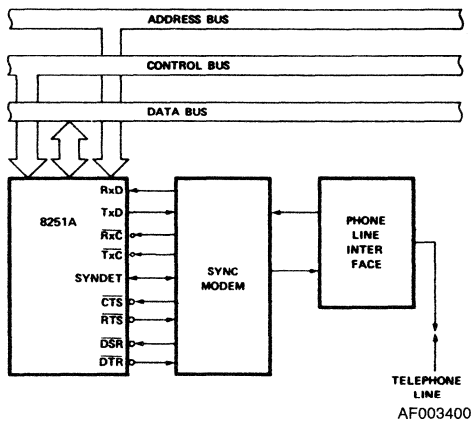


Figure 17. Synchronous Interface to Telephone Lines

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin
with Respect to Ground ..... −0.5 to +7 V
Power Dissipation ..... 1 Watt

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ±10%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ±10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified.

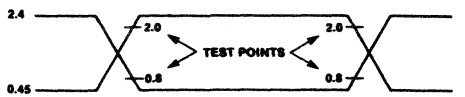
Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	0.8	V
V_{IH}	Input High Voltage		2.0	V_{CC}	V
V_{OL}	Output Low Voltage	I_{OL} = 2.2 mA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400 μA	2.4		V
I_{OFL}	Output Float Leakage	V_{OUT} = V_{CC} to 0.45 V		±10	μA
I_{IL}	Input Leakage	V_{IN} = V_{CC} to 0.45 V		±10	μA
I_{CC}	Power Supply Current	All Outputs = High		100	mA

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0V)

Parameters	Description	Test Conditions	Min	Max	Units
C_{IN}	Input Capacitance	fc = 1 MHz		10	pF
$C_{I/O}$	I/O Capacitance	Unmeasured Pins Returned to GND		20	pF

See Section 6 for Thermal Characteristics Information.

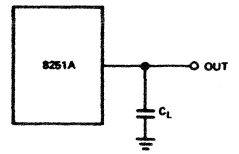
SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF006190

A.C. TESTING: INPUTS ARE DRIVEN AT 2.4 V FOR A LOGIC "1" AND 0.45 V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 2.0 V FOR A LOGIC "1" and 0.8 V FOR A LOGIC "0."

SWITCHING LOAD CIRCUIT



TC002030

C_L = 150 pF

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified.

Bus Parameters (Note 1)

Parameters	Description	Test Conditions	Min	Max	Units
Read Cycle					
t_{AR}	Address Stable Before $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	Note 2	0		ns
t_{RA}	Address Hold Time for $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	Note 2	0		ns
t_{RR}	$\overline{READ}$ Pulse Width		250		ns
t_{RD}	Data Delay from $\overline{READ}$	3, C_L = 150 pF		200	ns
t_{DF}	$\overline{READ}$ to Data Floating		10	100	ns
Write Cycle					
t_{AW}	Address Stable Before $\overline{WRITE}$		0		ns
t_{WA}	Address Hold Time for $\overline{WRITE}$		0		ns
t_{WW}	$\overline{WRITE}$ Pulse Width		250		ns
t_{DW}	Data Setup Time for $\overline{WRITE}$		150		ns
t_{WD}	Data Hold Time for $\overline{WRITE}$		20		ns
t_{RV}	Recovery Time Between WRITES	Note 4	6		t_{CY}
Other Timings					
t_{CY}	Clock Period	Notes 5, 6	320	1350	ns
t_0	Clock High Pulse Width		120	t_{CY}–90	ns
$t_{\overline{0}}$	Clock Low Pulse Width		90		ns
t_f t_r	Clock Rise and Fall Time			20	ns
t_{DTX}	TxD Delay from Falling Edge of TxC			1	μs
t_{Tx}	Transmitter Input Clock Frequency 1x Baud Rate		DC	64	kHz
	16x Baud Rate		DC	310	kHz
	64x Baud Rate		DC	615	kHz
t_{TPW}	Transmitter Input Clock Pulse Width 1x Baud Rate		12		t_{CY}
	16x and 64x Baud Rate		1		t_{CY}
t_{TPD}	Transmitter Input Clock Pulse Delay 1x Baud Rate		15		t_{CY}
	16x and 64x Baud Rate		3		t_{CY}

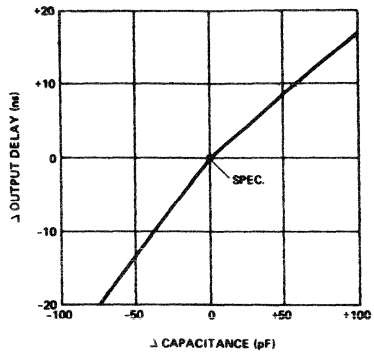
This table is continued on the next page.

SWITCHING CHARACTERISTICS (Cont'd.)

Parameters	Description	Test Conditions	Min	Max	Units
Other Timings (Cont'd.)					
t_{Rx}	Receiver Input Clock Frequency 1x Baud Rate		DC	64	kHz
	16x Baud Rate		DC	310	kHz
	64x Baud Rate		DC	615	kHz
t_{RPW}	Receiver Input Clock Pulse Width 1x Baud Rate		12		t_{CY}
	16x and 64x Baud Rate		1		t_{CY}
t_{RPD}	Receiver Input Clock Pulse Delay 1x Baud Rate		15		t_{CY}
	16x and 64x Baud Rate		3		t_{CY}
t_{TxRDY}	TxRDY Pin Delay from Center of Last Bit	Note 7		14	t_{CY}
t_{TxRDY} CLEAR	TxRDY ↓ from Leading Edge of $\overline{WR}$	Note 7		400	ns
t_{RxRDY}	RxRDY Pin Delay from Center of Last Bit	Note 7		26	t_{CY}
t_{RxRDY} CLEAR	RxRDY ↓ from Leading Edge of $\overline{RD}$	Note 7		400	ns
t_{IS}	Internal SYNDET Delay from Rising Edge of $\overline{RxC}$	Note 7		26	t_{CY}
t_{ES}	External SYNDET Set-up Time After Rising Edge of $\overline{RxC}$	Note 7	16 t_{CY}	$t_{RPD}-t_{CY}$	ns
$t_{TxEMPTY}$	TxEMPTY Delay from Center of Last Bit	Note 7		20	t_{CY}
t_{WC}	Control Delay from Rising Edge of WRITE (TxEn, $\overline{DTR}$, $\overline{RTS}$)	Note 7		8	t_{CY}
t_{CR}	Control to READ Set-up Time ($\overline{DSR}$, $\overline{CTS}$)	Note 7	20		t_{CY}

Notes:
1. AC timings measured $V_{OH} = 2.0$, $V_{OL} = 0.8$, and with switching load circuit.
2. Chip Select ($\overline{CS}$) and Command/Data ($C/\overline{D}$) are considered as Addresses.
3. Assumes that Address is valid before $R_D\downarrow$.
4. This recovery time is for after a Mode Instruction only. Write Data is allowed only when TxRDY = 1. Recovery time between Writes for Asynchronous Mode is 8 t_{CY} and for Synchronous Mode is 16 t_{CY}.
5. The TxC and RxC frequencies have the following limitations with respect to CLK: for 1x Baud Rate, f_{Tx} or $f_{Rx} \leqslant 1/(30\ t_{CY})$: for 16x and 64x Baud Rate, f_{Tx} or $f_{Rx} \leqslant 1/(4.5\ t_{CY})$.
6. Reset Pulse Width = 6 t_{CY} minimum; System Clock must be running during Reset.
7. Status update can have a maximum delay of 28 clock periods from the event affecting the status.

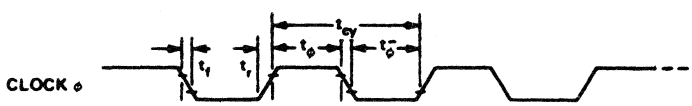
TYPICAL Δ OUTPUT DELAY VS. Δ CAPACITANCE (pF)



OP006180

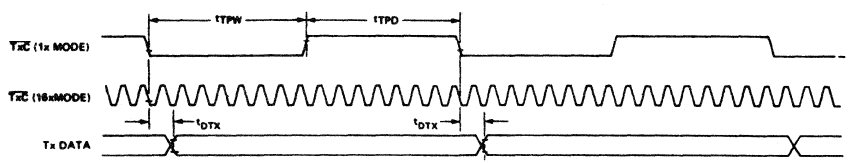
SWITCHING WAVEFORMS

SYSTEM CLOCK INPUT



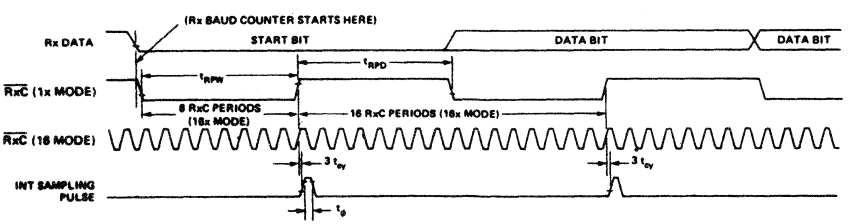
WF008150

TRANSMITTER CLOCK AND DATA



WF006200

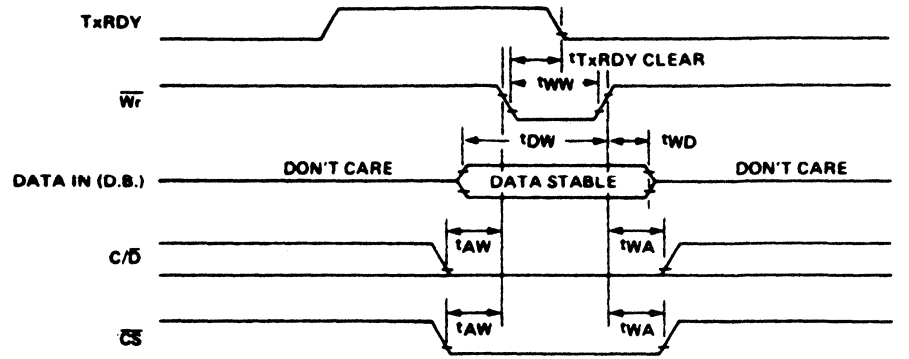
RECEIVER CLOCK AND DATA



WF008140

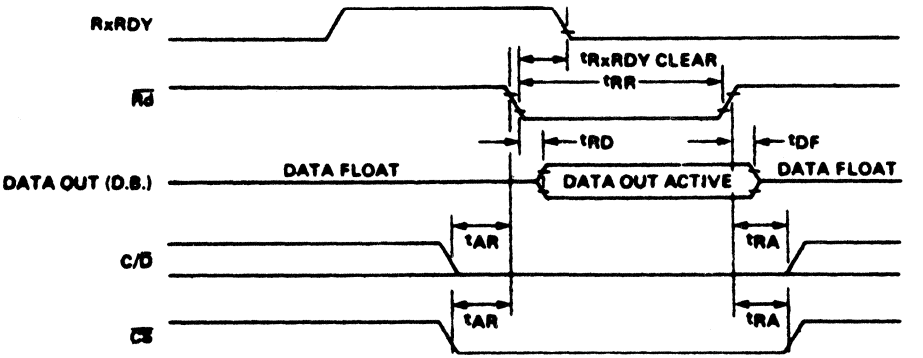
SWITCHING WAVEFORMS (Cont'd.)

WRITE DATA CYCLE (CPU → USART)



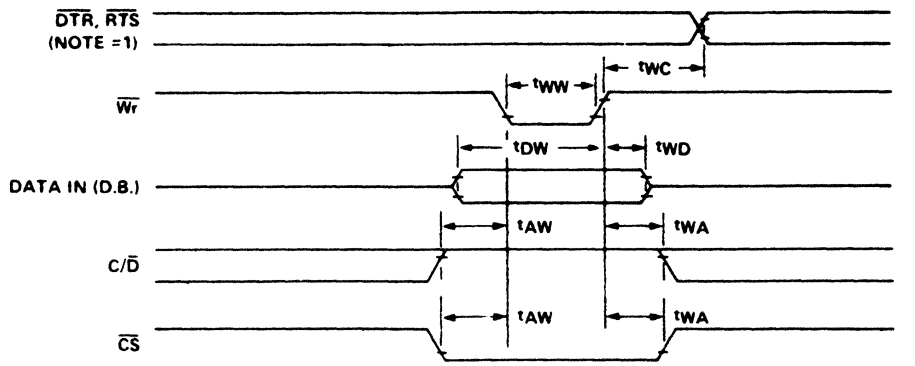
WF006140

READ DATA CYCLE (CPU ← USART)



WF006150

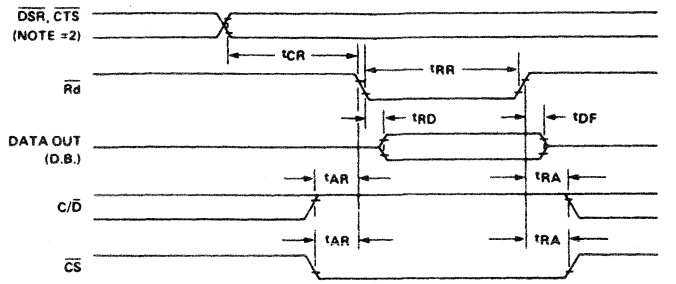
WRITE CONTROL OR OUTPUT PORT CYCLE (CPU → USART)



WF006160

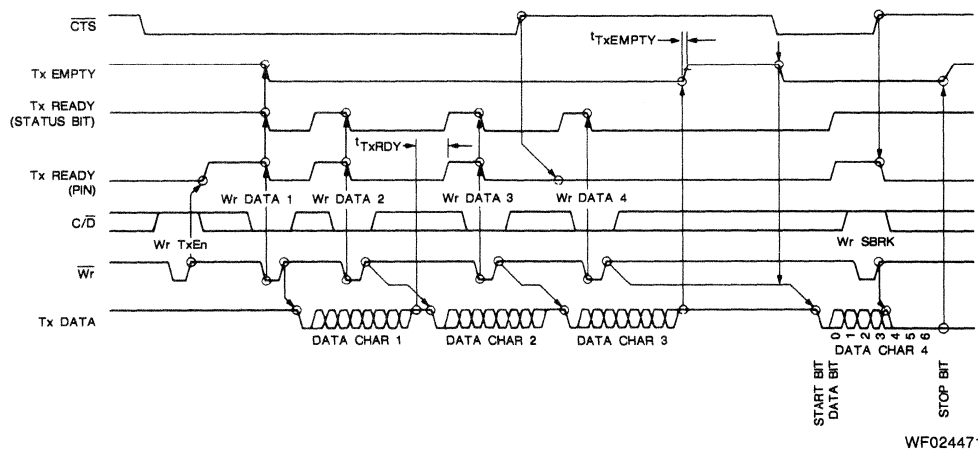
SWITCHING WAVEFORMS (Cont'd.)

READ CONTROL OR INPUT PORT (CPU ← USART)



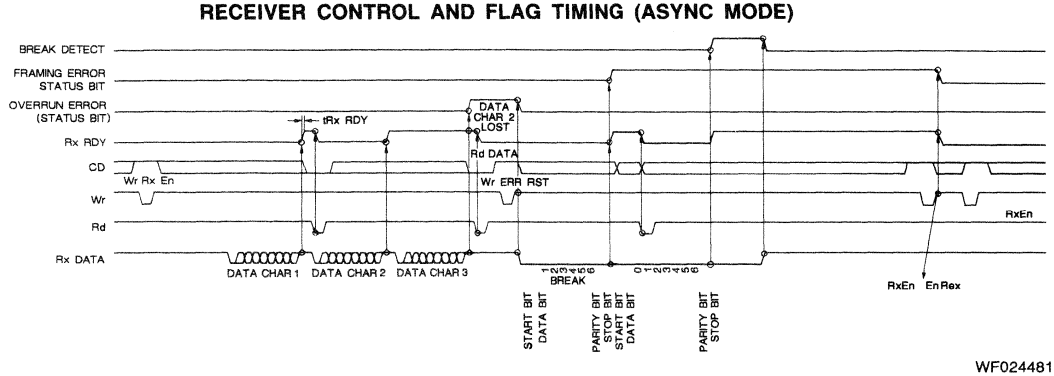
Notes: 1. T_{WC} INCLUDES THE RESPONSE TIMING OF A CONTROL BYTE.
2. T_{CR} INCLUDES THE EFFECT OF CTS ON THE TxENBL CIRCUITY.

TRANSMITTER CONTROL AND FLAG TIMING (ASYNC MODE)



EXAMPLE FORMAT = 7 BIT CHARACTER WITH PARITY & 2 STOP BITS.

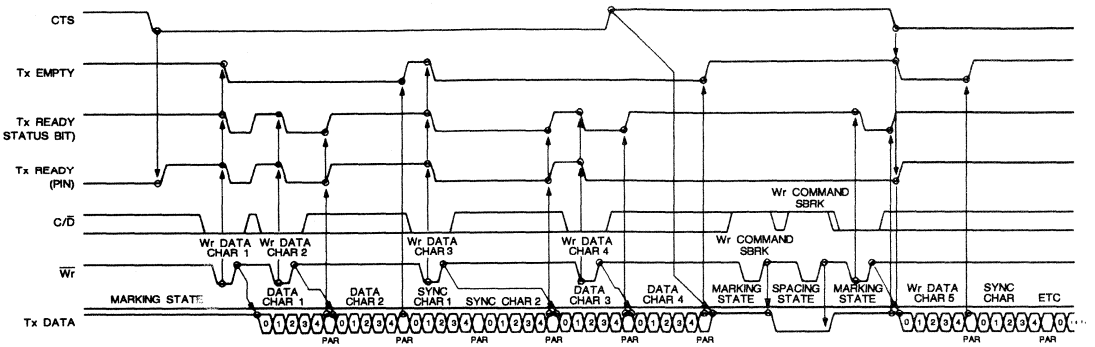
RECEIVER CONTROL AND FLAG TIMING (ASYNC MODE)



EXAMPLE FORMAT = 7 BIT CHARACTER WITH PARITY 2 & STOP BITS.

SWITCHING WAVEFORMS (Cont'd.)

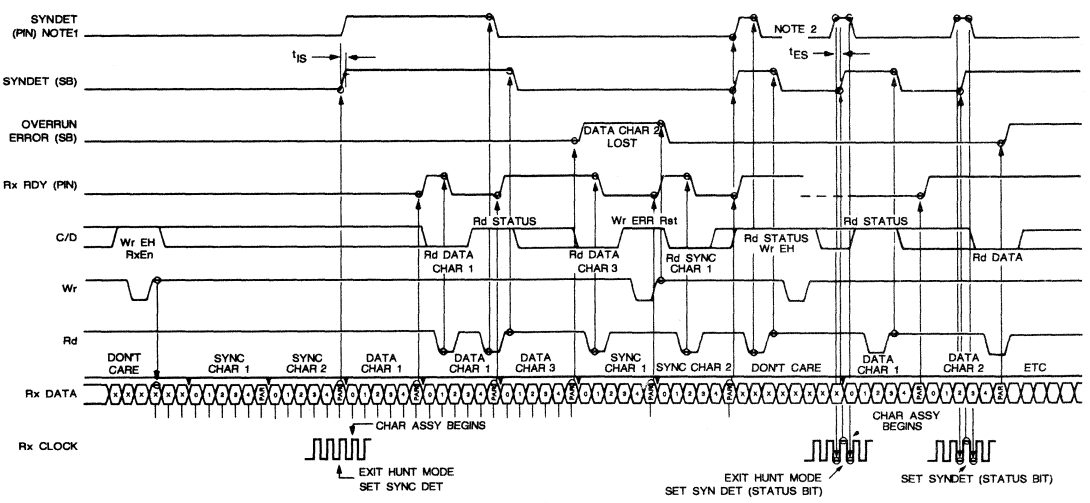
TRANSMITTER CONTROL AND FLAG TIMING (SYNC MODE)



WF024491

EXAMPLE FORMAT = 5 BIT CHARACTER WITH PARITY 2 SYNC CHARACTERS.

RECEIVER CONTROL AND FLAG TIMING (SYNC MODE)



WF024501

Notes: 1. INTERNAL SYNC. 2 SYNC CHARACTERS. 5 BITS WITH PARITY
2. EXTERNAL SYNC. 5 BITS WITH PARITY

8253

Programmable Interval Timer
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

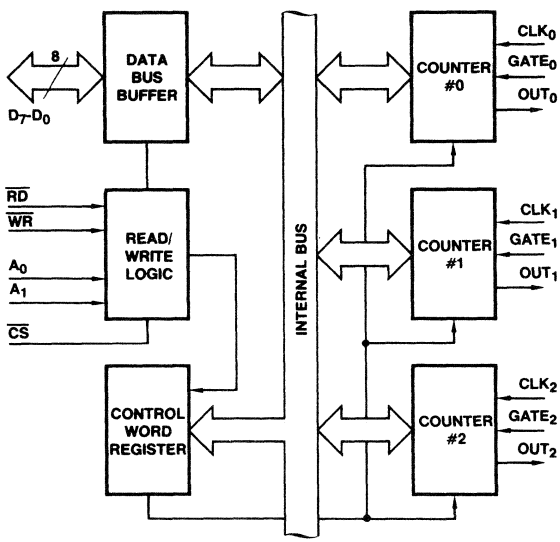
- Both Binary and BCD counting
- Single +5V supply
- Three independent 16-bit counters
- DC to 5MHz
- Programmable counter modes
- Bus oriented I/O

GENERAL DESCRIPTION

The 8253 is a programmable counter/timer chip designed for use with 8080A/8085A microprocessors. It uses NMOS technology with a singie +5V supply and is a direct replacement for Intel's 8253/8253-5.

Each device is organized as three independent 16-bit counters, each counter having a rate of up to 5MHz. All modes of operation are software programmable. For improved performance devices see the Am9513A System Timing Controller.

BLOCK DIAGRAM

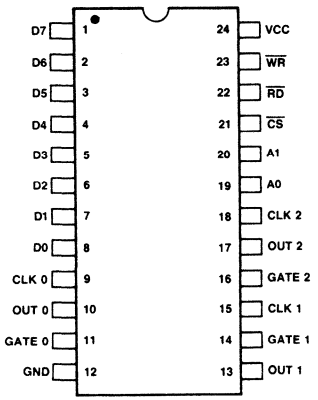


BD003760

Publication # 04126 | Rev. B | Amendment /0
Issue Date: May 1987

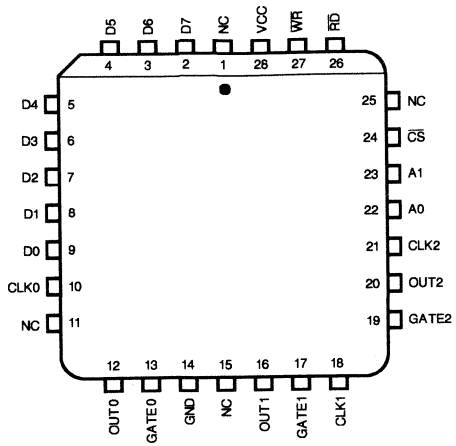
CONNECTION DIAGRAMS
Top View

DIPs



CD005672

PLCC



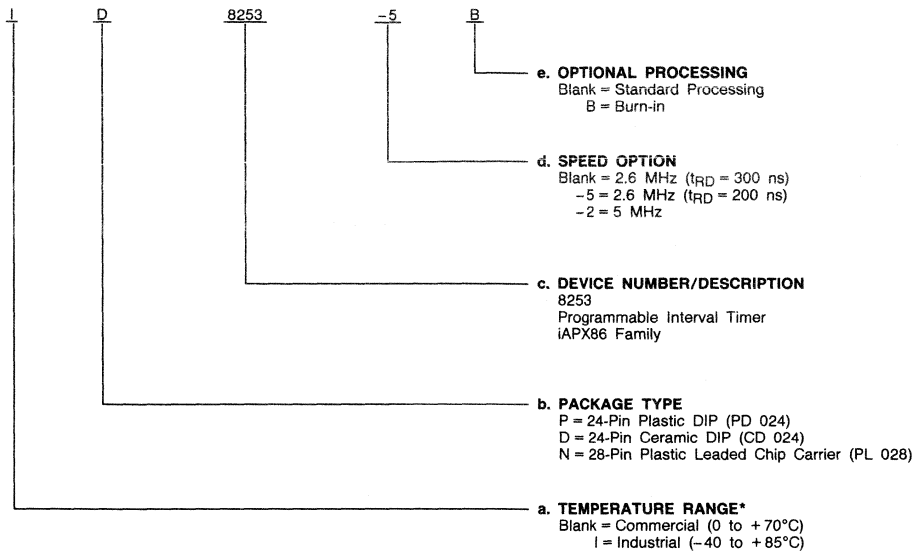
CD010071

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D, N	8253
	8253-5
	8253-2
D, ID	8253B
	8253-5B
D	8253-2B

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Handbook (Order #09275A/0) for electrical performance characteristics.

PIN DESCRIPTION

Pin No.*	Name	I/O	Description
1-8	D7-D0	I/O	Data bus (8-bit).
9, 15, 18	CLK N	I	Counter clock inputs.
11, 14, 16	GATE N	I	Counter gate inputs.
10, 13, 17	OUT N	O	Counter outputs.
22	$\overline{RD}$	I	Read counter.
23	$\overline{WR}$	I	Write command or data.
21	$\overline{CS}$	I	Chip select.
19, 20	A0-A1	I	Counter select.
24	VCC		+5 Volts.
12	GND		Ground.

* Pin numbers correspond to DIPs only.

DETAILED DESCRIPTION

General

The 8253 is a programmable interval timer/counter specifically designed for use with 8080A Microcomputer systems. Its function is that of a general-purpose, multitiming element that can be treated as an array of I/O ports in the system's 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 the system's software, the programmer configures the 8253 to match his requirements and initializes one of the counters of the 8253 with a desired quantity. Then upon command the 8253 will count out the delay and interrupt the CPU when it has completed its task. 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

This 3-state, bidirectional, 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, and
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 Read or Write operation can occur 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 counter's value.

$\overline{WR}$ (Write)

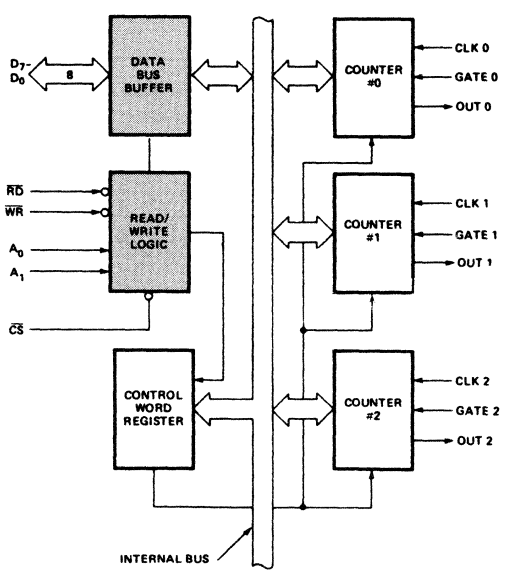
A "LOW" on this input informs the 8253 that the CPU is outputting data in the form of MODE information or loading counters.

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.



BD005101

Figure 1. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

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 operational MODE of each counter, the 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, presettable, 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 a different 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.

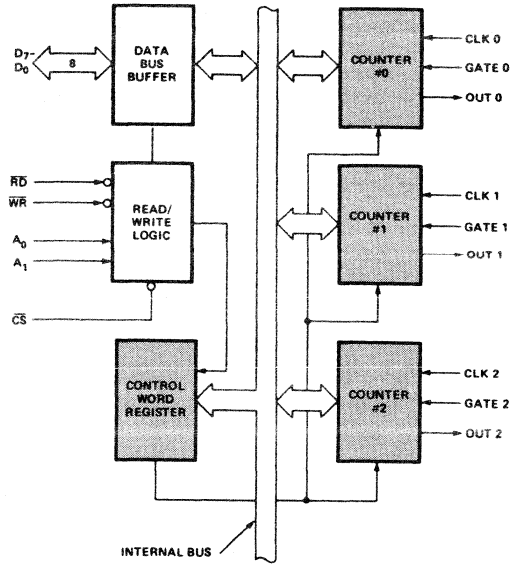
The reading of the contents of each counter is available to the programmer with simple READ operations for event counting applications, and special commands and logic are included in the 8253 so that the contents of each counter can be read "on the fly" without having to inhibit the clock input.

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A1	A0	
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

8253 SYSTEM INTERFACE

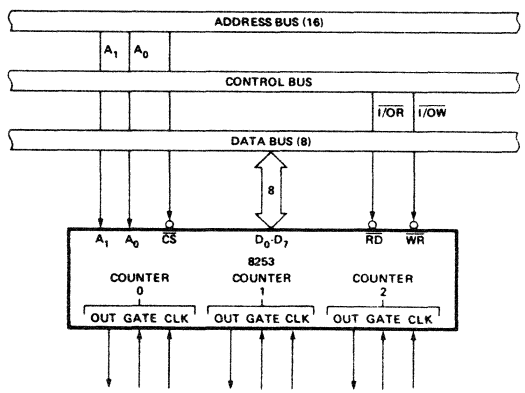
The 8253 is a component of the iAPX Family and interfaces in the same manner as all other peripherals of the family. It is treated by the system's software as an array of peripheral I/O ports; three are counters, and the fourth is a control register for MODE programming.

Basically, the select inputs 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 AMD Am25LS2548 or Am25LS2538 for larger systems.



BD005101

Figure 2. Block Diagram Showing Control Word Register and Counter Functions



BD005120

Figure 3. 8253 System Interface

8253 READ/WRITE PROCEDURE

Write Operations

The system's 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 zeroes into a count register will result in the maximum count (2^{16} for Binary or 10^4 for BCD). In MODE 0 a new count will not start until the load has been completed. The count register will accept one or two bytes depending on how the MODE control words (RL0, RL1) are programmed.

Programming Format

MODE Control Word
Counter n

LSB	Count Register byte Counter n
MSB	Count 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.

Alternate Programming Formats

			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 utilized.

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 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

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 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 are available.

MODE Register for Latching Count

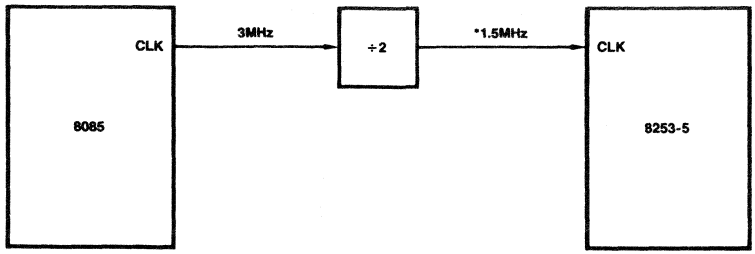
A0, A1 = 11

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	0	0	X	X	X	X

SCl,SCO – 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.

Clock Interface*



*If an 8085 clock output is to drive an 8253-5 clock input, it must be reduced to 2MHz or less.

PROGRAMMING INFORMATION

General

The complete functional definition of the 8253 is programmed by the system's 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. 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.

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 system's 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

D7	D6	D5	D4	D3	D2	D1	D0
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.
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 Code 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 (for 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 one. Subsequent clock pulses decrement the clock by two. After timeout, the output goes LOW and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by three. Subsequent clock pulses decrement the count by two until time-out. 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.

MODE 4: Software-Triggered Strobe

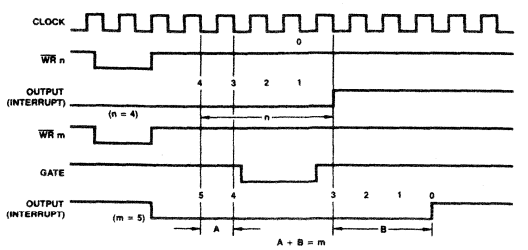
After the mode is set, the output will be HIGH. When the count is loaded, the counter will begin counting. On terminal count, the output will go LOW for one input clock period, then will go HIGH again.

If the count register is reloaded between output pulses, the present period will not be affected, but the subsequent period will reflect the new value. The count will be inhibited while the gate input is LOW. Reloading the counter register will restart counting beginning with the new number.

MODE 5: Hardware-Triggered Strobe

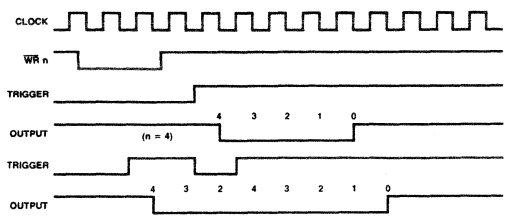
The counter will start counting after the rising edge of the trigger input and will go LOW for one clock period when the terminal count is reached. The counter is retriggerable. The output will not go LOW until the full count after the rising edge of any trigger.

MODE 0. Interrupt on Terminal Count.



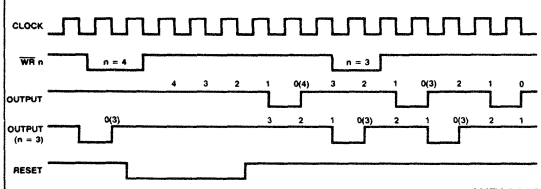
WF006860

MODE 1. Programmable One-Shot.



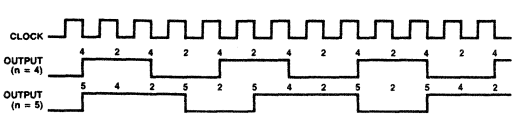
WF006870

MODE 2. Rate Generator.



WF006880

MODE 3. Square Wave Generator.

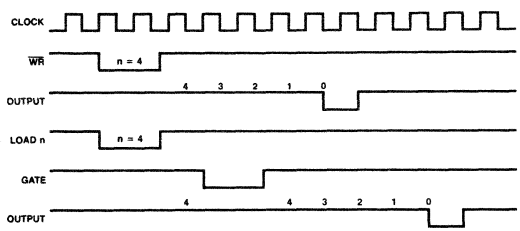


WF006890

Gate Pin Operations Summary

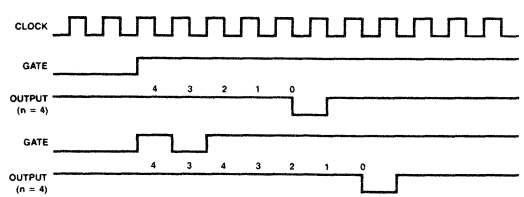
Modes	Signal Status		
	Low Or Going Low	Rising	High
0	Disables counting	–	Enables counting
1	–	1) Initiates counting 2) Resets output after next clock	–
2	1) Disables counting 2) Sets output immediately high	1) Reloads counter 2) Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	–	Enables counting
5	–	Initiates counting	–

MODE 4. Software-Triggered Strobe.



WF006900

MODE 5. Hardware-Triggered Strobe.



WF006910

8253 Timing Diagrams

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage On Any Pin
with Respect to Ground ..... −0.5 to +7.0V
Power Dissipation 1W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5V ± 10%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5V ± 10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

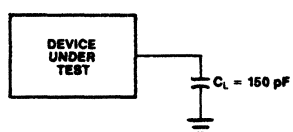
DC CHARACTERISTICS over operating ranges unless otherwise specified.

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−.5	.8	V
V_{IH}	Input High Voltage		2.2	V_{CC} + .5V	V
V_{OL}	Output Low Voltage	I_{OL} = 2.2 mA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400 µA	2.4		V
I_{IL}	Input Load Current	V_{IN} = V_{CC} to 0V		±10	µA
I_{OFL}	Output Float Leakage	V_{OUT} = V_{CC} to 0V		±10	µA
I_{CC}	V_{CC} Supply Current			140	mA

CAPACITANCE T_A = 25°C; V_{CC} = GND = 0V

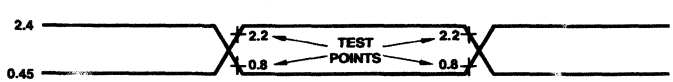
Parameters	Description	Test Conditions	Min	Typ	Max	Units
C_{IN}	Input Capacitance	f_c = 1MHz			10	pF
$C_{I/O}$	I/O Capacitance	Unmeasured pins returned to V_{SS}			20	pF

SWITCHING TEST CIRCUIT



TC002192

SWITCHING TEST INPUT WAVEFORM



WF006951

AC Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at 2.2V for a logic "1" and 0.8V for a logic "0".

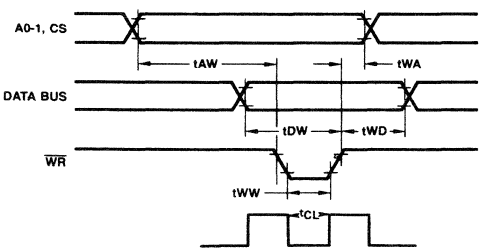
See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified.

Parameters	Description	8253		8253-5		8253-2		Units
		Min	Max	Min	Max	Min	Max	
Read Cycle								
t_{AR}	Address Stable Before $\overline{READ}$	50		30		25		ns
t_{RA}	Address Hold Time for $\overline{READ}$	5		5		5		ns
t_{RR}	$\overline{READ}$ Pulse Width	400		300		150		ns
t_{RD}	Data Delay from $\overline{READ}$ (Note 2)		300		200		120	ns
t_{DF}	$\overline{READ}$ to Data Floating	25	125	25	100	25	100	ns
t_{RV}	Recovery Time Between $\overline{READ}$ and Any Other Control Signal	1		1		0.5		μs
Write Cycle								
t_{AW}	Address Stable Before $\overline{WRITE}$	50		30		0		ns
t_{WA}	Address Hold Time for $\overline{WRITE}$	30		30		0		ns
t_{WW}	$\overline{WRITE}$ Pulse Width	400		300		150		ns
t_{DW}	Data Set-up Time for $\overline{WRITE}$	300		250		100		ns
t_{WD}	Data Hold Time for $\overline{WRITE}$	40		30		0		ns
t_{RV}	Recovery Time Between $\overline{WRITE}$ and Any Other Control Signal (Note 3)	1		1		0.5		μs

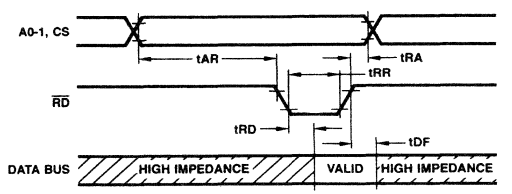
Notes: 1. AC timings measured at $V_{OH} = 2.2$, $V_{OL} = 0.8$.

Write Timing



WF006931

Read Timing

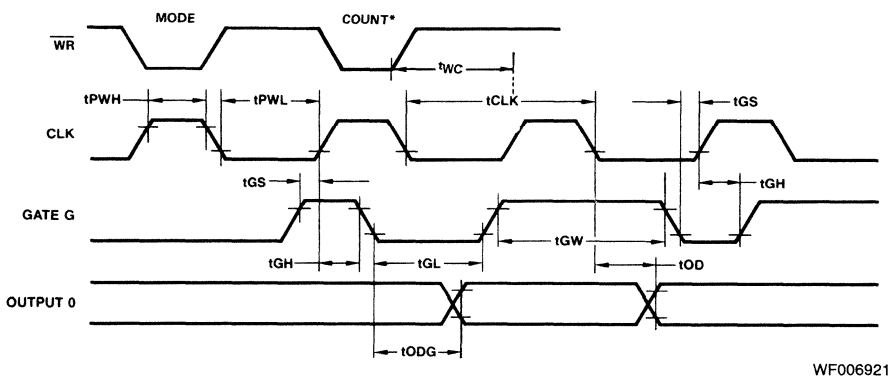


WF006940

CLOCK AND GATE TIMING

Parameters	Description	8253		8253-5		8253-2		Units
		Min	Max	Min	Max	Min	Max	
t_{CLK}	Clock Period	380	DC	380	DC	200	DC	ns
t_{PWH}	High Pulse Width	230		230		90		ns
t_{PWL}	Low Pulse Width	150		150		90		ns
t_{GW}	Gate Width High	150		150		120		ns
t_{GL}	Gate Width Low	100		100		80		ns
t_{GS}	Gate Set-up Time to CLK↑	100		100		60		ns
t_{GH}	Gate Hold Time After CLK↑	50		50		50		ns
t_{OD}	Output Delay from CLK↓ (Note 1)		400		400		250	ns
t_{ODG}	Output Delay from Gate↓ (Note 1)		300		300		150	ns

CLOCK AND GATE TIMING



*last byte of count being written

82C54

CMOS Programmable Interval Timer

DISTINCTIVE CHARACTERISTICS

- Compatible with all Intel and most other microprocessors
- High-speed, zero-wait-state operation with 10-MHz 8086/88 and 80186/188
- Three independent 16-bit counters
- Handles inputs from DC to 8 MHz
  - 10 MHz for 82C54-2
  - 12.5 MHz for 82C54-12
- Low-power CMOS
  - I_{CC} = 10 μA commercial standby current I_{CC}
- Completely TTL compatible
- Six programmable counter modes
- Binary or BCD counting
- Status read-back command
- Available in 24-pin DIP and 28-lead PLCC

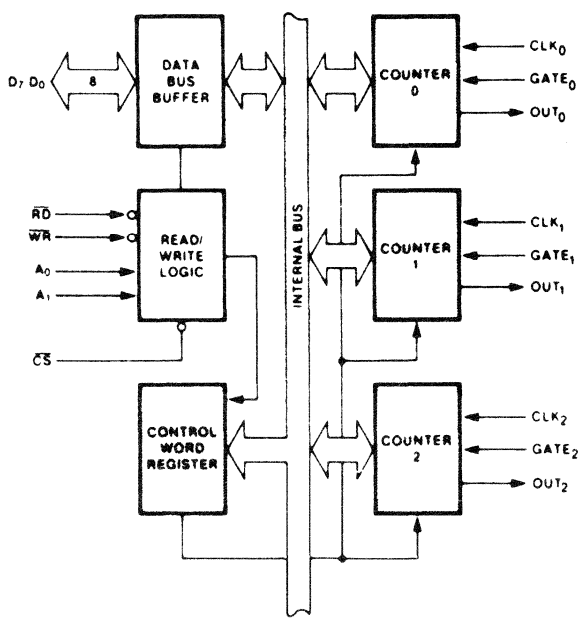
GENERAL DESCRIPTION

The AMD 82C54 is a high-performance, CMOS 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 12.5 MHz. All modes are software-programmable. The 82C54 is pin-compatible with the NMOS 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 as well.

The 82C54 is fabricated with AMD's CMOS technology providing low-power consumption with performance equal to or greater than the equivalent NMOS product. The 82C54 is available in 24-pin DIPs (plastic and ceramic) and 28-pin plastic leaded chip carrier (PLCC) packages.

BLOCK DIAGRAM

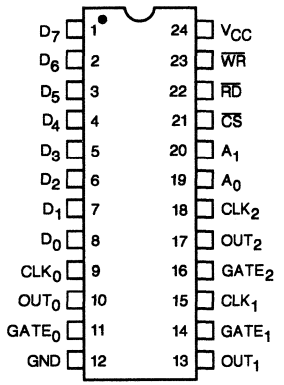


BD006111

Publication # 07840 | Rev. B | Amendment /0
Issue Date: April 1987

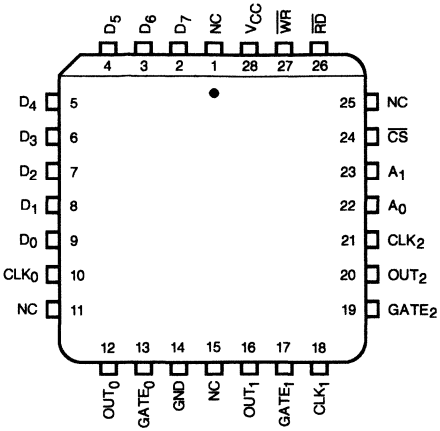
CONNECTION DIAGRAMS
Top View

DIPs



CD009390

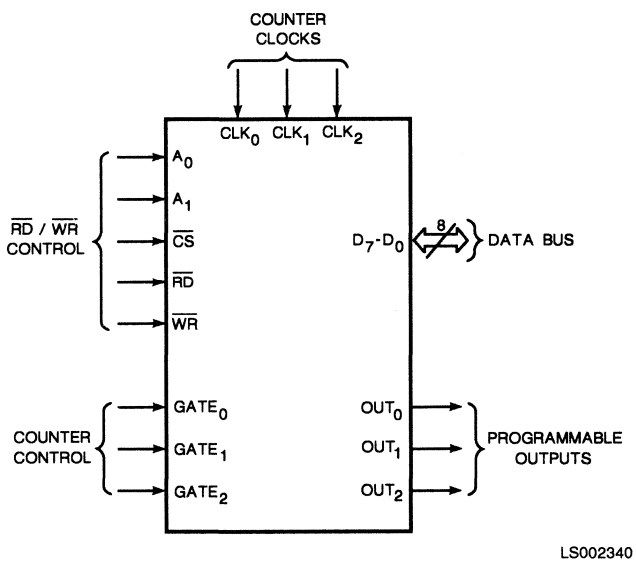
PLCC



CD009401

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL

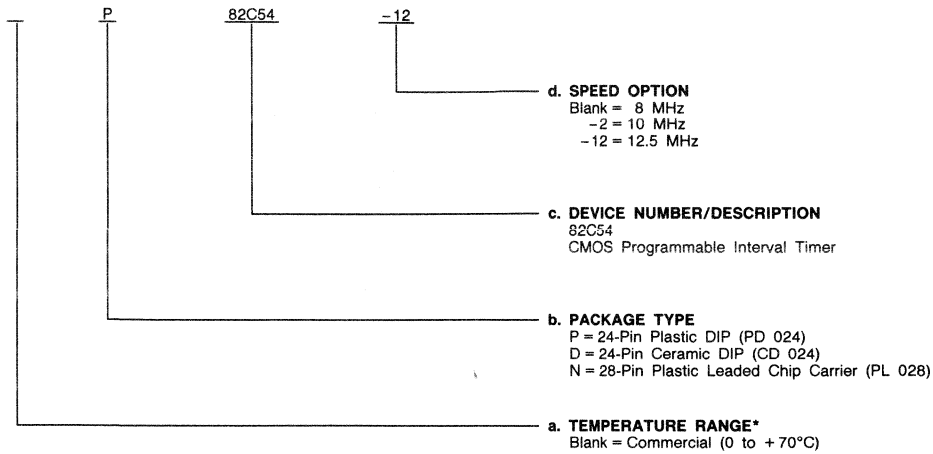


LS002340

V_{CC} = Power Supply
GND = Ground

ORDERING INFORMATION

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option



Valid Combinations	
P, D, N	82C54
	82C54-2
	82C54-12

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (order # 09275A/0) for electrical performance characteristics.

PIN DESCRIPTION

$D_7 - D_0$ Data Bus Lines (Bidirectional, Three-state)
Connected to system data bus.

CLK_0 Clock 0 (Input)
Clock input of Counter 0.

OUT_0 Out 0 (Output)
Output of Counter 0.

$GATE_0$ Gate 0 (Input)
Gate input of Counter 0.

CLK_1 Clock 1 (Input)
Clock input of Counter 1.

OUT_1 Out 1 (Output)
Output of Counter 1.

$GATE_1$ Gate 1 (Input)
Gate input of Counter 1.

CLK_2 Clock 2 (Input)
Clock input of Counter 2.

OUT_2 Out 2 (Output)
Output of Counter 2.

$GATE_2$ Gate 2 (Input)
Gate input of Counter 2.

A_1, A_0 Addresses (Input)
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

$\overline{CS}$ Chip Select (Input, Active LOW)
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.

$\overline{RD}$ Read Control (Input, Active LOW)
This input is LOW during CPU read operations.

$\overline{WR}$ Write Control (Input, Active LOW)
This input is LOW during CPU write operations.

V_{CC} +5-Volt Power Supply

GND Ground

NC No Connect

FUNCTIONAL DESCRIPTION

General

The 82C54 is a programmable interval counter/timer designed for use with AMD 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.

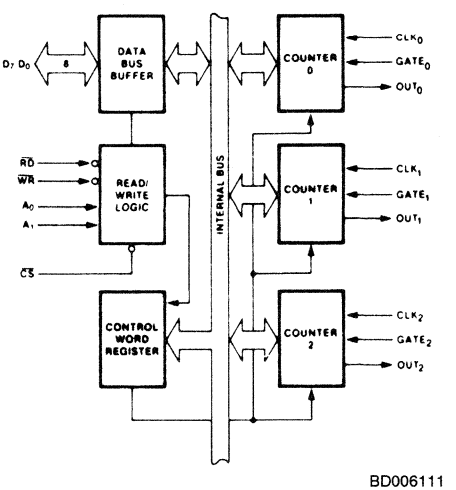
The 82C54 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in software, the programmer configures the 82C54 to match his/her requirements and programs one of the counters for the desired delay; after which 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
- Event counter
- Digital one-shot
- Programmable-rate generator
- Square-wave generator
- Binary-rate multiplier
- Complex-waveform generator
- Complex-motor controller

Data Bus Buffer

This three-state, bidirectional 8-bit buffer is used to interface the 82C54 to the system bus (see Figure 1).



BD006111

Figure 1. 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 2) 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.

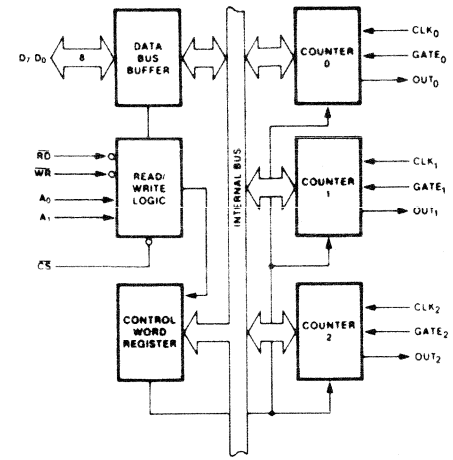


Figure 2. Block Diagram Showing Control Word and Register Counter Functions

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 3.

The Counters are fully independent; each may operate in a different mode.

The Control Word Register is shown in the figure; it is not part of the Counter, but its contents determine how the Counter operates.

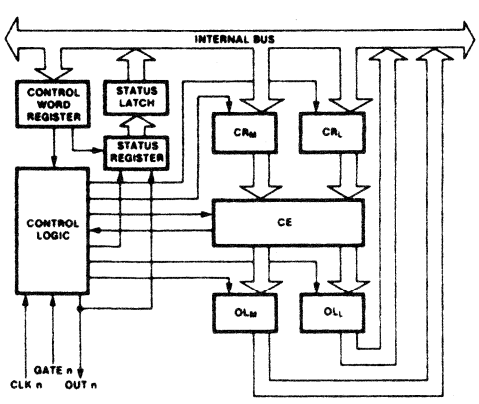


Figure 3. Internal Block Diagram of a Counter

The Status Register — shown in Figure 3 — 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 labeled 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 signficant 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 (see Figure 4).

Basically, the select inputs, A_0 and 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 AMD 8205, for larger systems.

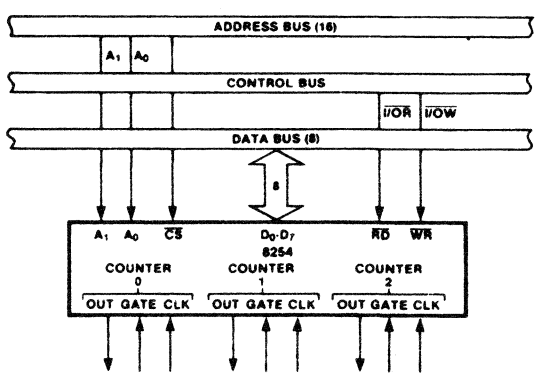


Figure 4. 82C54 System Interface

Operational Description

General

After power-up, the state of the 82C54 is undefined. The mode, count values, 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 5.

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 and 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)

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 compatiblity with future AMD products.

Figure 5. 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

section. 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 6. 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: 1) a simple read operation, 2) the Counter-Latch Command, or 3) the Read-Back Command.

The first method is to perform a simple read operation. To read the Counter, which is selected with the A_1, A_0 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, D_5 and D_4, 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

D_5, D_4 – 00 designates Counter-Latch Command
X – Don't Care

NOTE: Don't care bits (X) should be 0 to insure compatibility with future AMD products.

Figure 7. 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 re-programmed). The count is then unlatched automatically and the OL return 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, 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 may 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 8. The command applies to the Counters selected by setting their corresponding bits D_3, D_2, $D_1 = 1$.

A_0, $A_1 = 11$ $\overline{CS} = 0$ $\overline{RD} = 1$ $\overline{WR} = 0$

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 8. Read-Back Command Format

The Read-Back Command may be used to latch multiple Counter Output Latches (OL) by setting the Count bit $D_5 = 0$ 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. In other words, 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 Status bit $D_4 = 0$. 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 9. Bits D_5 through D_0 contain the Counter's programmed mode exactly as written in the last mode Control Word. Output bit D_7 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 5)

Figure 9. Status Byte

Null-Count bit D_6 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 section, but until the count is loaded into the CE, it cannot 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 10.

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 $\longrightarrow$ 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 10. Null Count Operation

If multiple Status-Latch operations of the Counter(s) are performed without reading the status, all but the first are ignored. In other words, 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 Count and Status bits D_5, $D_4 = 0$. 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 Counters) without any intervening reads, all but the first are ignored. This is illustrated in Figure 11.

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 11. 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 12. 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 section).

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 CLK 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.

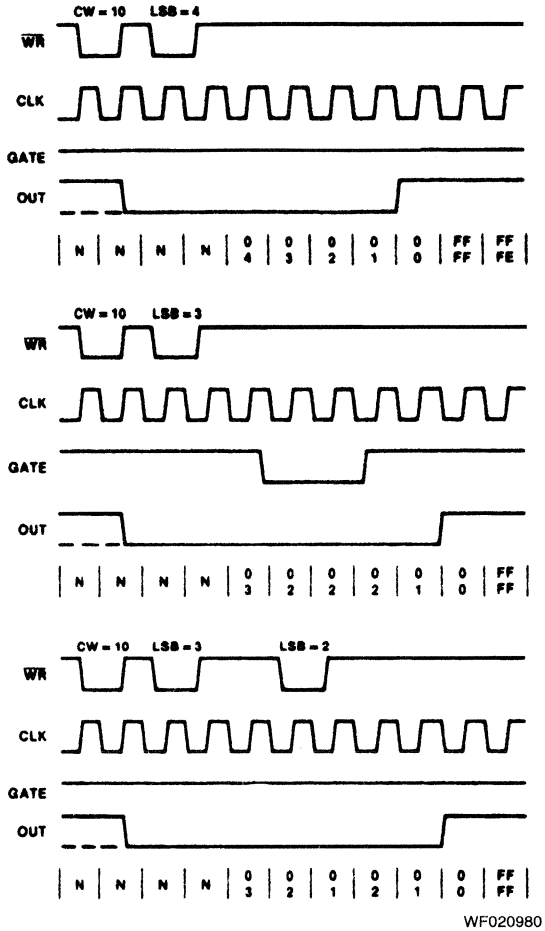


Figure 13. Mode 0

Notes: The following conventions apply to all Mode-Timing Diagrams:

1. Counters are being 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 LSB; the upper number is the MSB. Since the Counter is programmed to read/write LSB only, the MSB cannot be read. "N" stands for an undefined count; vertical lines show transitions between count values.

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

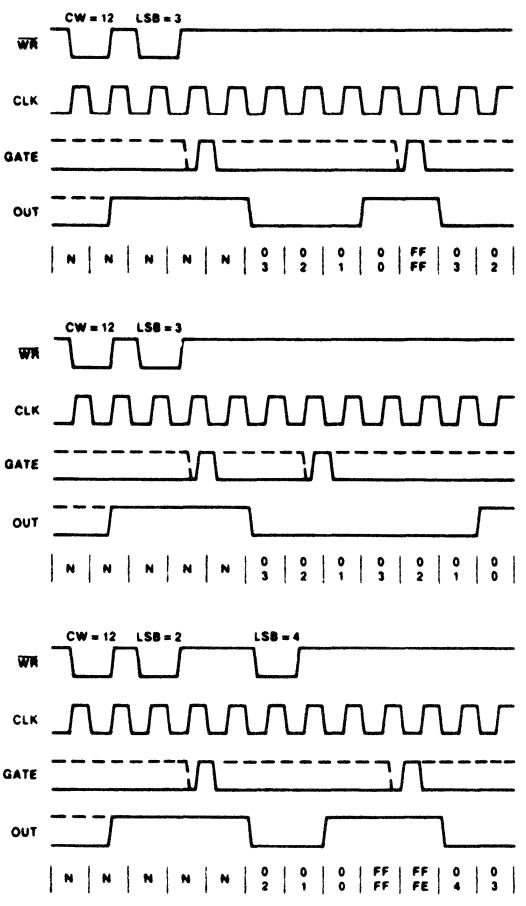
This mode functions like a divide-by-N Counter. It is typically used to generate a real-time clock interrupt. Out will initially be

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 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.

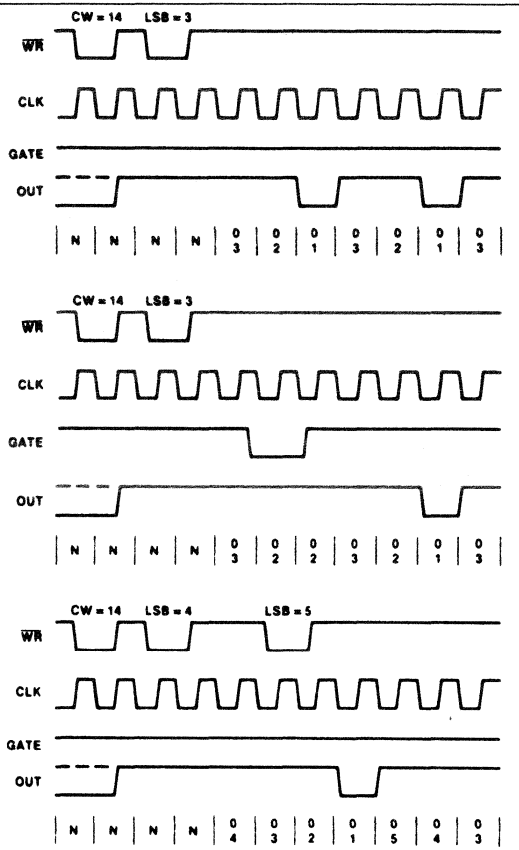
After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. Out goes LOW N CLK pulses after the initial count is written. This allows the Counter to be synchronized by software also.

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.



WF020990

Figure 14. Mode 1



WF021000

NOTE: A Gate transition should not occur one clock prior to terminal count.

Figure 15. Mode 2

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.

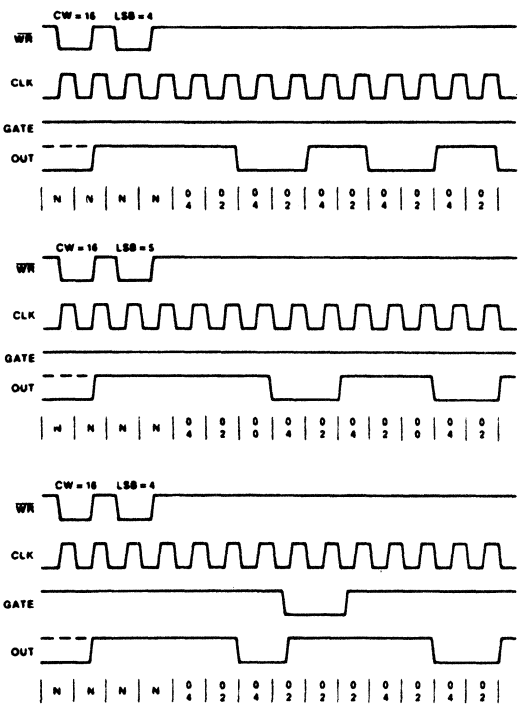
After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This allows the Counter to be synchronized by software also.

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. For odd counts, Out will be HIGH for (N + 1)/2 counts and LOW for (N – 1)/2 counts.



WF021010

NOTE: A Gate transition should not occur one clock prior to terminal count.

Figure 16. 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.

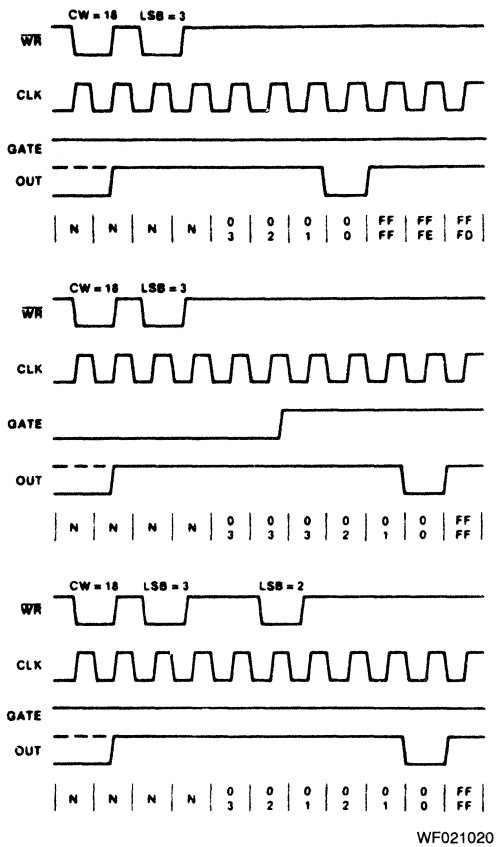
After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, Out does not strobe LOW until N + 1 CLK pulses after the initial count is written.

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.



WF021020

Figure 17. Mode 4

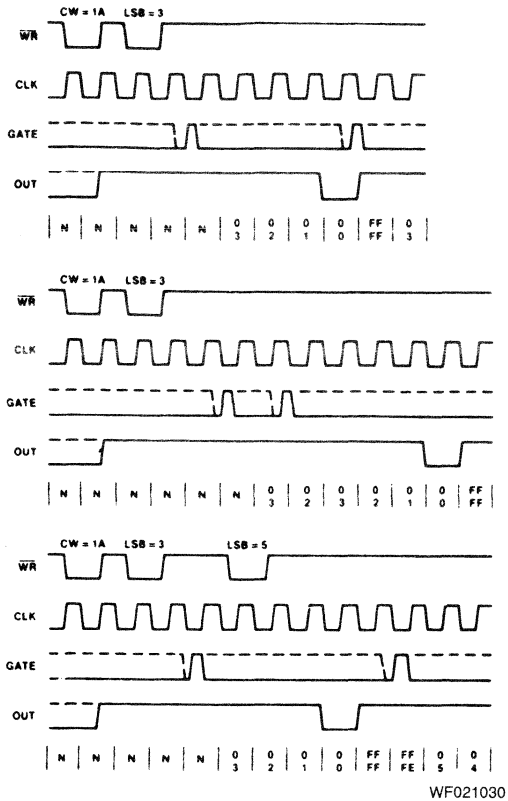
Mode 5: Hardware-Triggered Strobe (Retriggerable)

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.



WF021030

Figure 18. 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, 3	1) Disables counting 2) Sets output immediately HIGH	Initiates counting	Enables counting
4	Disables counting	—	Enables counting
5	—	Initiates counting	—

Figure 19. 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 20. 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 to 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

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin
with Respect to GND ..... −0.5 to +7.0 V
Power Dissipation ..... 1 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.5 to +5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

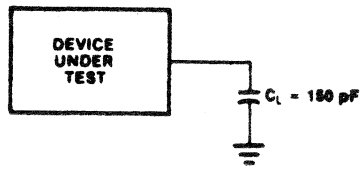
Parameter Symbol	Parameter Description	Test Conditions			Min.	Max.	Units
V_{IL}	Input LOW Voltage			C Devices	−0.5	0.8	V
V_{IH}	Input HIGH Voltage			C Devices	2.0	V_{CC}+0.5 V	V
V_{OL}	Output LOW Voltage	I_{OL} = 2.0 mA				.45	V
V_{OH}	Output HIGH Voltage	I_{OH} = −400 µA			2.4		V
I_{IL}	Input Load Current	V_{IN} = V_{CC} to 0 V				±10	µA
I_{OFL}	Output Float Leakage Current	V_{OUT} = V_{CC} to 0.45 V				±10	µA
I_{CC}	Operating Power-Supply Current	CLK Freq =	8 MHz	C Devices		20	mA
			10 MHz	C Devices		20	
			12.5 MHz	C Devices		20	
I_{CCSB}	Standby Power-Supply Current	CLK Freq = DC, $\overline{CS}$ = HIGH, All Inputs/Data Bus HIGH, All Outputs Floating		C Devices		10	µA
I_{CCSB1}	V_{CC} Standby Power-Supply Current	CLK Freq = DC, $\overline{CS}$ = HIGH, All Other Inputs, Outputs, I/O Plus Floating				150	µA

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0 V)

Parameter Symbol	Parameter Description	Test Conditions		Min.	Max.	Units
C_{IN}	Input Capacitance	fc = 1 MHZ Unmeasured pins returned to GND	C Devices		10	pF
$C_{I/O}$	I/O Capacitance		C Devices		20	pF
C_{OUT}	Output Capacitance		C Devices		20	pF

See Section 6 for Thermal Characteristics Information.

SWITCHING TEST CIRCUIT

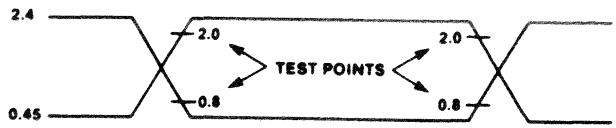


TC003430

C_L = 150 pF
C_L includes jig capacitance

SWITCHING TEST WAVEFORM

INPUT/OUTPUT



WF021040

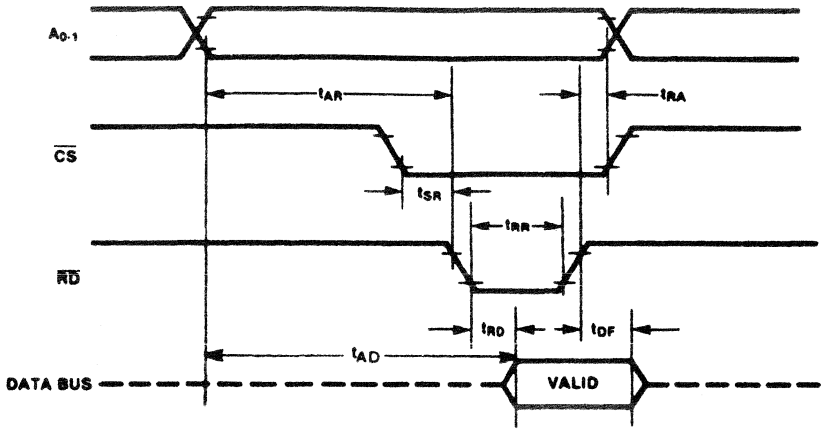
A.C. Testing: Inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0." Timing measurements are made at 2.0 V for a logic "1" and 0.8 V for a logic "0."

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

No.	Parameter Symbol	Parameter Description		8 MHz Min.	8 MHz Max.	10 MHz Min.	10 MHz Max.	12.5 MHz Min.	12.5 MHz Max.	Units
Read Cycle										
1	t_{AR}	Address Stable Before $\overline{RD}$ ↓		45		30		25		ns
2	t_{SR}	$\overline{CS}$ Stable Before $\overline{RD}$ ↓		0		0		0		ns
3	t_{RA}	Address Hold Time After $\overline{RD}$ ↑		0		0		0		ns
4	t_{RR}	$\overline{RD}$ Pulse Width	C Devices	150		95		80		ns
5	t_{RD}	Data Delay from $\overline{RD}$ ↓	C Devices		120		85		70	ns
6	t_{AD}	Data Delay from Address			220		185		150	ns
7	t_{DF}	$\overline{RD}$ ↑ to Data Floating		5	90	5	65	5	55	ns
8	t_{RV}	Command Recovery Time		200		165		135		ns
Write Cycle										
9	t_{AW}	Address Stable Before $\overline{WR}$ ↓		0		0		0		ns
10	t_{SW}	$\overline{CS}$ Stable Before $\overline{WR}$ ↓		0		0		0		ns
11	t_{WA}	Address Hold Time After $\overline{WR}$ ↑		0		0		0		ns
12	t_{WW}	$\overline{WR}$ Pulse Width		150		95		80		ns
13	t_{DW}	Data Setup Time Before $\overline{WR}$ ↑		120		95		80		ns
14	t_{WD}	Data Hold Time After $\overline{WR}$ ↑		0		0		0		ns
15	t_{RV}	Command Recovery Time		200		165		135		ns
Clock and Gate Cycle										
16	t_{CLK}	Clock Period		125	DC	100	DC	80	DC	ns
17	t_{PWH}	HIGH Pulse Width (Note 3)	C Devices	60		30		25		ns
18	t_{PWL}	LOW Pulse Width (Note 3)		60		50		40		ns
19	t_R	Clock Rise Time			25		25		25	ns
20	t_F	Clock Fall Time			25		25		25	ns
21	t_{GW}	Gate Width HIGH		50		50		40		ns
22	t_{GL}	Gate Width LOW		50		50		40		ns
23	t_{GS}	Gate Setup Time to CLK ↑		50		40		30		ns
24	t_{GH}	Gate Hold Time After CLK ↑ (Note 2)		50		50		40		ns
25	t_{OD}	Output Delay from CLK ↓			150		100		80	ns
26	t_{ODG}	Output Delay from Gate ↓			120		100		80	ns
27	t_{WC}	CLK Delay for Loading		0	55	0	55	0	45	ns
28	t_{WG}	Gate Delay for Sampling		−5	50	−5	40	−5	35	ns
29	t_{WO}	Out Delay from Mode Write			260		240		200	ns
30	t_{CL}	CLK Set Up for Count Latch		−4	45	−4	40	−4	35	ns

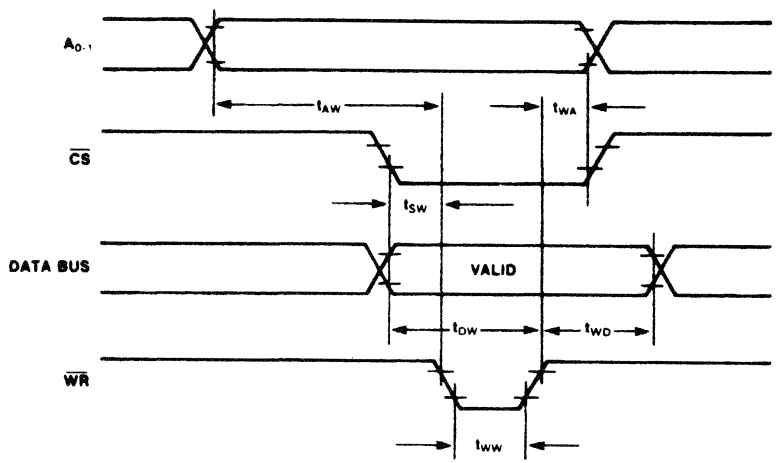
Notes: 1. Timings measured at $V_{OH} = 2.0$ V, $V_{OL} = 0.8$ V.
2. In Mode 1 and 5, triggers are sampled on each rising clock edge. A second trigger within 120 ns (70 ns for the 82C54-2) of the rising clock edge may not be detected.
3. LOW-going glitches that violate t_{PWH}, t_{PWL} may cause errors requiring Counter re-programming.

SWITCHING WAVEFORMS



WF021051

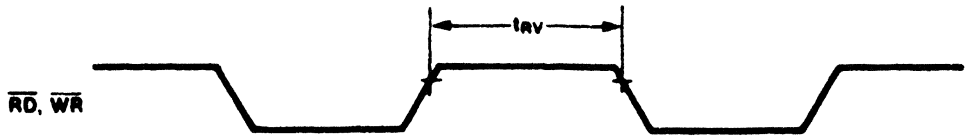
Read Cycle



WF021061

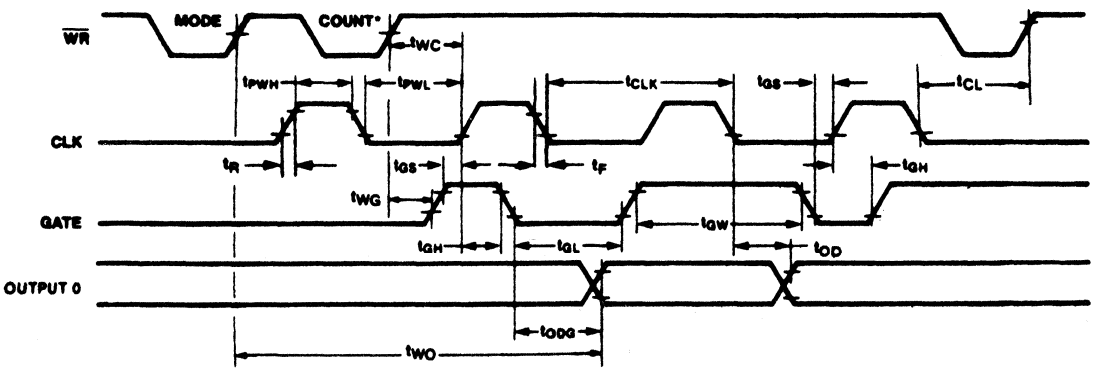
Write Cycle

SWITCHING WAVEFORMS (Cont'd.)



WF021070

Recovery Cycle



WF021080

*Last byte of count being written

Clock and Gate Cycle

8255A

Programmable Peripheral Interface
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Direct bit set/reset capability easing control application interface
- Reduces system package count
- Improved DC driving capability
- 24 programmable I/O pins
- Completely TTL compatible
- Fully compatible with the iAPX86 microprocessor family
- Improved timing characteristics

GENERAL DESCRIPTION

The 8255A is a general purpose programmable I/O device designed for use with iAPX Family microprocessors. It has 24 I/O pins which may be individually programmed in two groups of twelve and used in three major modes of operation. In the first mode, each group of twelve I/O pins may be programmed in sets of 4 and 8 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 four pins, three are used for handshaking and interrupt control signals. The third mode of operation (Mode 2) is a bidirectional bus mode which uses eight lines for a bidirectional bus, and five lines, borrowing one from the other group, for handshaking.

BLOCK DIAGRAM

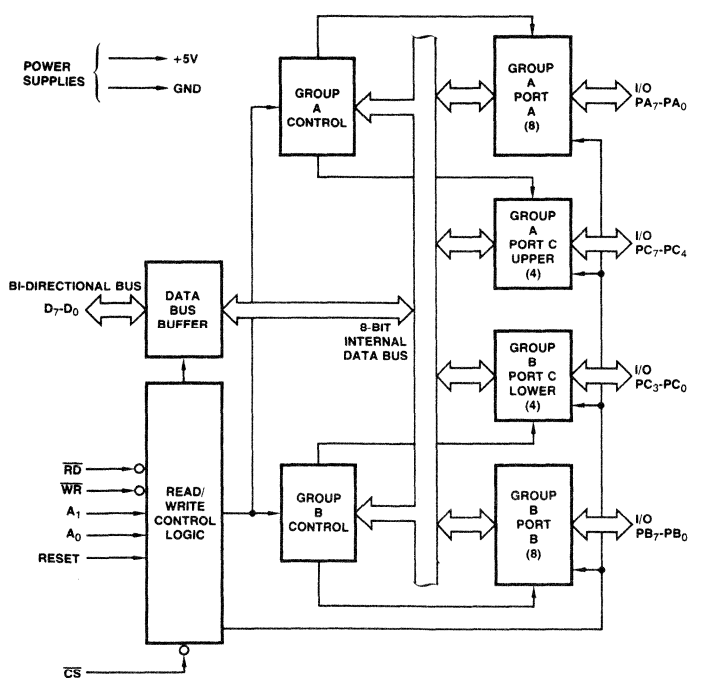
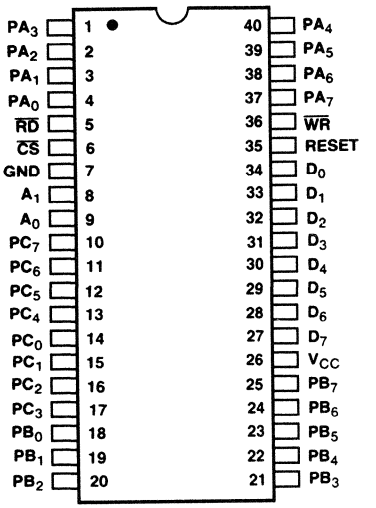


Figure 1

Publication # 04122 | Rev. B | Amendment /0
Issue Date: May 1987

CONNECTION DIAGRAMS
Top View

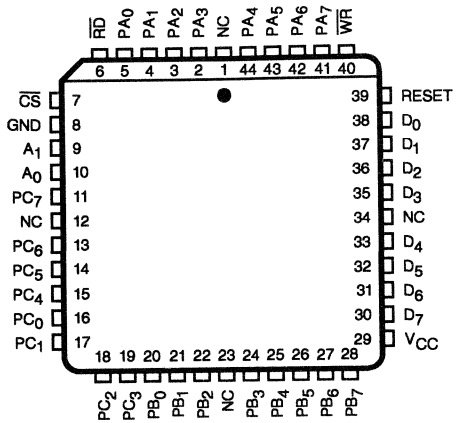
DIPs



CD005402

Figure 2.1

PLCC



CD010660

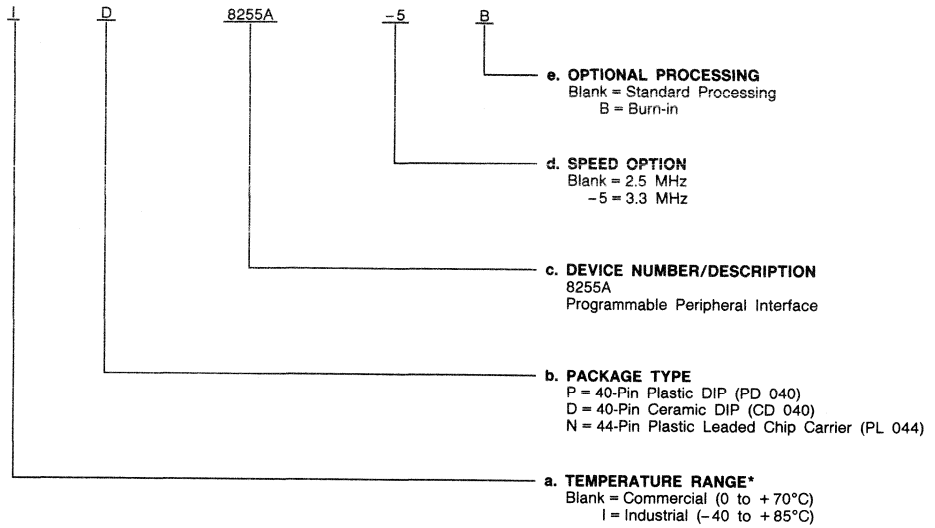
Figure 2.2

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D, N	8255A
	8255A-5
D, ID	8255AB
	8255A-5B

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Handbook (Order #09275A/0) for electrical performance characteristics.

PIN DESCRIPTION

Pin No.*	Name	I/O	Pin Description
27 – 34	$D_7 - D_0$	I/O	Data Bus (Bidirectional).
35	Reset	I	Reset Input.
6	$\overline{CS}$	I	Chip Select.
5	$\overline{RD}$	I	Read Input.
36	$\overline{WR}$	I	Write Input.
9, 8	A_0, A_1	I	Port Address.
37 – 40, 1 – 4	$PA_7 - PA_0$	I/O	Port A (Bit).
25 – 18	$PB_7 - PB_0$	I/O	Port B (Bit).
10 – 13, 17 – 14	$PC_7 - PC_0$	I/O	Port C (Bit).
26	V_{CC}		+5 Volts.
7	GND		0 Volts.

*Pin numbers correspond to DIPs only.

DETAILED DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in 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).

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

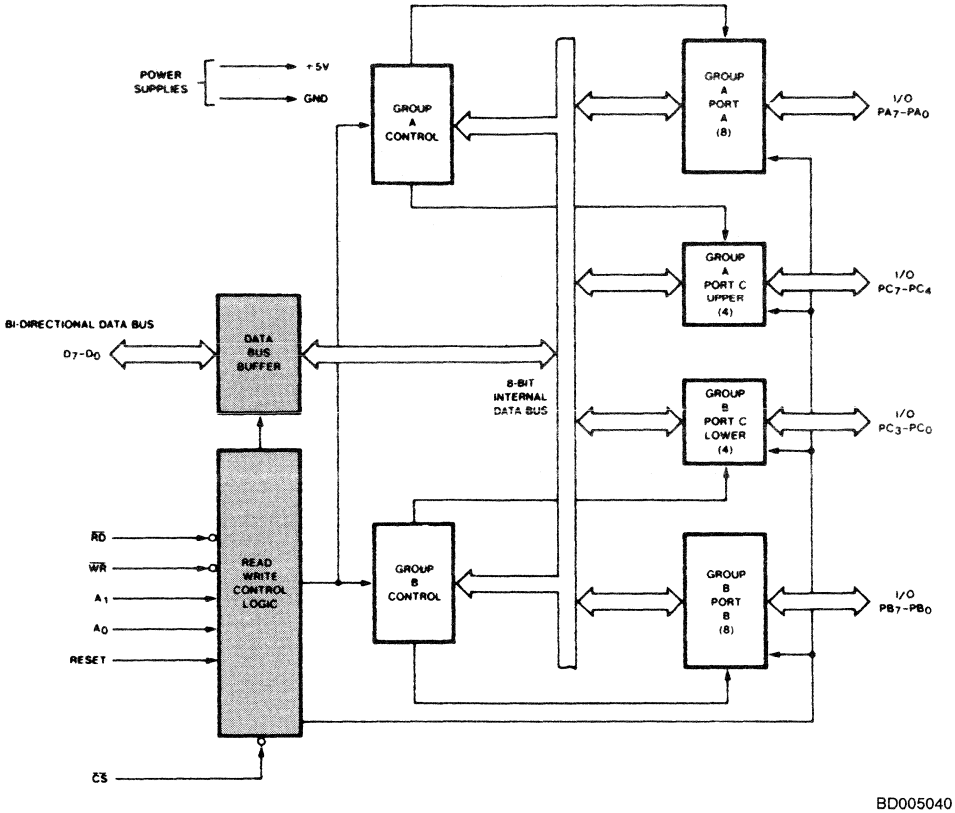


Figure 3. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

(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.

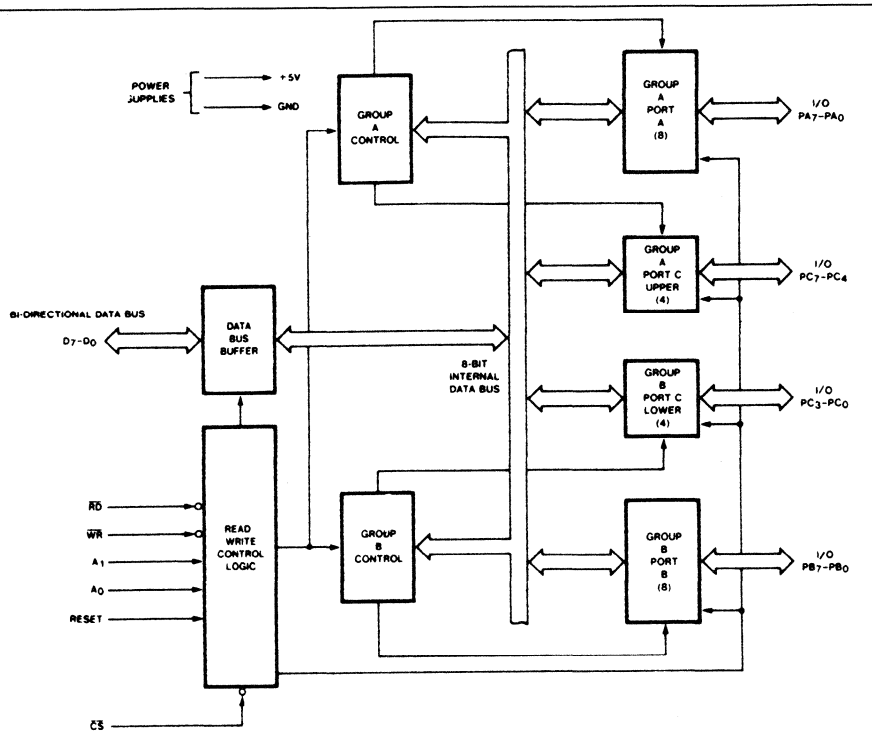


Figure 4. 8225A Block Diagram Showing Group A and Group B Control Functions

PROGRAMMING INFORMATION

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, and Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

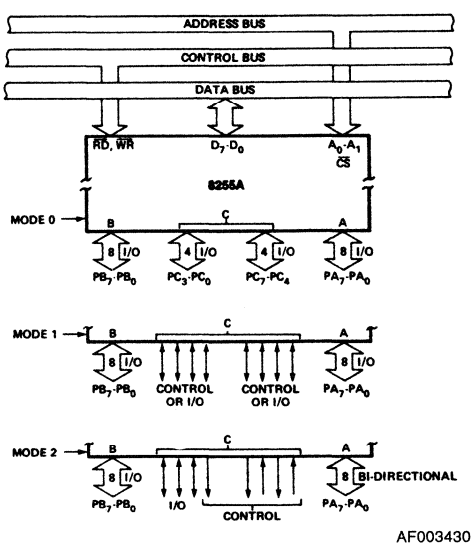


Figure 5. Basic Mode Definitions and Bus Interface

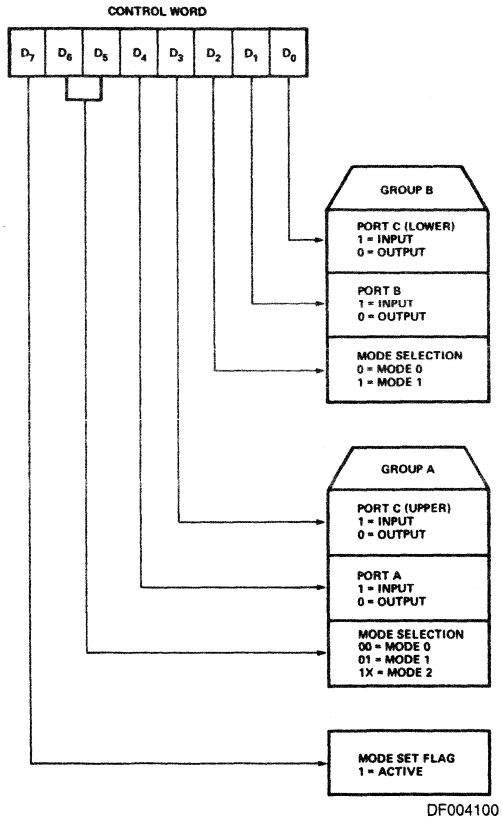


Figure 6. Mode Definition Format

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.

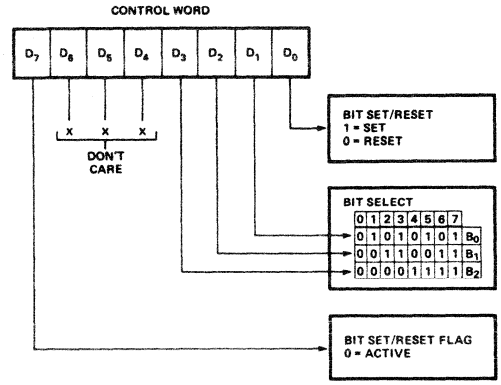


Figure 7. Bit Set/Reset Format

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

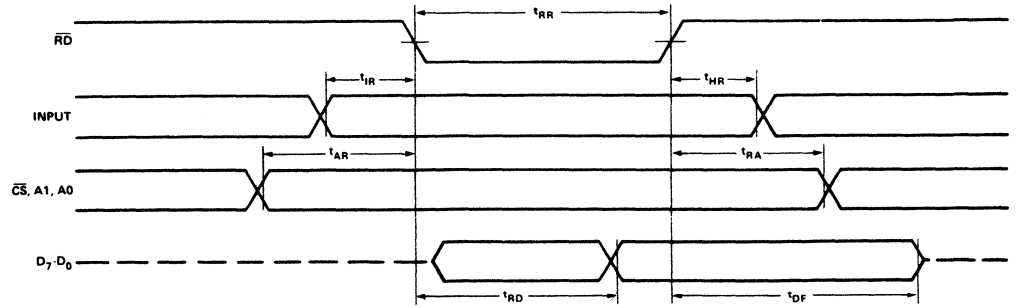
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

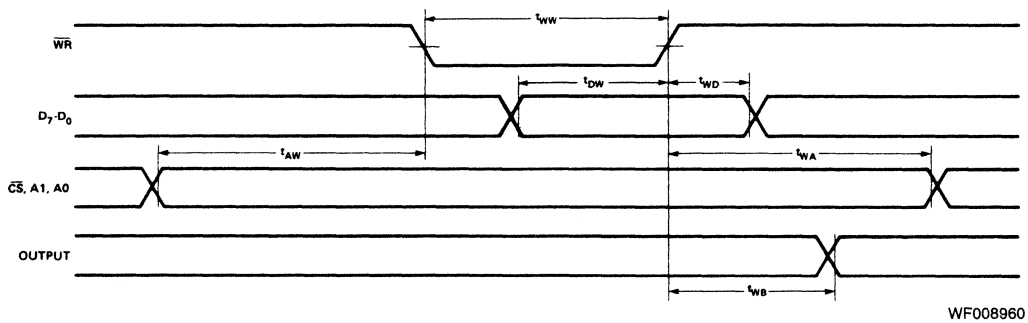
MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

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)

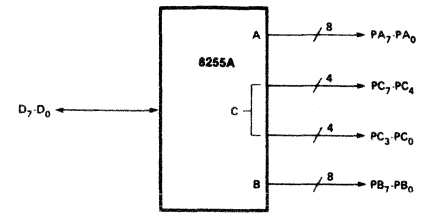
MODE 0 Port Definition

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 0 Configurations

CONTROL WORD =0

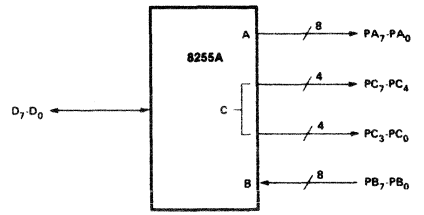
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	0



LS001460

CONTROL WORD =2

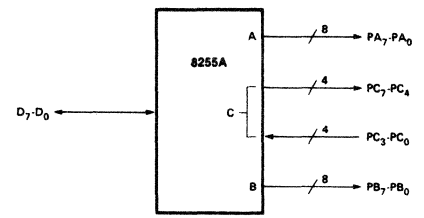
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	0



LS001470

CONTROL WORD =1

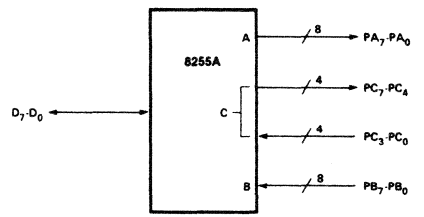
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	1



LS001480

CONTROL WORD =3

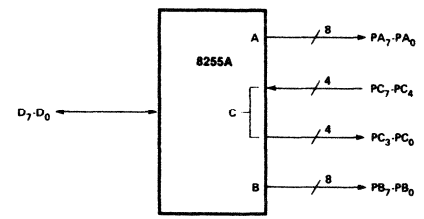
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	1



LS001490

CONTROL WORD =4

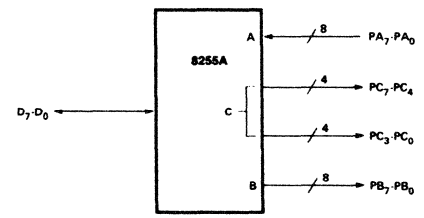
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



LS001500

CONTROL WORD =8

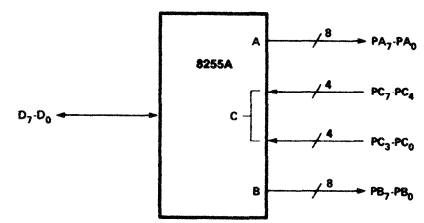
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0



LS001510

CONTROL WORD =5

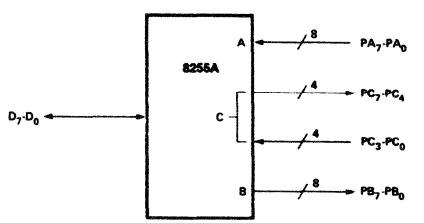
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



LS001520

CONTROL WORD =9

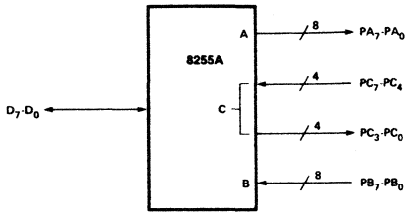
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



LS001530

CONTROL WORD =6

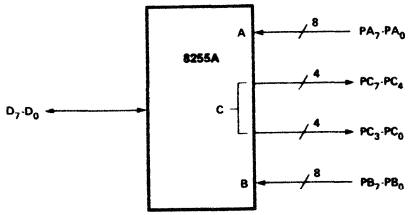
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



LS001540

CONTROL WORD =10

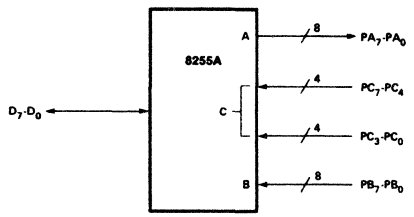
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0



LS001550

CONTROL WORD =7

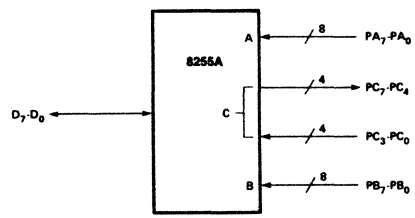
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



LS001560

CONTROL WORD =11

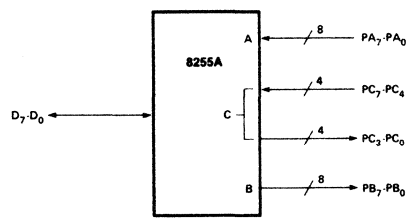
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1



LS001570

CONTROL WORD =12

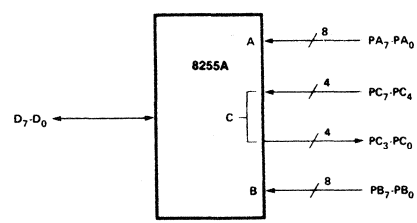
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	0	0



LS001580

CONTROL WORD =14

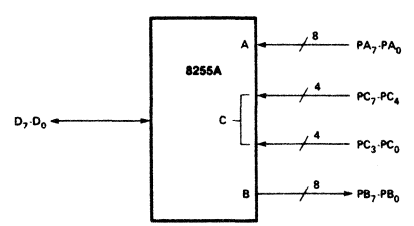
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	1	0



LS001590

CONTROL WORD =13

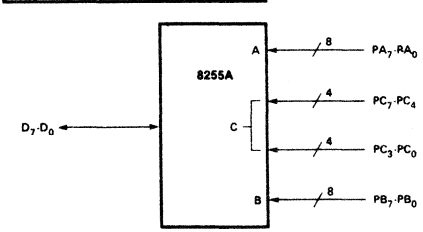
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	0	1



LS001600

CONTROL WORD =15

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	1	1



LS001610

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

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 when $\overline{STB}$ 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.

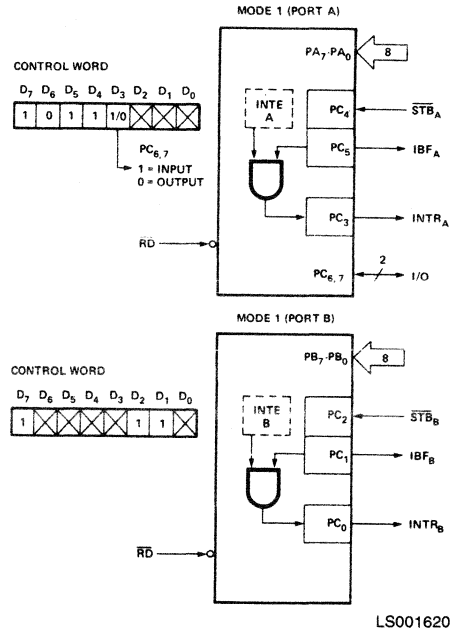


Figure 8. MODE 1 Input

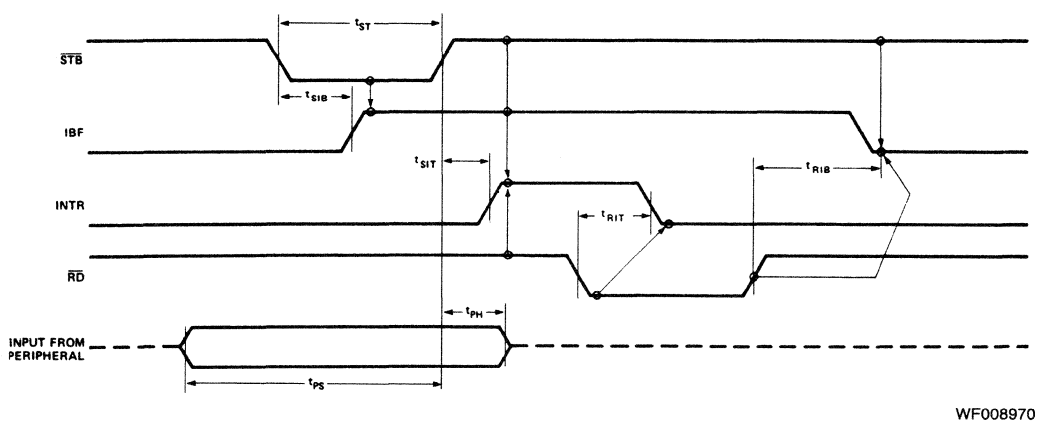


Figure 9. MODE 1 (Strobed Input)

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The OBF output will go "low" to indicate that the CPU has written data out to the specified port. The OBF F/F will be set by the rising edge of the WR input and reset by 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 ACK is a "one," OBF is a "one" and INTE is a "one." It is reset by the falling edge of WR.

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

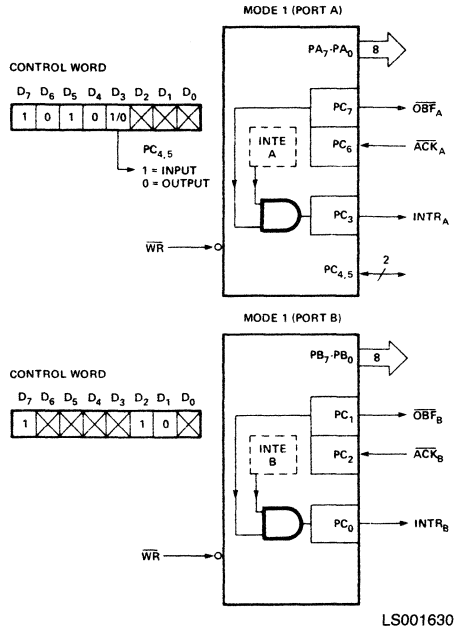


Figure 10. MODE 1 Output

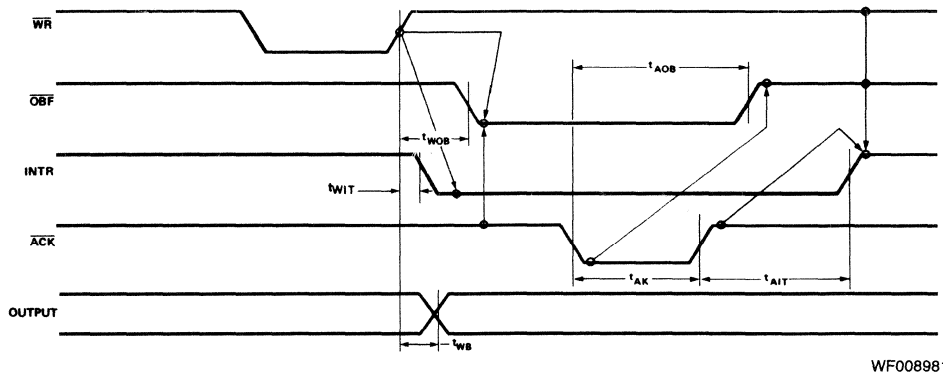


Figure 11. Mode 1 (Strobed Output)

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.

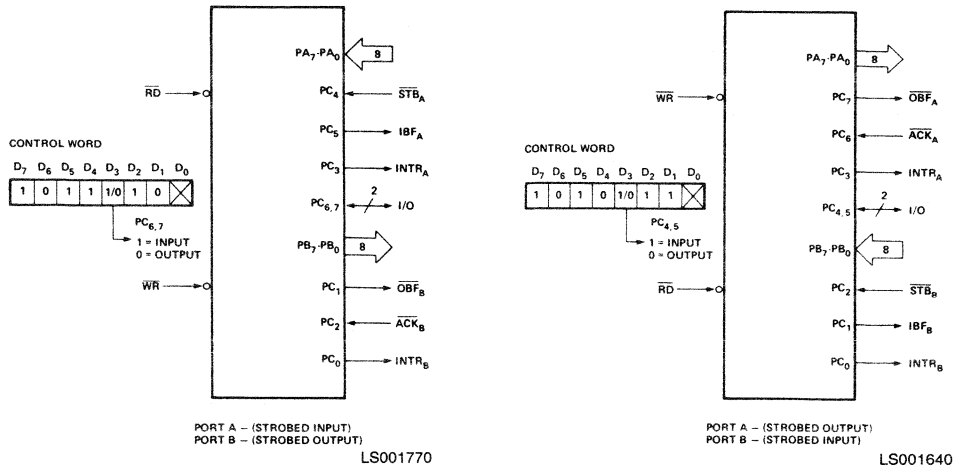


Figure 12. Combinations of MODE 1

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 both input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Full). The 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)

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.

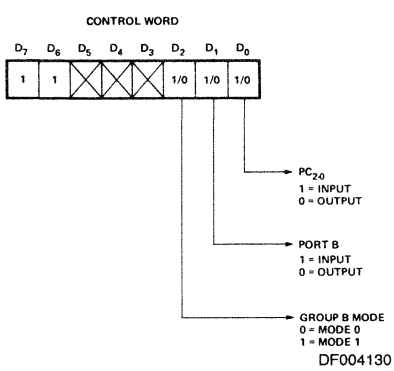


Figure 13. MODE Control Word

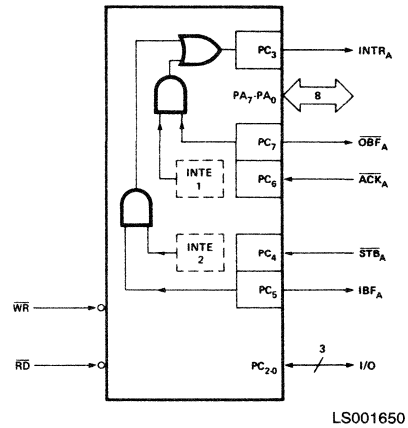


Figure 14. MODE 2

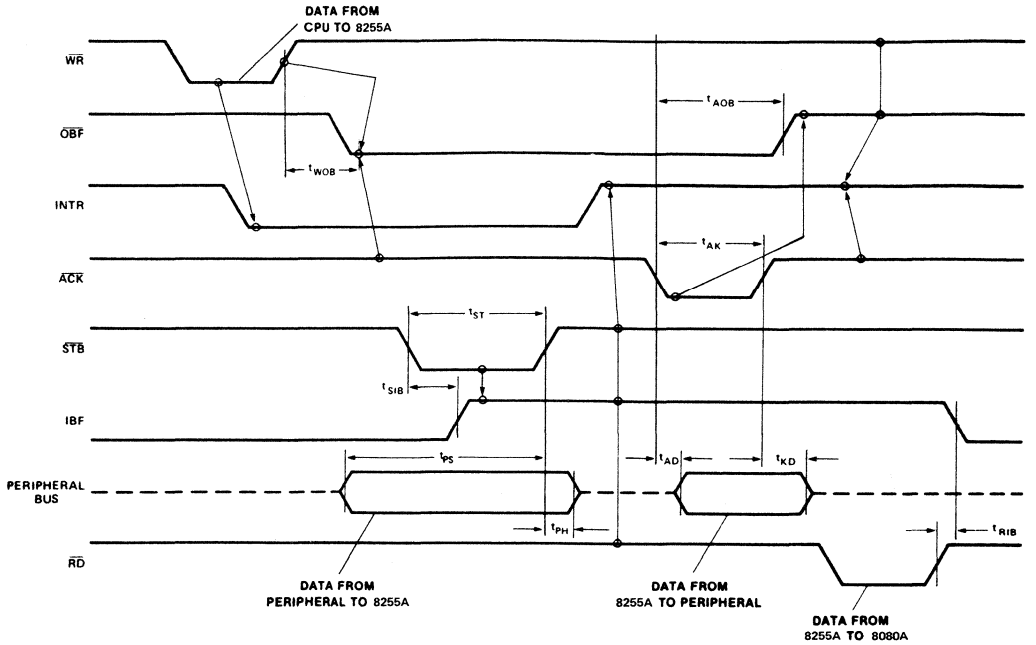
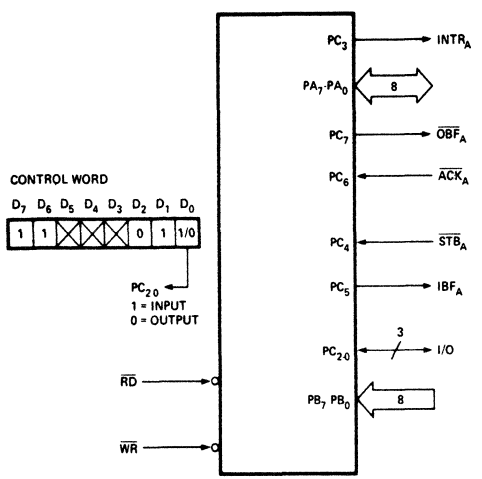


Figure 15. MODE 2 (Bidirectional)

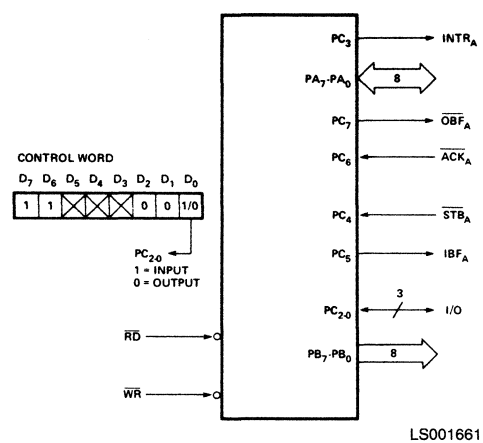
Note: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
($INTR = IBF \cdot \overline{MASK} \cdot \overline{STB} \cdot \overline{RD} + \overline{OBF} \cdot \overline{MASK} \cdot \overline{ACK} \cdot \overline{WR}$)

MODE 2 AND MODE 0 (INPUT)



LS001671

MODE 2 AND MODE 0 (OUTPUT)



LS001661

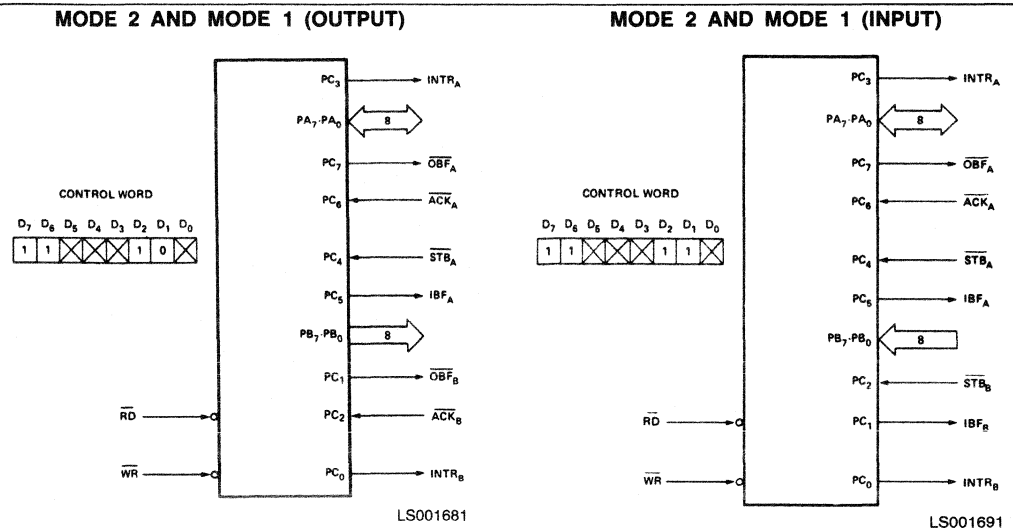


Figure 16. MODE 1/4 Combinations

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	$\overline{STB}_B$	$\overline{AKB}_B$	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 8255A 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 8255A is programmed to function in Modes 1 or 2, Port C generates or accepts "handshaking" 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.

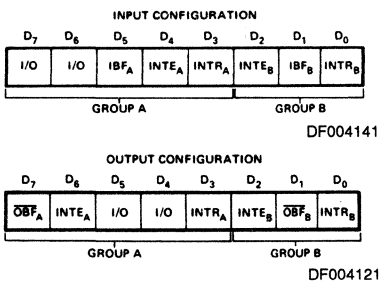


Figure 17. MODE 1 Status Word Format

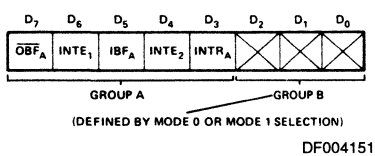


Figure 18. MODE 2 Status Word Format

APPLICATIONS INFORMATION

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 present a few examples of typical applications of the 8255A.

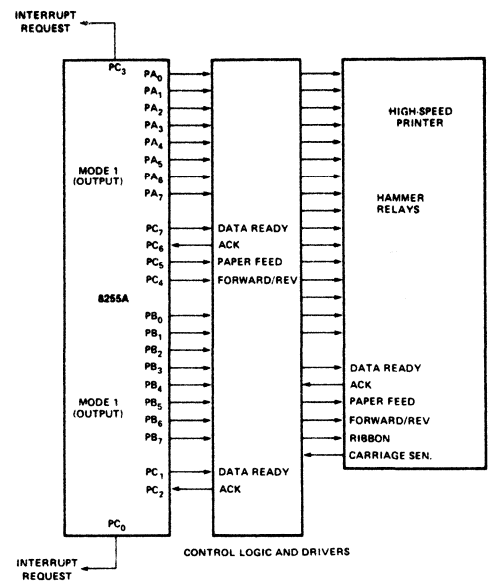


Figure 19. Printer Interface

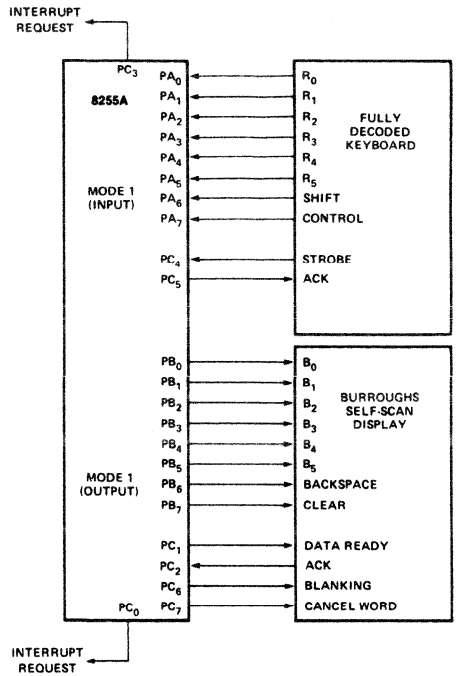


Figure 20. Keyboard and Display Interface

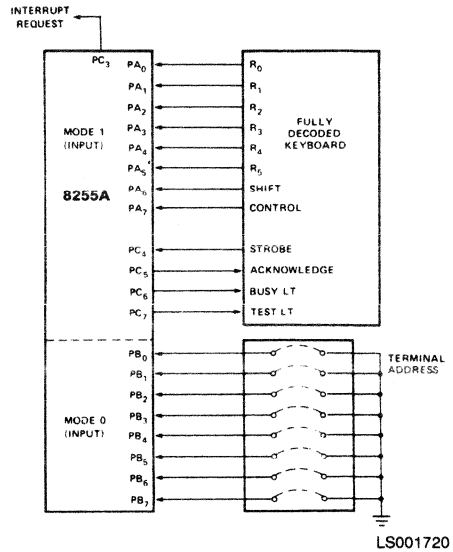


Figure 21. Keyboard and Terminal Address Interface

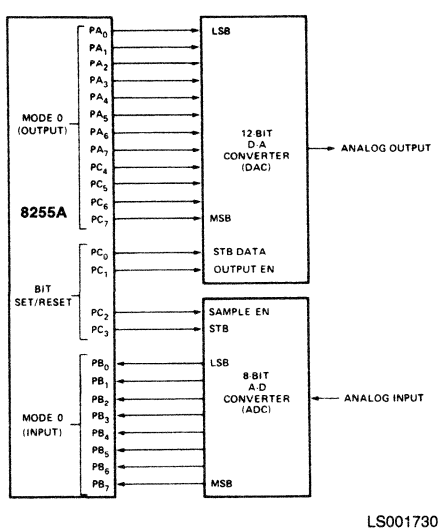


Figure 22. Digital to Analog, Analog to Digital

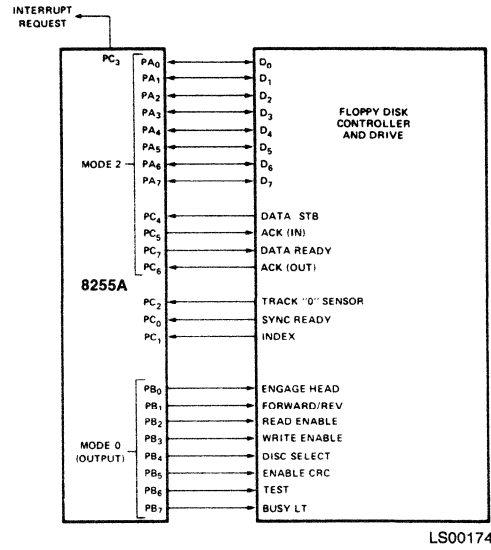
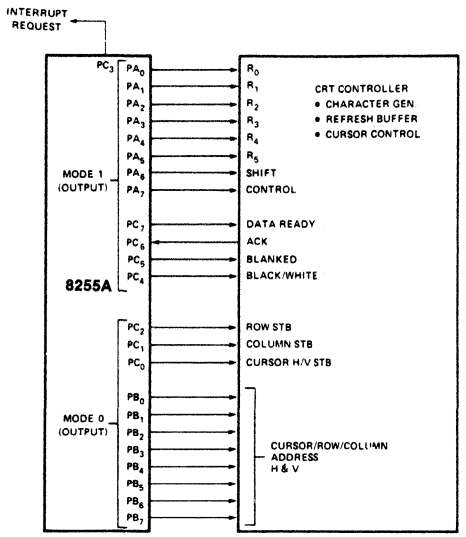
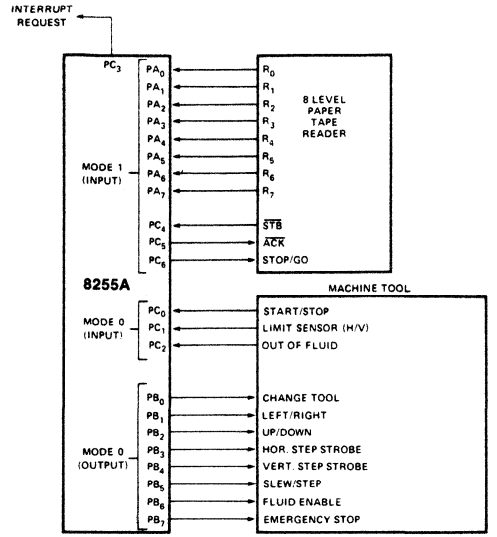


Figure 23. Basic Floppy Disk Interface



LS001750

Figure 24. Basic CRT Controller Interface



LS001760

Figure 25. Machine Tool Controller Interface

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to 7.0 V
All Signal Voltages
with Respect to V_{SS} ..... −0.5 to +7.0 V
Power Dissipation ..... 1.0 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ±10%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ±10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS (over Operating Ranges)

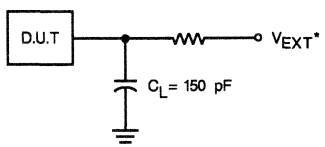
Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	0.8	Volts
V_{IH}	Input High Voltage		2.0	V_{CC}	Volts
V_{OL}(DB)	Output Low Voltage (Data Bus)	I_{OL} = 2.5 mA		0.45	Volts
V_{OL}(PER)	Output Low Voltage (Peripheral Port)	I_{OL} = 1.7 mA		0.45	Volts
V_{OH}(DB)	Output High Voltage (Data Bus)	I_{OH} = −400 μA	2.4		Volts
V_{OH}(PER)	Output High Voltage (Peripheral Port)	I_{OH} = −200 μA	2.4		Volts
I_{DAR} (Note 1)	Darlington Drive Current	R_{EXT} = 750Ω; V_{EXT} = 1.5 V	−1.0	−4.0	mA
I_{CC}	Power Supply Current			120	mA
I_{IL}	Input Load Current	V_{IN} = V_{CC} to 0 V		±10	μA
I_{OFL}	Output Float Leakage	V_{OUT} = V_{CC} to 0.45 V		±10	μA

Note 1: Available on any 8 pins from Port B and C.

CAPACITANCE T_A = 25°C; V_{CC} = GND = 0 V

Parameters	Description	Test Conditions	Min	Typ	Max	Units
C_{IN}	Input Capacitance	fc = 1 MHz			10	pF
$C_{I/O}$	I/O Capacitance	Unmeasured pins returned to GND			20	pF

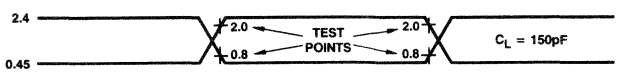
SWITCHING TEST LOAD CIRCUIT



TC002142

*V_{EXT} is set at various voltages during testing to guarantee the specification.
C_L includes jig capacitance.

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF006350

AC testing: Inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0."
Timing measurements are made at 2.0 V for a logic "1" and 0.8 V for a logic "0."

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

BUS PARAMETERS

READ

Parameters	Description	8255A		8255A-5		Units
		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 (Note 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

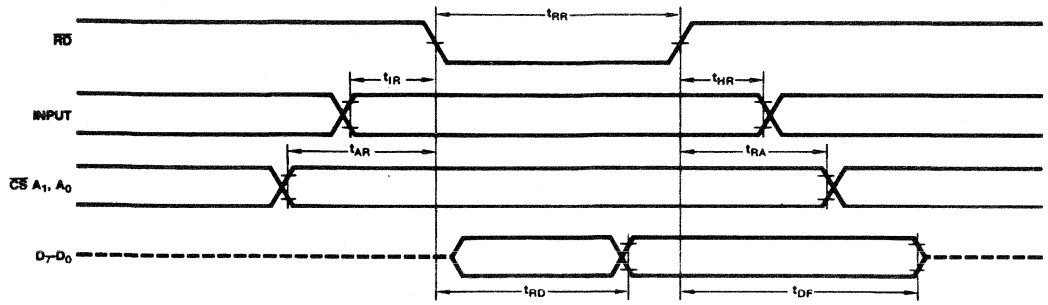
Parameters	Description	8255A		8255A-5		Units
		Min	Max	Min	Max	
t_{AW}	Address Stable Before WRITE	0		0		ns
t_{WA}	Address Stable After WRITE	20		20		ns
t_{WW}	WRITE Pulse Width	400		300		ns
t_{DW}	Data Valid to WRITE (T.E.)	100		100		ns
t_{WD}	Data Valid After WRITE	30		30		ns

OTHER TIMINGS

Parameters	Description	8255A		8255A-5		Units
		Min	Max	Min	Max	
t_{WB}	WR = 1 to Output (Note 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 (Note 1)		300		300	ns
t_{KD}	ACK = 1 to Output Float	20	250	20	250	ns
t_{WOB}	WR = 1 to OBF = 0 (Note 1)		650		650	ns
t_{AOB}	ACK = 0 to OBF = 1 (Note 1)		350		350	ns
t_{SIB}	STB = 0 to IBF = 1 (Note 1)		300		300	ns
t_{RIB}	RD = 1 to IBF = 0 (Note 1)		300		300	ns
t_{RIT}	RD = 0 to INTR = 0 to (Note 1)		400		400	ns
t_{SIT}	STB = 1 to INTR = 1 (Note 1)		300		300	ns
t_{AIT}	ACK = 1 toINTR = 1 (Note 1)		350		350	ns
t_{WIT}	WR = 0 to INTR = 0 (Notes 1, 3)		450		450	ns

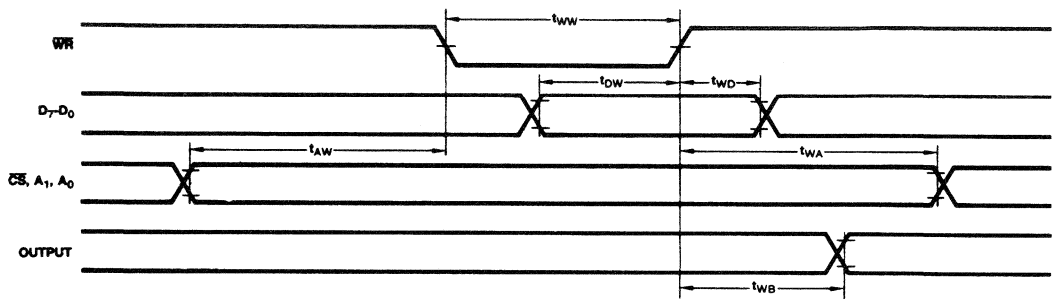
Notes: 1. Test Conditions: 8255A: $C_L = 150$ pF
2. Period of Reset pulse must be at least 50 μ during or after power on. Subsequent Reset pulse can be 500 ns min.
3. INTR↑ may occur as early as WR↓.

SWITCHING WAVEFORMS



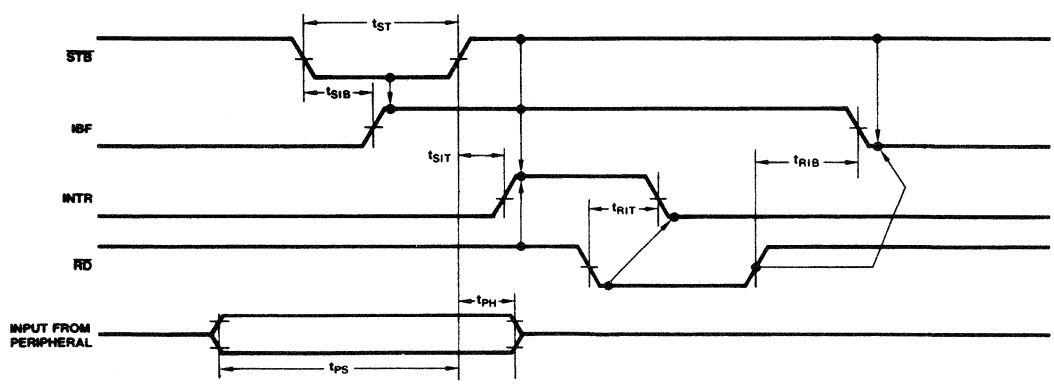
WF006320

Mode 0 (Basic Input)



WF006330

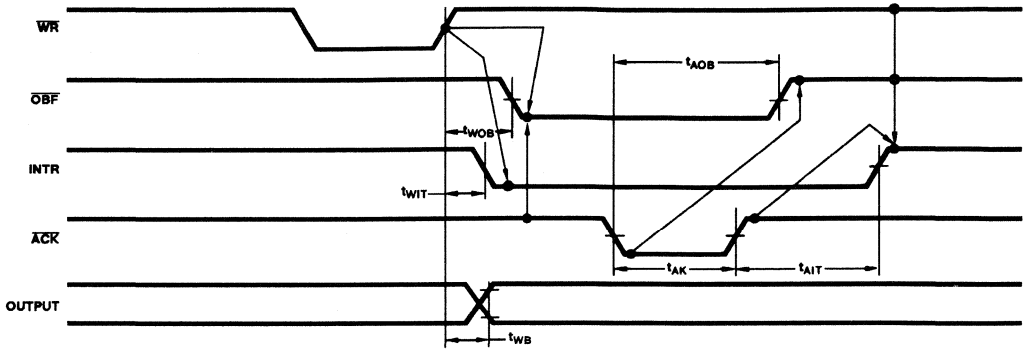
Mode 0 (Basic Output)



WF006340

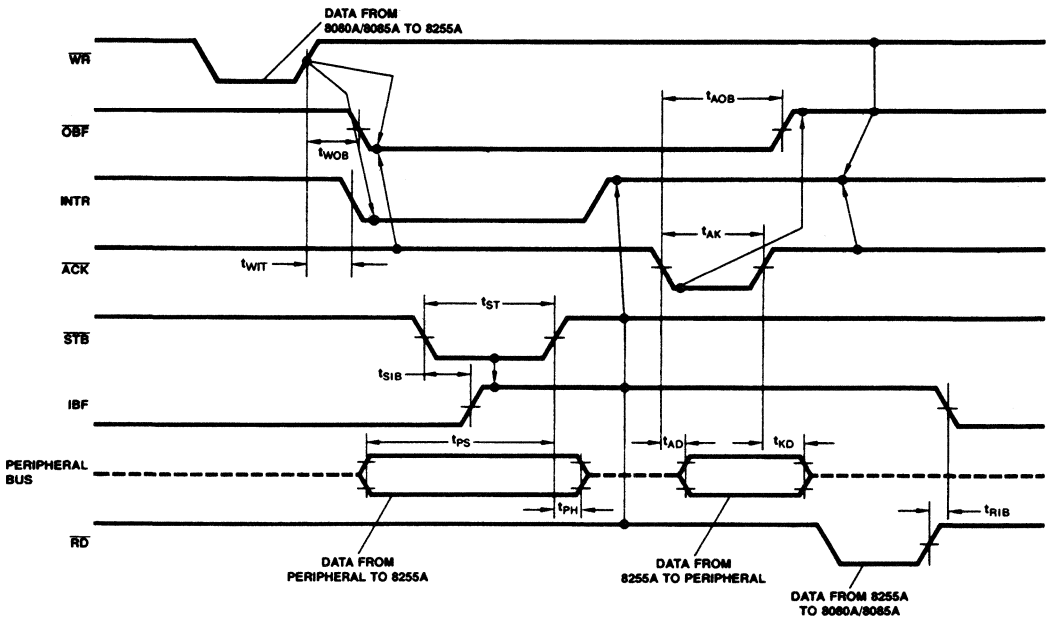
Mode 1 (Strobed Input)

SWITCHING WAVEFORMS (Cont'd.)



WF006362

Mode 1 (Strobed Output)



WF006370

Mode 2 (Bidirectional)

Note: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occur before $\overline{RD}$ is permissible (INTR = IBF · $\overline{MASK}$ · $\overline{STB}$ · $\overline{RD}$ + $\overline{OBF}$ · $\overline{MASK}$ · $\overline{ACK}$ · $\overline{WR}$).

8259A

Programmable Interrupt Controller
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Eight-Level Priority Controller
- Expandable to 64 Levels
- Programmable Interrupt Modes
- Individual Request Mask Capability
- Single +5V Supply (No Clocks)
- 28-Pin Dual-In-Line Package

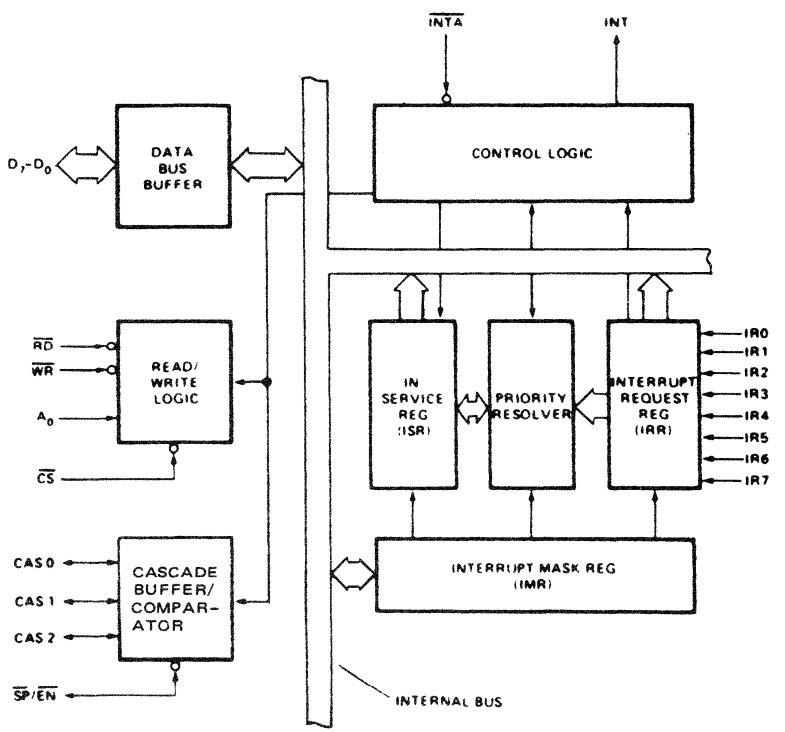
GENERAL DESCRIPTION

The 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 8259. Software originally written for the 8259 will operate the 8259A in all 8259 equivalent modes.

BLOCK DIAGRAM



BD003540

Figure 1.

Publication # 06102 | Rev. B | Amendment /0
Issue Date: April 1987

CONNECTION DIAGRAMS
Top View

DIPs

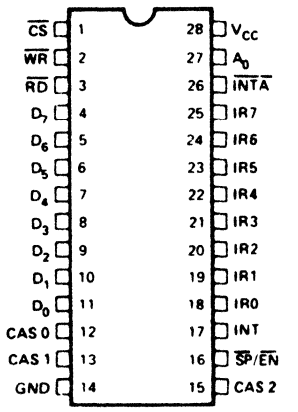


Figure 2.1

PLCC

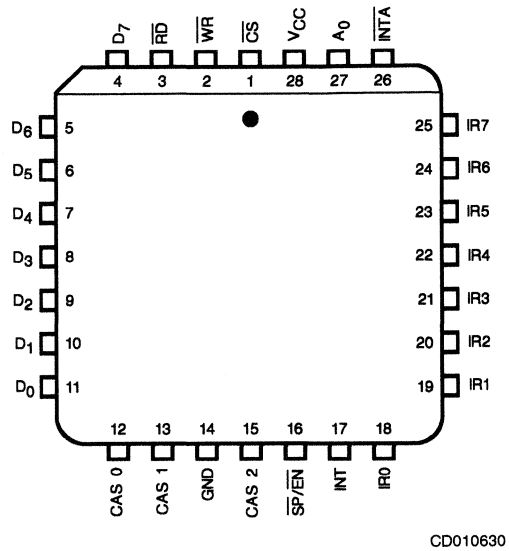


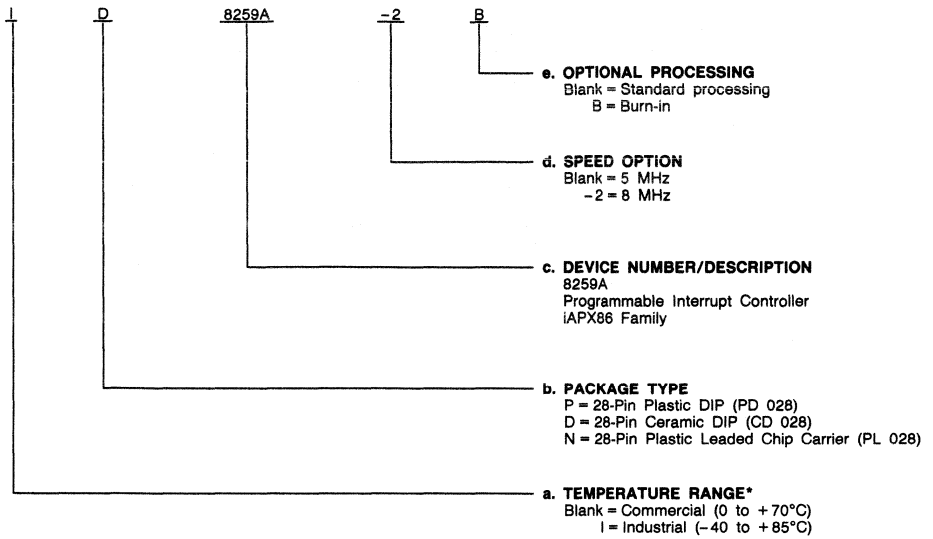
Figure 2.2

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D, N	8259A
	8259A-2
D, ID	8259AB
	8259A-2B

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
28	V_{CC}	I	**Supply:** +5V Supply.
14	GND	I	**Ground.**
1	$\overline{CS}$	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 $\overline{CS}$.
2	$\overline{WR}$	I	**Write:** A low on this pin when $\overline{CS}$ is low enables the 8259A to accept command words from the CPU.
3	$\overline{RD}$	I	**Read:** A low on this pin when $\overline{CS}$ is low enables the 8259A to release status onto the data bus for the CPU.
4-11	D_7-D_0	I/O	**Bidirectional Data Bus:** Control, status and interrupt-vector information are transferred via the bus.
12, 13, 15	CAS_0-CAS_2	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 abd inputs for a slave 8259A.
16	$\overline{SP}/\overline{EN}$	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).
17	INT	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.
18-25	IR_0-IR_7	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).
26	$\overline{INTA}$	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.
27	A_0	I	**AO 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 iAPX 86, 88).

DETAILED 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.

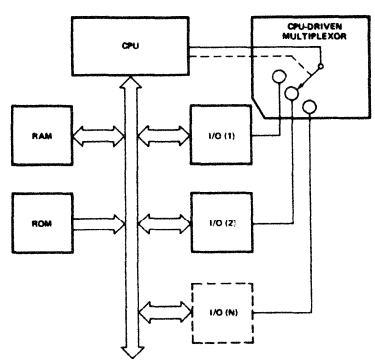
This method is called *Interrupt*. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

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.

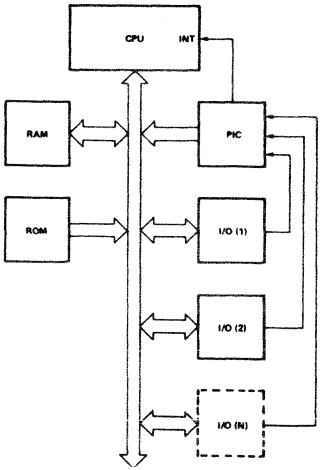
The 8259A

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.



AF003320

Figure 3a. Polled Method



AF003330

Figure 3b. Interrupt Method

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, 8085AH 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)

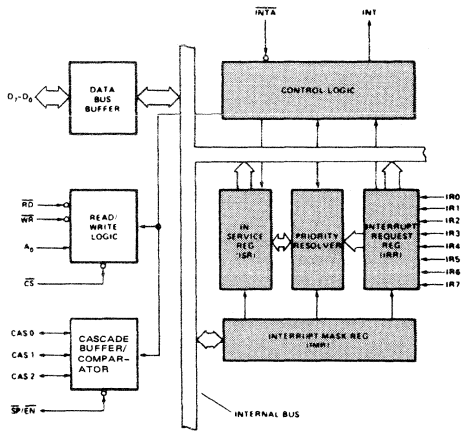
A LOW on this input enables the 8259A. No reading or writing of the chip will occur unless the device is selected.

$\overline{WR}$ (WRITE)

A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 8259A.

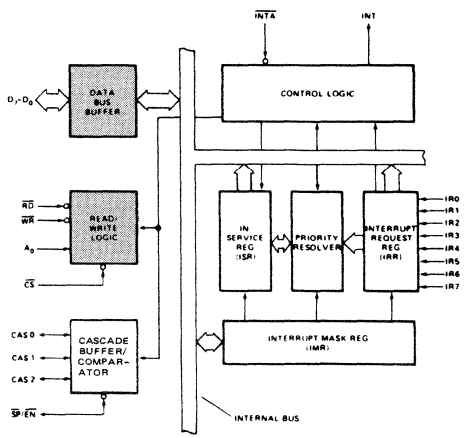
$\overline{RD}$ (READ)

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.



BD003540

Figure 4a. 8259A Block Diagram



BD003540

Figure 4b. 8259A Block Diagram

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 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 $\overline{INTA}$ 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 8080A/85AH 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 $\overline{INTA}$ 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 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 $\overline{INTA}$ pulses to be sent to the 8259A from the CPU group.

6. These two $\overline{INTA}$ pulses allow the 8259A to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{INTA}$ 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 8259A. 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.

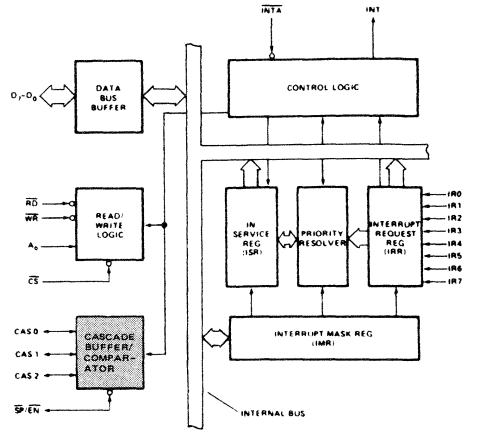
The events occurring in an 8086 system are the same until step 4.

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 8259A does not drive the Data Bus during this cycle.

5. The 8086 will initiate a second $\overline{INTA}$ pulse. During this pulse, the 8259A releases an 8-bit pointer onto the Data Bus where it is read by the CPU.

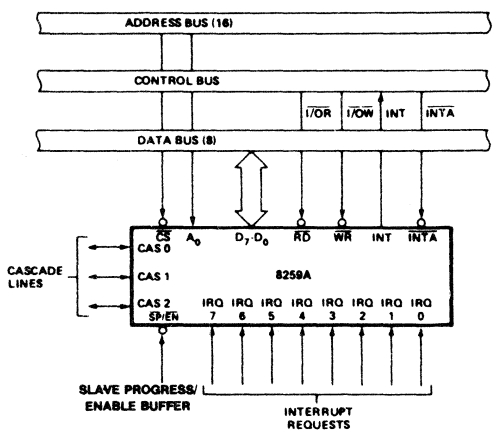
6. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

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.



BD003540

Figure 4c. 8259A Block Diagram



AF003300

Figure 5. 8259A Interface to Standard System Bus

Interrupt Sequence Outputs

8080A/85AH

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	S5	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

8086, 8088

8086 mode is similar to 8080A 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 8080A/85AH 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 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 INFORMATION

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 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, 8080A/85AH 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 8080A/85AH 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 8080A/85AH 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.

Initialization Command Word 4 (ICW4)

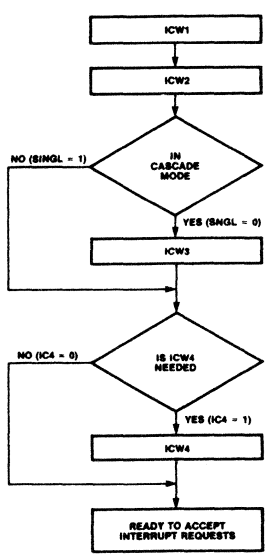
SFNM: If SFNM = 1 the special fully nested mode is programmed.

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.

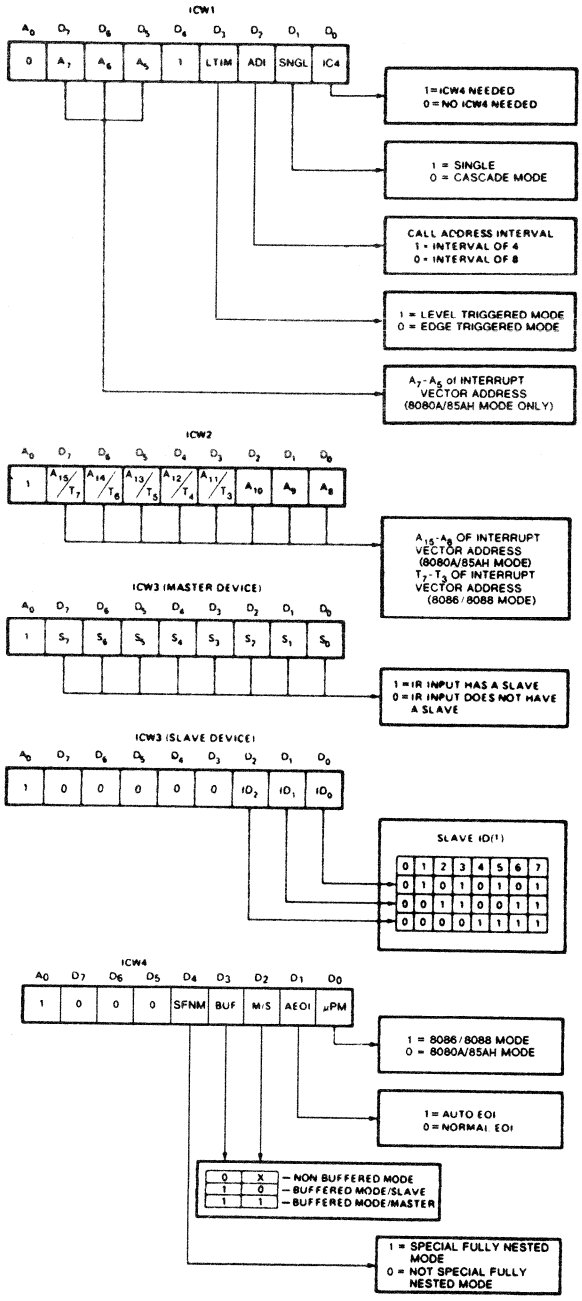
AEOI: If AEOI = 1 the automatic end of interrupt mode is programmed.

μPM: Microprocessor mode: μPM = 0 sets the 8259A for 8080A/85AH system operation, μPM = 1 sets the 8259A for 8086 system operation.



PF001310

Figure 6. Initialization Sequence



Note 1: 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 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	EO1	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)

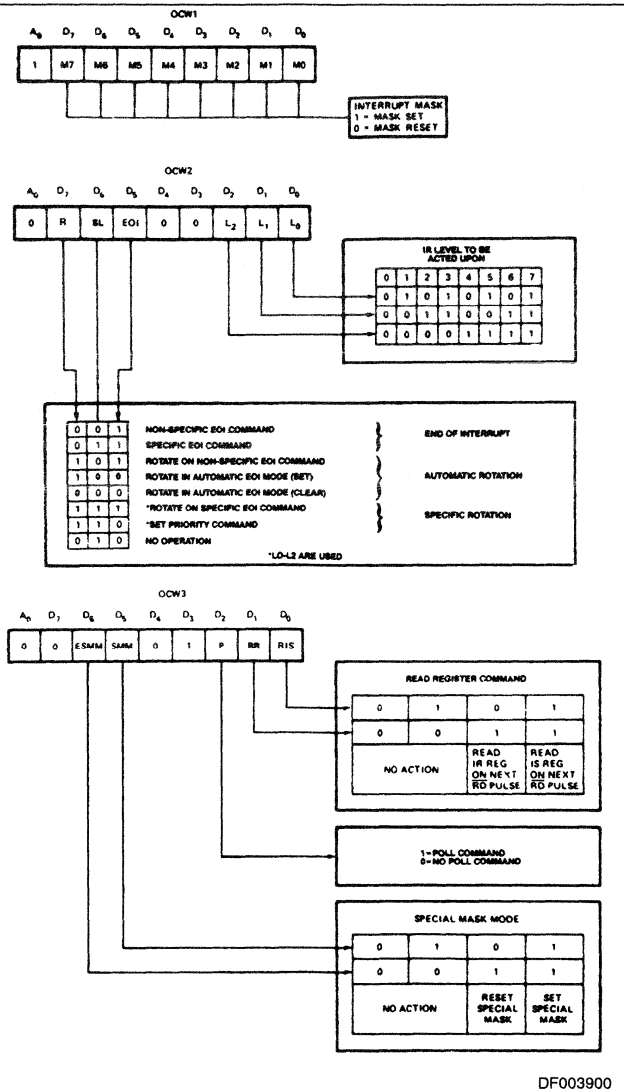
R, SL, EOI – These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

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 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.



DF003900

Figure 8. Operation Command Word Format

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 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 preserve 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 LO – 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

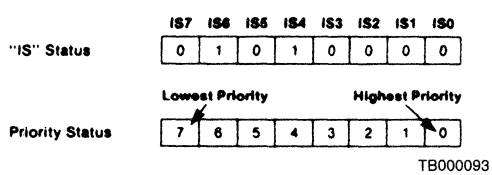
If AEOI = 1 in ICW4, then the 8259A will operate in AEOI mode continuously until reprogrammed by ICW4. In this mode the 8259A will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in 8080A/85AH, second in 8086). Note that from 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.

AUTOMATIC ROTATION (Equal Priority Devices)

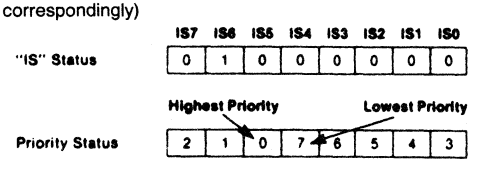
In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most *once*. For example, if the priority and "in service" status is:

Before Rotate (IR4 the highest priority requiring service)



TB000093

After Rotate (IR4 was serviced, all other priorities rotated correspondingly)



TB000094

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 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 OCW3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

Poll Command

In this mode the INT output is not used or 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 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 $\overline{RD}$.

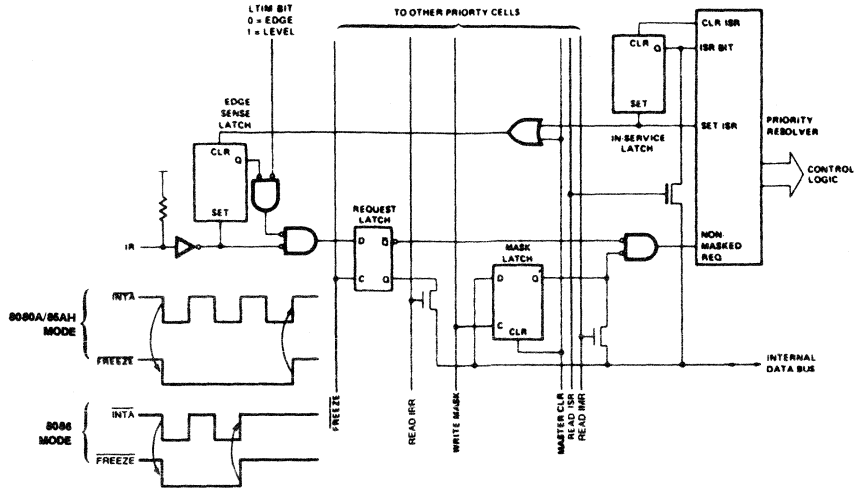
The word enabled onto the data bus during $\overline{RD}$ is:

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 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.



LS001361

NOTES1. MASTER CLEAR ACTIVE ONLY DURING ICW1
2. FREEZE/ IS ACTIVE DURING INTA/ AND POLL SEQUENCES ONLY
3. TRUTH TABLE FOR D-LATCH

C	D	Q	OPERATION
1	D_I	Q_I	FOLLOW
0	X	Qn-1	HOLD

Figure 9. Priority Cell - Simplified Logic Diagram

Reading The 8259A 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 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 $\overline{RD}$ 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 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 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 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.

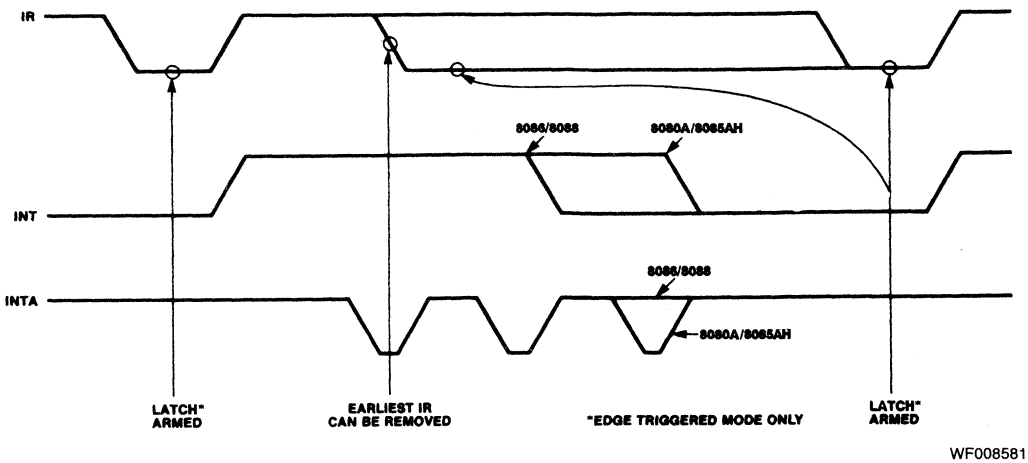


Figure 10. IR Triggering Timing Requirements

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 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

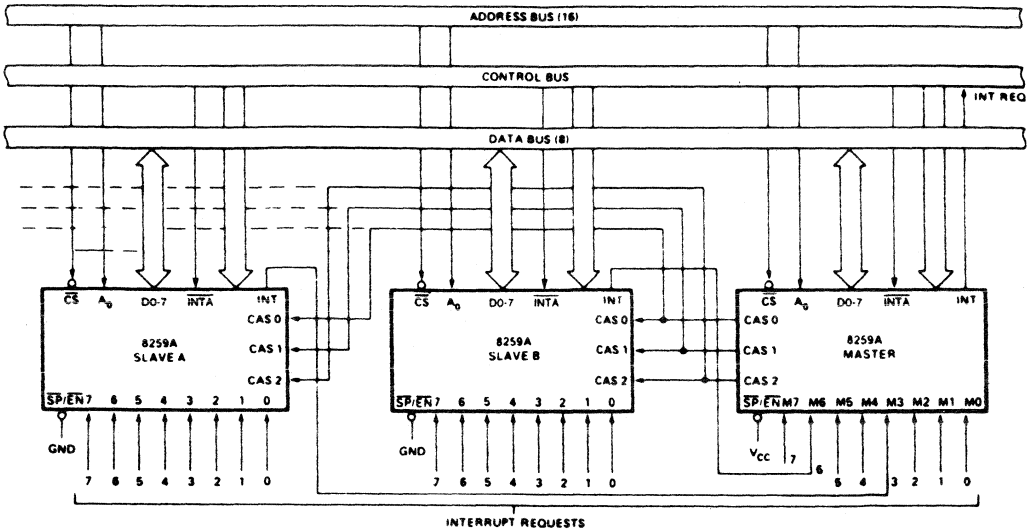
The 8259A can be easily interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels.

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 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 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).



AF003310

Figure 11. Cascading the 8259A

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin
with Respect to Ground ..... −0.5V to +7V
Power Dissipation ..... 1 Watt

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ±10%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ±10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

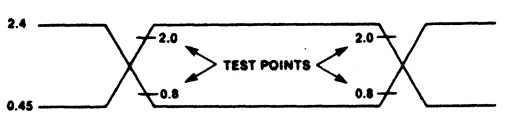
DC CHARACTERISTICS over operating ranges unless otherwise specified.

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	0.8	V
V_{IH}	Input High Voltage	Commercial	2.0	V_{CC} + 0.5 V	V
		Industrial	2.3		
V_{OL}	Output Low Voltage	I_{OL} = 2.2 mA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400 μA	2.4		V
$V_{OH(INT)}$	Interrupt Output High Voltage	I_{OH} = −100 μA	3.5		V
		i_{OH} = −400 μA	2.4		V
I_{LI}	Input Load Current	$0V \leqslant V_{IN} \leqslant V_{CC}$	−10	+10	μA
I_{LOL}	Output Leakage Current	$0.45V \leqslant V_{OUT} \leqslant V_{CC}$	−10	+10	μA
I_{CC}	V_{CC} Supply Current			85	mA
I_{LIR}	IR Input Load Current	V_{IN} = 0		−300	μA
		$V_{IN} = V_{CC}$		10	μA

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0V)

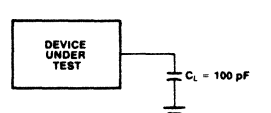
Parameters	Description	Test Conditions	Min	Typ	Max	Units
C_{IN}	Input Capacitance	fc = 1MHz			10	pF
$C_{I/O}$	I/O Capacitance	Unmeasured pins returned to V_{SS}			20	pF

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF007100

SWITCHING TEST LOAD CIRCUIT



TC001840

Note: 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."

C_L = 100 pF.
C_L includes jig capacitance.

Refer to Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified.

Timing Requirements

Parameters	Description	Test Conditions	8259A		8259A-2		Units
			Min	Max	Min	Max	
TAHRL	A_0/$\overline{CS}$ Setup to $\overline{RD}$/$\overline{INTA}$↓		0		0		ns
TRHAX	A_0/$\overline{CS}$ Hold after $\overline{RD}$/$\overline{INTA}$↑		0		0		ns
TRLRH	$\overline{RD}$ Pulse Width		235		160		ns
TAHWL	A_0/$\overline{CS}$ Setup to $\overline{WR}$↓		0		0		ns
TWHAX	A_0/$\overline{CS}$ 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)	(Note 1)	100		100		ns
TCVIAL	Cascade Setup 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		160		ns
TWHWL	End of $\overline{WR}$ to next $\overline{WR}$		190		190		ns
*TCHCL	End of Command to next Command (Not same command type)		500		500		ns
	End of $\overline{INTA}$ sequence to next $\overline{INTA}$ sequence						

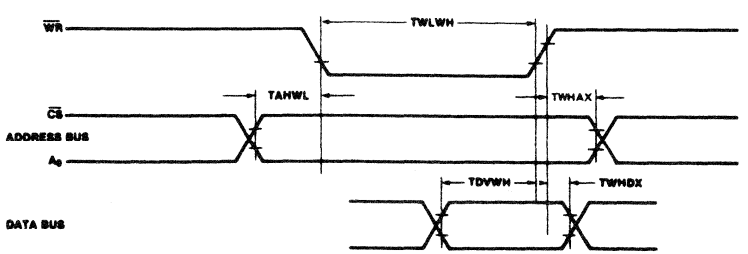
*Worst-case timing for TCHCL in an actual microprocessor system is typically much greater than 500ns (i.e. 8085A = 1.6μs, 8085A-2 = 1μs, 8086 = 1μs, 8086-2 = 625ns)
Note 1: This is the low time required to clear the input latch in the edge triggered mode.

Timing Responses

Parameters	Description	Test Conditions	8259A		8259A-2		Units
			Min	Max	Min	Max	
TRLDV	Data Valid from $\overline{RD}$/$\overline{INTA}$↓	C of Data Bus = 100pF		200		120	ns
TRHDZ	Data Float after $\overline{RD}$/$\overline{INTA}$↑	C of Data Bus	10	100	10	85	ns
TJHIH	Interrupt Output Delay	Max test C = 100pF Min test C = 15pF		350		300	ns
TIALCV	Cascade Valid from First $\overline{INTA}$↓ (Master Only)	C_{INT} = 100pF		565		360	ns
TRLEL	Enable Active from $\overline{RD}$↓ or $\overline{INTA}$↓	$C_{CASCADE}$ = 100pF		125		100	ns
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

SWITCHING WAVEFORMS

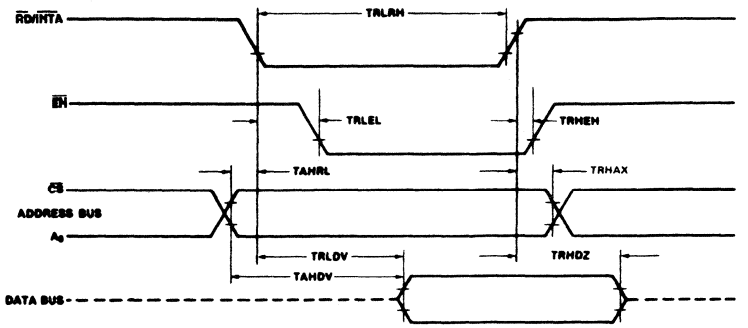
WRITE



WF006070

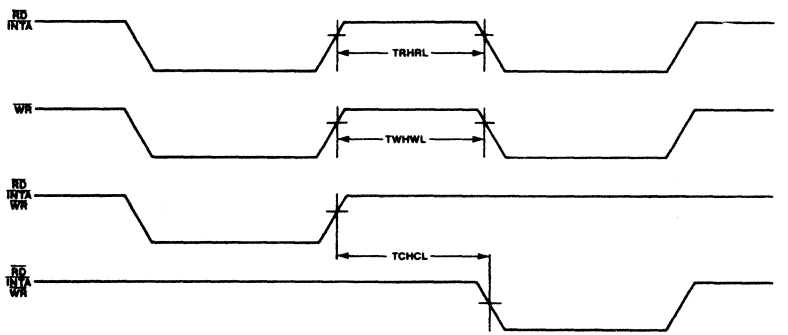
SWITCHING WAVEFORMS (Continued)

READ/INTA



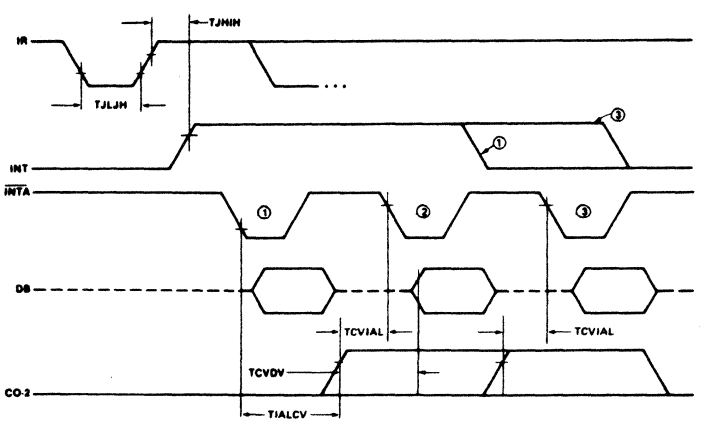
WF006080

OTHER TIMING



WF006090

INTA SEQUENCE



WF006100

Notes: Interrupt output must remain HIGH at least until leading edge of first INTA.
1. Cycle 1 in iAPX86, iAPX88 systems, the Data Bus is not active.

8284A/8284A-1

Clock Generator and Driver for 8086, 8088 Processors

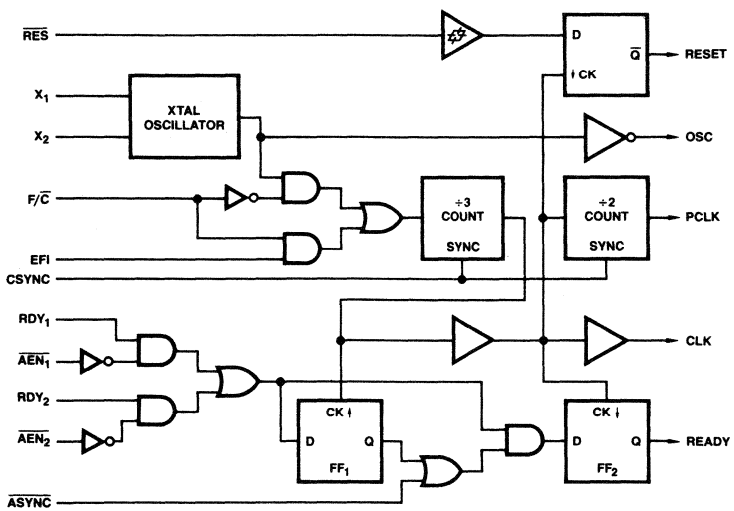
DISTINCTIVE CHARACTERISTICS

- Generates the System Clock for the 8086, 8088 Processors: 5MHz, 8MHz with 8284A; 10MHz with 8284A-1
- Uses a crystal or a TTL signal for frequency source
- Provides local READY and Multibus* READY synchronization
- Generates system reset output from Schmitt trigger input
- Capable of clock synchronization with other 8284As

GENERAL DESCRIPTION

The 8284A is a single chip clock generator/driver for the 8086, 8088 processors. The chip contains a crystal-controlled oscillator, a divide-by-three counter, complete MULTIBUS* "Ready" synchronization and reset logic.

BLOCK DIAGRAM



BD001440

RELATED AMD PRODUCTS

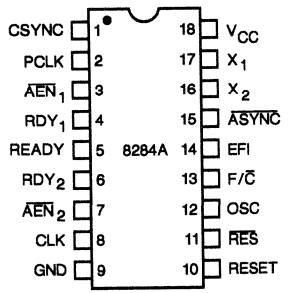
Part No.	Description
Am8086	16-Bit Microprocessor
8288	Bus Controller

*MULTIBUS is a registered trademark of Intel Corp.

Publication # 03359 Rev. D Amendment /0
Issue Date: April 1987

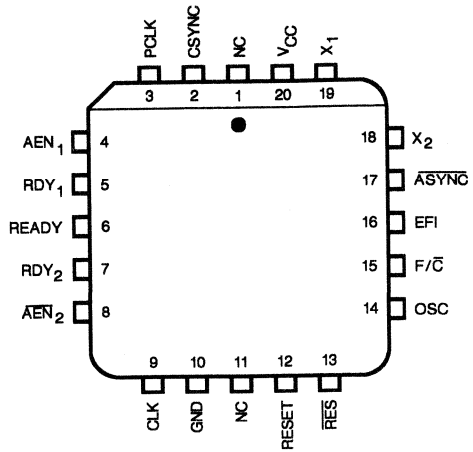
CONNECTION DIAGRAMS
Top View

DIPs



CD001582

PLCC



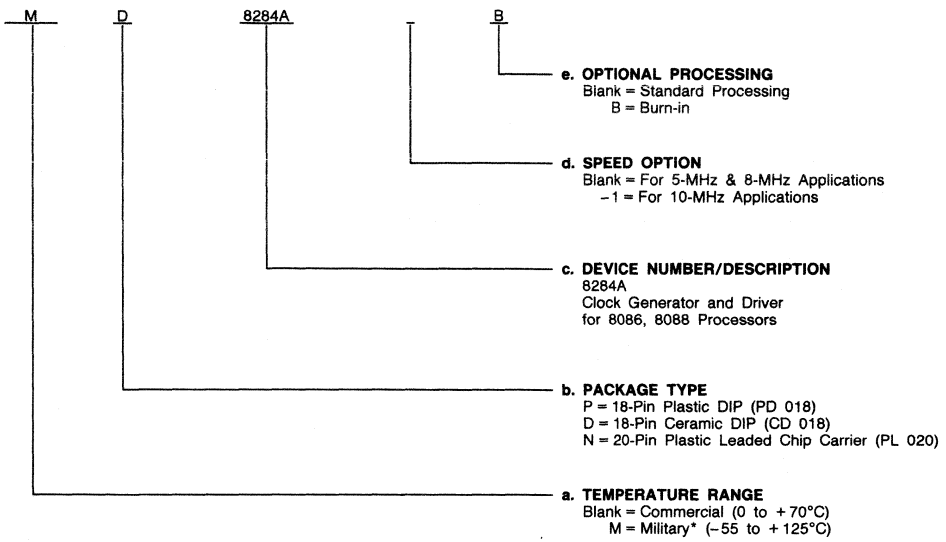
CD009272

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
MD, D, P, N	8284A
MD, D, P	8284AB
D	8284A-1
	8284A-1B

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

* Military temperature range products are NPL (Non-Compliant Products List) or Non-MIL-STD-883C Compliant products only.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
3, 7	$\overline{AEN}_1$, $\overline{AEN}_2$	I	Address Enable. The $\overline{AEN}$ signal is used to qualify the Bus Ready signal (RDY_1 or RDY_2). $\overline{AEN}_1$ validates RDY_1 while $\overline{AEN}_2$ validates RDY_2. It is possible for the processor to access two Multi-Master System Busses if you use both signals. Both signals are tied LOW in non Multi-Master Systems.
4, 6	RDY_1, RDY_2	I	Bus Ready. These signals are indications from a device located on the system bus that it is available or data has been received. RDY_1 and RDY_2 are qualified by $\overline{AEN}_1$ and $\overline{AEN}_2$ respectively.
15	$\overline{ASYNC}$	I	Ready Synchronous Select. The $\overline{ASYNC}$ signal defines the synchronization mode of the READY logic. When $\overline{ASYNC}$ is open (internal pull-up resistor is provided) or pulled HIGH, there is one stage of READY Synchronization. When $\overline{ASYNC}$ is LOW, there are two stages of READY Synchronization.
5	READY	O	Ready. READY is the synchronized RDY signal input. After the guaranteed hold time to the processor has been met, the READY signal is cleared.
7, 16	X_1,X_2	I	Crystal In. These are the input pins for the attached crystal. The crystal frequency is 3 times the desired process clock frequency.
13	F/$\overline{C}$	I	Frequency/Crystal Select. When F/$\overline{C}$ is strapped HIGH, CLK is generated from the EFI input. When strapped LOW, the F/$\overline{C}$ allows the processor clock to be generated by the crystal.
14	EFI	I	External Frequency. Used in conjunction with a HIGH signal on F/$\overline{C}$, CLK is generated from the input frequency appearing on this pin. The input signal is a square wave 3 times the frequency of the desired CLK output.
8	CLK	O	Processor Clock. CLK is the clock output used by the processor and all devices which directly connect to the processor's local bus (including bipolar support chips and other MOS devices). An output HIGH of 4.5V (V_{CC} = 5V) is provided on this pin to drive MOS devices. The output frequency of CLK is 1/3 of the crystal on EFI input frequency and a 1/3 duty cycle.
2	PCLK	O	Peripheral Clock. This signal is a TTL level peripheral clock signal whose output frequency is 1/2 that of CLK and has a 50% duty cycle.
12	OSC	O	Oscillator Output. This signal is the TTL level output of the internal oscillator circuitry. Its frequency is equal to that of the crystal.
11	$\overline{RES}$	I	Reset In. This signal is used to generate a RESET. The 8284A provides a Schmitt trigger input so that an RC connection can be used to establish the power-up reset of proper duration.
10	RESET	O	Reset. This signal is used to reset the 8086 family processors.
1	CSYNC	I	Clock Synchronization. This signal is designed to allow multiple 8284As to be synchronized to provide clocks that are in phase. CSYNC HIGH will reset the internal counters, when CSYNC goes LOW the counters will resume counting. CSYNC needs to be externally synchronized to EFI. When used with the internal oscillator, CSYNC should be hard wired to ground.

DETAILED DESCRIPTION

OSCILLATOR

The oscillator circuit of the 8284A is designed primarily for use with a fundamental mode, series resonant crystal from which the operating frequency is derived.

The crystal frequency should be selected at three times the required CPU clock. X_1 and X_2 are the two crystal input crystal connections. The output of the oscillator is buffered and brought out on OSC so that other system timing signals can be derived from this stable, crystal-controlled source.

Two 510Ω series resistors are optional for systems which have a V_{CC} ramp time greater than (or equal to) 1V/ms and/or inherent board capacitance between X_1 or X_2 exceeding 10pF. This capacitance value should not include the 8284A's pin capacitance. By limiting the stray capacitance to less than 10pF on X_1 or X_2, the deviation from the desired fundamental frequency is minimized.

CLOCK GENERATOR

The clock generator consists of a synchronous divide-by-three counter with a special clear input that inhibits the counting. This clear input, (CSYNC), allows the output clock to be synchronized with an external event (such as another 8284A clock). It is necessary to synchronize the CSYNC input to the EFI clock external to the 8284A (see Figure 1). This is accomplished with two Schottky flip-flops. The counter output is a 33% duty cycle clock at one-third the input frequency.

The F/$\overline{C}$ input is a strapping pin that selects either the EFI input or the crystal oscillator as the clock for the ÷ 3 counter. If the EFI input is selected as the clock source, the oscillator section can be used independently for another clock source. Output is taken from OSC.

CLOCK OUTPUTS

The CLK output is a 33% duty cycle MOS clock driver designed to drive the 8086 or 8088 processors directly. PCLK is a TTL level peripheral clock signal whose output frequency is ½ that of CLK. PCLK has a 50% duty cycle.

RESET LOGIC

Reset logic for the 8284A is provided by a Schmitt trigger input ($\overline{RES}$) and a synchronizing flip-flop to generate the reset timing.

The reset signal is synchronized to the falling edge of CLK. A simple RC network can be used to provide power-on reset by utilizing this function of the 8284A.

READY SYNCHRONIZATION

Two READY inputs (RDY_1, RDY_2) are provided to accommodate two Multi-Master system busses. Each input has a qualifier ($\overline{AEN}_1$ and $\overline{AEN}_2$, respectively). The $\overline{AEN}$ signals validate their respective RDY signals. If a Multi-Master system is not being used the $\overline{AEN}$ pin should be tied LOW.

To assure RDY setup and hold times are met, synchronization is required for all asynchronous active going edges of either RDY input. Inactive-going edges of RDY (in normally ready systems) do not require synchronization, but must satisfy RDY setup and hold as a matter of proper system design.

The two modes of RDY synchronization operation are defined by the $\overline{ASYNC}$ input.

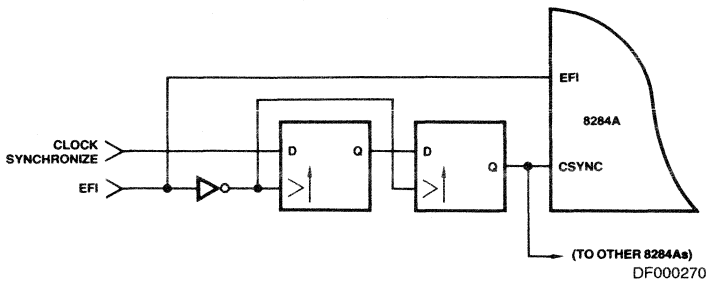
When $\overline{ASYNC}$ is LOW, two stages of synchronization are provided for active RDY input signals. Positive-going asynchronous RDY inputs will first be synchronized to flip-flop one at the rising edge of CLK and then synchronized to flip-flop two at the next falling edge of CLK; after which time the READY output will go active (HIGH). Negative-going asynchro-

nous RDY inputs will be synchronized directly to flip-flop two at the falling edge of CLK, after which time the READY output will go inactive. This mode of operation is intended for use by asynchronous, (normally not ready), devices in the system which cannot be guaranteed by design to meet the required RDY setup timing t_{R1VCL} on each bus cycle.

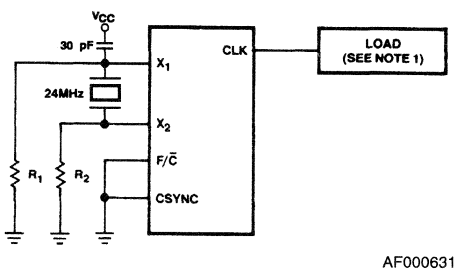
When $\overline{ASYNC}$ is high or left open, the first READY flip-flop is bypassed in the READY synchronization logic. RDY inputs are synchronized by flip-flop two on the falling edge of CLK before they are presented to the processor. This mode is available for synchronous devices that can be guaranteed to meet the required RDY setup time.

$\overline{ASYNC}$ can be changed on every bus cycle to select the appropriate mode of synchronization for each device in the system.

Figure 1. CSYNC Synchronization

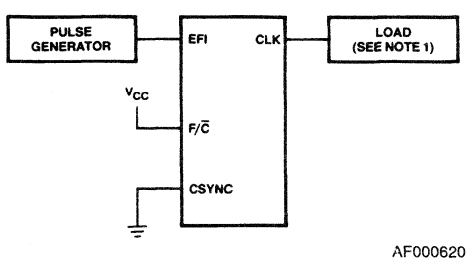


CLOCK HIGH AND LOW TIME (USING X_1, X_2)

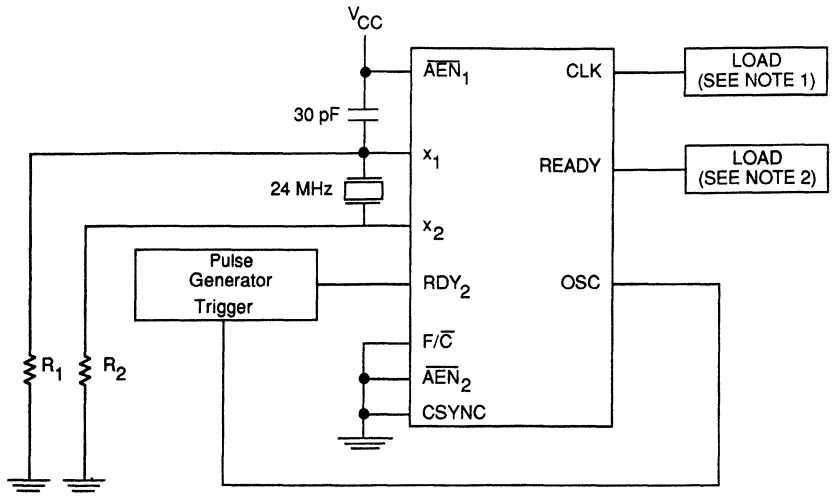


$R_1 = R_2 = 510\Omega$.

CLOCK HIGH AND LOW TIME (USING EFI)

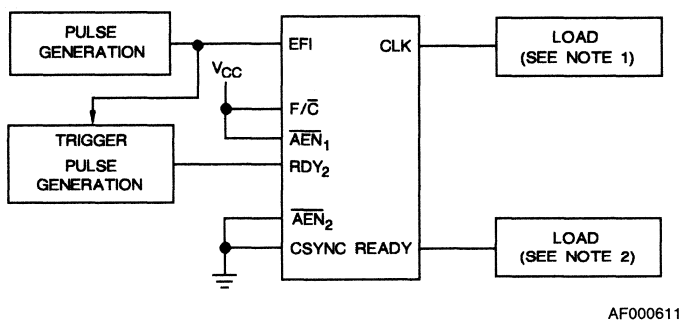


READY TO CLOCK (USING X_1, X_2)



$R_1 = R_2 = 510\Omega$.

READY TO CLOCK (USING EFI)



Notes: 1. $C_L = 100pF$
2. $C_L = 30pF$

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Ambient Temperature with Powers Applied
(COML, A-1) ..... 0°C to +70°C
(MIL) ..... −55°C to +125°C
All Output and Supply Voltages ..... −0.5V to +7.0V
All Input Voltage ..... −1.0V to +5.5V
Power Dissipation 1W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature ..... 0°C to +70°C
Supply Voltage ..... +4.75V to +5.25V

Military (M) Devices
Temperature ..... −55°C to +125°C
Supply Voltage ..... +4.5V to +5.5V

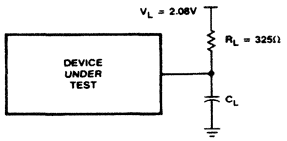
Operating ranges define those limits between which the functionality and parameters of the device are guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
I_F	Forward Input Current ($\overline{ASYNC}$)	$V_F = 0.45V$		−1.3	mA
	Other Inputs	$V_F = 0.45V$		−0.5	
I_R	Reverse Input Current ($\overline{ASYNC}$)	$V_R = V_{CC}$		50	μA
	Other Inputs	$V_R = 5.25V$		50	
V_C	Input Forward Clamp Voltage	$I_C = -5mA$		−1.0	Volts
I_{CC}	Power Supply Current			162	mA
V_{IL}	Input LOW Voltage			0.8	Volts
V_{IH}	Input HIGH Voltage		2.0		Volts
V_{IHR}	Reset Input HIGH Voltage		2.6		Volts
V_{OL}	Output LOW Voltage	5mA		0.45	Volts
V_{OH}	Output HIGH Voltage CLK	−1mA	4.0	2.5	Volts
	Other Outputs	−1mA	2.4		
V_{IHR}–V_{ILR}	$\overline{RES}$ Input Hysteresis (Note 1)		0.25		Volts

Note 1. This specification is provided for reference only.

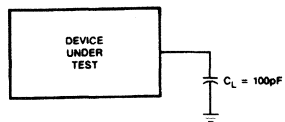
SWITCHING TESTING CIRCUIT (CLK, READY)



TC000670

C_L = 100pF for CLK
C_L = 30pF for READY

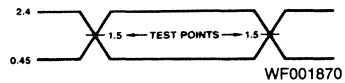
SWITCHING TESTING CIRCUIT (CLK, READY)



TC000680

C_L = 100pF

SWITCHING TESTING WAVEFORM (input, output)



WF001870

AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "O". Timing measurements are made at 1.5V for both a logic "1" and "0".

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified

TIMING REQUIREMENTS

Parameters	Description	Test Conditions	Min	Typ	Max	Units
t_{EHEL}	External Frequency HIGH Time	90% – 90%V_{IN}	13			ns
t_{ELEH}	External Frequency LOW Time	10% – 10%V_{IN}	13			ns
t_{ELEL}	EFI Period	MIL (Note 1)	$t_{EHEL} + t_{ELEH} + \delta$			ns
		COM'L, A-1	33			
	XTAL Frequency		12		25	MHz
t_{R1VCL}	RDY_1, RDY_2 Active Setup to CLK	$\overline{ASYNC}$ = HIGH	35			ns
t_{R1VCH}	RDY_1, RDY_2 Active Setup to CLK	$\overline{ASYNC}$ = LOW	35			ns
t_{R1VCL}	RDY_1, RDY_2 Inactive Setup to CLK		35			ns
t_{CLR1X}	RDY_1, RDY_2 Hold to CLK		0			ns
t_{AYVCL}	$\overline{ASYNC}$ Setup to CLK		50			ns
t_{CLAYX}	$\overline{ASYNC}$ Hold to CLK		0			ns
t_{A1VR1V}	$\overline{AEN_1}$, $\overline{AEN_2}$ Setup to RDY_1, RDY_2		15			ns
t_{CLA1X}	$\overline{AEN_1}$, $\overline{AEN_2}$ Hold to CLK		0			ns
t_{YHEH}	CSYNC Setup to EFI		20			ns
t_{EHYL}	CSYNC Hold to EFI	MIL	20			ns
		COM'L, A-1	10			
t_{YHYL}	CSYNC Width		$2 \cdot t_{ELEL}$			ns
t_{I1HCL}	$\overline{RES}$ Setup to CLK	(Note 2)	65			ns
t_{CLI1H}	$\overline{RES}$ Hold to CLK	(Note 2)	20			ns
t_{ILIH}	Input Rise Time	From 0.8V to 2.0V			20	ns
t_{ILIL}	Input Fall Time	From 2.0V to 0.8V			12	ns

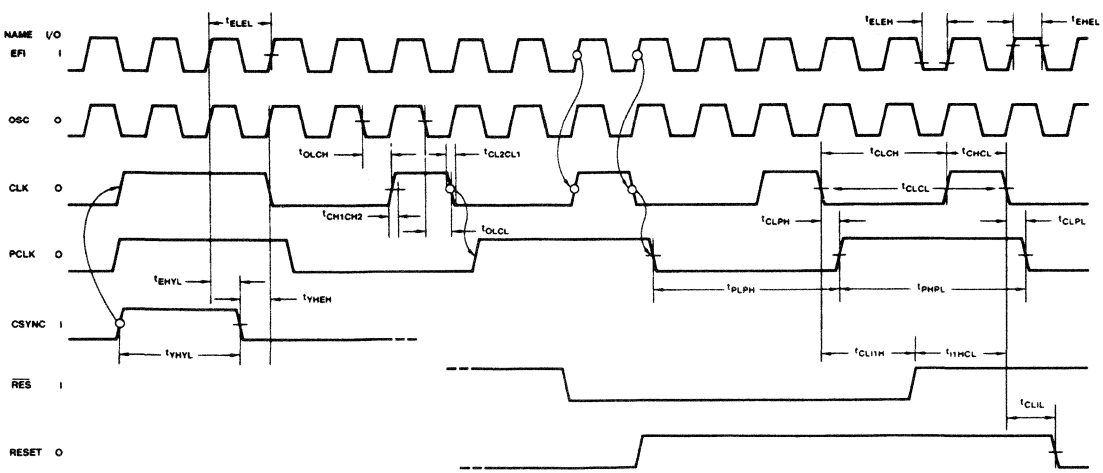
TIMING RESPONSES

Parameters	Description	Test Conditions	Min	Typ	Max	Units
t_{CLCL}	CLK Cycle Period	MIL, COM'L	125			ns
		A-1	100			
t_{CHCL}	CLK HIGH Time	MIL, COM'L	(1/3 t_{CLCL}) + 2			ns
		A-1	39			
t_{CLCH}	CLK LOW Time	MIL, COM'L	(2/3 t_{CLCL}) − 15			ns
		A-1	53			
t_{CH1CH2}	CLK Rise or Fall Time	1.0V to 3.5V			10	ns
t_{CL2CL1}						
t_{PHPL}	PCLK HIGH Time		t_{CLCL} − 20			ns
t_{PLPH}	PCLK LOW Time		t_{CLCL} − 20			ns
t_{RYLCL}	Ready Inactive to CLK (See Note 4)		−8			ns
t_{RYHCH}	Ready Active to CLK (See Note 3)	MIL, COM'L	(2/3 t_{CLCL}) − 15			ns
		A-1	53			
t_{CLIL}	CLK to Reset Delay				40	ns
t_{CLPH}	CLK to PCLK HIGH Delay				22	ns
t_{CLPL}	CLK to PCLK LOW Delay				22	ns
t_{OLCH}	OSC to CLK HIGH Delay		−5		22	ns
t_{OLCL}	OSC to CLK LOW Delay		2		35	ns
t_{OLOH}	Output Rise Time (except CLK)	From 0.8V to 2.0V			20	ns
t_{OHOL}	Output Fall Time (except CLK)	From 2.0V to 0.8V			12	ns

Notes:
1. δ = EFI rise (5ns max) + EFI fall (5ns max).
2. Setup and hold necessary only to guarantee recognition at next clock.
3. Applies only to T_3 and T_W states.
4. Applies only to T_2 states.

SWITCHING WAVEFORMS

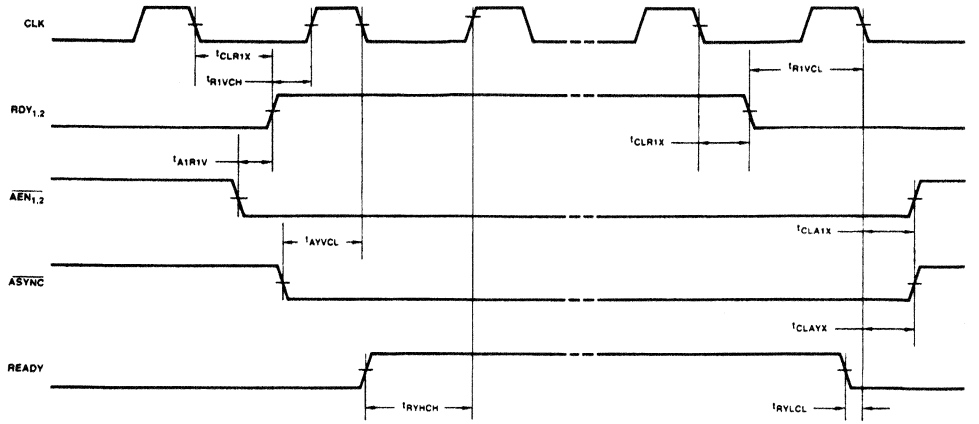
CLOCKS AND RESET SIGNALS



WF002530

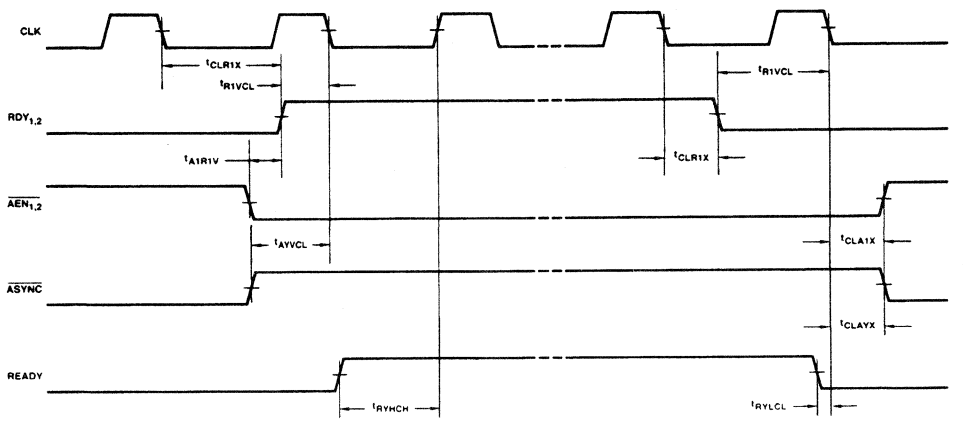
Note: All timing requirements are made at 1.5 volts, unless otherwise noted.

READY SIGNALS (FOR ASYNCHRONOUS DEVICES)



WF002520

READY SIGNALS (FOR SYNCHRONOUS DEVICES)



WF002510

8288

Bus Controller

DISTINCTIVE CHARACTERISTICS

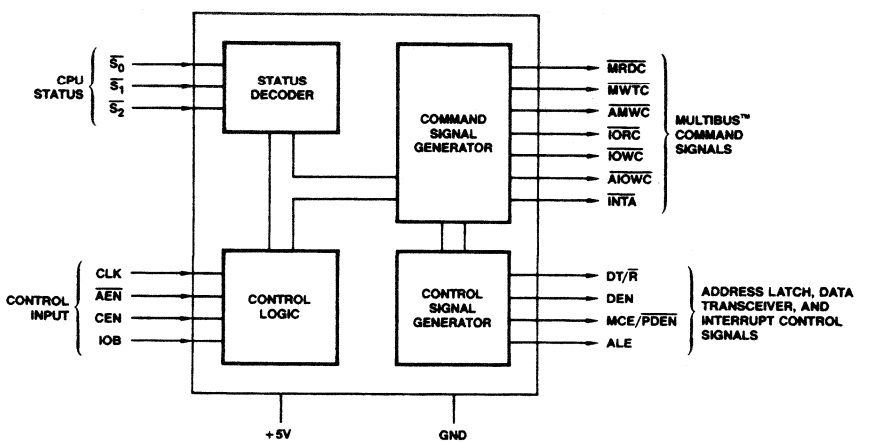
- Bipolar drive capability
- Three-state output drivers
- Multi-master or I/O bus interface
- Flexible system configurations

GENERAL DESCRIPTION

The 8288 optimizes 8086 or 8088 operations by providing command and control timing generation when the CPU is in maximum mode. It provides for highly flexible configurations for larger systems. It also adds powerful bipolar drive capability to the system.

The 8288 is implemented in bipolar technology in a 20-pin plastic or ceramic DIP.

BLOCK DIAGRAM

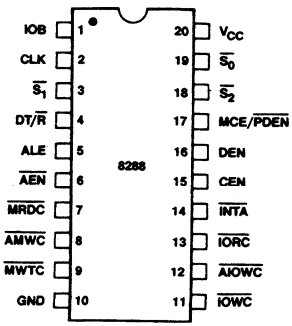


BD001570

Publication # 03358 | Rev. C | Amendment /0
Issue Date: April 1987

CONNECTION DIAGRAM
Top View

DIPs



CD001840

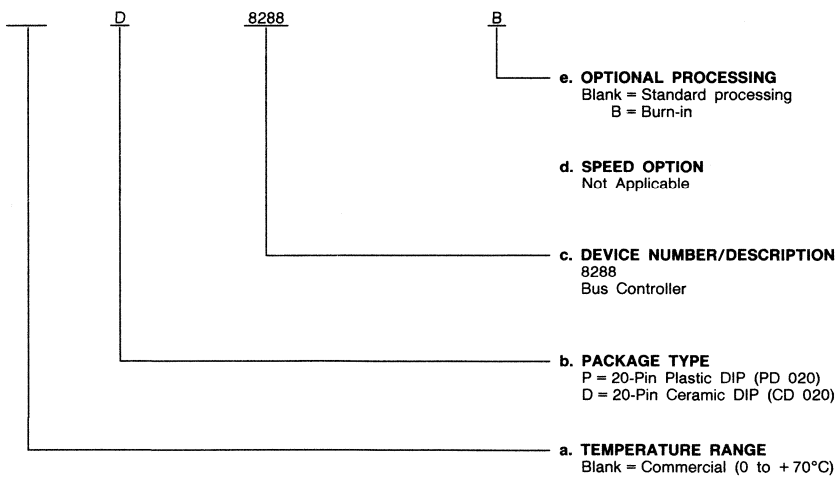
Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Temperature Range
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D	8288
	8288B

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
19, 3, 18	$\overline{S}_0$, $\overline{S}_1$, $\overline{S}_2$	I	Status. These signals are the status input pins from the microprocessor. The 8288 decodes these inputs to generate command and control signals.
2	CLK	I	Clock. Clock signal from the clock generator.
5	ALE	O	Address Latch Enable. This signal strobes an address into the address latches. The latching occurs on the falling edge (HIGH to LOW) transition.
16	DEN	O	Data Enable. This signal enables the data transceivers onto the data bus (local or system).
4	DT/$\overline{R}$	O	Data Transmit/Receive. This signal determines the direction of data flow through the transceivers.
6	$\overline{AEN}$	I	Address Enable. This signal enables the 8288 command outputs at least 115ns after it becomes active LOW. When this pin goes inactive, it 3-states the command output drivers.
15	CEN	I	Command Enable. This signal, when LOW, enables all command outputs and the DEN and $\overline{PDEN}$ control outputs are forced to their inactive states.
1	IOB	I	Input/Output Bus Mode. When strapped HIGH, the 8288 functions in the I/O Bus mode. When LOW, the 8288 functions in the System Bus mode.
12	$\overline{AIOWC}$	O	Advanced I/O Write Command. The $\overline{AIOWC}$ gives I/O devices early indication of a write instruction by issuing an I/O Write Command earlier in the machine cycle.
11	$\overline{IOWC}$	O	I/O Write. This signal tells an I/O device to read the data on the data bus.
13	$\overline{IORC}$	O	I/O Read. This signal tells an I/O device to drive its data onto the data bus.
8	$\overline{AMWC}$	O	Advanced Memory Write. The AMWC gives memory devices an early indication of a write instruction by issuing a memory write command earlier in the machine cycle.
9	$\overline{MWTC}$	O	Memory Write. This signal instructs the memory to record the data present on the data bus.
7	$\overline{MRDC}$	O	Memory Read. This signal instructs the memory to drive its data onto the data bus.
14	$\overline{INTA}$	O	Interrupt Acknowledge. This signal informs the interrupting device that its interrupt has been acknowledged and drives the vectoring information onto the data bus.
17	MCE/ $\overline{PDEN}$	O	Master Cascade Enable/ Peripheral Data Enable. Dual Function pin: MCE (IOB LOW): This signal occurs during an interrupt sequence. Its function is to read a Cascade Address from a master Priority Interrupt Controller onto the data bus. $\overline{PDEN}$ (IOB HIGH): This signal enables the data bus transceiver for the I/O Bus during I/O instructions. It performs the same function for the I/O Bus that DEN performs for the system bus.

DETAILED DESCRIPTION

COMMAND AND CONTROL LOGIC

The command logic decodes the three CPU status lines ($\overline{S}_0$, $\overline{S}_1$, $\overline{S}_2$) to determine what command is to be issued.

This chart shows the meaning of each status "word."

$\overline{s}_2$	$\overline{s}_1$	$\overline{s}_0$	Processor State	8288 Command
0	0	1	Interrupt Acknowledge	$\overline{INTA}$
0	0	1	Read I/O Port	$\overline{IORC}$
0	1	0	Write I/O Port	$\overline{IOWC}$, $\overline{AIOWC}$
0	1	1	Halt	None
1	0	0	Code Access	$\overline{MRDC}$
1	0	1	Read Memory	$\overline{MRDC}$
1	1	0	Write Memory	$\overline{MWTC}$, $\overline{AMWC}$
1	1	1	Passive	None

I/O BUS MODE

The 8288 is put into the I/O Bus mode by strapping the IOB pin HIGH. This mode allows one 8288 Bus Controller to handle two external buses. This allows the CPU to access the I/O Bus with no waiting involved. In the I/O Bus Mode, all I/O command lines ($\overline{INTA}$, $\overline{IORC}$, $\overline{IOWC}$, $\overline{AIOWC}$) are always enabled. When the processor initiates an I/O Command, the 8288 immediately activates the command lines using $\overline{PDEN}$ and DT/$\overline{R}$ to control the I/O bus transceiver. There is no arbitration present in this system, so the I/O command lines should not be used to control the system bus. Normal memory access requires a "Bus Ready" signal ($\overline{AEN}$ LOW) before it will proceed. The IOB mode is recommended if I/O or peripherals dedicated to one processor exist in a multiprocessor based system.

SYSTEM BUS MODE

The 8288 is put into the System Bus mode by strapping the IOB pin LOW. This mode is used when only one bus exists. No command is issued until 115ns after the $\overline{AEN}$ line is activated. Bus arbitration is assumed, and this logic will inform the bus controller via the $\overline{AEN}$ line when the bus is free for use. Both I/O commands and memory wait for bus arbitration.

COMMAND OUTPUTS

To prevent the processor from entering unnecessary wait states, the advanced write commands initiate write procedures early in the machine cycle.

The command outputs are:
$\overline{MRDC}$ – Memory Read Command
$\overline{MWTC}$ – Memory Write Command
$\overline{IORC}$ – I/O Read Command
$\overline{IOWC}$ – I/O Write Command
$\overline{AMWC}$ – Advanced Memory Write Command
$\overline{AIOWC}$ – Advanced I/O Write Command
$\overline{INTA}$ – Interrupt Acknowledge

$\overline{INTA}$ (Interrupt Acknowledge) acts as an I/O read during an interrupt cycle. Its purpose is to inform an interrupting device that its interrupt is being acknowledged and that it should place vectoring information onto the data bus.

CONTROL OUTPUTS

The Data Enable (DEN), Data Transmit/$\overline{Receive}$ (DT/$\overline{R}$) and Master Cascade Enable/Peripheral Data Enable (MCE/$\overline{PDEN}$) are the control outputs of the 8288. The DEN signal determines when the external bus should be enabled onto the local bus while the DT/$\overline{R}$ determines the direction of the data transfer. These two signals usually go to the chip select and direction pins of a transceiver.

The MCE/$\overline{PDEN}$ function is determined by the IOB selection. When IOB is HIGH the $\overline{PDEN}$ serves as a dedicated data enable signal for the I/O or Peripheral System Bus.

INTERRUPT ACKNOWLEDGE AND MCE

The MCE signal is used during an interrupt acknowledge cycle if the 8288 is in the System Bus mode (IOB Low). An interrupt sequence consists of two interrupt acknowledge cycles occurring back to back. No data or address transfers take place during the first cycle. Logic should be provided to mask off MCE during this cycle. Just before the second cycle begins the MCE signal gates a master Priority Interrupt Controller's (PIC) cascade address onto the processor's local bus where ALE (Address Latch Enable) strobes it into the address latches. On the leading edge of the second interrupt cycle the addressed slave PIC gates an interrupt vector onto the system data bus where it is read by the processor.

The MCE signal is not used if the system only contains one PIC. If this is the case the second Interrupt Acknowledge signal gates the interrupt vector onto the processor bus.

ADDRESS LATCH ENABLE AND HALT

Address Latch Enable (ALE) occurs during each machine cycle and serves to strobe the current address into the address latches. ALE also serves to strobe the status ($\overline{S}_0$, $\overline{S}_1$, $\overline{S}_2$) into a latch for halt state decoding.

COMMAND ENABLE

The Command Enable (CEN) input acts as a command qualifier for the 8288. If the CEN pin is HIGH the 8288 functions normally. If the CEN pin is pulled LOW, all command lines are held in their inactive state (not 3-state). This feature can be used to implement memory partitioning and to eliminate address conflicts between system bus devices and resident bus devices.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Temperature (Ambient) Under Bias ..... 0°C to +70°C
All Output and Supply Voltages ..... −0.5V to +7.0V
All Input Voltage..... −1.0V to +5.5V
Power Dissipation 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature ..... 0°C to +70°C
Supply Voltage ..... +4.75V to +5.25V

Military (M) Devices
Temperature ..... −55°C to +125°C
Supply Voltage ..... +4.5V to +5.5V

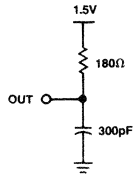
Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
V_C	Input Clamp Voltage	$I_C = -5mA$		−1	V
I_{CC}	Power Supply Current			230	mA
I_F	Forward Input Current	$V_F = 0.45V$		−0.7	mA
I_R	Reverse Input Current	$V_R = V_{CC}$		50	μA
V_{OL}	Output Low Voltage Command Outputs	$I_{OL} = 32mA$		0.5	V
	Control Outputs	$I_{OL} = 16mA$		0.5	V
V_{OH}	Output High Voltage Command Outputs	$I_{OH} = -5mA$	2.4		V
	Control Outputs	$I_{OH} = -1mA$	2.4		V
V_{IL}	Input Low Voltage			0.8	V
V_{IH}	Input High Voltage		2.0		V
I_{OFF}	Three-State Leakage	$V_{OFF} = 0.4$ to $5.25V$		100	μA

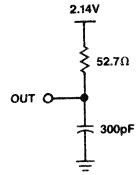
SWITCHING TEST CIRCUIT

3-State to High



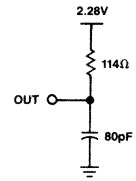
TC000810

Command Output Test Load



TC000800

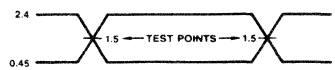
Control Output Test Load



TC000790

SWITCHING TEST INPUT, OUTPUT WAVEFORM

Input/Output



WF001870

AC Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0." The clock is driven at 4.3V and 0.25V. Timing measurements are made at 1.5V for both a logic "1" and "0."

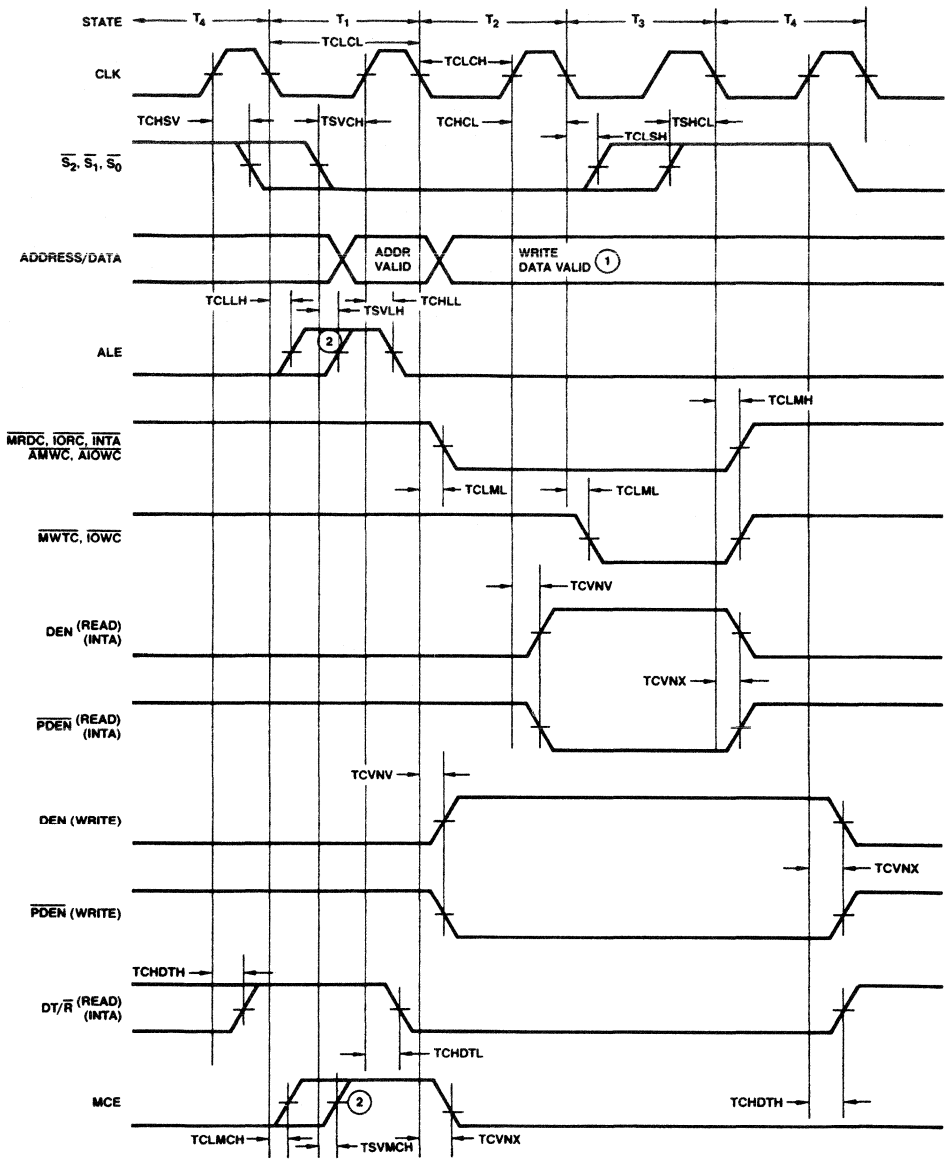
SWITCHING CHARACTERISTICS ($T_A = +25°C$, $V_{CC} = 5.0V$)

Parameters	Description	Test Conditions	Min	Max	Units
Timing Requirements					
TCLCL	CLK Cycle Period		100		ns
TCLCH	CLK Low Time		50		ns
TCHCL	CLK High Time		30		ns
TSVCH	Status Active Setup Time		35		ns
TCHSV	Status Active Hold Time		10		ns
TSHCL	Status Inactive Setup Time		35		ns
TCLSH	Status Inactive Hold Time		10		ns
TILIH	Input Rise Time	From 0.8V to 2.0V		20	ns
TIHIL	Input Fall Time	From 2.0V to 0.8V		12	ns
Timing Responses					
TCVNV	Control Active Delay		5.0	45	ns
TCVNX	Control Inactive Delay		5	45	ns
TCLLH TCMCH	ALE MCE Active Delay (from CLK)			20	ns
TSVLH TSVMCH	ALE MCE Active Delay (from Status)			20	ns
TCHLL	ALE Inactive Delay	$\overline{MRDC}$	4.0	15	ns
TCLML	Command Active Delay	$\overline{IORC}$	7	35	ns
TCLMH	Command Inactive Delay	$\overline{MWTC}$	10	35	ns
TCHDTL	Direction Control Active Delay	$\overline{IOWC}$		50	ns
TCHDTH	Direction Control Inactive Delay	$\overline{INTA}$		30	ns
TAELCH	Command Enable Time	$\overline{AMWC}$		40	ns
TAEHCZ	Command Disable Time	$\overline{AIOWC}$		40	ns
TAELCV	Enable Delay Time	Other	95	200	ns
TAEVNV	AEN to DEN			20	ns
TCEVNV	CEN to DEN, PDEN			25	ns
TCELRH	CEN to Command			TCLML	ns
TOLOH	Output Rise Time	From 0.8V to 2.0V		20	ns
TOHOL	Output Fall Time	From 2.0V to 0.8V		12	ns

For $\overline{MRDC}$, $\overline{IORC}$, $\overline{MWTC}$, $\overline{IOWC}$, $\overline{INTA}$, $\overline{AMWC}$, $\overline{AIOWC}$: $I_{OL} = 32mA$, $I_{OH} = -5mA$, $C_L = 300pF$

For Other: $I_{OL} = 16mA$, $I_{OH} = -1.0mA$, $C_L = 80pF$

SWITCHING WAVEFORMS

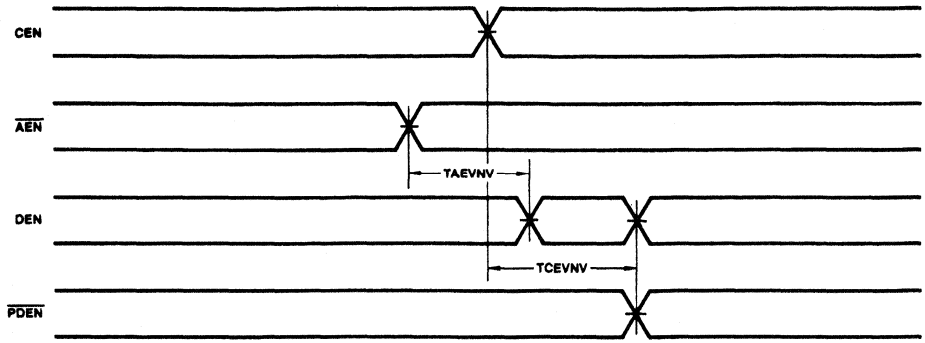


WF002110

Notes: 1. Address/data bus is shown only for reference purposes.
2. Leading edge of ALE and MCE is determined by the falling edge of CLK or status going active, whichever occurs last.
3. All timing measurements are made at 1.5V unless specified otherwise.

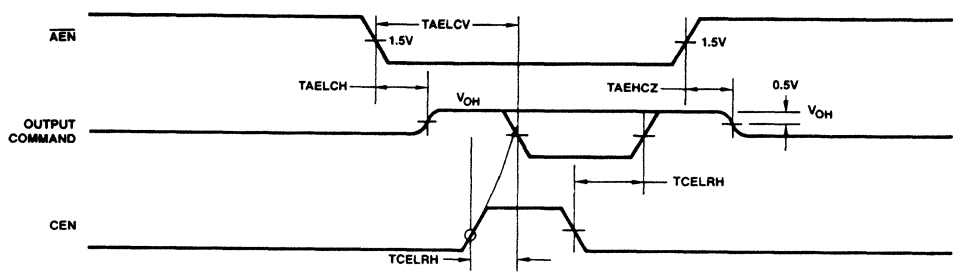
SWITCHING WAVEFORMS (Cont.)

DEN, PDEN QUALIFICATION TIMING



WF002040

ADDRESS ENABLE (AEN) TIMING (3-STATE ENABLE/DISABLE)



WF002050

Note: CEN must be low or valid prior to T_2 to prevent the command from being generated.

SINGLE-CHIP MICROCONTROLLERS

AMD LITERATURE

To obtain literature in the U.S., write or call the AMD Literature Distribution Center, 901 Thompson Place, P.O. Box 3453 — M/S 82, Sunnyvale, CA 94088; (408) 732-2400, TOLL FREE (800) 538-8450. To obtain literature from international locations, contact the nearest AMD sales office or distributor (see listings in the back of this publication).

The 8051 Family

of 8-Bit Single-Chip Microcontrollers

8031AH/8051AH/8053AH
8751H/8753H
80C31BH/80C51BH
80C321/80C521
80535/80515

- Boolean Processor
- Bit-Addressable RAM
- Programmable Full-Duplex Serial Channel
- 64K Bytes Program Memory space
- 64K Bytes Data Memory space
- 32 I/O Lines (Four 8-bit Ports)
  - Six 8-Bit Ports on 80515/535
- Dual Data Pointers on 80C521/321
- 8-Bit A/D Converter on 80515/535
- Security Feature on EPROM Devices
- On-Chip Oscillator/Clock Circuitry

DEVICE	TECH	INTERNAL MEMORY			Timers/ Counters (16-bit)	Watchdog Timer
		ROM (bytes)	EPROM (bytes)	RAM (bytes)		
8753H	NMOS	–	8K	128	2	
8053AH	NMOS	8K	–	128	2	
8751H	NMOS	–	4K	128	2	
8051AH	NMOS	4K	–	128	2	
8031AH	NMOS	–	–	128	2	
80515	NMOS	8K	–	256	3	yes
80535	NMOS	–	–	256	3	yes
80C521	CMOS	8K	–	256	2	yes
80C321	CMOS	–	–	256	2	yes
80C51	CMOS	4K	–	128	2	
80C31	CMOS	–	–	128	2	

INTRODUCTION

The 8051 is a stand-alone, high-performance, single-chip microcontroller intended for use in sophisticated real-time applications such as intelligent computer peripherals, instrumentation and industrial control. It provides the hardware features, architectural enhancements and instructions that make it a powerful and cost-effective controller.

AMD is using the 8051 core as a basic building block, to which we are adding new functions to address the demand for increased integration and performance. The 80C521 has double the on-chip memory around an 80C51 core — 8K bytes of ROM and 256 bytes of RAM. A sophisticated, programmable watchdog timer is provided as a means of graceful recovery from unexpected input conditions, external events and programming anomalies. A second data pointer was also added which facilitates external memory block moves, saving both execution time and code space. The 80C321 is the ROMless version of the 80C521.

The 80515 is an application-specific version with an on-board A/D converter and watchdog timer, which makes it ideal for motor control applications — whether for automotive engines, VCR motors, sewing machines, vending machines, etc. The 80535 is the ROMless version of the 80515.

The 8751 and 8753 are EPROM versions of the 8051 and 8053 respectively. These socket-compatible EPROM versions allow the user to accelerate end product introduction by avoiding the leadtimes associated with ROM devices.

The 8053 is an NMOS device with 8K bytes of on-chip ROM. It is pin-compatible with the 8051 and retains all the 8051 features. It is designed for those 8051 applications where 4K bytes of program memory just aren't enough. The 8753, formerly the 9761, is the 8K byte EPROM version of the 8053.

The 80C31 and 80C51 are CMOS versions of the popular 8031 and 8051. Naturally the 80C31 and 80C51 offer lower power consumption and faster speeds. In addition to the power savings of normal operation, these CMOS devices feature an Idle and a Power-Down Mode to further conserve power. Both the 80C31 and 80C51 are offered in the standard 12 MHz and the faster 16 MHz versions.

AMD's Microcontroller Family — the industry-standard 8051 microcontroller with added functionality, integration and performance. Matching the market needs with AMD's products offering.

8031AH/8051AH/8053AH

Single-Chip 8-Bit Microcontroller

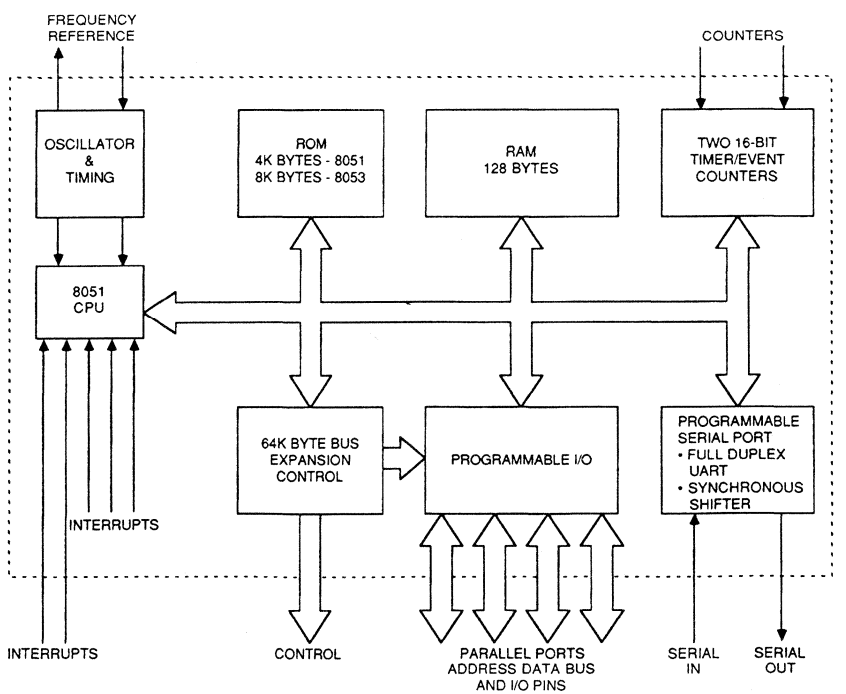
DISTINCTIVE CHARACTERISTICS

- 4K x 8 ROM (8051 only)
- 8K x 8 ROM (8053 only)
- 128 x 8 RAM
- Four 8-bit ports, 32 I/O lines
- Two 16-bit timer/event counters
- 64K addressable Program Memory
- All versions are pin-compatible
- Boolean processor
- Programmable Serial Port
- Five interrupt sources/two priority levels
- On-chip Oscillator/Clock Circuit
- 64K addressable Data Memory

GENERAL DESCRIPTION

The 8051 Family is optimized for control applications. Byte processing and numerical operations on small data structures are facilitated by a variety of fast addressing modes for accessing the internal RAM. The instruction set provides a convenient menu of 8-bit arithmetic instructions, including multiply and divide instructions. Extensive on-chip support is provided for 1-bit variables as a separate data type. This allows direct bit manipulation and testing in control and logic systems that require Boolean processing. Efficient use of program memory results from an instruction set consisting of 44% 1-byte, 41% 2-byte, and 15% 3-byte instructions. With a 12 MHz crystal, 58% of the instructions execute in 1 μs, 40% in 2 μs, and multiply and divide require only 4 μs.

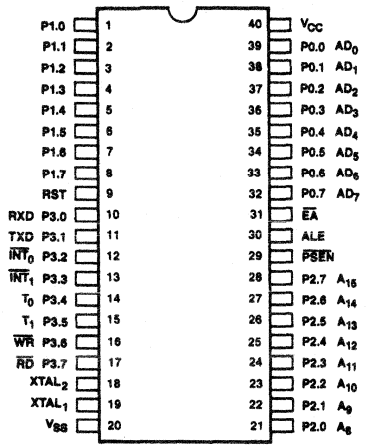
BLOCK DIAGRAM



Publication # 07839 | Rev. B | Amendment /0
Issue Date: June 1987

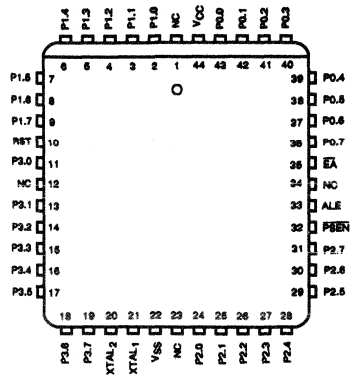
CONNECTION DIAGRAMS
Top View

DIPS



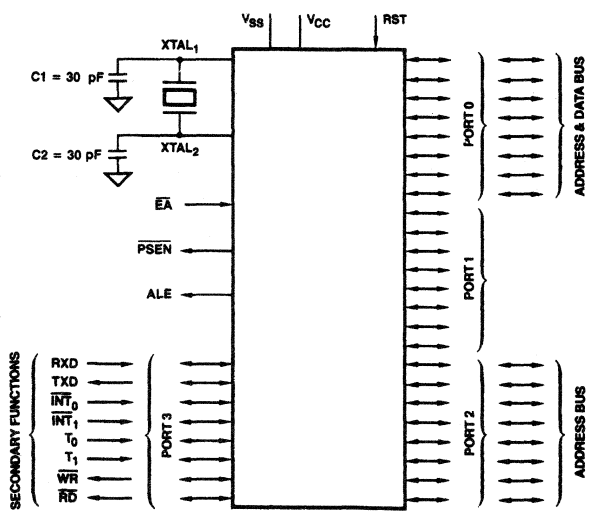
CD005551

PLCC



CD009440

LOGIC SYMBOL



LS001322

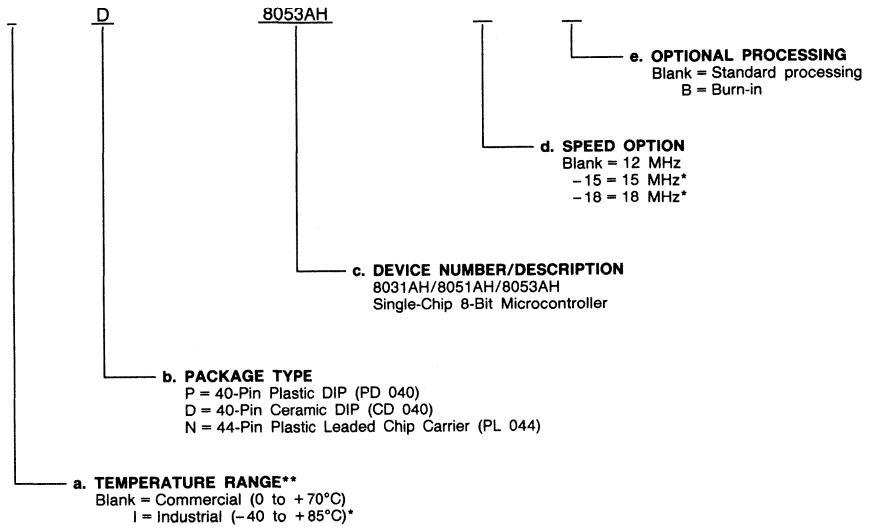
4

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Temperature Range
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*Available only for the 8031AH at time of printing.

**This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order # 09275A/0) for electrical performance characteristics.

Valid Combinations	
P, D, N	8031AH-18
	8031AH-15
	8031AH
	8051AH
	8053AH
ID	8031AHB

PIN DESCRIPTION

Port 0 (Bidirectional, Open Drain)

Port 0 is an open-drain I/O port. As an Output Port, each pin can sink eight LS TTL inputs. Port 0 pins that have "1"s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed LOW-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting "1"s. Port 0 also outputs the code bytes during program verification in the 8051AH and 8053AH. External pullups are required during program verification.

Port 1 (Bidirectional)

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four LS TTL inputs. Port 1 pins that have "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 1 pins that are externally being pulled LOW will source current (I_{IL} on the data sheet) because of the internal pullups.

Port 1 also receives the LOW-order address bytes during program verification.

Port 2 (Bidirectional)

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four LS TTL inputs. Port 2 pins having "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 2 pins externally being pulled LOW will source current (I_{IL}) because of the internal pullups.

Port 2 emits the HIGH-order address byte during fetches from external Porgram Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting "1"s. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function register.

Port 2 also receives the HIGH-order address bits during ROM verification.

Port 3 (Bidirectional)

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four LS TTL inputs. Port 3 pins having "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 3 pins externally being pulled LOW will source current (I_{IL}) because of pullups.

Port 3 also serves the functions of various special features as listed below:

Port Pin	Alternate Function
$P_{3.0}$	RxD (Serial Input Port)
$P_{3.1}$	TxD (Serial Output Port)
$P_{3.2}$	$\overline{INT_0}$ (External Interrupt 0)
$P_{3.3}$	$\overline{INT_1}$ (External Interrupt 1)
$P_{3.4}$	T_0 (Timer 0 External Input)
$P_{3.5}$	T_1 (Timer 1 External Input)
$P_{3.6}$	$\overline{WR}$ (External Data Memory Write Strobe)
$P_{3.7}$	$\overline{RD}$ (External Data Memory Read Strobe)

RST Reset (Input; Active HIGH)

A HIGH on this pin — for two machine cycles while the oscillator is running — resets the device.

ALE Address Latch Enable (Output; Active HIGH)

Address Latch Enable output pulse for latching the LOW byte of the address during accesses to external memory. ALE can drive eight LS TTL inputs.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, allowing use for external-timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$ Program Store Enable (Output; Active LOW)

$\overline{PSEN}$ is the read strobe to external Program Memory. $\overline{PSEN}$ can drive eight LS TTL inputs. When the device is executing code from an external program memory, $\overline{PSEN}$ is activated twice each machine cycle — except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory. $\overline{PSEN}$ is not activated during fetches from internal Program Memory.

$\overline{EA}$ External Access Enable (Input; Active LOW)

$\overline{EA}$ must be externally held LOW to enable the device to fetch code from external Program Memory locations 0000H to 0FFFH (0000H to 1FFFH in the 8053AH). If $\overline{EA}$ is held HIGH, the 8051AH executes from internal Program Memory unless the program counter contains an address greater than 0FFFH (1FFFH in the 8053AH).

$XTAL_1$ Crystal (Input)

Input to the oscillator's high-gain amplifier. Required when a crystal is used. Connect to V_{SS} when external source is used on $XTAL_2$.

$XTAL_2$ Crystal (Output)

Output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal or external source can be used.

V_{CC} Power Supply

V_{SS} Circuit Ground

FUNCTIONAL DESCRIPTION

The term "8051" shall be used to refer collectively to the 8051AH, 8031AH, and 8053AH.

8051 CPU Architecture

The 8051 CPU manipulates operands in three memory spaces. These are the 64K-byte Program Memory, 64K-byte External Data Memory and 256-byte Internal Data Memory. Of the 64K bytes of Program Memory space, the lower 4K bytes on the 8051AH (addr. 0000H to 0FFFH) and the lower 8K bytes of the 8053AH (addr. 0000H to 1FFFH) may reside on-chip. The Internal Data Memory address space is further divided into the 128-byte Internal Data RAM and 128-byte Special Function Register (SFR) address spaces shown in Figure 1.

Four Register Banks (each with eight registers), 128 addressable bits and the stack reside in the Internal Data RAM. The stack depth is limited only by the available Internal Data RAM and its location is determined by the 8-bit stack pointer. All registers except the four 8-Register Banks reside in the Special Function Register address space. These memory mapped registers include arithmetic registers, pointers, I/O ports, interrupt system registers, timers, and a serial port. Ninety-two bit locations in the SFR address space are addressable as bits. The 8051 contains 128 bytes of Internal Data RAM and 20 SFRs.

The 8051 provides a non-paged Program Memory address space to accommodate relocatable code. Conditional branches are performed relative to the Program Counter. The base-register-plus-index register-indirect jump permits branching relative to a 16-bit base register with an offset provided by an 8-bit index register. Sixteen-bit jumps and calls permit branching to any location in the contiguous 64K Program Memory address space.

The 8051 has five methods for addressing source operands: Register, Direct, Register-Indirect, Immediate, and Base-Register-plus-Index-Register-Indirect Addressing. The first three methods can be used for addressing destination operands. Most instructions have a "destination, source" field that specifies the data type, addressing methods, and operands involved. For operations other than moves, the destination operand is also a source operand.

Registers in the four 8-Register Banks can be accessed through Register, Direct, or Register-Indirect Addressing; the 128 bytes of Internal Data RAM through Direct or Register-Indirect Addressing; and the Special Function Registers through Direct Addressing. External Data Memory is accessed through Register-Indirect Addressing. Look-Up-Tables resident in Program Memory can be accessed through Base-Register-plus-Index-Register-Indirect Addressing.

The 8051 is classified as an 8-bit machine since the internal ROM, RAM, Special Function Registers, Arithmetic Logic Unit, and external data bus are each 8-bits wide. The 8051 performs operations on bit, nibble, byte, and double-byte data types.

The 8051 has extensive facilities for byte transfer, logic, and integer arithmetic operations. It excels at bit handling since data transfer, logic, and conditional branch operations can be performed directly on Boolean variables.

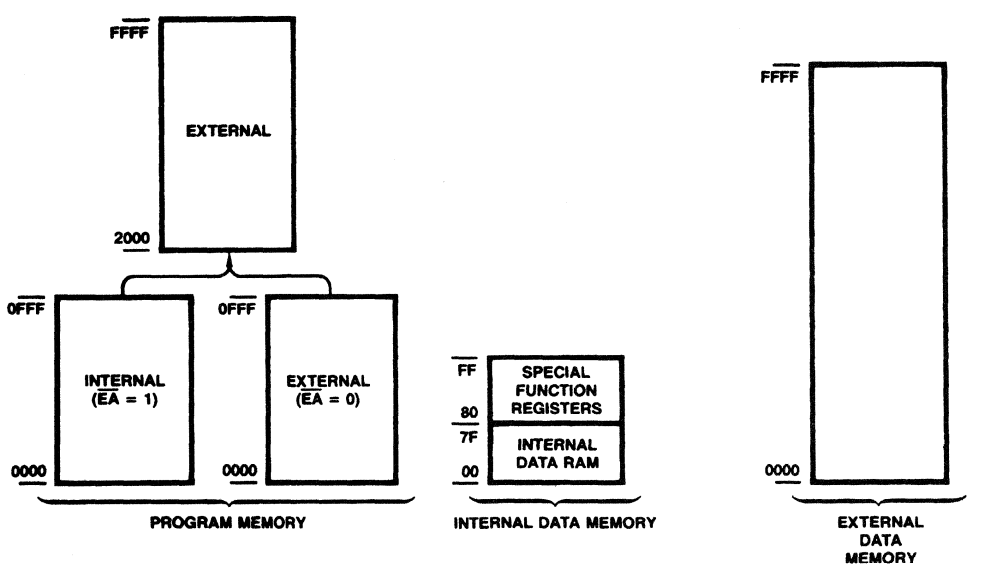


Figure 1. 8051 Memory Organization

Special Function Register Map

Addr (Hex)		Symbol	Name	Default Power-On Reset
80	*	P0	Port 0	11111111
81		SP	Stack Pointer	00000111
82		DPL	Data Pointer Low	00000000
83		DPH	Data Pointer High	00000000
87		PCON	Power Control	0XXX0000
88	*	TCON	Timer/Counter Control	00000000
89		TMOD	Timer/Counter Mode Control	00000000
8A		TLO	Timer/Counter 0 Low Byte	00000000
8B		TL1	Timer/Counter 1 Low Byte	00000000
8C		TH0	Timer/Counter 0 High Byte	00000000
8D		TH1	Timer/Counter 1 High Byte	00000000
90	*	P1	Port 1	11111111
98	*	SCON	Serial Control	00000000
99		SBUF	Serial Data Buffer	Indeterminate
A0	*	P2	Port 2	11111111
A8	*	IE	Interrupt Enable Control	0XX00000
B0	*	P3	Port 3	11111111
B8	*	IP	Interrupt Priority Control	XXX00000
D0	*	PSW	Program Status Word	00000000
E0	*	ACC	Accumulator	00000000
F0	*	B	B Register	00000000

* Bit Addressable

8051 Instruction Set

The 8051AH, 8031AH, and 8053AH share the same instruction set. It allows expansion of on-chip CPU peripherals and optimizes byte efficiency and execution speed. Efficient use of program memory results from an instruction set consisting of 49 single-byte, 45 two-byte,and 17 three-byte instructions. When using a 12-MHz oscillator, 64 instructions execute in 1 μs and 45 instructions execute in 2 μs. The remaining instructions (multiply and divide) execute in only 4 μs. The number of bytes in each instruction and the number of cycles required for execution are listed in Table 1.

On-Chip Peripheral Functions

In addition to the CPU and memories, an interrupt system, extensive I/O facilities, and several peripheral functions are integrated on-chip to relieve the CPU of repetitious, complicated, or time-critical tasks and to permit stringent real-time control of external system interfaces. The extensive I/O facilities include the I/O pins, parallel I/O ports, bidirectional address/data bus, and the serial port for I/O expansion. The CPU peripheral functions intergrated on-chip are the two 16-bit counters and the serial port. All of these work together to boost system performance.

Interrupt System

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution, a sophisticated multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response latency ranges from 3 μs to 7 μs when using a 12 MHz crystal.

The 8051 acknowledges interrupt request from five sources: Two from external sources via the $\overline{INT_0}$ and $\overline{INT_1}$ pins, one from each of the two internal counters and one from the serial I/O port. Each interrupt vectors to a separate location in Program Memory for its service program. Each of the five sources can be assigned to either of two priority levels and can be independently enabled and disabled. Additionally all enabled sources can be globally disabled or enabled. Each external interrupt is programmable as either level- or transition-activated and is active-LOW to allow the "wire or-ing" of several interrupt sources to the input pin. The interrupt system is shown diagrammatically in Figure 2.

I/O Facilities

The 8051 has instructions that treat its 32 I/O lines as 32 individually addressable bits and as four parallel 8-bit ports addressable as Ports 0, 1, 2 and 3. Ports 0, 2, and 3 can also assume other functions. Port 0 provides the multiplexed low-order address and data bus used for expanding the 8051 with standard memories and peripherals. Port 2 provides the high-order address bus when expanding the 8051 with External Program Memory or External Data Memory. The pins of Port 3 can be configured individually to provide external interrupt request inputs, counter inputs, the serial port's receiver input and transmitter output, and to generate the control signals used for reading and writing External Data Memory. The generation or use of an alternate function on a Port 3 pin is done automatically by the 8051 as long as the pin is configured as an input. The configuration of the ports is shown on the 8051 Logic Symbol.

Open-Drain I/O Pins

Each pin of Port 0 can be configured as an open drain output or as a high-impedance input. Resetting the microcomputer programs each pin as an input by writing a one (1) to the pin. If a zero (0) is later written to the pin it becomes configured as an output and will continuously sink current. Rewriting a one (1) to the pin will place its output driver in a high-impedance state and configure the pin as an input. Each I/O pin of Port 0 can sink/source eight LS TTL loads.

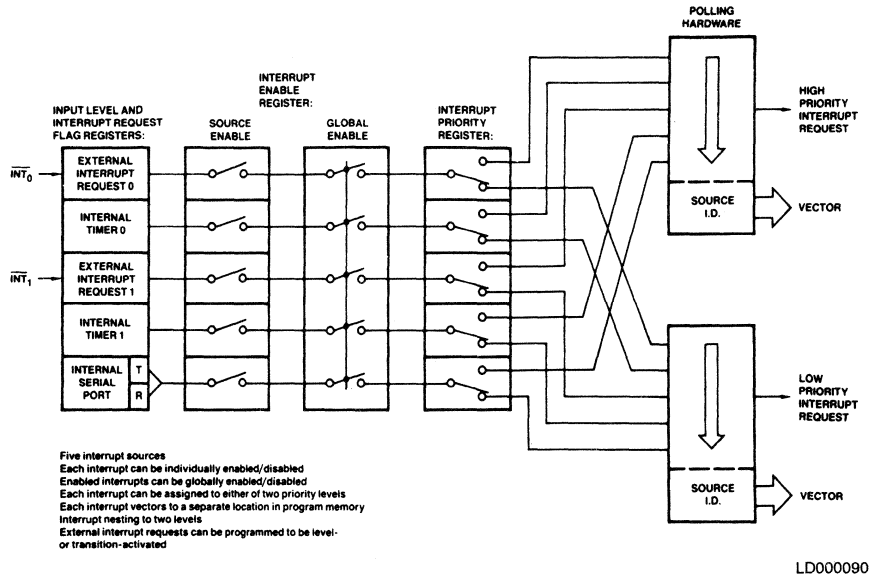


Figure 2. 8051 Interrupt System

Quasi-Bidirectional I/O Pins

Ports 1, 2 and 3 are quasi-bidirectional buffers. Resetting the microcomputer programs each pin as an input by writing a one (1) to the pin. If a zero (0) is later written to the pin it becomes configured as an output and will continuously sink current. Any pin that is configured as an output will be reconfigured as an input when a one (1) is written to the pin. Simultaneous to this reconfiguration, the output driver of the quasi-bidirectional port will source current for two oscillator periods. Since current is sourced only when a bit previously written to a zero (0) is updated to a one (1), a pin programmed as an input will not source current into the TTL gate that is driving it if the pin is later written with another one (1). Since the quasi-bidirectional output driver sources current for only two oscillator periods, an internal pull-up resistor of approximately 20 to 40 kΩ is provided to hold the external driver's loading at a TTL HIGH level. Ports 1, 2, and 3 can sink/source four LS TTL loads.

Microprocessor Bus

When accessing external memory the HIGH-order address is emitted on Port 2 and the LOW-order address on Port 0. The ALE signal is provided for strobing the address into an external latch. The program store enable ($\overline{PSEN}$) signal is provided for enabling an external memory device to Port 0 during a read from the Program Memory address space. When the MOVX instruction is executed, Port 3 automatically generates the read ($\overline{RD}$) signal for enabling an External Data Memory device to Port 0 or generates the write ($\overline{WR}$) signal for strobing the external memory device with the data emitted by Port 0. Port 0 emits the address and data to the external memory through a push/pull driver that can sink/source eight LS TTL loads. At the end of the read/write bus cycle, Port 0 is automatically reprogrammed to its high-impedance state and Port 2 is returned to the state it had prior to the bus cycle. The 8053AH generates the address, data, and control signals needed by memory and I/O devices in a manner that minimizes the requirements placed on external program and data memories.

Timer Event Counters

The 8051 contains two 16-bit counters for measuring time intervals and pulse widths, for counting events, as well as for generating precise, periodic interrupt requests. Each can be programmed independently to one of the following three modes:

Mode 0 – similar to an 8048 8-bit timer or counter with divide by 32 prescaler.

Mode 1 – 16-bit time-interval or event counter.

Mode 2 – 8-bit time-interval or event counter with automatic reload upon overflow.

Additionally, counter 0 can be programmed to a mode that divides it into one 8-bit time-interval or event counter and one 8-bit time-interval counter (Mode 3). When counter 0 is in Mode 3, counter 1 can be programmed to any of the three aforementioned modes, although it cannot set an interrupt request flag or generate an interrupt. This mode is useful because counter 1's overflow can be used to pulse the serial port's transmission-rate generator. Along with their multiple operating modes and 16-bit precision, the counters can also handle very high input frequencies. These range from 0.1 MHz to 1.0 MHz (from 1.2 MHz to 12 MHz crystal) when programmed to increment once every machine cycle and from 0 Hz to an upper limit of 50 kHz to 0.5 MHz (for 1.2 MHz to 12 MHz crystal) when programmed for external inputs. Both internal and external inputs can be gated to the counter by a second external source for directly measuring pulse widths.

The counters are started and stopped under software control. Each counter sets its interrupt request flag when it overflows from all ones to all zeroes (or auto-reload value). The operating modes and input sources are summarized in Figures 3 and 4. The effects of the configuration flags and the status flags are shown in Figures 5 and 6.

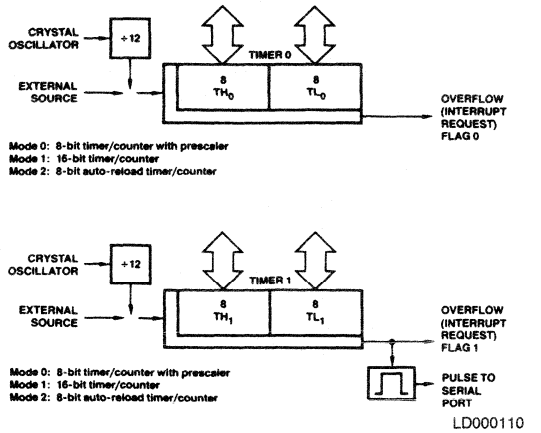


Figure 3. Timer/Event Counter Modes 0, 1 and 2

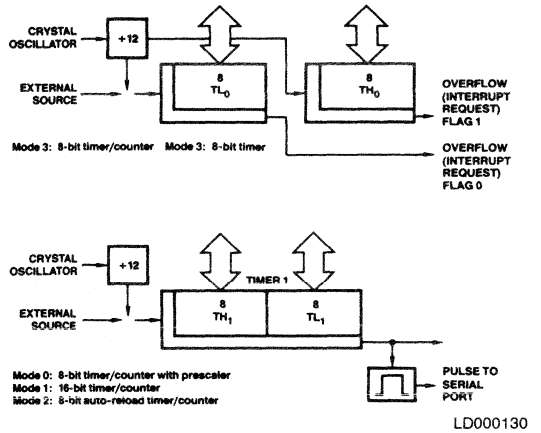


Figure 4. Timer/Event Counter 0 in Mode 3

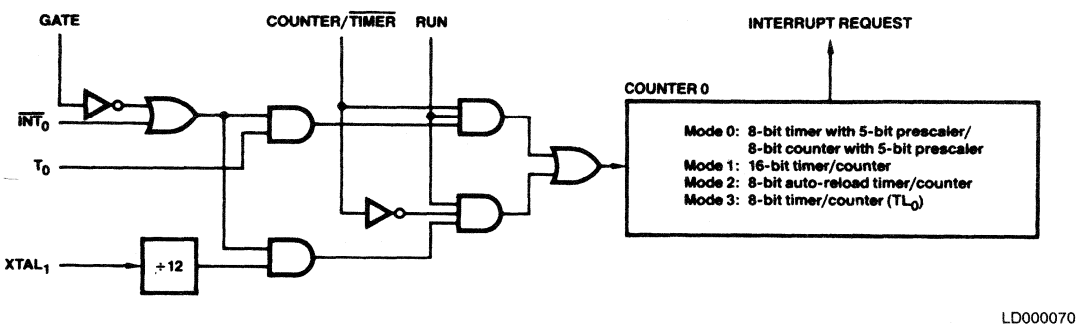


Figure 5. Timer/Counter 0 Control and Status Flag Circuitry

Serial Communications

The 8051's serial I/O port is useful for serially linking peripheral devices as well as multiple 8051s through standard asynchronous protocols with full-duplex operation. The serial port also has a synchronous mode for expansion of I/O lines using CMOS and TTL shift registers. This hardware serial communications interface saves ROM code and permits a much higher tranmission rate than could be achieved through software. In response to a serial port interrupt request, the CPU has only to read/write the serial port's buffer to service the serial link. A block diagram of the serial port is shown in Figures 7 and 8. Methods for linking UART (universal asynchronous receiver/transmitter) devices are shown in Figure 9 and a method for I/O expansion is shown in Figure 10.

The full-duplex serial I/O port provides asynchronous modes to faciliate communications with standard UART devices, such as printers and CRT terminals, or communications with other 8051s in multi-processor systems. The receiver is double buffered to eliminate the overrun that would occur if the CPU failed to respond to the receiver's interrupt before the beginning of the next frame. The 8051 can generally maintain the serial link at its maximum rate so double buffering of the transmitter is not needed. A minor degradation in transmission rate can occur in rare events such as when the servicing of the transmitter has to wait for a lengthy interrupt service program to complete. In asynchronous modes, false start-bit rejection is provided on received frames. For noise rejection a best two-out-of-three vote is taken on three samples near the center of each received bit.

When interfacing with standard UART devices, the serial channel can be programmed to Mode 1 which transmits/receives a ten-bit frame or programmed to Mode 2 or 3 which transmits/receives an eleven-bit frame as shown in Figure 11. The frame consists of a start bit, eight or nine data bits, and one stop bit. In modes 1 and 3, the transmission-rate timing circuitry receives a pulse from counter 1 each time the counter overflows. The input to counter 1 can be an external source or a division by 12 of the oscillator frequency. The auto-reload mode of the counter provides communication rates of 0.05 to 62,500 bits per second (including start and stop bits) for a 12-MHz crystal. In Mode 2 the communication rate is a division by 64 or 32 of the oscillator frequency yielding a transmission rate of 187,500 bits per second or 375,000 bits per second (including start and stop bits) for a 12-MHz crystal.

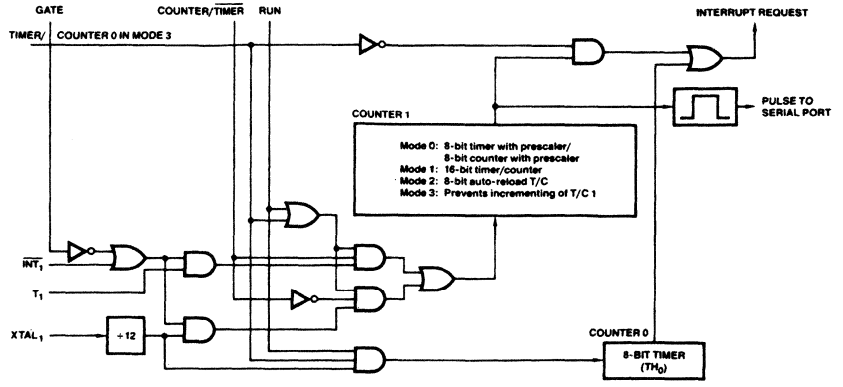


Figure 6. Timer/Counter 1 Control and Status Flag Circuitry

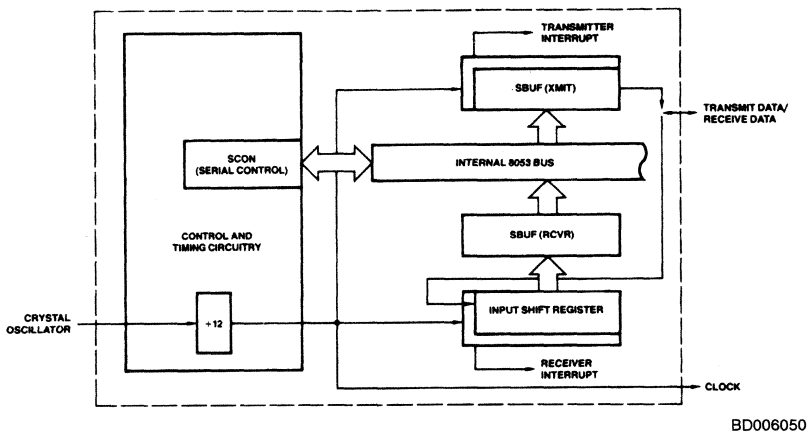


Figure 7. Serial Port — Synchronous Mode 0

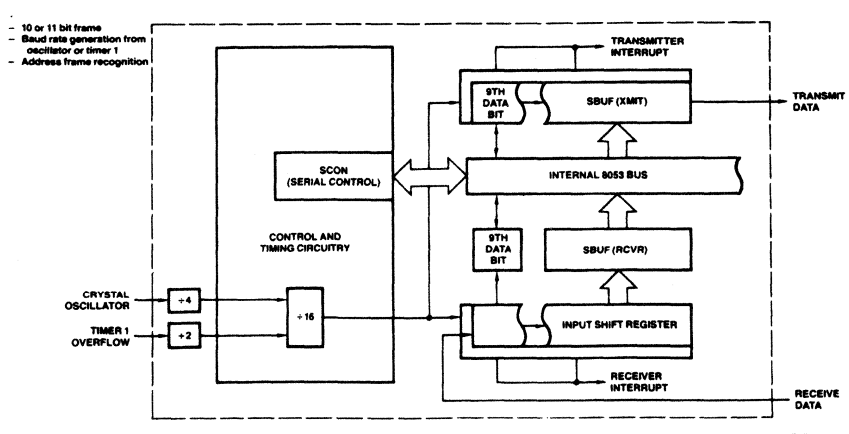


Figure 8. Serial Port — UART Modes 1, 2 and 3

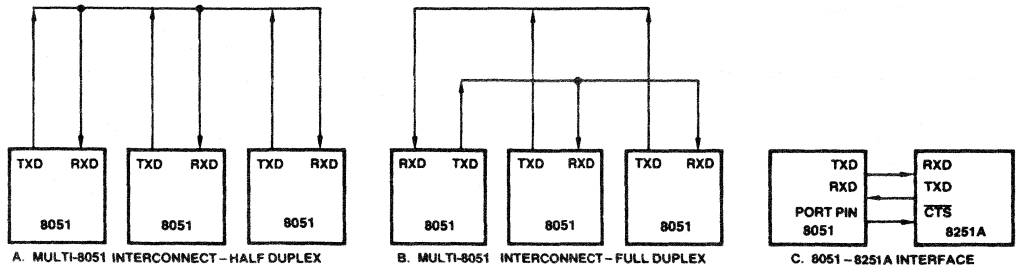


Figure 9. UART Interfacing Schemes

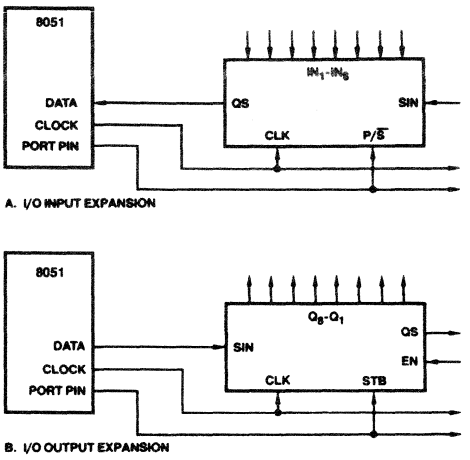


Figure 10. I/O Expansion Technique

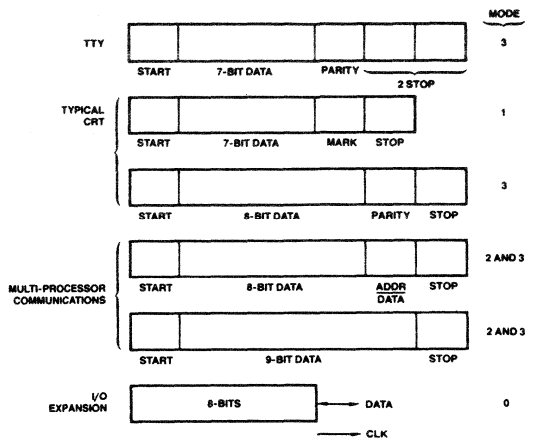


Figure 11. Typical Frame Formats

Distributed processing offers a faster, more powerful system than a single CPU can provide. This results from hierarchy of interconnected processsors, each with its own memories and I/O. In a multiprocessing environment, a single host 8051 controls other slave 8051s configured to operate simultaneously on separate portions of a program. The interconnected 8051s reduce the load on the host processor and result in a lower-cost system of data transmission. This form of distributed processing is especially effective in a complex process where controls are required at physically separated locations.

In Modes 2 and 3 interprocessor communication is facilitated by the automatic wake-up of slave processors through interrupt driven address-frame recognition. The protocol for interprocessor communications is shown in Table 1. In synchronous mode (Mode 0) the high speed serial port provides an efficient, low-cost method of expanding I/O lines using standard TTL and CMOS shift registers. The serial channel provides a clock output for synchronizing the shifting of bits to/from an external register. The data rate is a division by 12 of the oscillator frequency and hence is 1M bits per second at 12 MHz.

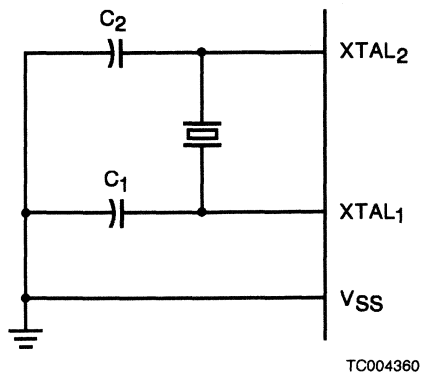
TABLE 1. PROTOCOL FOR MULTI-PROCESSOR COMMUNICATIONS

Slaves	Configure serial port to interrupt CPU if the received ninth data bit is a one (1).
Master	Transmit frame containing address in first 8 data bits and set ninth data bit (i.e., ninth data bit designates address frame).
Slaves	Serial port interrupts CPU when address frame is received. Interrupt service program compares received address to its address. The slave which has been addressed reconfigures its serial port to interrupt the CPU on all subsequent transmissions.
Master	Transmit control frames and data frames (these will be accepted only by the previously addressed slave).

Oscillator Characteristics

$XTAL_1$ and $XTAL_2$ are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 12. Either a quartz crystal or ceramic resonator may be used.

To drive the device from an external clock source, $XTAL_1$ should be grounded, while $XTAL_2$ is driven, as shown in Figure 13. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum HIGH and LOW times specified on the data sheet must be observed.



C_1, C_2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 12. Oscillator Connections

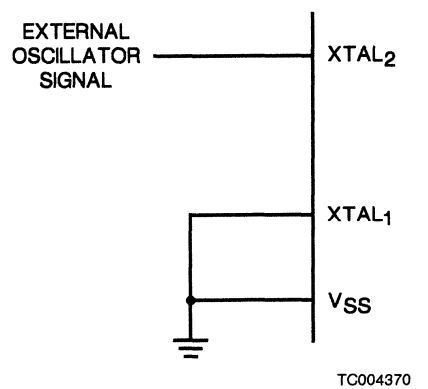


Figure 13. External Drive Configuration

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin
with Respect to Ground ..... −0.5 to +7.0 V
Power Dissipation ..... 1 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.5 to +5.5 V
Ground (V_{SS}) ..... 0 V

Industrial (I) Devices (8031AH only)
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... +4.5 to +5.5 V
Ground (V_{SS}) ..... 0 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameters	Description		Test Conditions	Min.	Max.	Units
V_{IL}	Input LOW Voltage			−0.5	0.8	V
V_{IH}	Input HIGH Voltage (Except RST/V_{PD} and $XTAL_2$)			2.0	V_{CC} +0.5	V
V_{IH1}	Input HIGH Voltage to RST/V_{PD}, $XTAL_2$		$XTAL_1 = V_{SS}$	2.5	V_{CC} +0.5	V
V_{PD}	Power-Down Voltage to RST/V_{PD}		V_{CC} = 0 V	4.5	5.5	V
V_{OL}	Output LOW Voltage, Ports 1, 2, 3 (Note 1)		I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output LOW Voltage, Port 0, ALE, $\overline{PSEN}$ (Note 1)		I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output HIGH Voltage, Ports 1, 2, 3		I_{OH} = −80 μA	2.4		V
V_{OH1}	Output HIGH Voltage, Port 0, ALE, $\overline{PSEN}$		I_{OH} = −400 μA	2.4		V
I_{IL}	Logical 0 Input Current, Ports 1, 2, 3		V_{IL} = 0.45 V		−500	μA
I_{IL2}	Logical 0 Input Current for $XTAL_2$		$XTAL_1 = V_{SS}$ V_{IN} = 0.45 V		−3.2	mA
I_{IH1}	Input HIGH Current to RST/V_{PD} for Reset		$V_{IN} < (V_{CC} - 1.5\ V)$		500	μA
I_{LI}	Input Leakage Current to Port 0, $\overline{EA}$		0.45 < V_{IN} < V_{CC}		±10	μA
I_{CC}	Power-Supply Current	8051AH/8031AH/ 8053AH	$\overline{EA} = V_{CC}$		125	mA
			All Outputs Disconnected		160	
I_{PD}	Power-Down Current		V_{CC} = 0 V; V_{PO} = 5.0 V		10	mA
C_{IO}	Capacitance of I/O Buffer		fc = 1 MHz		10	pF

Notes: 1. Capacitive load on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s 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 0.8 V. 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.

4

See Section 6 for Thermal Characteristics Information.

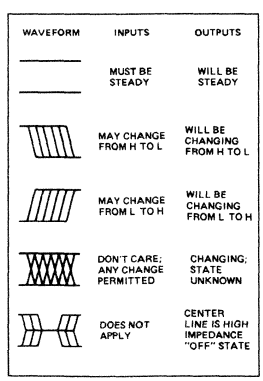
SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified (Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100 pF; Load Capacitance for all other outputs = 80 pF)

Parameter Symbol	Parameter Description	12 MHz Clock		18 MHz Clock (Note 1)		Variable Clock		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
TCY 1/TCLCL	Oscillator Frequency					1.2	18	MHz
TLHLL	ALE Pulse Width	127		71		2TCLCL-40		ns
TAVLL	Address Setup to ALE	43		15		TCLCL-40		ns
TLLAX	Address Hold After ALE	48		20		TCLCL-35		ns
TLLIV	ALE to Valid Instruction In		233		122		4TCLCL-100	ns
TLLPL	ALE to $\overline{PSEN}$	58		30		TCLCL-25		ns
TPLPH	$\overline{PSEN}$ Pulse Width	215		131		3TCLCL-35		ns
TPLIV	$\overline{PSEN}$ to Valid Instruction In		125		41		3TCLCL-125	ns
TPXIX	Input Instruction Hold After $\overline{PSEN}$	0		0		0		ns
TPXIZ	Input Instruction Float After $\overline{PSEN}$		63		35		TCLCL-20	ns
TPXAV	Address Valid After $\overline{PSEN}$	75		47		TCLCL-8		ns
TAVIV	Address to Valid Instruction In		302		162		5TCLCL-115	ns
TPLAZ	Address Float After $\overline{PSEN}$		20		20		20	ns
TRLRH	$\overline{RD}$ Pulse Width	400		233		6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		233		6TCLCL-100		ns
TRLDV	$\overline{RD}$ to Valid Data In		250		112		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		41		2TCLCL-70	ns
TLLDV	ALE to Valid Data In		517		294		8TCLCL-150	ns
TAVDV	Address to Valid Data In		585		334		9TCLCL-165	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	200	300	116	216	3TCLCL-50	3TCLCL+50	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	203		92		4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	23		0		TCLCL-60		ns
TQVWH	Data Setup Before $\overline{WR}$	433		238		7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	33		5		TCLCL-50		ns
TRLAZ	Address Float After $\overline{RD}$		20		20		20	ns
TWHLH	$\overline{WR}$ or $\overline{RD}$ High to ALE High	43	123	16	96	TCLCL-40	TCLCL+40	ns

Notes: 1. 18 MHz clock pertains only to 8031AH in the Commercial operating range.

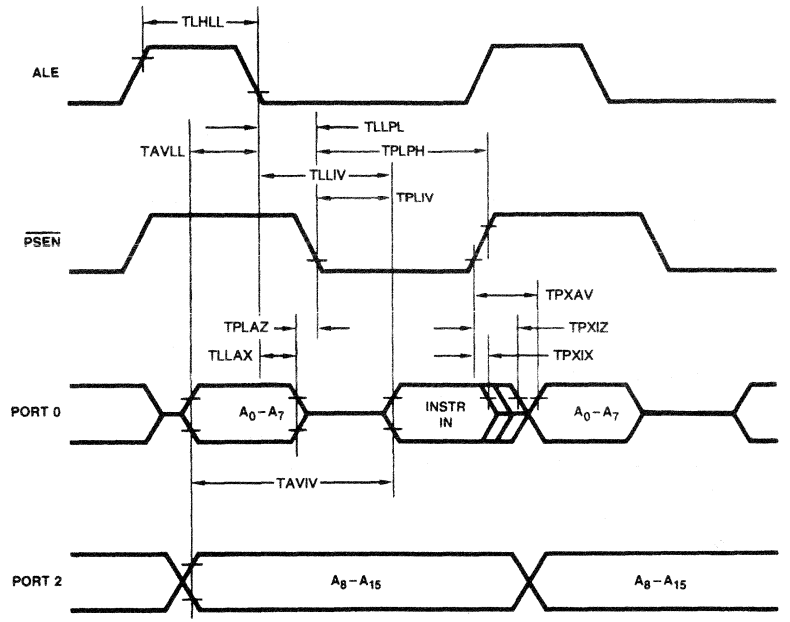
SWITCHING WAVEFORMS

KEY TO SWITCHING WAVEFORMS



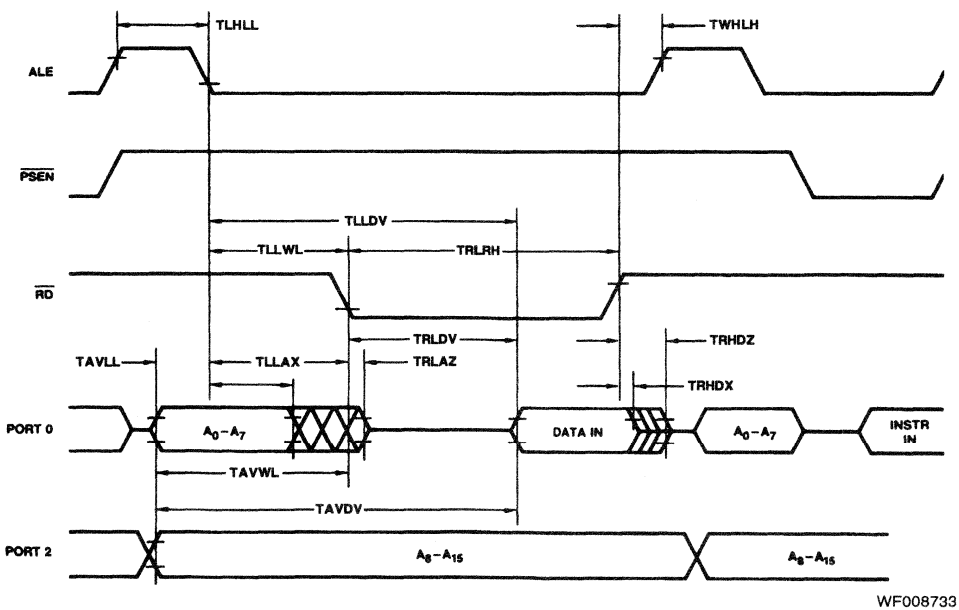
KS000010

SWITCHING WAVEFORMS (Cont'd.)



WF008743

External Program Memory Read Cycle

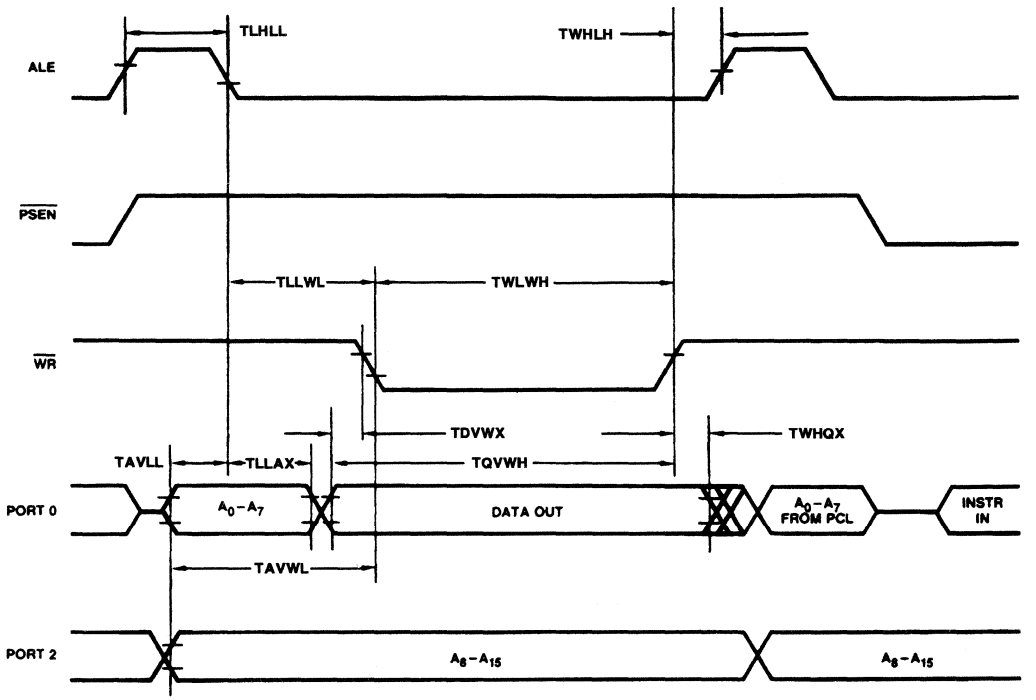


WF008733

External Data Memory Read Cycle

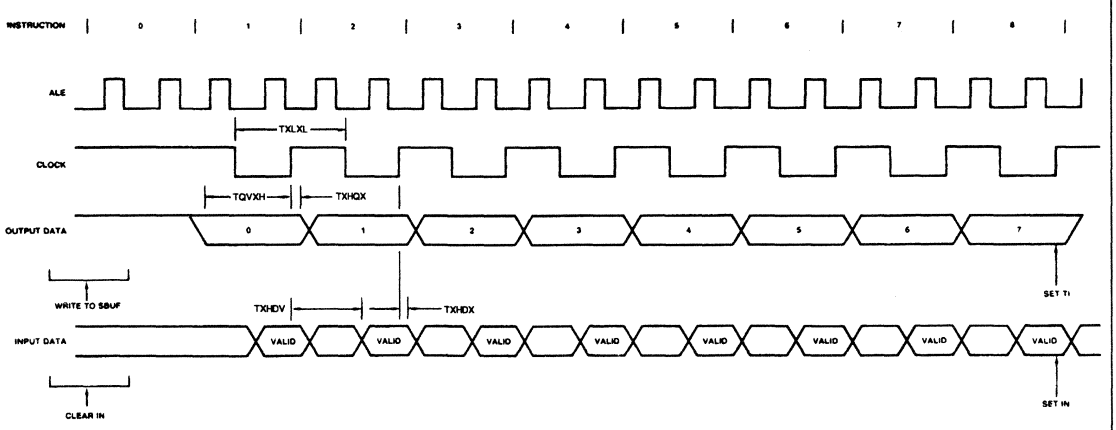
4

SWITCHING WAVEFORMS (Cont'd.)



WF008754

External Data Memory Write Cycle

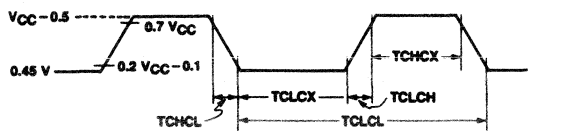


WF008722

Shift Register Timing Waveforms

EXTERNAL CLOCK DRIVE

Parameter Symbol	Parameter Description	Min.	Max.	Units
1/TCLCL	Oscillator Frequency	1.2	12	MHz
TCHCX	HIGH Time	20		ns
TCLCX	LOW Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns



WF020910

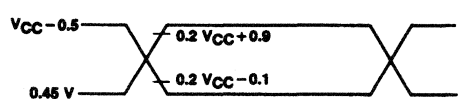
External Clock Drive Waveform

SERIAL PORT TIMING — SHIFT REGISTER MODE

(Load Capacitance = 80 pF)

Parameter Symbol	Parameter Description	12 MHz Osc.		Variable Oscillator		Units
		Min.	Max.	Min.	Max.	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL – 133		ns
TXHQX	Output Data Hold After Clock Rising Edge	50		2TCLCL – 117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL – 133	ns

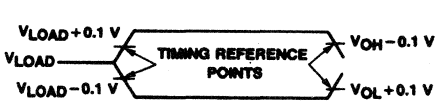
AC Testing



WF020900

AC INPUTS DURING TESTING ARE DRIVEN AT V_{CC} – 0.5 FOR A LOGIC "1" AND 0.45 V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT V_{IH} MIN. FOR A LOGIC "1" AND V_{IL} MAX. FOR A LOGIC "0."

Input/Output Waveform

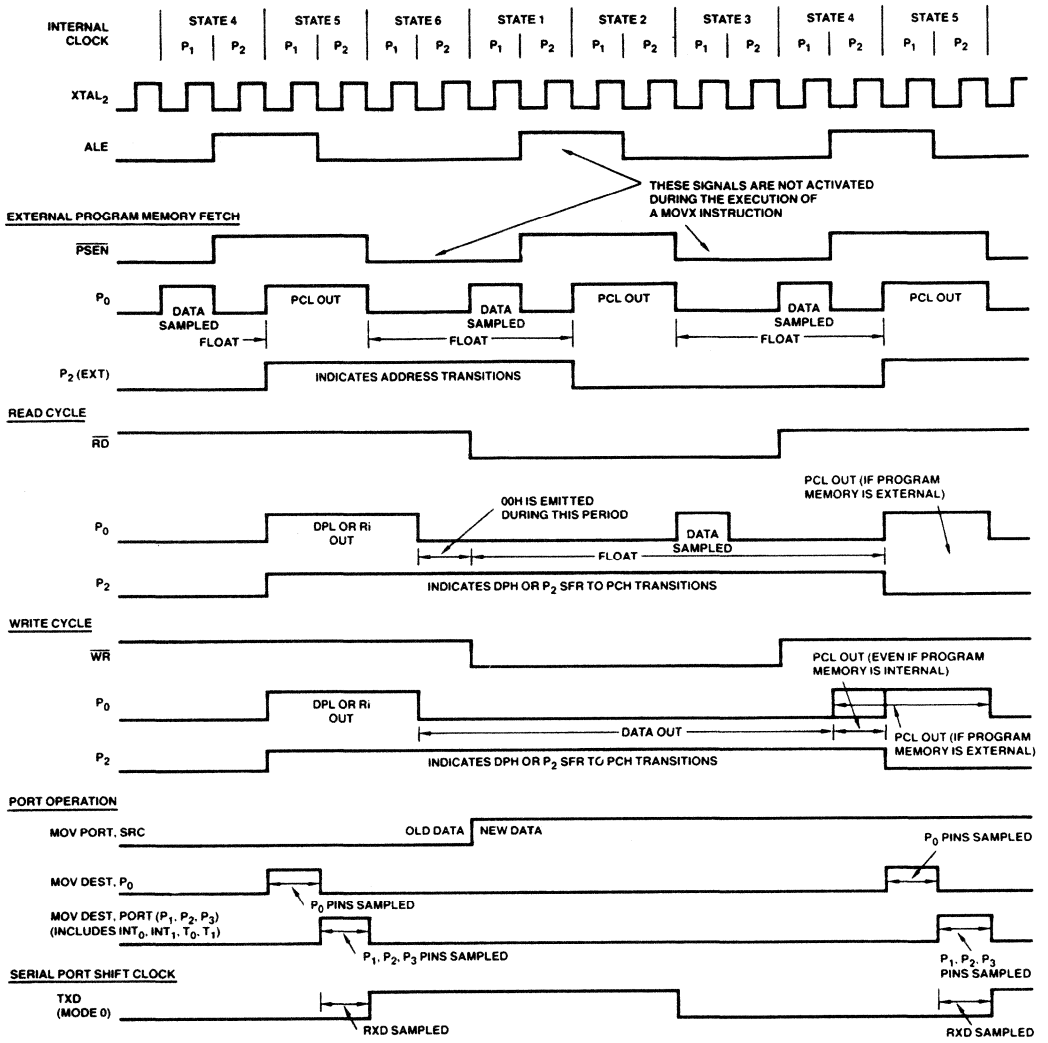


WF020940

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.

Float Waveform

CLOCK WAVEFORMS



WF007070

All internal timing is referenced to the internal time state shown on the top of the page. This waveform represents the signal on the X_2 input of the oscillator. This diagram represents when these signals are actually clocked within the chip. However, the time it takes a signal to propagate to the pins is in the range of 25 to 125 ns. Prop delays are dependent on many variables, such as temperature, pin loading. Propagation also varies from output to output and component to component. Typically though, /RD and /WR have prop delays of approximately 50 ns and the other timing signals approximately 85 ns, at room temperature, fully loaded. These differences in prop delays between signals have been integrated into the timing specs.

TABLE 3. 8051 FAMILY INSTRUCTION SET

Instructions That Affect Flag Setting*

Instruction	Flag			Instruction	Flag		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	O	X		ANL C,/bit	X		
DIV	O	X		ORL C, bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C, bit	X		
RLC	X			CJNE	X		
SETB C	1						

Interrupt Response Time: To finish execution of current instruction, respond to the interrupt request and push the PC; to vector to the first instruction of the interrupt service program requires 38 to 81 oscillator periods (3 to 7 μs @ 12 MHz).

*Note that operations on SFR byte address 208 or bit addresses 209 – 215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

DATA TRANSFER

Mnemonic		Description	Byte	Cyc
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct byte	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Mode direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Move 16-bit constant to Data Pointer	3	2
MOVC	A,@A + DPTR	Move Code byte relative to DPTR to Accumulator	1	2
MOVC	A,@A + PC	Move Code byte relative to PC to Accumulator	1	2
MOVX	A,@Ri	Move External RAM (8-bit address) to Accumulator	1	2
MOVX	A,@DPTR	Move External RAM (16-bit address) to Accumulator	1	2
MOVX	@Ri,A	Move Accumulator to External RAM (8-bit address)	1	2
MOVX	@DPTR,A	Move Accumulator to External RAM (16-bit address)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte off of stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	1
XCHD	A,@Ri	Exchange indirect RAM's least sig nibble with A's LSN	1	1

BOOLEAN VARIABLE MANIPULATION

Mnemonic		Description	Byte	Cyc
CLR	C	Clear Carry Flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry Flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement Carry Flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry Flag	2	2
ANL	C,/bit	AND complement of direct bit to Carry	2	2
ORL	C,bit	OR direct bit to Carry Flag	2	2
ORL	C,/bit	OR complement of direct bit to Carry	2	2
MOV	C,bit	Move direct bit to Carry Flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2

LOGIC (Cont'd.)

Mnemonic		Description	Byte	Cyc
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1

LOGIC (Cont'd.)

Mnemonic		Description	Byte	Cyc
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	1
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate Accumulator Left through Carry Flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate Accumulator Right through Carry Flag	1	1
SWAP	A	Exchange nibbles within the Accumulator	1	1

ARITHMETIC

Mnemonic		Description	Byte	Cyc
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with carry	1	1
ADDC	A,direct	Add direct byte to Accumulator with Carry Flag	2	1
ADDC	A,@Ri	Add indirect RAM and Carry Flag to Accumulator	1	1
ADDC	A,#data	Add immediate data and Carry Flag to Accumulator	2	1
SUBB	A,Rn	Subtract register from Accumulator with Borrow	1	1
SUBB	A,direct	Subtract direct byte from Accumulator with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from Accumulator with Borrow	1	1
SUBB	A,#data	Subtract immediate data from Accumulator with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
MUL	AB	Multiply Accumulator times B	1	4
DIV	AB	Divide Accumulator by B	1	4
DA	A	Decimal Adjust Accumulator	1	1

OTHER

Mnemonic	Description	Byte	Cyc
NOP	No Operation	1	1

CONTROL TRANSFER (BRANCH)

Mnemonic		Description	Byte	Cyc
AJMP	addr11	Absolute Jump	2	2
LJMP	addr16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is zero	2	2
JNZ	rel	Jump if Accumulator is not zero	2	2
JC	rel	Jump if Carry Flag is set	2	2
JNC	rel	Jump if carry is not set	2	2
JB	bit,rel	Jump relative if direct bit is set	3	2
JNB	bit,rel	Jump relative if direct bit is not set	3	2
JBC	bit,rel	Jump relative if direct bit is set, then clear bit	3	2
CJNE	A,direct,rel	Compare direct byte to Accumulator and Jump if not Equal	3	2
CJNE	A,#data,rel	Compare immediate to Accumulator and Jump if not Equal	3	2
CJNE	Rn,#data,rel	Compare immediate to reg and Jump if not Equal	3	2
CJNE	@Ri,#data,rel	Compare immediate to indirect RAM and Jump if not Equal	3	2
DJNZ	Rn,rel	Decrement register and Jump if not zero	2	2
DJNZ	direct,rel	Decrement direct byte and Jump if not zero	3	2

CONTROL TRANSFER (SUBROUTINE)

Mnemonic		Description	Byte	Cyc
ACALL	addr11	Absolute Subroutine Call	2	2
LCALL	addr16	Long Subroutine Call	3	2
RET		Return from Subroutine Call	1	2
RETI		Return from Interrupt Call	1	2

Notes on Data Addressing Modes:

- Rn –Working register R0 – R7 of the currently selected Register bank.
- direct –128 internal RAM locations, any I/O port, control, or status register.
- @Ri –Indirect internal RAM location addressed by register R0 or R1.
- #data –8-bit constant included in instruction.
- #data16 –16-bit constant included as bytes 2 and 3 of instruction.
- bit –128 software flags, any I/O pin, control, or status bit.

Notes on Program Addressing Modes:

- addr16 –Destination address for LCALL and LJMP may be anywhere within the 64-Kilobyte program memory address space.
- addr11 –Destination address for ACALL and AJMP will be within the same 2-Kilobyte page of program memory as the first byte of the following instruction.
- rel –SJMP and all conditional jumps include as 8-bit offset by Range is +127, –128 bytes relative to first byte of the following instruction.

TABLE 4. INSTRUCTION OPCODES IN HEXADECIMAL ORDER (Cont'd.)

Hex Code	Bytes	Mnemonic	Operands	Hex Code	Bytes	Mnemonic	Operands
00	1	NOP		2E	1	ADD	A,R6
01	2	AJMP	Code addr	2F	1	ADD	A,R7
02	3	LJMP	Code addr	30	3	JNB	Bit addr,code addr
03	1	RR	A	31	2	ACALL	Code addr
04	1	INC	A	32	1	RETI	
05	2	INC	Data addr	33	1	RLC	A
06	1	INC	@R0	34	2	ADDC	A,#data
07	1	INC	@R1	35	2	ADDC	A,data addr
08	1	INC	R0	36	1	ADDC	A,@R0
09	1	INC	R1	37	1	ADDC	A,@R1
0A	1	INC	R2	38	1	ADDC	A,R0
0B	1	INC	R3	39	1	ADDC	A,R1
0C	1	INC	R4	3A	1	ADDC	A,R2
0D	1	INC	R5	3B	1	ADDC	A,R3
0E	1	INC	R6	3C	1	ADDC	A,R4
0F	1	INC	R7	3D	1	ADDC	A,R5
10	3	JBC	Bit addr,code addr	3E	1	ADDC	A,R6
11	2	ACALL	Code addr	3F	1	ADDC	A,R7
12	3	LCALL	Code addr	40	2	JC	Code addr
13	1	RRC	A	41	2	AJMP	Code addr
14	1	DEC	A	42	2	ORL	Data addr,A
15	2	DEC	Data addr	43	3	ORL	Data addr,#data
16	1	DEC	@R0	44	2	ORL	A,#data
17	1	DEC	@R1	45	2	ORL	A,data addr
18	1	DEC	R0	46	1	ORL	A,@R0
19	1	DEC	R1	47	1	ORL	A,@R1
1A	1	DEC	R2	48	1	ORL	A,R0
1B	1	DEC	R3	49	1	ORL	A,R1
1C	1	DEC	R4	4A	1	ORL	A,R2
1D	1	DEC	R5	4B	1	ORL	A,R3
1E	1	DEC	R6	4C	1	ORL	A,R4
1F	1	DEC	R7	4D	1	ORL	A,R5
20	3	JB	Bit addr,code addr	4E	1	ORL	A,R6
21	2	AJMP	Code addr	4F	1	ORL	A,R7
22	1	RET		50	2	JNC	Code addr
23	1	RL	A	51	2	ACALL	Code addr
24	2	ADD	A,#data	52	2	ANL	Data addr,A
25	2	ADD	A,data addr	53	3	ANL	Data addr,#data
26	1	ADD	A,@R0	54	2	ANL	A,#data
27	1	ADD	A,@R1	55	2	ANL	A,data addr
28	1	ADD	A,R0	56	1	ANL	A,@R0
29	1	ADD	A,R1	57	1	ANL	A,@R1
2A	1	ADD	A,R2	58	1	ANL	A,R0
2B	1	ADD	A,R3	59	1	ANL	A,R1
2C	1	ADD	A,R4	5A	1	ANL	A,R2
2D	1	ADD	A,R5	5B	1	ANL	A,R3

Hex Code	Bytes	Mnemonic	Operands
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	Code addr
61	2	AJMP	Code addr
62	2	XRL	Data addr,A
63	3	XRL	Data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	Code addr
71	2	ACALL	Code addr
72	2	ORL	C,bit addr
73	1	JMP	@A +DPTR
74	2	MOV	A,#data
75	3	MOV	Data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	Code addr
81	2	AJMP	Code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A +PC
84	1	DIV	AB
85	3	MOV	Data addr,data addr
86	2	MOV	Data addr,@R0
87	2	MOV	Data addr,@R1
88	2	MOV	Data addr,R0
89	2	MOV	Data addr,R1
8A	2	MOV	Data addr,R2
8B	2	MOV	Data addr,R3
8C	2	MOV	Data addr,R4
8D	2	MOV	Data addr,R5
8E	2	MOV	Data addr,R6
8F	2	MOV	Data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	Code addr
92	2	MOV	Bit addr,C
93	1	MOVC	A,@A +DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,/bit addr
A1	2	AJMP	Code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		Reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,/bit addr
B1	2	ACALL	Code addr
B2	2	CPL	Bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	Data addr
C1	2	AJMP	Code addr
C2	2	CLR	Bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	Data addr
D1	2	ACALL	Code addr
D2	2	SETB	Bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	Data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	Code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	Code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	Data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

80C51BH/80C31BH

CMOS Single-Chip Microcontroller

DISTINCTIVE CHARACTERISTICS

- CMOS versions of 8051 and 8031
- 80C51 = 80C31 + 4K bytes ROM
- 128 bytes of RAM
- 32 programmable I/O lines
- CMOS and TTL compatible
- Two 16-bit timer/counters
- Low-power consumption:
  - Normal operation: 16 mA @ 5 V, 12 MHz
  - Idle mode: 3.7 mA @ 5 V, 12 MHz
  - Power-Down mode: 50 μA @ 2 V to 6 V
- 64K bytes Program Memory space
- 64 K bytes Data Memory space
- Boolean processor

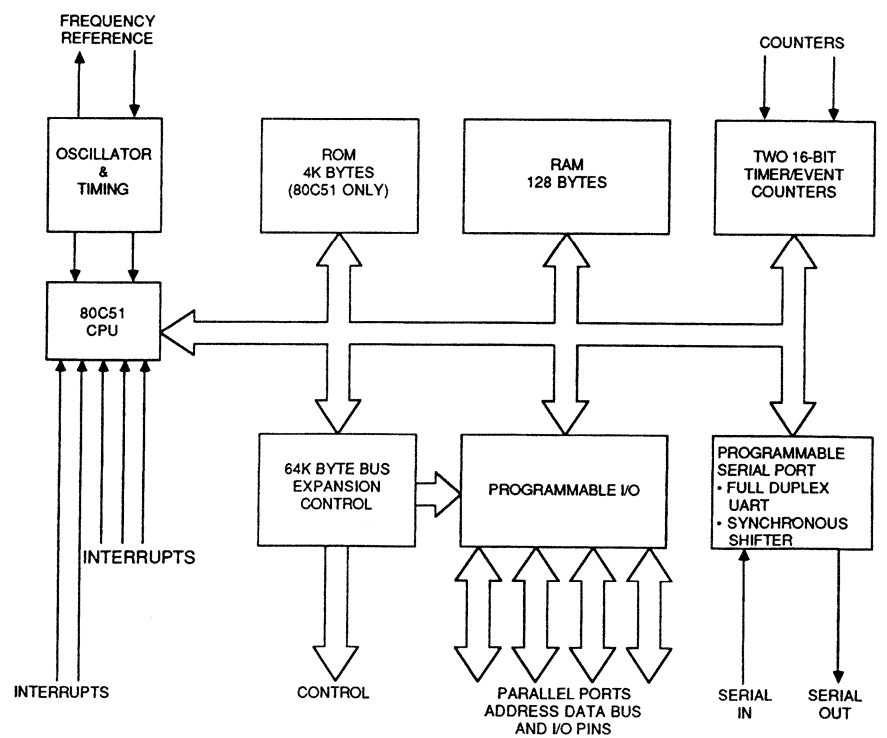
GENERAL DESCRIPTION

The AMD 80C51 and 80C31 are CMOS versions of the 8051 and 8031 8-bit microcontrollers. They combine the power savings of CMOS with the powerful 8051/31 microcontroller.

These CMOS versions retain all the features of their NMOS counterparts: 4K bytes on-chip ROM (80C51 only); 128 bytes RAM; 32 I/O lines; two 16-bit timers; a five-source, two level interrupt structure; a full-duplex serial port; and on-chip oscillator and clock circuits.

In addition, the 80C51/31 has two software-selectable modes of reduced activity for further power conservation — Idle and Power-Down. In the Idle mode, the CPU is frozen while the RAM, timers, serial port, and the interrupt system continue to function. In the Power-Down mode, the RAM is saved and all other functions are inoperative.

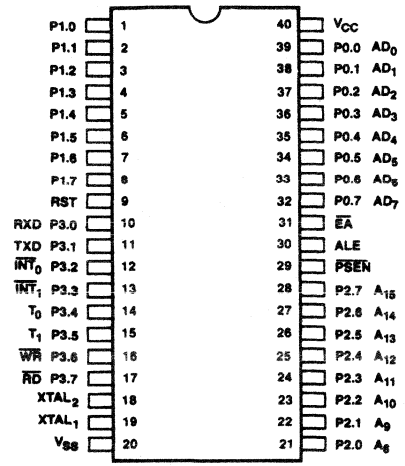
BLOCK DIAGRAM



Publication # 04815 | Rev. C | Amendment /0
Issue Date: June 1987

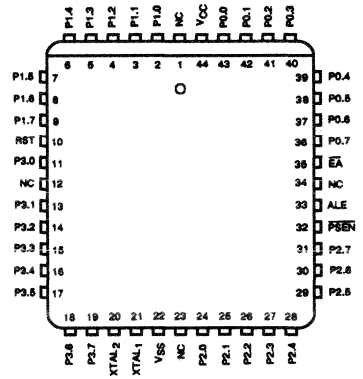
CONNECTION DIAGRAMS
Top View

DIPs



CD005551

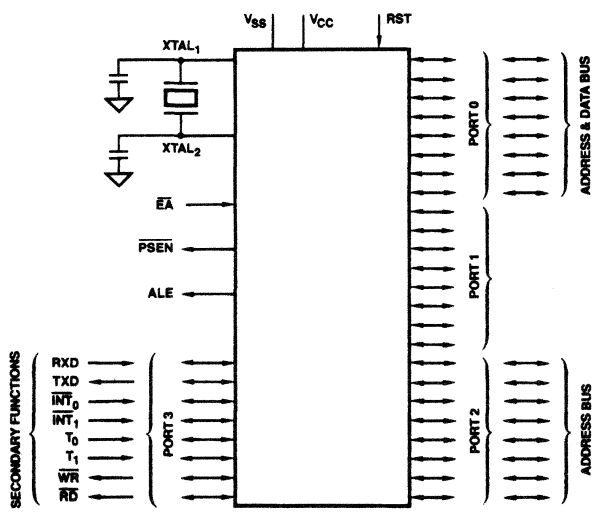
PLCC



CD009440

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL

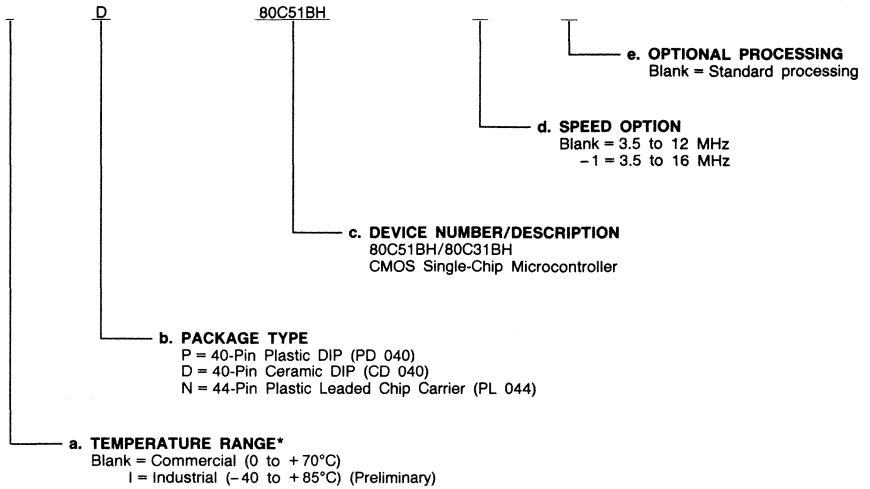


LS001323

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D, N	80C51BH
	80C51BH-1
	80C31BH
	80C31BH-1

*This device will also be available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for preliminary electrical performance characteristics.

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Port 0 (Bidirectional, Open Drain)

Port 0 is an open-drain bidirectional I/O port. Port 0 pins that have "1"s wriiten to them float, and in that state can allow them to be used as high-impedance inputs.

Port 0 is also the multiplexed LOW-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting "1"s. Port 0 also outputs the code bytes during program verification in the 80C51BH. External pullups are required during program verification.

Port 1 (Bidirectional)

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four LS TTL inputs. Port 1 pins that have "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 1 pins that are externally being pulled LOW will source current (I_{IL} on the data sheet) because of the internal pullups.

Port 1 also receives the LOW-order address bytes during program verification.

Port 2 (Bidirectional)

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four LS TTL inputs. Port 2 pins having "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 2 pins externally being pulled LOW will source current (I_{IL}) because of the internal pullups.

Port 2 emits the HIGH-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting "1"s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function register.

Port 2 also receives the HIGH-order address bits during ROM verification.

Port 3 (Bidirectional)

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four LS TTL inputs. Port 3 pins that have "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 3 pins externally being pulled LOW will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features as listed below:

Port Pin	Alternate Function
$P_{3.0}$	RxD (serial input port)
$P_{3.1}$	TxD (serial output port)
$P_{3.2}$	$\overline{INT_0}$ (External interrupt 0)
$P_{3.3}$	$\overline{INT_1}$ (external interrupt 1)
$P_{3.4}$	T_0 (Timer 0 external input)
$P_{3.5}$	T_1 (Timer 1 external input)
$P_{3.6}$	$\overline{WR}$ (external Data Memory write strobe)
$P_{3.7}$	$\overline{RD}$ (external Data Memory read strobe)

RST Reset (Input, Active HIGH)

A HIGH on this pin — for two machine cycles while the oscillator is running — resets the device. An internal diffused resistor to V_{SS} permits power-on reset, using only an external capacitor to V_{CC}.

ALE Address Latch Enable (Output, Active HIGH)

Address Latch Enable output pulse for latching the LOW byte of the address during accesses to external memory.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, allowing use for external-timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$ Program Store Enable (Output, Active LOW)

$\overline{PSEN}$ is the read strobe to external Program Memory. When the 80C51BH is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle — except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory. $\overline{PSEN}$ is not activated during fetches from internal Program Memory.

$\overline{EA}$ External Access Enable (Input, Active LOW)

$\overline{EA}$ must be externally held LOW to enable the device to fetch code from external Program Memory locations 0000H to 0FFFH. If $\overline{EA}$ is held HIGH, the device executes from internal Program Memory unless the program counter contains an address greater than 0FFFH.

$XTAL_1$ Crystal (Input)

Input to the inverting-oscillator amplifier, and input to the internal clock-generator circuits.

$XTAL_2$ Crystal (Output)

Output from the inverting-oscillator amplifier.

V_{CC} Power Supply

Supply voltage during normal, idle, and power-down operations.

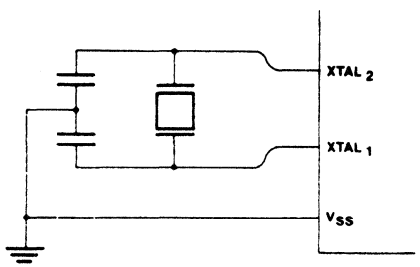
V_{SS} Circuit Ground

FUNCTIONAL DESCRIPTION

Oscillator Characteristics

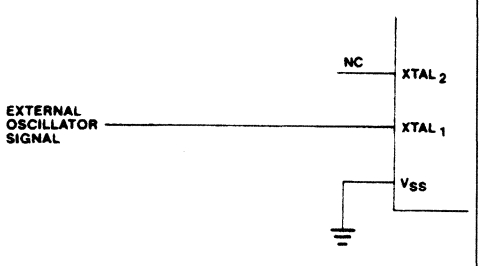
$XTAL_1$ and $XTAL_2$ are the input and output, respectively, of an inverting amplifier which is configured for use as an on-chip oscillator (see Figure 1). Either a quartz crystal or ceramic resonator may be used.

To drive the device from an external clock source, $XTAL_1$ should be driven while $XTAL_2$ is left unconnected (see Figure 2). There are no requirements on the duty cycle of the external-clock signal since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum HIGH and LOW times specified on the data sheet must be observed.



TC003411

Figure 1. Crystal Oscillator



TC003391

Figure 2. External Drive Configuration

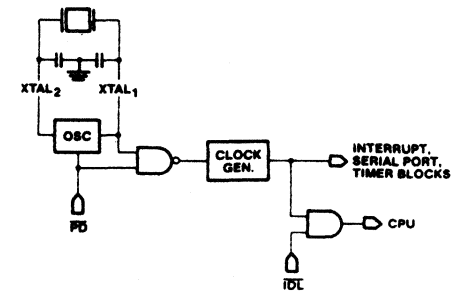
Note: Different from NMOS configuration.

Idle and Power-Down Operation

Figure 3 shows the internal Idle and Power-Down clock configuration. As illustrated, Power-Down operation freezes the oscillator. Idle mode operation shows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is halted.

These special modes are activated by software via the Special Function Register, PCON (Table 1). Its hardware address is 87H; PCON is not bit-addressable.

If "1"s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is "0XXX0000".



TC003381

Figure 3. Idle and Power-Down Hardware

TABLE 1. PCON (Power Control Register)

(MSB)							(LSB)
SMOD	–	–	–	GF1	GF0	PD	IDL

Symbol	Position	Name and Description
SMOD	PCON.7	Double-baud-rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2, or 3.
–	PCON.6	(Reserved)
–	PCON.5	(Reserved)
–	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit
GF0	PCON.2	General-purpose flag bit
PD	PCON.1	Power-Down bit. Setting this bit activates power-down operation.
IDL	PCON.0	Idle-mode bit. Setting this bit activates idle-mode operation.

Idle Mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM, and all other registers maintain their data during Idle. Table 2 describes the status of the external pins during Idle mode.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following RETI, the next instruction to be executed will be the one following the instruction that wrote a 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an enabled interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with a hardware reset. Since the oscillator is still running, the

hardware reset needs to be active for only 2 machine cycles (24 oscillator periods) to complete the reset operation.

Power-Down Mode

The instruction that sets PCON.1 is the last executed prior to going into Power-Down. Once in Power-Down, the oscillator is stopped. Only the contents of the on-chip RAM are preserved. The Special Function Registers are not saved. A hardware reset is the only way of exiting the Power-Down mode.

In the Power-Down mode, V_{CC} may be lowered to minimize circuit power consumption. Care must be taken to ensure the voltage is not reduced until the Power-Down mode is entered, and that the voltage is restored before the hardware reset is applied, which frees the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

Table 2 describes the status of the external pins while in the Power-Down mode. It should be noted that if the Power-Down mode is activated while in external program memory, the port data that is held in the Special Function Register P_2 is restored to Port 2. If the data is a 1, the port pin is held HIGH during the Power-Down mode by the strong pullup, P_1, shown in Figure 4.

80C51BH I/O Ports

The I/O port drive of the 80C51BH is similar to the 8051. The I/O buffers for Ports 1, 2, and 3 are implemented as shown in Figure 4.

When the port latch contains a 0, all pFETS in Figure 4 are off while the nFET is turned on. When the port latch makes a 0-to-1 transition, the nFET turns off. The strong pullup pFET, P_1, turns on for two oscillator periods, pulling the output HIGH very rapidly. As the output line is drawn HIGH, pFET P_3 turns on through the inverter to supply the I_{OH} source current. This inverter and P_3 form a latch which holds the 1 and is supported by P_2.

When Port 2 is used as an address port, for access to external program of data memory, any address bit that contains a 1 will have its strong pullup turned on for the entire duration of the external memory access.

When an I/O pin on Ports 1, 2, or 3 is used as an input, the user should be aware that the external circuit must sink current during the logical 1-to-0 transition. The maximum sink current is specified as I_{TL} under the D.C. Specifications. When the input goes below approximately 2 V, P_3 turns off to save I_{CC} current. Note, when returning to a logical 1, P_2 is the only internal pullup that is on. This will result in a slow rise time if the user's circuit does not force the input line HIGH.

DESIGN CONSIDERATIONS

- At power on, the voltage on V_{CC} and RST must come up at the same time for a proper start-up.
- Before entering the Power Down mode, the contents of the Carry Bit and B.7 must be equal.

TABLE 2. STATUS OF THE EXTERNAL PINS DURING IDLE AND POWER-DOWN MODES

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data
Power-Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power-Down	External	0	0	Floating	Port Data	Port Data	Port Data

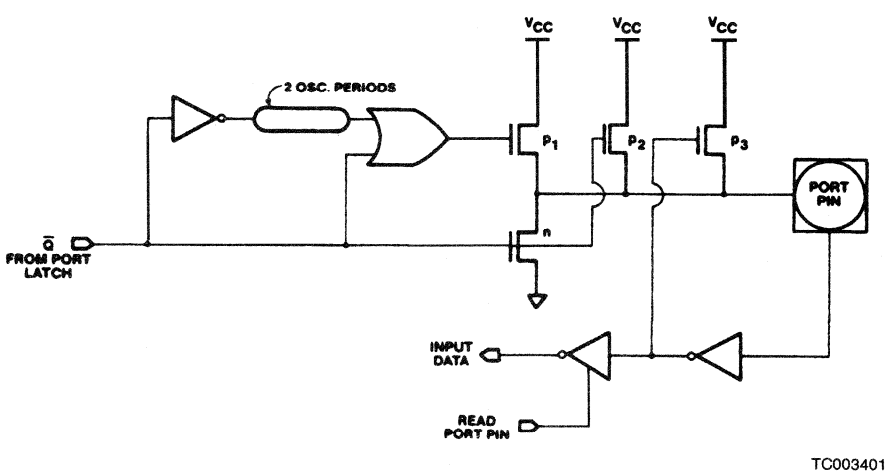


Figure 4. I/O Buffers in the 80C51BH (Ports 1, 2, 3)

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any
Pin to V_{SS} ..... −0.5 V to V_{CC} +0.5 V
Voltage on V_{CC} to V_{SS} ..... −0.5 V to 6.5 V
Power Dissipation ..... 200 mW

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4 V to +6 V
Ground (V_{SS}) ..... 0 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
V_{IL}	Input LOW Voltage (Except $\overline{EA}$)		−0.5	.2 V_{CC} − .1	V
V_{IL1}	Input LOW Voltage ($\overline{EA}$)		−0.5	.2 V_{CC} − .3	V
V_{IH}	Input HIGH Voltage (Except $XTAL_1$, RST)		.2 V_{CC} + .9	V_{CC} + 0.5	V
V_{IH1}	Input HIGH Voltage ($XTAL_1$ RST)		.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output LOW Voltage (Ports 1, 2, 3)	I_{OL} = 1.6 mA (Note 1)		0.45	V
V_{OL1}	Output LOW Voltage (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA (Note 1)		0.45	V
V_{OH}	Output HIGH Voltage (Ports 1, 2, 3)	I_{OH} = −60 μA, V_{CC} = 5 V±10%	2.4		V
		I_{OH} = −25 μA	.75 V_{CC}		V
		I_{OH} = −10 μA	.9 V_{CC}		V
V_{OH1}	Output HIGH Voltage (Port 0 in External Bus Mode, ALE, $\overline{PSEN}$)	I_{OH} = −400 μA, V_{CC} = 5 V±10%	2.4		V
		I_{OH} = −150 μA	.75 V_{CC}		V
		I_{OH} = −40 μA (Note 2)	.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)	V_{IN} = 0.45 V		−50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3)	V_{IN} = 2 V		−650	μA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	μA
RRST	Reset Pulldown Resistor		50	150	kΩ
CIO	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{PD}	Power Down Current	V_{CC} = 2 to 6 V (Note 3)		50	μA

MAXIMUM I_{CC} (mA)

	Operating (Note 4)			Idle (Note 5)		
Freq. / V_{CC}	4 V	5 V	6 V	4 V	5 V	6 V
3.5 MHz	4.3	5.7	7.5	1.1	1.6	2.2
8.0 MHz	8.3	11	14	1.8	2.7	3.7
12 MHz	12	16	20	2.5	3.7	5
16 MHz	16	20.5	25	3.5	5	6.5

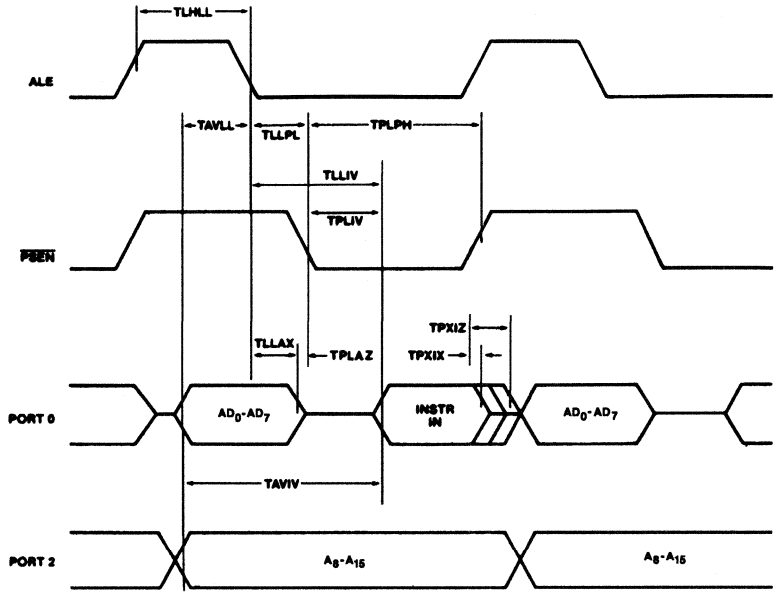
Notes:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OLS} of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8 V. 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 $\overline{PSEN}$ to momentarily fall before the .9 V_{CC} specification when the address bits are stabilizing.
3. Power-Down I_{CC} is measured with all outputs pins disconnected: $\overline{EA}$ = Port 0 = V_{CC}; $XTAL_2$ N.C.; RST = V_{SS}.
4. I_{CC} is measured with all output pins disconnected; $XTAL_1$ driven with TCLCH, TCHCL = 5 ns, V_{IL} = V_{SS} + .5 V, V_{IH} = V_{CC} − .5 V; $XTAL_2$ N.C.; $\overline{EA}$ = RST = Port 0 = 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; $XTAL_1$ driven with TCLCH, TCHCL = 5 ns, V_{IL} = V_{SS} + .5 V, V_{IH} = V_{CC} − .5 V; $XTAL_2$ N.C.; Port 0 = V_{CC}; $\overline{EA}$ = RST = V_{SS}.

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
(C_L for Port 0, ALE and $\overline{PSEN}$ Outputs = 100 pF; C_L for All Other Outputs = 80 pF)

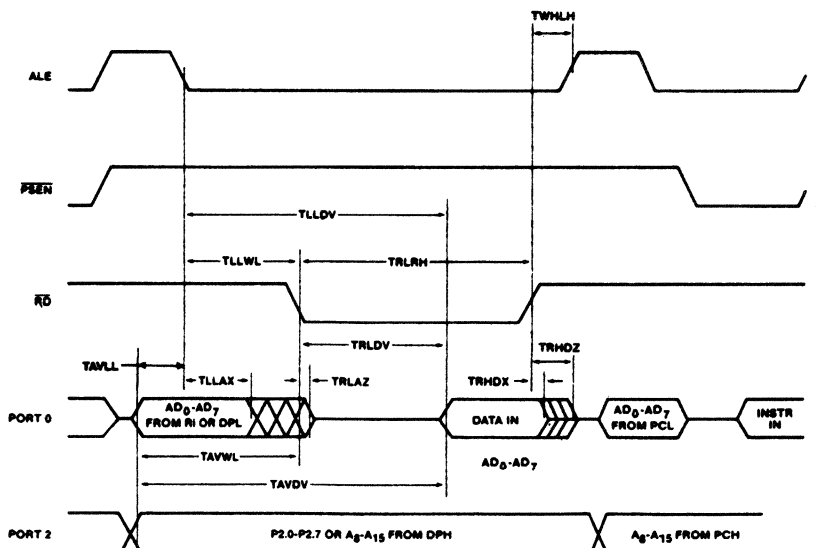
Parameter Symbol	Parameter Description	16 MHz Osc.		12 MHz Osc.		Variable Oscillator		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
External Program and Data Memory Characteristics								
1/TCLCL	Oscillator Frequency					3.5	16	MHz
TLHLL	ALE Pulse Width	85		127		2TCLCL – 40		ns
TAVLL	Address Valid to ALE LOW	7		28		TCLCL – 55		ns
TLLAX	Address Hold After ALE LOW	27		48		TCLCL – 35		ns
TLLIV	ALE LOW to Valid Instr. In		150		234		4TCLCL – 100	ns
TLLPL	ALE LOW to $\overline{PSEN}$ LOW	22		43		TCLCL – 40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	142		205		3TCLCL – 45		ns
TPLIV	$\overline{PSEN}$ LOW to Valid Instr. In		83		145		3TCLCL – 105	ns
TPXIX	Input Instr. Hold After $\overline{PSEN}$	0		0		0		ns
TPXIZ	Input Instr. Float After $\overline{PSEN}$		38		59		TCLCL – 25	ns
TAVIV	Address to Valid Instr. In		208		312		5TCLCL – 105	ns
TPLAZ	$\overline{PSEN}$ LOW to Address Float		10		10		10	ns
TRLRH	$\overline{RD}$ Pulse Width	275		400		6TCLCL – 100		ns
TWLWH	$\overline{WR}$ Pulse Width	275		400		6TCLCL – 100		ns
TRLDV	$\overline{RD}$ LOW to Valid Data In		148		252		5TCLCL – 165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		0		ns
TRHDZ	Data Float After $\overline{RD}$		55		97		2TCLCL – 70	ns
TLLDV	ALE LOW to Valid Data In		350		517		8TCLCL – 150	ns
TAVDV	Address to Valid Data In		398		585		9TCLCL – 165	ns
TLLWL	ALE LOW to $\overline{RD}$ or $\overline{WR}$ LOW	137	238	200	300	3TCLCL – 50	3TCLCL + 50	ns
TAVWL	Address Valid to Read or Write LOW	120		203		4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transiiton	2		23		TCLCL – 60		ns
TQVWH	Data Valid to Write HIGH	287		433		7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	12		33		TCLCL – 50		ns
TRLAZ	$\overline{RD}$ LOW to Address Float		0		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	22	103	43	123	TCLCL – 40	TCLCL + 40	ns

SWITCHING WAVEFORMS



WF021961

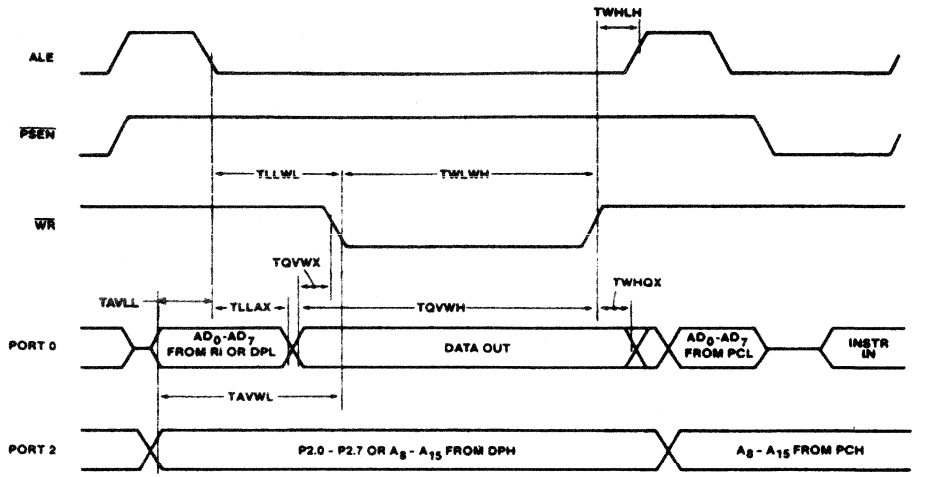
External Program Memory Read Cycle



WF020961

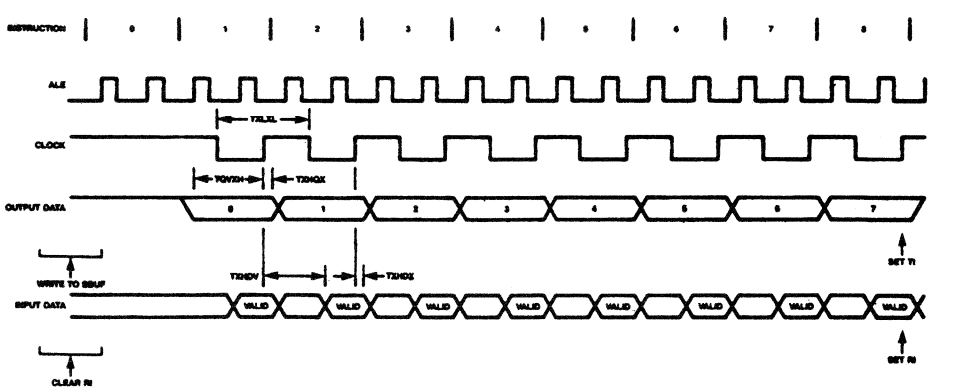
External Data Memory Read Cycle

SWITCHING WAVEFORMS (Cont'd.)



WF020931

External Data Memory Write Cycle

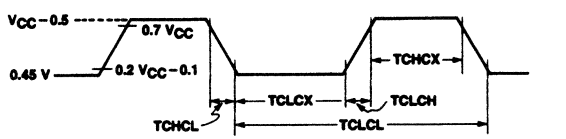


WF020950

Shift Register Timing Waveforms

EXTERNAL CLOCK DRIVE

Parameter Symbol	Parameter Description	Min.	Max.	Units
1/TCLCL	Oscillator Frequency	3.5	16	MHz
TCHCX	HIGH Time	20		ns
TCLCX	LOW Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns



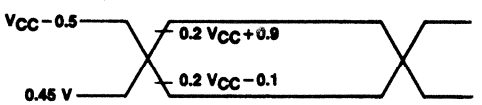
WF020910

External Clock Drive Waveform

SERIAL PORT TIMING — SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to 70°C; V_{CC} = 5 V ±20%; V_{SS} = 0 V; Load Capacitance = 80 pF

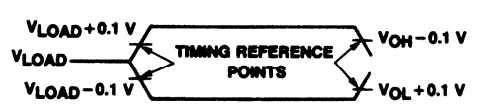
Parameter Symbol	Parameter Description	16 MHz Osc.		Variable Oscillator		Units
		Min.	Max.	Min.	Max.	
TXLXL	Serial Port Clock Cycle Time	750		12TCLCL		ns
TQVXH	Output Data Setup to Clock Rising Edge	492		10TCLCL – 133		ns
TXHQX	Output Data Hold After Clock Rising Edge	8		2TCLCL – 117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		492		10TCLCL – 133	ns



WF020900

AC INPUTS DURING TESTING ARE DRIVEN AT V_{CC} – 0.5 FOR A LOGIC "1" AND 0.45 V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT V_{IH} MIN. FOR A LOGIC "1" AND V_{IL} MAX. FOR A LOGIC "0."

AC Testing Input/Output Waveforms



WF020940

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.

Float Waveform

80515/80535

8-Bit Single-Chip Microcontroller

ADVANCE INFORMATION

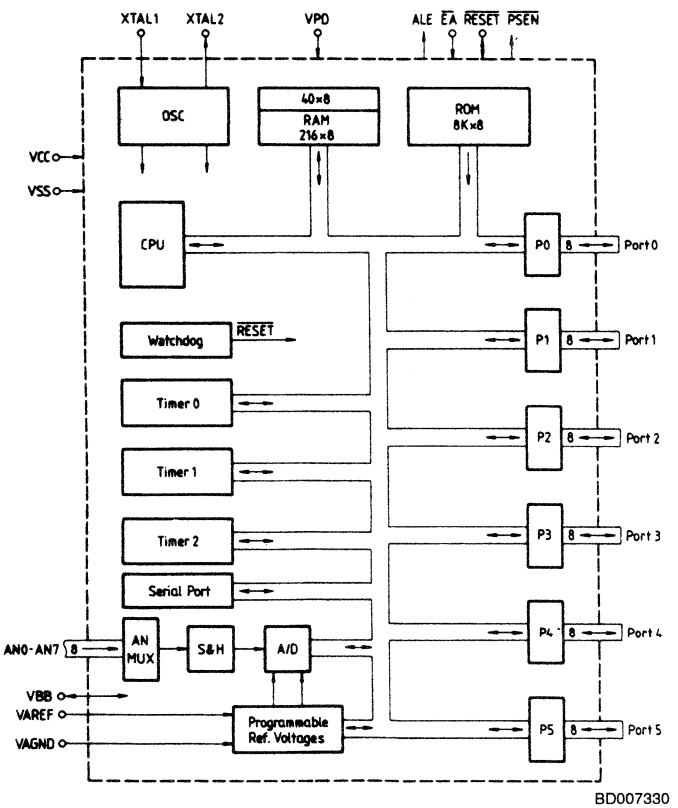
DISTINCTIVE CHARACTERISTICS

- 8K x 8 ROM (80515 only)
- 256 x 8 RAM
- Six 8-bit ports; 48 I/O lines
- Three 16-bit Timer/Event Counters
- Reload, capture, compare capabilities on Timer 2
- Full-Duplex Serial Channel
- Twelve Interrupt Sources; four priority levels
- 8-bit A/D Converter
- Upward-compatible with 8051
- 16-bit Watchdog Timer
- VPD provides standby current for 40 bytes of RAM
- Boolean processor
- 256 bit-addressable locations
- Most instructions execute in 1 μs
- 64K bytes Program Memory space
- 64K bytes Data Memory space

GENERAL DESCRIPTION

The 80515/80535 is a stand-alone, high-performance single-chip microcontroller based on the 8051 architecture. While maintaining all the 8051 operating characteristics, the 80515/80535 incorporates several enhancements which significantly increase design flexibility and overall system performance. With onboard A/D Converter and Watchdog Timer, the 80515 is ideal for motor control applications ranging from automotive engines to vending machines. The 80535 is identical to the 80515 except that it lacks the on-chip ROM.

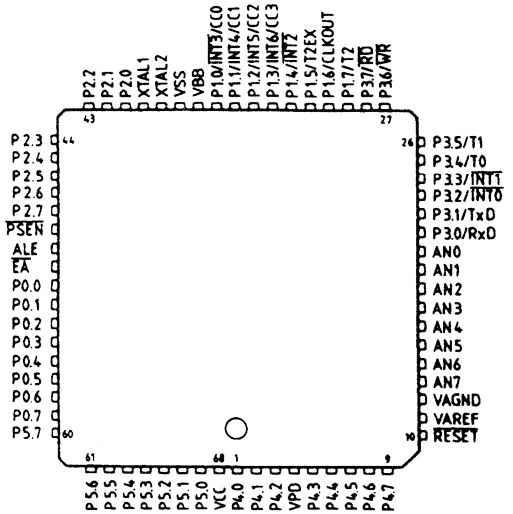
BLOCK DIAGRAM



BD007330

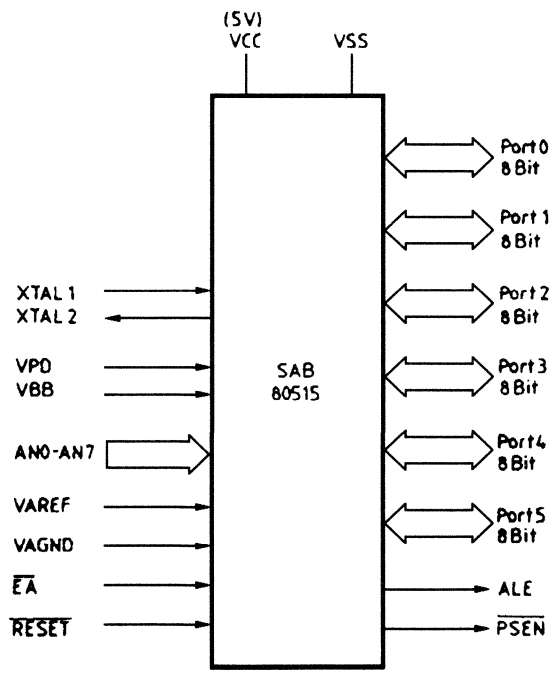
Publication # 09137 | Rev. A | Amendment /0
Issue Date: July 1987

CONNECTION DIAGRAM
Top View



CD010840

LOGIC SYMBOL



LS003051

PIN DESCRIPTION

Port 0 Port 0 (Input/Output; Open Drain)
Port 0 is an open-drain bidirectional I/O port. Port 0 pins that have "1"s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed LOW-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting "1"s. Port 0 can sink/source eight LS TTL inputs. Port 0 also outputs the code bytes during program verification in the 80515. External pullups are required during program verification.

Port 1 Port 1 (Input/Output)
Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 output buffers can sink/source four LS TTL inputs. Port 1 pins that have "1"s written to them are pulled HIGH by the internal pullups and — when in this state — can be used as inputs. As inputs, Port 1 pins that are externally being pulled LOW will source current (I_{IL} on the data sheet) because of the internal pullups. Port 1 also receives the LOW-order address bytes during program verification.

Port 1 also serves the functions of various special features as listed below:

Port	Symbol	Alternate Function
P1.0	$\overline{INT3}$/CC0	External interrupt 3 input, compare 0 output, capture 0 input
P1.1	INT4/CC1	External interrupt 4 input, compare 1 output, capture 1 input
P1.2	INT5/CC2	External interrupt 5 input, compare 2 output, capture 2 input
P1.3	INT6/CC3	External interrupt 6 input, compare 3 output, capture 3 input
P1.4	$\overline{INT2}$	External interrupt 2 input
P1.5	T2EX	Timer 2 external reload trigger input
P1.6	CLKOUT	System clock output
P1.7	T2	Timer 2 external counter input

Port 2 Port 2 (Input/Output)
Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four LS TTL inputs. Port 2 pins having "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 2 pins externally being pulled LOW will source current (I_{IL}) because of the internal pullups.

Port 2 emits the HIGH-order address byte during fetches from External Program Memory and during accesses to External Data Memory that use 16-bit addresses (MOVX @ DPTR). In this application it uses strong internal pullups when emitting "1"s. During accesses to External Data Memory that use 8-bit addresses (MOVX @ Ri), Port 2 emits the contents of the P2 Special Function register.

Port 2 also receives the HIGH-order address bits during ROM verification.

Port 3 Port 3 (Input/Output)
Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four LS TTL inputs. Port 3 pins that have "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 3 pins externally being pulled LOW will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features as listed below:

Port	Symbol	Alternate Function
P3.0	RXD	Serial input port
P3.1	TXD	Serial output port
P3.2	$\overline{INT0}$	External interrupt 0 input, timer 0 gate control
P3.3	$\overline{INT1}$	External interrupt 1 input, timer 1 gate control
P3.4	T0	Timer 0 external counter input
P3.5	T1	Timer 1 external counter input
P3.6	$\overline{WR}$	External Data Memory write strobe
P3.7	$\overline{RD}$	External Data Memory read strobe

Port 4 Port 4 (Input/Output)
Port 4 is an 8-bit quasi-bidirectional I/O port. Port 4 can sink/source four LS-TTL loads.

Port 5 Port 5 (Input/Output)
Port 5 is an 8-bit quasi-bidirectional I/O port. Port 5 can sink/source four LS-TTL loads.

$\overline{RST}$ $\overline{Reset}$ (Input; Active LOW)
A LOW level on this pin for the duration of two machine cycles while the oscillator is running resets the 80515. A small internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.

ALE Address Latch Enable (Output; Active HIGH)
Address Latch Enable output pulse for latching the LOW byte of the address during accesses to external memory.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, allowing use for external-timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$ Program Store Enable (Input; Active LOW)
$\overline{PSEN}$ is the read strobe to External Program Memory. When the 80515 is executing code from External Program Memory, $\overline{PSEN}$ is activated twice each machine cycle — except that two $\overline{PSEN}$ activations are skipped during each access to External Data Memory. $\overline{PSEN}$ is not activated during fetches from Internal Program Memory.

$\overline{EA}$ External Access Enable (Input; Active LOW)
$\overline{EA}$ must be externally held LOW to enable the device to fetch code from external Program Memory locations 0000H to 1FFFH. If $\overline{EA}$ is held HIGH, the device executes from Internal Program Memory unless the program counter contains an address greater than 1FFFH. For the 80535, $\overline{EA}$ must be LOW.

$XTAL_1$ Crystal (Input)
Input to the inverting oscillator amplifier. When an external oscillator is used, $XTAL_1$ should be grounded.

$XTAL_2$ Crystal (Output)
Output of the inverting oscillator amplifier. $XTAL_2$ is also the input for the oscillator signal when using an external oscillator.

V_{CC} Power Supply
Supply voltage during normal operations.

V_{SS} Circuit Ground

V_{PD} Power-Down Supply
If V_{PD} is held within its specs while V_{CC} drops below specs, V_{PD} will provide standby power to 40 bytes of the internal RAM. When V_{PD} is LOW, the RAM's current is drawn from V_{CC}.

V_{AREF} Reference Voltage for the A/D Converter

V_{AGND} Reference Ground for the A/D Converter

AN_0 — AN_7 Multiplexed Analog Inputs

V_{BB} Substrate Pin
Must be connected to V_{SS} through a capacitor (100 to 1000 nF) for proper operation of the A/D converter.

FUNCTIONAL DESCRIPTION

The architecture of the 80515 is based on the 8051 Microcontroller. The following 8051 features are retained in the 80515:

- Instruction set
- External memory expansion interface (Port 0 and Port 2)
- Full-duplex serial port
- Timer/counters 0 and 1
- Alternate functions on Port 3
- The lower 128 bytes of internal RAM and the lower 4 Kbytes of internal ROM.

The 80515 contains an additional 128 byte of internal RAM and 4 Kbyte of internal ROM; thus a total of 256 byte RAM and 8 Kbyte ROM on-chip. The 80515 has a third 16-bit timer/controller with a 2:1 prescaler, reload mode, compare and capture capability. It also contains a 16-bit watchdog timer, an 8-bit A/D converter with 8 analog inputs and programmable reference voltages, two additional quasi-bidirectional 8-bit ports, a programmable clock output (fosc/12), a RAM power-down supply, which supplies 40 byte with a typical current of 1 mA, and a powerful interrupt structure with 12 sources and 4 priority levels.

Figure 2 shows a detailed block diagram of the 80515.

CPU

The 80515 is efficient both as a controller and as an arithmetic processor. It has extensive facilities for binary and BCD arithmetic and excels in bit-handling capabilities. Efficient use of Program Memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 12-MHz crystal, 58% of the instructions execute in 1.0 μs.

Memory Organization

The 80515 manipulates operands in the four memory address spaces described below:

Program Memory

The 80515 has 8 Kbyte of on-chip ROM, while the 80535 has no internal ROM. The Program Memory can be externally expanded up to 64 Kbyte. If the $\overline{EA}$ pin is held HIGH, the 80515 executes out of internal ROM unless the address exceeds 1FFFH. Locations 2000H through FFFFH are then fetched from the External Program Memory. If the $\overline{EA}$ pin is held LOW, the 80515 fetches all instructions from the External Program Memory. Since the 80535 has no internal ROM, pin $\overline{EA}$ must be tied LOW when using this device.

Data Memory

The Data Memory address space consists of an internal and an external memory space. The Internal Data Memory is divided into three physically separate and distinct blocks: the lower 128 byte of RAM; the upper 128 byte of RAM; and the 128-byte special function register (SFR) area. While the upper 128 byte of Data Memory and the SFR area share the same address locations, they are accessed only through different addressing modes. The lower 128 byte of Data Memory can be accessed through direct or register-indirect addressing; the upper 128 byte of RAM can be accessed through register-indirect addressing; and the special function registers are accessible only through direct addressing.

Four 8-register banks occupy locations 0 through 1FH in the lower RAM area. The next 16 bytes, locations 20H through 2FH, contain 128 directly accessible bit locations. The stack can be located anywhere in the Internal Data Memory address space, and the stack depths can be expanded up to 256 byte.

The External Data Memory can be expanded up to 64 Kbyte and can be accessed by instructions that use a 16-bit or 8-bit address.

All registers, except the program counter and the four 8-register banks, reside in the special function register area. The 41 special function registers (SFRs) include arithmetic registers, pointers, and registers that provide an interface between the CPU and the on-chip peripheral functions. There are also 128 directly addressable bits within the SFR area. The special function registers are listed in Table 1.

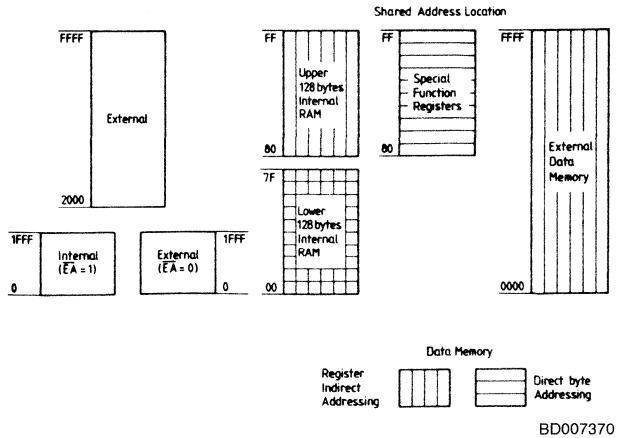
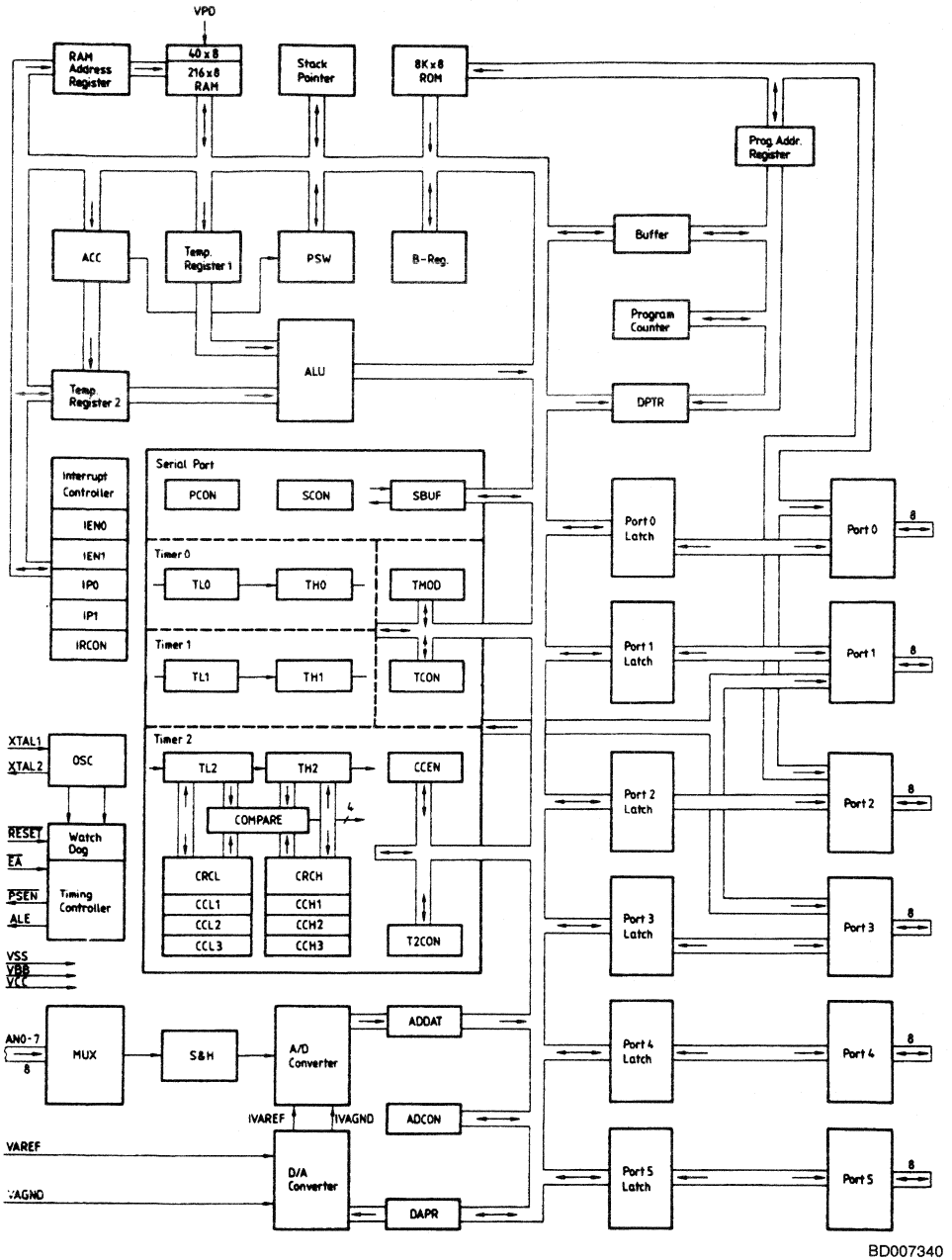


Figure 1. Memory Address Spaces



BD007340

Figure 2. Detailed Block Diagram

TABLE 1. SPECIAL FUNCTION REGISTERS

Addr (HEX)	Symbol	Name	Default After Power-On Reset
* 80	P0	Port 0	11111111
81	SP	Stack Pointer	00000111
82	DPL	Data Pointer, LOW Byte	00000000
83	DPH	Data Pointer, HIGH Byte	00000000
87	PCON	Power Control Register	0XXXXXXX
* 88	TCON	Timer Control Register	00000000
89	TMOD	Timer Mode Register	00000000
8A	TL0	Timer 0, LOW Byte	00000000
8B	TL1	Timer 1, LOW Byte	00000000
8C	TH0	Timer 0, HIGH Byte	00000000
8D	TH1	Timer 1, HIGH Byte	00000000
* 90	P1	Port 1	11111111
* 98	SCON	Serial Port Control Register	00000000
99	SBUF	Serial Port Buffer Register	Indeterminate
* 0A0	P2	Port 2	11111111
* 0A9	IEN0	Interrupt Enable Register 0	00000000
0A9	IP0	Interrupt Priority Register 0	00000000
* 0B0	P3	Port 3	11111111
* 0B9	IEN1	Interrupt Enable Register 1	00000000
0B9	IP1	Interrupt Priority Register 1	00000000
* 0C0	IRCON	Interrupt Request Control Register	00000000
0C1	CCEN	Compare/Capture Enable Register	00000000
0C2	CCL1	Compare/Capture Register 1, LOW Byte	00000000
0C3	CCH1	Compare/Capture Register 1, HIGH Byte	00000000
0C4	CCL2	Compare/Capture Register 2, LOW Byte	00000000
0C5	CCH2	Compare/Capture Register 2, HIGH Byte	00000000
0C6	CCL3	Compare/Capture Register 3, LOW Byte	00000000
0C7	CCH3	Compare/Capture Register 3, HIGH Byte	00000000
* 0C8	T2CON	Timer 2 Control Register	00000000
0CA	CRCL	Compare/Reload/Capture Register, LOW Byte	00000000
0CB	CRCH	Compare/Reload/Capture Register, HIGH Byte	00000000
0CC	TL2	Timer 2, LOW Byte	00000000
0CD	TH2	Timer 2, HIGH Byte	00000000
* 0D0	PSW	Program Status Word Register	00000000
* 0D8	ADCON	A/D-Converter Control Register	00000000
0D9	ADDAT	A/D-Converter Data Register	00000000
0DA	DAPR	D/A-Converter Program Register	00000000
* 0E0	ACC	Accumulator	00000000
* 0E8	P4	Port 4	11111111
* 0F0	B	B Register	00000000
* 0F8	P5	Port 5	11111111

The SFRs marked with an asterisk (*) are both bit and byte-addressable.
Figure 1 illustrates the memory address spaces of the 80515.

I/O Ports

The 80515 has six 8-bit ports. Port 0 is an open-drain bidirectional I/O port, while Ports 1 through 5 are quasi-bidirectional I/O ports with internal pullups. That means, when configured as inputs, Ports 1 through 5 will pull HIGH and will source current when externally pulled LOW. Port 0 will float when configured as input.

Port 0 and Port 2 can be used to expand the Program and Data Memory externally. During an access to external memory, Port 0 emits the LOW-order address byte and reads/writes the data byte, while Port 2 emits the HIGH-order address byte. In this function, Port 0 is an open-drain port, but uses a strong internal pullup FET.

Timer/Counters

The 80515 contains three 16-bit timer/counters which are useful in many applications for timing and counting. The input clock for each timer/counter is 1/12 of the oscillator frequency in the timer operation or can be taken from an external clock source for the counter operation (maximum count rate is 1/24 of the oscillator frequency).

Timer/Counters 0 and 1

These timer/counters can operate in four modes:

Mode 0: 8-bit timer/counter with 32:1 prescaler

Mode 1: 16-bit timer/counter

Mode 2: 8-bit timer/counter with 8-bit auto-reload

Mode 3: Timer/counter 0 is configured as one 8-bit timer/counter and one 8-bit timer; timer/counter 1 in this mode holds its count.

External inputs $\overline{INT_0}$ and $\overline{INT_1}$ can be programmed to function as a gate for timer/counters 0 and 1 to facilitate pulse width measurements.

Timer/Counter 2

Timer/counter 2 of the 80515 is a 16-bit timer/counter with several additional features. It offers a 2:1 prescaler, a select-

able gate function, and compare, capture, and reload functions. Corresponding to the 16-bit timer register there are four 16-bit capture/compare registers; one of them can be used to perform a 16-bit reload on a timer overflow or external event. Each of these registers corresponds to a pin on Port 1 for capture input/compare output.

Figure 3 shows a block diagram of the timer/counter 2.

Reload: With the 16-bit CRC register, which is a concatenation of the 8-bit registers CRCL and CRCH, a 16-bit reload can be performed. There are two modes from which to select:

Mode 0: Reload is caused by a timer 2 overflow (auto-reload).

Mode 1: Reload is caused in response to a negative transition at pin T2EX (P1.5), which can also request an interrupt.

Compare: In the compare mode, the 16-bit values stored in the dedicated compare registers are compared to the contents of the timer 2 registers. If the count value in the timer 2 registers matches one of the stored values, an appropriate output signal is generated and an interrupt is requested. Two compare modes are provided:

Mode 0: Upon a match, the output signal changes from LOW to HIGH. It goes back to a LOW level when timer 2 overflows.

Mode 1: The transition of the output signal can be determined by software. A timer 2 overflow causes no output change.

Capture: This feature permits saving the actual timer/counter contents into a selected register upon an external event or a software write operation. Two modes are provided to latch the current 16-bit value in timer 2 registers into a dedicated capture register:

Mode 0: Capture is performed in response to a transition at the corresponding Port 1 pins CC0 to CC3.

Mode 1: Write operation into the LOW-order byte of the dedicated capture register causes the timer 2 contents to be latched into this register.

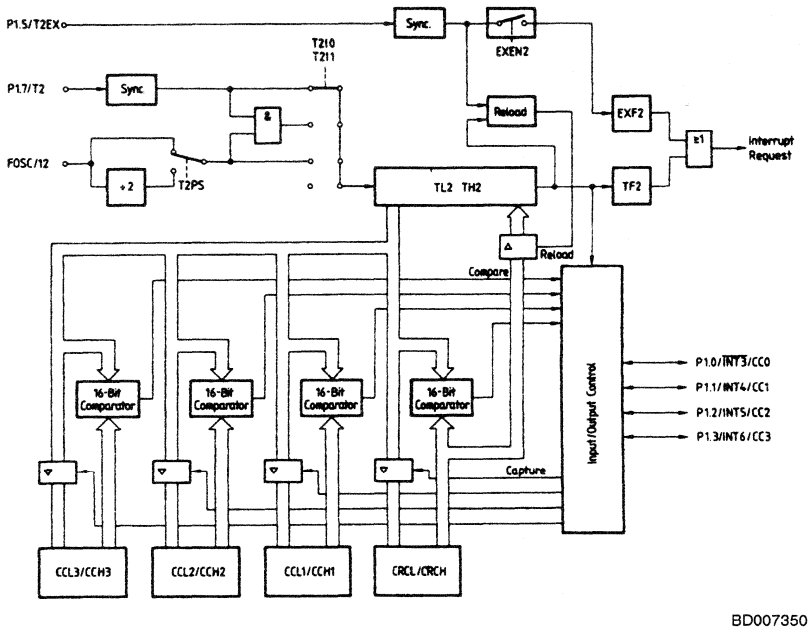


Figure 3. Timer/Counter 2 Block Diagram

Watchdog Timer

As a means of safe recovery from software or hardware upset, a watchdog timer is provided in the 80515. If the software fails to clear the watchdog timer at least every 65,532 µs, an internal hardware reset will be initiated. The software can be designed such that the watchdog times out if the program does not progress properly. The watchdog will also time out if the software error was due to hardware-related problems. This prevents the controller from malfunctioning for longer than 65 ms if a 12 MHz oscillator is used.

The watchdog timer is a 16-bit counter which is incremented once every machine cycle. After an external reset, the watchdog timer is disabled and cleared to 0000H. The counter is started by setting bit SWDT (bit 6 in SFR IEN1). After having been started, the watchdog timer 0000H by cannot be stopped by software. It can only be cleared to 0000H by first setting bit WDT (IEN0.6) and with the next instruction setting SWDT. Bit WDT will automatically be cleared during the third machine cycle after having been set. This double instruction clearing of the watchdog timer was implemented to minimize the chance of unintentionally clearing the watchdog. To prevent the watchdog from overflowing, it must be cleared periodically.

If the software fails to clear the watchdog in time, an internally generated watchdog reset is entered at the counter state FFFCH, which lasts four machine cycles. This internal reset differs from an external reset only to the extent that the

watchdog timer is not disabled and bit WDTS (watchdog timer status, bit 6 in SFR IP0) is set. Bit WDTS allows the software to examine from which source the reset was initiated. If it is set, the reset was caused by a watchdog timer overflow.

Serial Port

The serial port of the 80515 permits the full-duplex communication between microcontrollers or between microcontrollers and peripheral devices. The serial port can operate in four modes:

Mode 0: Shift register mode. Serial data enters and exits through RxD. TxD outputs the shift clock. Eight bits are transmitted/received — eight data bits (LSB) first. The baud rate is fixed at 1/12 of the oscillator frequency.

Mode 1: Ten bits are transmitted (through RxD) or received (through TxD) — a start bit (0), eight data bits (LSB first), and a stop bit (1). The baud rate is variable.

Mode 2: Eleven bits are transmitted (through RxD) or received (through TxD) — a start bit (0), eight data bits (LSB first), a programmable 9th data bit, and a stop bit (1). The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

Mode 3: Eleven bits are transmitted (through TxD) or received (through RxD) — a start bit (0), eight data bits (LSB first), a programmable 9th data bit, and a stop bit (1). Mode 3 is the same as mode 2 in all respects except the baud rate; the baud rate in mode 3 is variable.

The variable baud rates can be generated by timer 1 or an internal baud rate generator.

A/D Converter

The 8-bit A/D converter of the 80515 has eight multiplexed analog inputs and uses the successive approximation method. The sampling of an analog signal takes 5 machine cycles, the total conversion time is 15 machine cycles (15 μs at 12 MHz oscillator frequency). Conversion can be programmed to be single or continuous; at the end of a conversion an interrupt can be generated.

The internal reference voltages IVAREF and IVAGND for the A/D converter are programmable in 16 steps with respect to the external reference voltages. This feature permits a second conversion with changed internal reference voltages to gain a higher resolution. In addition, the internal reference voltages can easily be adapted by software to the desired analog voltage range. The D/A conversion of the divide factors for the reference voltages takes 7 machine cycles each (7 μs at 12 MHz oscillator frequency).

Figure 4 shows a block diagram of the A/D converter of the 80515.

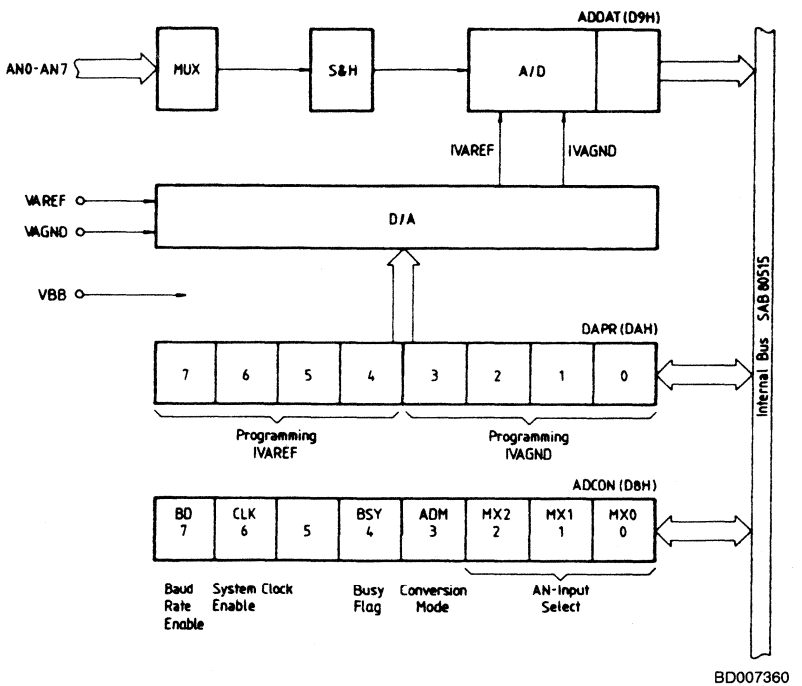


Figure 4. A/D Converter Block Diagram

Interrupt Structure

The twelve interrupt sources of the 80515 are organized in six pairs:

1) External interrupt to 0 — A/D converter interrupt
2) Timer 0 interrupt — External interrupt 2
3) External interrupt 1 — External interrupt 3
4) Timer 1 interrupt — External interrupt 4
5) Serial port interrupt — External interrupt 5
6) Timer 2 interrupt — External interrupt 6

Each interrupt source has its own vector address. It can be programmed to one of four priority levels and can individually be enabled/disabled. The minimum interrupt response time is 3 to 8 machine cycles.

External interrupts 0 and 1 can be activated by a LOW-level or a negative transition (selectable) at their corresponding input pin; external interrupts 2 and 3 can be programmed to be activated by a negative or a positive transition. The external interrupts 4 to 6 are activated by a positive transition. The interrupts 3 to 6 can be combined with the corresponding alternate functions compare (output) and capture (input) on Port 1.

Figure 5 shows the interrupt request sources, and Figure 6 illustrates the priority level structure of the 80515.

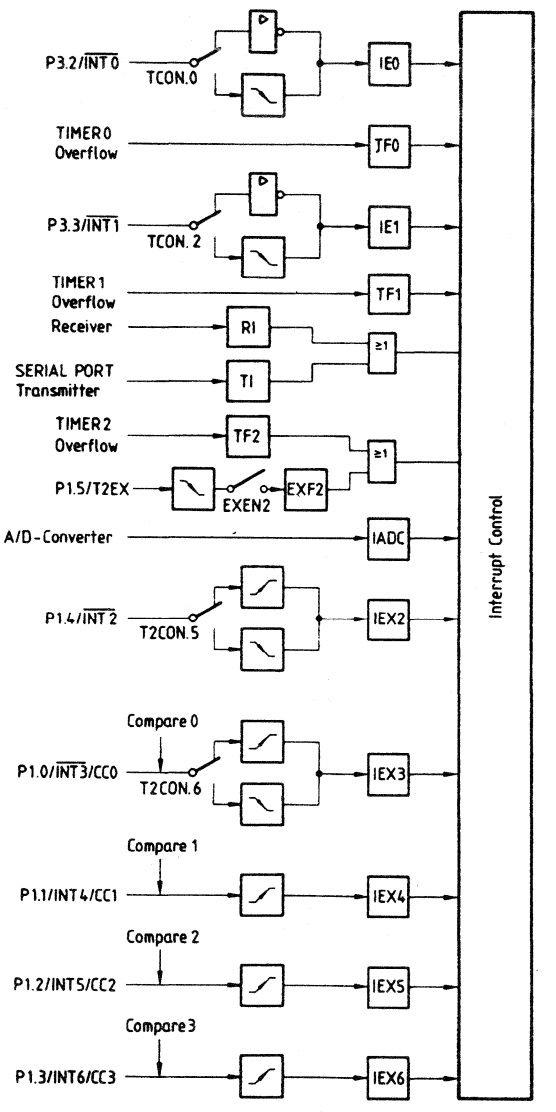


Figure 5. Interrupt Request Sources

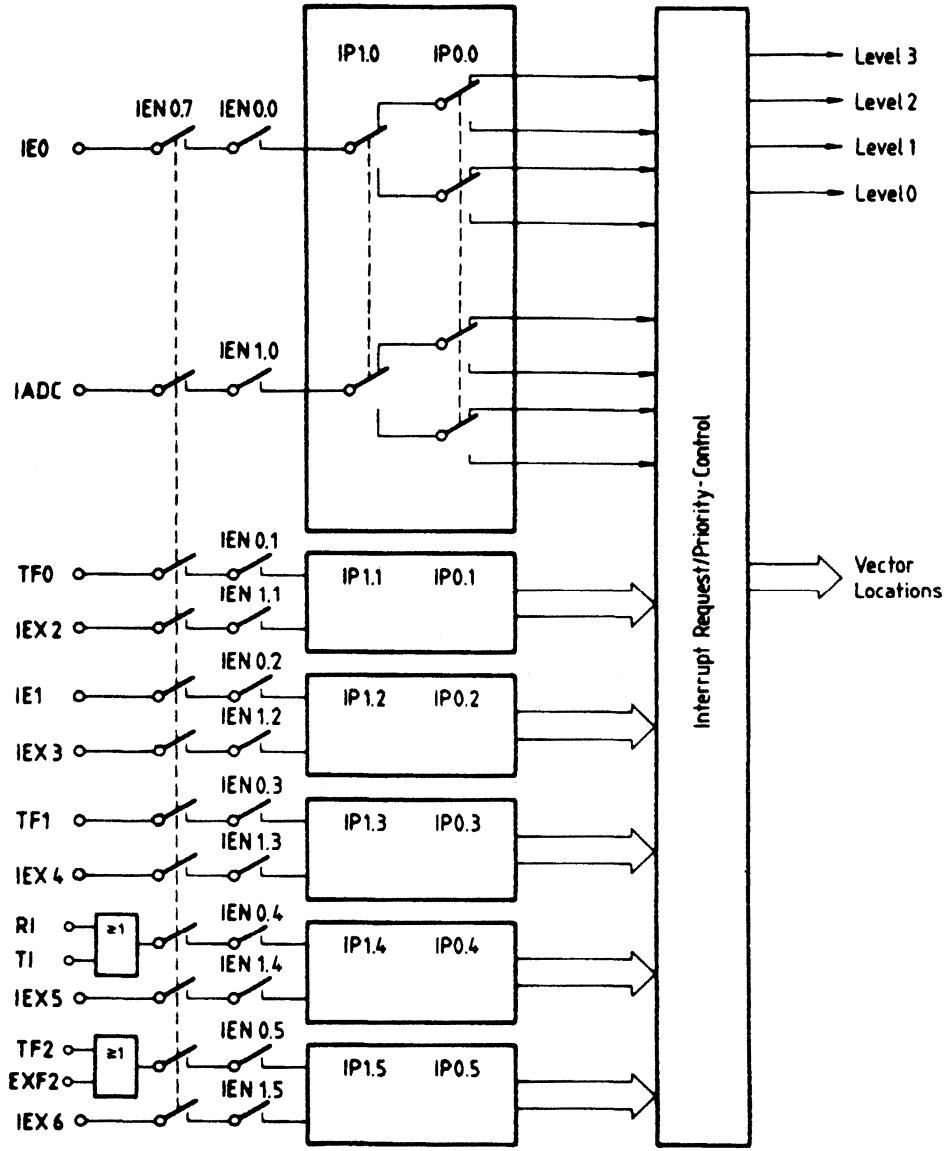


Figure 6. Priority Level Structure

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin
with Respect to Ground(V_{SS}) ..... −0.5 to +7.0 V
Power Dissipation ..... 2 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5.0 V ± 10%
Ground (V_{SS}) ..... 0 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
V_{IL}	Input LOW Voltage		−0.5	0.8	V
V_{IH}	Input HIGH Voltage (Except $\overline{RESET}$ and $XTAL_2$)		2.0	V_{CC} +0.5	V
V_{IH1}	Input HIGH Voltage to $XTAL_2$)	$XTAL_1$ to V_{SS}	2.5	V_{CC} +0.5	V
V_{IH2}	Input HIGH Voltage to $\overline{RESET}$		3.0		V
V_{PD}	Power-Down Voltage	V_{CC} = 0 V	3	5.5	V
V_{OL}	Output LOW Voltage, Ports 1, 2, 3, 4, 5	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output LOW Voltage, Port 0, ALE, $\overline{PSEN}$	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output HIGH Voltage, Ports 1, 2, 3, 4, 5	I_{OH} = −80 μA	2.4		V
V_{OH1}	Output HIGH Voltage, Port 0, ALE, $\overline{PSEN}$	I_{OH} = −400 μA	2.4		V
I_{IL}	Logic 0 Input Current, Ports 1, 2, 3, 4, 5	V_{IL} = 0.45 V		−800	μA
I_{IL2}	Logic 0 Input Current, $XTAL_2$	$XTAL_1 = V_{SS}$ V_{IL} = 0.45 V		−2.5	mA
I_{IL3}	Input LOW Current to $\overline{RESET}$ for Reset	V_{IL} = 0.45 V		−500	μA
I_{L1}	Input Leakage Current to Port 0, $\overline{EA}$	$0V < V_{IN} < V_{CC}$		±10	μA
I_{CC}	Power Supply Current 80515/80535	All Outputs Disconnected		210	mA
I_{PD}	Power-Down Current	V_{CC} = 0 V		3	mA
C_{IO}	Capacitance of I/O Buffer	f_c = 1 MHz		10	pF

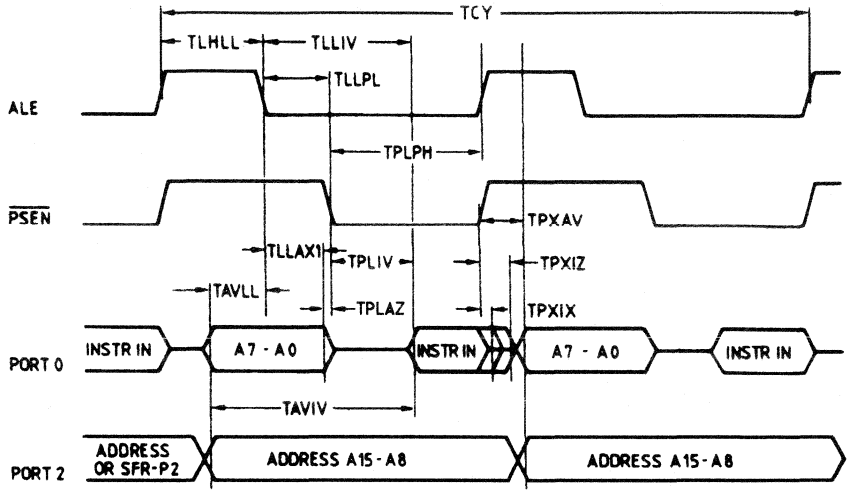
See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified (C_L for Port 0, ALE, and $\overline{PSEN}$ Outputs = 100 pF; C_L For All Other Outputs = 80 pF)

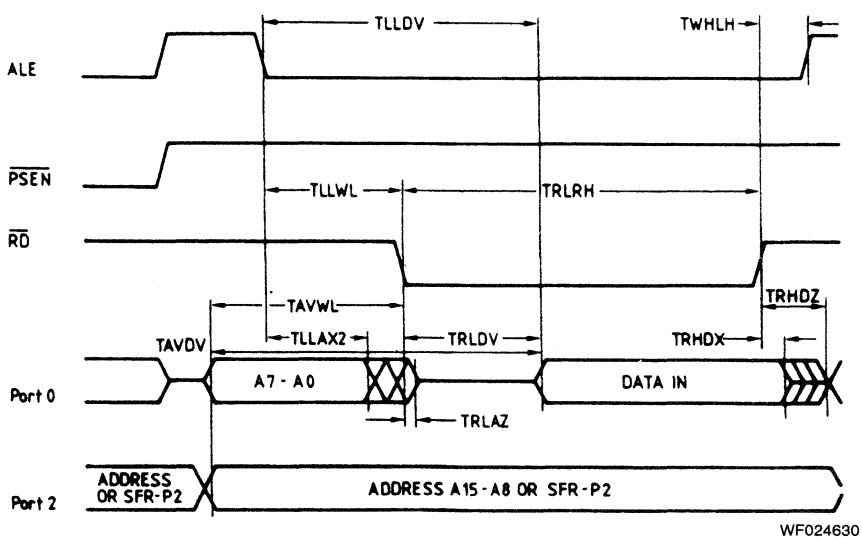
Parameter Symbol	Parameter Description	12 MHz Clock		Variable Clock		Units
		Min.	Max.	Min.	Max.	
1/TCLCL	Cycle Time			1.2	12	MHz
TLHLL	ALE Pulse Width	127		2TCLCL-40		ns
TAVLL	Address Setup to ALE	53		TCLCL-30		ns
TLLAX1	Address Hold After ALE	48		TCLCL-35		ns
TLLIV	ALE to Valid Instruction In		233		4TCLCL-100	ns
TLLPL	ALE to $\overline{PSEN}$	58		TCLCL-25		ns
TPLPH	$\overline{PSEN}$ Pulse Width	215		3TCLCL-35		ns
TPLIV	$\overline{PSEN}$ to Valid Instruction In		150		3TCLCL-100	ns
TPXIX	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
TPXIZ*	Input Instruction Float After $\overline{PSEN}$		63		TCLCL-20	ns
TPXAV*	Address Valid After $\overline{PSEN}$	75		TCLCL-8		ns
TAVIV	Address to Valid Instruction In		302		5TCLCL-115	ns
TPLAZ	Address Float to $\overline{PSEN}$	20		20		
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL-100		ns
TLLAX2	Address Hold After ALE	132		2TCLCL-35		ns
TRLDV	$\overline{RD}$ to Valid Data In		250		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		2TCLCL-70	ns
TLLDV	ALE to Valid Data In		517		8TCLCL-150	ns
TAVDV	Address to Valid Data In		585		9TCLCL-165	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	200	300	3TCLCL-50	3TCLCL + 50	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	203		4TCLCL-130		ns
TWHLH	$\overline{WR}$ or $\overline{RD}$ HIGH to ALE HIGH	43	123	TCLCL-40	TCLCL + 40	ns
TQVWX	Data Valid to $\overline{WR}$ Transition	33		TCLCL-50		ns
TQVWH	Data Setup Before $\overline{WR}$	433		7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	33		TCLCL-50		ns
TRLAZ	Address Float After $\overline{RD}$		20		20	ns

* Interfacing the 80515 to devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damage to Port 0 drivers.

SWITCHING WAVEFORMS

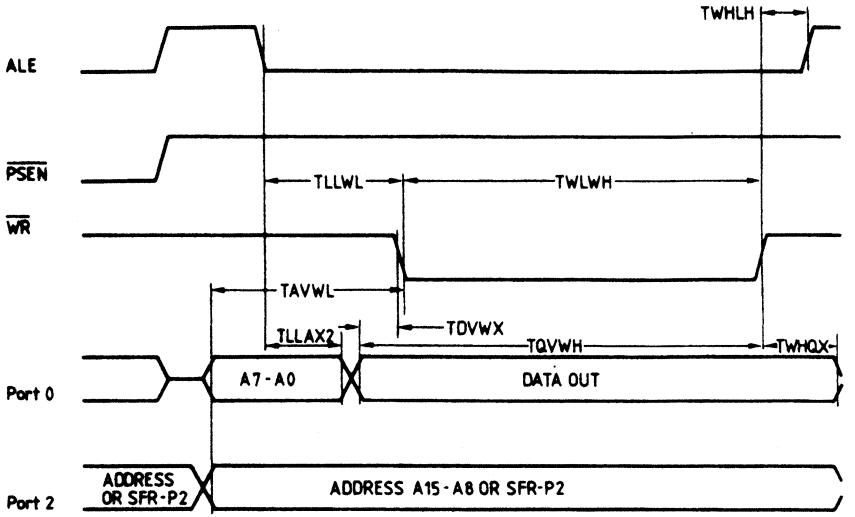


Program Memory Read Cycle



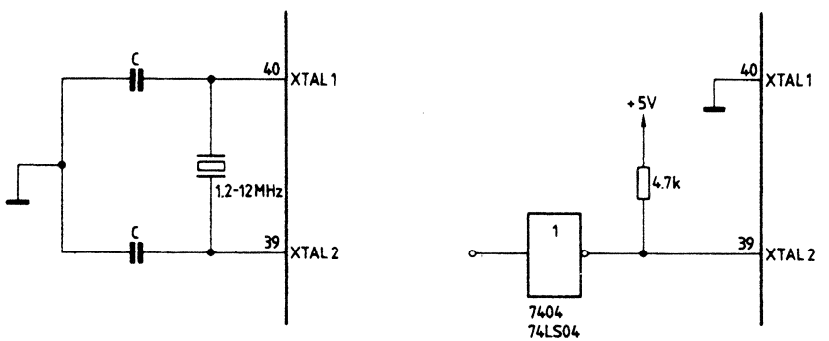
Data Memory Read Cycle

SWITCHING WAVEFORMS (Cont'd.)



WF024640

Data Memory Write Cycle



C = 30 pF ± 10 pF

Crystal Oscillator Mode

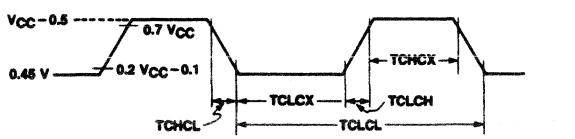
Driving from External Source

TC004310

Recommended Oscillator Circuits

EXTERNAL CLOCK DRIVE

Parameter Symbol	Parameter Description	Min.	Max.	Units
1/TCLCL	Oscillator Frequency	1.2	12	MHz
TCHCX	HIGH Time	20		ns
TCLCX	LOW Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns



WF020910

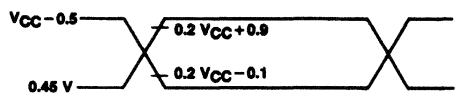
External Clock Drive Waveform

SERIAL PORT TIMING — SHIFT REGISTER MODE

(Load Capacitance = 80 pF)

Parameter Symbol	Parameter Description	12 MHz Osc.		Variable Oscillator		Units
		Min.	Max.	Min.	Max.	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL – 133		ns
TXHQX	Output Data Hold After Clock Rising Edge	50		2TCLCL – 117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL – 133	ns

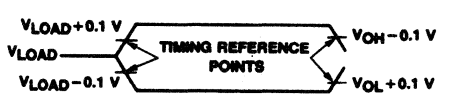
AC Testing



WF020900

AC INPUTS DURING TESTING ARE DRIVEN AT V_{CC} – 0.5 FOR A LOGIC "1" AND 0.45 V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT V_{IH} MIN. FOR A LOGIC "1" AND V_{IL} MAX. FOR A LOGIC "0."

Input/Output Waveform



WF020940

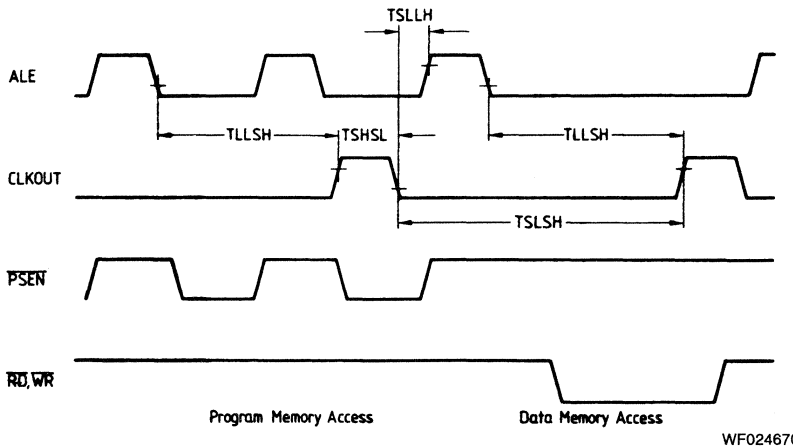
FOR TIMING PURPOSES A PORT PIN IS NO LONGER FLOATING WHEN A 100 mV CHANGE FROM LOAD VOLTAGE OCCURS, AND BEGINS TO FLOAT WHEN A 100 mV CHANGE FROM THE LOADED V_{OH}/V_{OL} LEVEL OCCURS. $I_{OL}/I_{OH} \geq \pm 20$ mA.

Float Waveform

SYSTEM CLOCK TIMING

Parameter Symbol	Parameter Description	12 MHz clock		Variable Clock 1/TCLCL = 1.2 MHz to 12 MHz		Units
		Min.	Max.	Min.	Max.	
TLLSH	ALE to CLKOUT	543		7TCLCL-40		ns
TSHSL	CLKOUT HIGH Time	127		2TCLCL-40		ns
TSLSH	CLKOUT LOW time	793		10TCLCL-40		ns
TSLLH	CLKOUT LOW to ALE HIGH	43	123	TCLCL-40	TCLCL + 40	ns

System Clock Timing



A/D Converter Characteristics (V_{CC} = 5 V ±10%; V_{SS} = 0 V; $V_{AREF} = V_{CC}$ ±5%; $V_{AGND} = V_{SS}$ ±0.2 V; T_A = 0 to +70°C)

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
V_{AINPUT}	Analog Input Voltage		V_{AGND}-0.2	V_{AREF}+0.2	V
C_I	Analog Input Capacitance				pF
T_S	Sample Time			5 TCY	µs
T_C	Conversion Time (Including Sample Time)	for $IV_{AREF} = V_{AREF}$ and $IV_{AGND} = V_{AGND}$		15 TCY	µs
		for $IV_{AREF} \neq V_{AREF}$ and $IV_{AGND} = V_{AGND}$ or for $IV_{AREF} = V_{AREF}$ and $IV_{AGND} \neq V_{AGND}$		22 TCY	µs
		for $IV_{AREF} \neq V_{AREF}$ and $IV_{AGND} \neq V_{AGND}$		29 TCY	µs
	Differential Non-Linearity	$IV_{AREF} = V_{AREF} = V_{CC}$ $IV_{AGND} = V_{AGND} = V_{SS}$ R_i of Analog Input Source ≤ 10 kΩ		±1	LSB
	Integral Non-Linearity			±1	LSB
	Offset Error			±1	LSB
	Gain Error			±1	LSB
I_{REF}	V_{AREF} Supply Current			5	mA

Notes: 1. The internal resistance of the analog source must be less than 10 kΩ to assure full loading of the sample capacitance during sample time.
2. The internal resistance of the analog reference voltage source must be less than 1 kΩ.

80C521/80C321

CMOS Single-Chip Microcontroller

ADVANCE INFORMATION

DISTINCTIVE CHARACTERISTICS

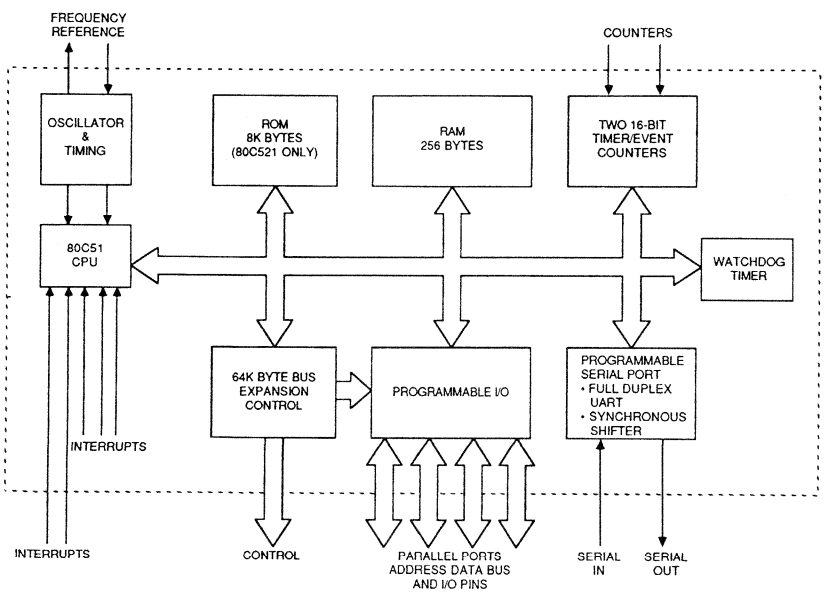
- **CMOS extensions to the 80C51**
- **256 Bytes of RAM**
- **8K Bytes of ROM (80C521 only)**
- **Programmable Watchdog Timer**
- **Dual Data Pointers**
- **Software Reset**
- **All Original 80C51 Features Are Retained**
  - 32 I/O Lines
  - Two 16-Bit Timer/Counters
  - Five Source, Two Level Interrupt
  - 64K Bytes Program Memory Space
  - Full-Duplex Serial Port
  - Power-Down & Idle Modes
  - On-Chip Oscillator/Clock Circuit
  - 64K Bytes Data Memory Space

GENERAL DESCRIPTION

The 80C521 and 80C321 Microcontrollers are fully instruction-set-compatible and pin-compatible enhancements of the 80C51/80C31. The 80C521 contains 8K bytes of ROM, 256 bytes of RAM, a programmable Watchdog Timer, and Dual Data Pointers. The Watchdog Timer can be programmed to times ranging from 128 microseconds to four full seconds at 12 MHz.

The Dual Data Pointer structure speeds access to external memory by providing two identical 16-bit data pointers with a fast switching mechanism, rather than a single data pointer as in the rest of the 8051 Family. The 80C321 is a ROM-less version of the 80C521.

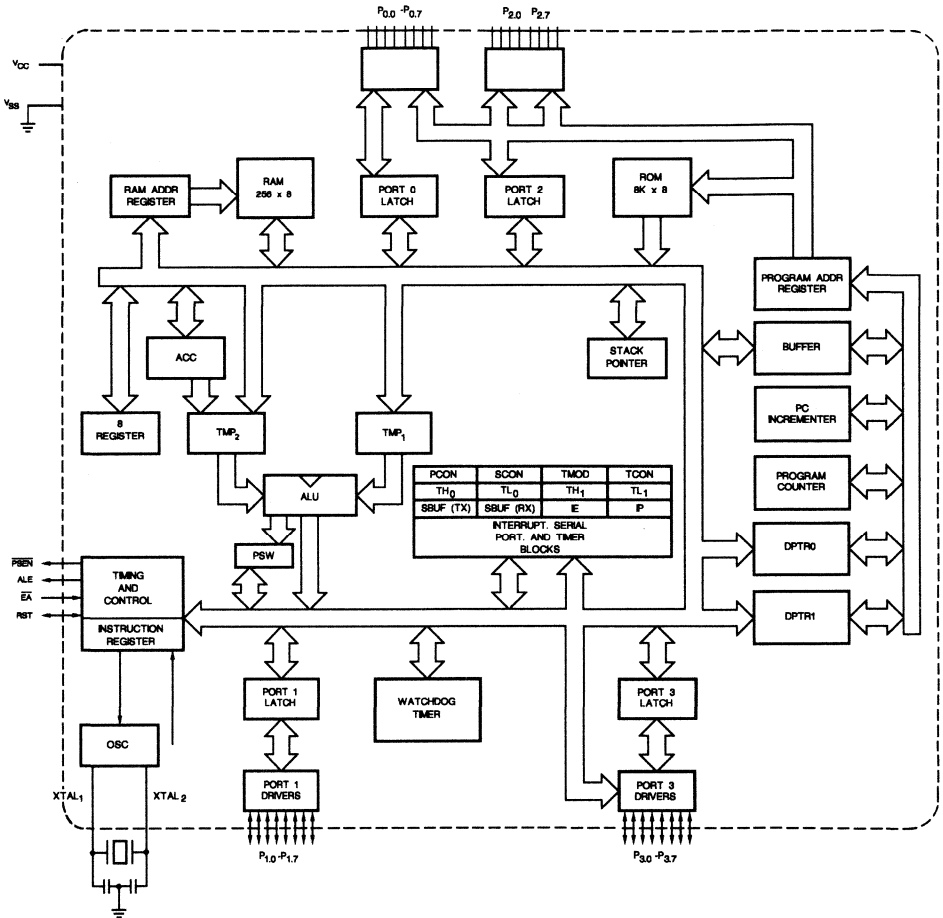
SIMPLIFIED BLOCK DIAGRAM



BD007211

Publication # 09136 | Rev. A | Amendment /0
Issue Date: July 1987

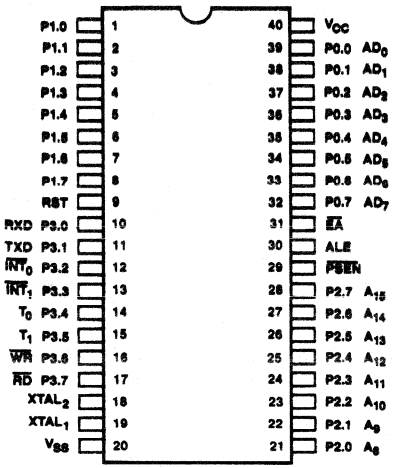
DETAILED BLOCK DIAGRAM



BD004096

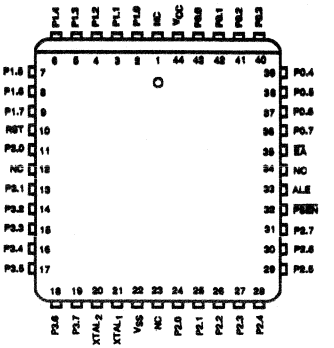
CONNECTION DIAGRAMS
Top View

DIPs



CD005551

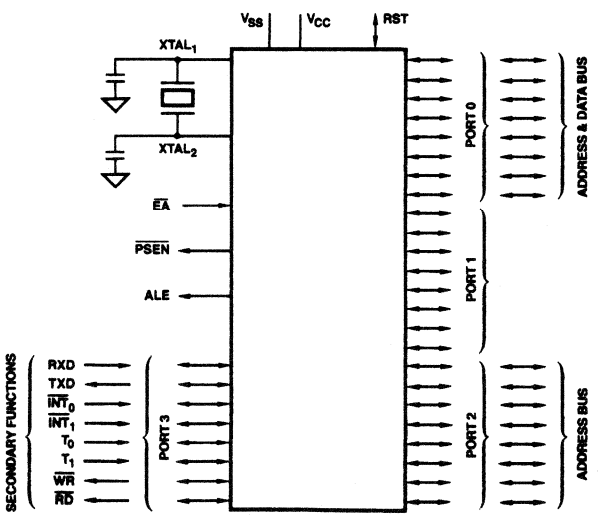
PLCC



CD009440

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL



LS001324

PIN DESCRIPTION

Port 0 (Bidirectional, Open Drain)

Port 0 is an open-drain bidirectional I/O port. Port 0 pins that have "1"s wriiten to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed LOW-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting "1"s. Port 0 also outputs the code bytes during program verification in the 80C521. External pullups are required during program verification.

Port 1 (Bidirectional)

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four LSTTL inputs. Port 1 pins that have "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 1 pins that are externally being pulled LOW will source current (I_{IL} on the data sheet) because of the internal pullups.

Port 1 also receives the LOW-order address bytes during program verification.

Port 2 (Bidirectional)

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four LSTTL inputs. Port 2 pins having "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 2 pins externally being pulled LOW will source current (I_{IL}) because of the internal pullups.

Port 2 emits the HIGH-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting "1"s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the HIGH-order address bits during ROM verification.

Port 3 (Bidirectional)

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four LSTTL inputs. Port 3 pins that have "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 3 pins externally being pulled LOW will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features as listed below:

Port Pin	Alternate Function
$P_{3.0}$	RxD (serial input port)
$P_{3.1}$	TxD (serial output port)
$P_{3.2}$	$\overline{INT_0}$ (External interrupt 0)
$P_{3.3}$	$\overline{INT_1}$ (external interrupt 1)
$P_{3.4}$	T_0 (Timer 0 external input)
$P_{3.5}$	T_1 (Timer 1 external input)
$P_{3.6}$	$\overline{WR}$ (external Data Memory write strobe)
$P_{3.7}$	$\overline{RD}$ (external Data Memory read strobe)

RST Reset (Input/Output, Active HIGH)

A HIGH on this pin — for two machine cycles while the oscillator is running — resets the device. An internal diffused resistor to V_{SS} permits power-on reset, using only an external capacitor to V_{CC}.

Immediately prior to a Watchdog Reset or Software Reset, this pin is pulled HIGH for one state time. The internal pull-up can be overdriven by an external driver capable of sinking/sourcing 2.5 mA (see Figure 6 for possible circuit configurations).

ALE Address Latch Enable (Output, Active HIGH)

Address Latch Enable output pulse for latching the LOW byte of the address during accesses to external memory.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, allowing use for external-timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$ Program Store Enable (Output, Active LOW)

$\overline{PSEN}$ is the read strobe to external Program Memory. When the 80C521 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle — except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory. $\overline{PSEN}$ is not activated during fetches from internal Program Memory.

$\overline{EA}$ External Access Enable (Input, Active LOW)

$\overline{EA}$ must be externally held LOW to enable the device to fetch code from external Program Memory locations 0000H to 1FFFH. If $\overline{EA}$ is held HIGH, the device executes from internal Program Memory unless the program counter contains an address greater than 1FFFH.

The 80C521 internally latches the value of the $\overline{EA}$ pin at the falling edge of the reset pulse on the RST pin during a Hardware or Power-on Reset. Once latched, the $\overline{EA}$ value cannot be changed except by a Hardware reset.

$XTAL_1$ Crystal (Input)

Input to the inverting-oscillator amplifier, and input to the internal clock-generator circuits.

$XTAL_2$ Crystal (Output)

Output from the inverting-oscillator amplifier.

V_{CC} Power Supply

Supply voltage during normal, idle, and power-down operations.

V_{SS} Circuit Ground

FUNCTIONAL DESCRIPTION

Program Memory

The 80C521 has 64K bytes of Program Memory space. The lower 8K bytes (addresses 0000H to 1FFF) may reside on-chip. Instructions residing at addresses beyond 1FFF will always be fetched externally. When the External Access ($\overline{EA}$) pin is held LOW, all code-fetch operations take place externally to the 80C521.

Data Memory

The 80C521 can address 64K bytes of Data Memory external to the chip. The "MOVX" instructions are used to access the external Data Memory.

The internal data memory is comprised of three physically distinct memory spaces. They are the lower 128 bytes of RAM, the upper 128 bytes of RAM, and the 128 byte Special Function Register (SFR) space. The lower 128 bytes of RAM can be accessed through direct addressing (i.e., MOV addr, data), or indirect addressing (i.e., MOV @ Ri). The upper 128 bytes of RAM (locations 80H through FFH) can be accessed only through indirect addressing modes. The Special Function Register space, while physically distinct from the upper 128 bytes of RAM, shares addresses with the upper 128 bytes of RAM. The SFR space may be accessed through direct addressing modes only.

The first 32 bytes of RAM contain four register banks, each of which contains eight general-purpose registers. The next 16 bytes (locations 20H through 2FH) contain 128 directly addressable bit locations. The stack may be located anywhere in the internal RAM space and may be up to 256 bytes in length.

SPECIAL FUNCTION REGISTER MAP

Addr (HEX)	Symbol	Name	Default After Power-On Reset
* 80	P0	Port 0	11111111
81	SP	Stack Pointer	00000111
82	DPL	Data Pointer Low	00000000
83	DPH	Data Pointer High	00000000
+ 84	DPL1	Data Pointer Low 1	00000000
+ 85	DPH1	Data Pointer High 1	00000000
+ 86	DPS	Data Pointer Selection	00000000
87	PCON	Power Control	0XXX0000
* 88	TCON	Timer/Counter Control	00000000
89	TMOD	Timer/Counter Mode Control	00000000
8A	TL0	Timer/Counter 0 Low Byte	00000000
8B	TL1	Timer/Counter 1 Low Byte	00000000
8C	TH0	Timer/Counter 0 High Byte	00000000
8D	TH1	Timer/Counter 1 High Byte	00000000
* 90	P1	Port 1	11111111
* 98	SCON	Serial Control	00000000
99	SBUF	Serial Data Buffer	Indeterminate
* A0	P2	Port 2	11111111
* A8	IE	Interrupt Enable Control	0XX00000
+ A9	WDS	Watchdog Selection	00000000
+ AA	WDK	Watchdog Key	00000000
* B0	P3	Port 3	11111111
* B8	IP	Interrupt Priority Control	XXX00000
* D0	PSW	Program Status Word	00000000
* E0	ACC	Accumulator	00000000
* F0	B	B Register	00000000

* Bit Addressable

\+ New SFRs defined on the 80C521/80C321

Basic Timing Definitions

Instructions in the 8051 Family execute in either one, two, or four machine cycles. A machine cycle is comprised of six state times with each state comprised of two clock cycles; thus, a machine cycle lasts 12 clock cycles. With an external oscillator running at 12 MHz, a machine cycle lasts 1 μs. At 16 MHz, a machine cycle lasts 750 ns.

Reset Operation

The 80C521/80C321 may be reset by four different methods: (1) Power-On Reset, (2) Hardware Reset, (3) Watchdog Reset, and (4) Software Reset.

1) **Power-on Reset** occurs when the RST pin is wired to V_{CC} using an external capacitor, and V_{CC} is activated.

2) **Hardware Reset** occurs when the oscillator is running and the RST pin is held HIGH for two or more machine cycles.

3) **Watchdog Reset** occurs when the count value of the Watchdog Timer is allowed to exceed the programmed value, resulting in an overflow signal that resets the chip in two machine cycles.

4) **Software Reset** occurs when the software writes a Keyed sequence to the Key register of the Watchdog Timer. This causes a Watchdog Reset to be immediately generated.

After Power-On Reset, the SFRs have the values indicated in the Special Function Register Map Section, and the contents of the Internal RAM are undefined. Hardware Reset is the same as Power-On Reset except that the contents of the Internal RAM are preserved. A Hardware Reset has priority over a Watchdog Reset or a Software Reset. The Watchdog Reset puts the 80C521 into the same state as the Hardware Reset except that the Reset Cause (RC) bit in the Watchdog Selection (WDS) register is set to a one. The Software Reset is functionally equivalent to the Watchdog Reset.

Watchdog Timer

The Watchdog Timer (WDT) is a specially designed timer unit that will reset the chip upon reaching a pre-programmed time interval. It operates independently of the two general purpose timer/counters and is dedicated specifically to the watchdog function. The Watchdog Timer allows safe recovery from problems resulting from unexpected input conditions, external events, or programming anomolies.

The WDT is disabled following any reset. While disabled, the WDT time interval may be programmed. The WDT is enabled by a sequence of two write operations.

Once enabled, the WDT cannot be stopped (i.e., disabled) except by one of the four Reset types described in the last section. Furthermore, while the WDT is enabled, the WDT time interval cannot be modified. The WDT, however, may be cleared by software at any time with the same sequence of two write operations. The clearing operation causes the present count of the WDT to be set to zero, but it does not stop the WDT from incrementing.

If the count in the WDT ever reaches the pre-programmed value, the WDT will overflow, resetting the chip in two machine cycles. This is a Watchdog Reset. Additionally, if a system error condition is discovered, software may intentionally generate an immediate reset via the WDT, using a special sequence of write operations. This is a Software Reset.

A Watchdog Reset or Software Reset will set a special "cause" bit, allowing differentiation between these two Reset types and the Hardware or Power-On Reset types. Neither Watchdog Reset nor the Software Reset modify the contents of the Internal RAM. The Watchdog Reset will cause the RST pin to be pulled high during S1P1 and S1P2 of the first cycle of the two-cycle reset, providing a hardware indication that a reset is imminent.

Two 8-bit Special Function Registers are associated with the WDT. They are as follows:

Watchdog Selection – – (WDS) – – Address: A9 (Hex)

Watchdog Key – – (WDK) – – Address: AA (Hex)

Watchdog Selection – – (WDS) – – Address: A9H

The Watchdog Selection register allows the time interval of the WDT to be programmed and retains the cause of the most recent reset. This register is Read/Write, but its contents cannot be changed once the WDT has been enabled. Its default value after a Hardware or Power-On Reset = 00H. Its default value after a Watchdog Reset or Software Reset = 80H. This is the only register on the 80C521 whose initialization value differs between the two reset groups.

(MSB) **(LSB)**

RC	–	TV	–	PT3	PT2	PT1	PT0
7	6	5	4	3	2	1	0

Bits 3-0 – – Programmed Time – – (PT3 – PT0)

The value contained in these bits at the time the Watchdog Timer is enabled determines the time interval of the WDT. The time interval is a multiple of the input clock period. The times are decoded as follows:

Programmable Watchdog Timing Intervals

PT3-PT0	12 MHz	16 MHz	Clock Divide Ratio
0 0000	128 μs	96 μs	1536
1 0001	256 μs	192 μs	3072
2 0010	512 μs	384 μs	6144
3 0011	1.024 ms	768 μs	12288
4 0100	2.048 ms	1.536 ms	24576
5 0101	4.096 ms	3.072 ms	49152
6 0110	8.192 ms	6.144 ms	98304
7 0111	16.384 ms	12.288 ms	196608
8 1000	32.768 ms	24.576 ms	393216
9 1001	65.536 ms	49.152 ms	786432
A 1010	131.072 ms	98.304 ms	1572864
B 1011	262.144 ms	196.608 ms	3145728
C 1100	524.288 ms	393.216 ms	6291456
D 1101	1.049 sec.	786.432 ms	12582912
E 1110	2.097 sec.	1.573 sec.	25165824
F 1111	4.194 sec.	3.146 sec.	50331648

If the Programmed Time bits are read while the WDT is disabled, they will show the last value written. Once the WDT is enabled, these bits will show the programmed time of the WDT and cannot be modified.

Bit 4

Reserved. Will return an unidentified value when read.

Bit 5 – – Timer Verification – – (TV)

This bit reflects Bit 12 of the internal counter within the Watchdog Timer. It will toggle every 8.192 ms at 12 MHz. This bit is Read-only.

Bit 6

Reserved. Will return an unidentified value when read.

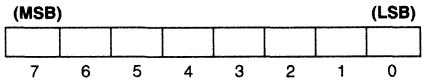
Bit 7 – – Reset Cause – – (RC)

The Reset Cause bit indicates the cause of the last reset of the 80C521. If a Power-On or Hardware Reset occurs, the bit is set to a zero by the reset circuitry. If a Watchdog or Software Reset occurs, the bit is set to a one by the reset circuitry. Like the Programmed Time bits, this bit may not be modified once the WDT is enabled. Writing this bit does not affect any chip function.

Watchdog Key – – (WDK) – – Address: AAH

This register controls the enabling and clearing of the Watchdog Timer. The writing of an A5H followed by the writing of a 5AH to this register enables the WDT to begin incrementing. It is not a requirement that the writes be on consecutive instructions, thus interrupts do not have to be disabled. Once the WDT is enabled, it may be cleared at any time by the writing of the same sequence. The clearing operation causes the present count of the WDT to be cleared, but does not stop the WDT from incrementing.

This is a Write-only register. Read operations are not defined and will not affect the WDT circuitry.



The enabling/clearing operation of the Watchdog Timer is accomplished by writing a keyed sequence of values to the WDK register. The Keyed Sequence is comprised of two stages (see Figure 1).

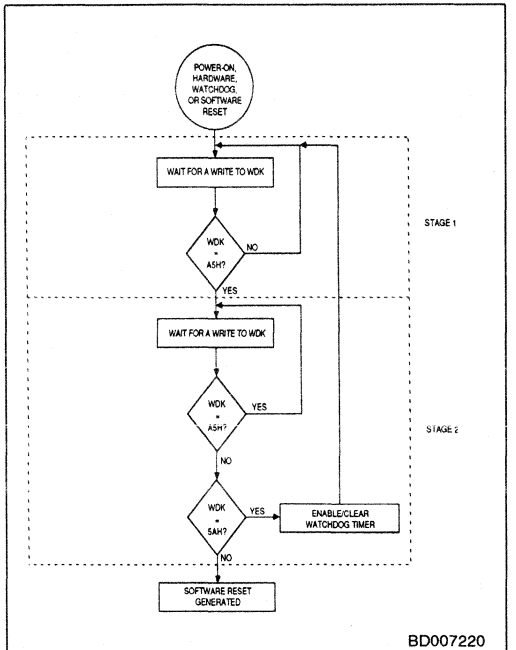


Figure 1. WDT Keyed Sequence Flowchart

The Keyed Sequence is in Stage 1 after all forms of reset, or following any Watchdog enable or clear operation. In Stage 1 all values written to the WDK register are ignored except A5H. An A5H causes the Keyed Sequence to enter Stage 2.

Once Stage 2 is entered, the next write to the WDK register prompts one of the following actions: 1) If the next write is again an A5H, the Keyed Sequence remains in Stage 2; 2) If the next write is a 5AH, the WDT is enabled/cleared, and the Keyed Sequence re-enters Stage 1; or, 3) If the next write is any other value, a Software Reset via the WDT is generated.

Example of Write Operations to WDK:

Write 1st	Write 2nd	Action Taken After Second Write
11	18	No action taken, Keyed Sequence still in Stage 1
A5	A5	Keyed Sequence enters Stage 2 and remains there
A5	5A	WDT is enabled/cleared, Sequence reenters Stage 1
A5	11	Software Reset occurs via the WDT

The two-stage feature, together with the Software Reset, greatly reduces the chance of an instruction sequence accidentally clearing the Watchdog Timer. Furthermore, while still allowing a Software Reset to be initiated, the two-stage feature reduces the chance of unintentionally generating a Software Reset.

Software Reset

A Software Reset may be accomplished through the Watchdog Timer. If an A5H is written to the Watchdog Key (WDK) register, followed by the write of a value other than A5H or 5AH, a Software Reset will be generated. This "software-generated" Watchdog Reset occurs regardless of whether or not the Watchdog Timer was previously enabled.

After the second value is written to the WDK register, program execution continues for one machine cycle before the reset operation begins. During S1P1 and S1P2 of this last machine cycle, the RST pin is pulled HIGH (see Figure 6). The reset operation lasts two machine cycles and does not modify the contents of the internal RAM.

The Software Reset is functionally equivalent to the Watchdog Reset. For instance, the Reset Cause bit in WDS will be set to one, indicating a Watchdog Reset occurred (see the Watchdog Timer section for more details).

The following code may be used to generate a Software Reset.

```
MOV WDK,#A5H ; Write A5 (Hex) to WDK
MOV WDK,#11H ; Write 11 (Hex) to WDK
               Software Reset generated via WDT
```

Dual Data Pointers

The Dual Data Pointer structure is the means by which the 80C521 Family may specify the address of an external Data Memory location. The Dual Data Pointer structure consists of two 16-bit registers that address external memory, and a single 8-bit register that allows the program code to selectively switch between them. They are located in the Special Function Register space at the following addresses:

82H Data Pointer Low	–(DPL)	Data Pointer 0 (DPTR0)
83H Data Pointer High	–(DPH)	
84H Data Pointer Low 1	–(DPL1)	Data Pointer 1 (DPTR1)
85H Data Pointer High 1	–(DPH1)	
86H Data Pointer Selection	–(DPS)	

Data Pointer 0 (DPTR0) is the original data pointer on the standard 80C51 (formerly referred to as DPTR). Data Pointer 1 (DPTR1) is an additional data pointer with identical characteristics. Instructions that refer to "DPTR" refer to the data pointer that is currently selected in the Data Pointer Selection (DPS) register. The six instructions that reference "DPTR" are as follows:

```
INC DPTR            ; Increments the data pointer by 1
MOV DPTR,#data16    ; Loads DPTR with a 16-bit constant
MOVC A, @A + DPTR   ; Move code byte relative to DPTR to
                      Acc
MOVX A, @DPTR       ; Move external RAM (16-bit address)
                      to Acc
MOVX @DPTR, A       ; Move Acc to external RAM (16-bit
                      address)
JMP @A + DPTR       ; Jump indirect relative to DPTR
```

It is also possible to access each data pointer on a byte-by-byte basis by specifying its low or high byte in an instruction that accesses the Special Function Registers. These instructions can be executed at any time regardless of which of the two data pointers is currently selected. Three examples are as follows:

```
MOV DPH,R3 ; Move the contents of Register 3 into DPH
MOV A,DPL1 ; Move the contents of DPL1 into the Acc
PUSH DPH1  ; Push the contents of DPH1 onto the stack
```

The Dual Data Pointer structure saves both time and code space by eliminating the need for frequent loading and unloading of a single data pointer. For instance, block move operations in external memory can be more efficiently implemented by using DPTR0 as the source address, and DPTR1 as the destination address. The Dual Data Pointer structure enhances this operation considerably.

Data Pointer Selection – –(DPS)– – Address: 86H

This register determines which of the two data pointers is currently "selected." Once a data pointer is selected, the six "DPTR" instructions refer only and always to that data pointer until another data pointer is selected. Upon reset, the default data pointer (DPTR0) will be selected, thus retaining compatibility with existing 8051 Family devices. The switch between data pointers may be accomplished with a single cycle instruction (such as: INC DPS or MOV DPS,A). The default value at reset = 00H. This is a Read/Write register.

(MSB)							(LSB)
0	0	0	0	0	0	0	SEL0
7	6	5	4	3	2	1	0

Bit 0 – –Select 0 – –(SEL0)

If this bit is 0, the original data pointer, DPTR0, is selected. If this bit is 1, DPTR1 is selected. This bit may be written by software at any time. When read, its current value is presented.

Bits 7-1

Reserved. Will return 0 when read.

Data Pointer Low – –(DPL)– – Address: 82H

DPL is a Read/Write register that contains the low byte of Data Pointer 0. It may be accessed at any time with an instruction that specifies a direct byte as a source of destination. However, SEL0 in the DPS register must be set to 0 before any of the six explicit "DPTR" instructions will access this register. The default at reset = 00H.

(MSB)							(LSB)
7	6	5	4	3	2	1	0

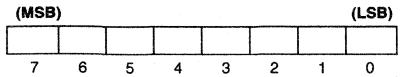
Data Pointer High – –(DPH)– – Address: 83H

DPH is a Read/Write register that contains the high byte of Data Pointer 0. It may be accessed at any time with an instruction that specifies a direct byte as a source or destination. However, SEL0 in the DPS register must be set to 0 before any of the six explicit "DPTR" instructions will access this register. The default at reset = 00H.

(MSB)							(LSB)
7	6	5	4	3	2	1	0

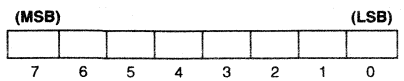
Data Pointer Low 1 - - (DPL1) - - Address: 84H

DPL1 is a Read/Write register that contains the low byte of Data Pointer 1. It may be accessed at any time with an instruction that specifies a direct byte as a source or destination. However, SEL0 in the DPS register must be set to 1 before any of the six explicit "DPTR" instructions will access this register. The default at reset = 00H.



Data Pointer High 1 - - (DPH1) - - Address: 85H

DPH1 is a Read/Write register that contains the high byte of Data Pointer 1. It may be accessed at any time with an instruction that specifies a direct byte as a source or destination. However, SEL0 in the DPS register must be set to 1 before any of the six explicit "DPTR" instructions will access this register. The default at reset = 00H.



Dual Data Pointer Example

To load both data pointers after reset:

```
Method-1:
MOV DPL ,#data8          ; load low byte of DPTR0
MOV DPH ,#data8          ; load high byte of DPTR0
MOV DPL1,#data8          ; load low byte of DPTR1
MOV DPH1,#data8          ; load high byte of DPTR1
 (Data Pointer 0 is still selected.)
Method-2:
MOV DPTR,#data16         ; load DPTR0 with 16-bit const.
INC DPS                  ; switch data pointers
MOV DPTR,#data16         ; load DPTR1 with 16-bit const.
 (Data Pointer 1 is now selected.)
```

Oscillator Characteristics

$XTAL_1$ and $XTAL_2$ are the input and output, respectively, of an inverting amplifier which is configured for use as an on-chip oscillator (see Figure 2). Either a quartz crystal or ceramic resonator may be used.

To drive the device from an external clock source, $XTAL_1$ should be driven while $XTAL_2$ is left unconnected (see Figure 3). There are no requirements on the duty cycle of the external-clock signal since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum HIGH and LOW times specified on the data sheet must be observed.

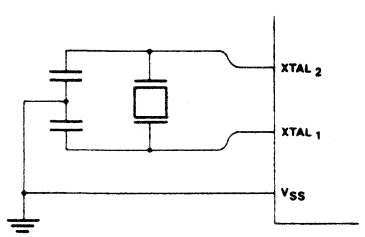
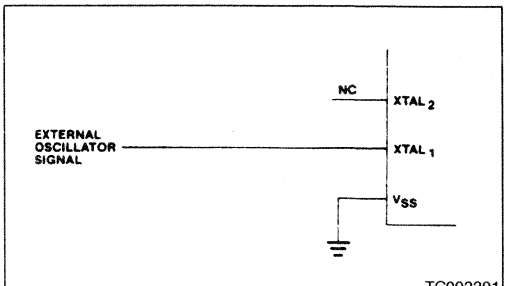


Figure 2. Crystal Oscillator



Note: Different from NMOS 8051

Figure 3. External Drive Configuration

Idle and Power-Down Operation

Figure 4 shows the internal operation of the Idle and Power-Down circuitry. Power-Down operation disconnects the clock source from all internal chip circuitry. Idle mode operation allows the interrupt, serial port, timers, and watchdog circuitry to continue to function, while the CPU is stopped. If the Watchdog Timer is enabled, Power-Down operation is not possible.

These special modes are activated by software via the Special Function Register, PCON (Table 1). Its hardware address is 87H; PCON is not bit-addressable.

If "1"s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is "0XXX0000".

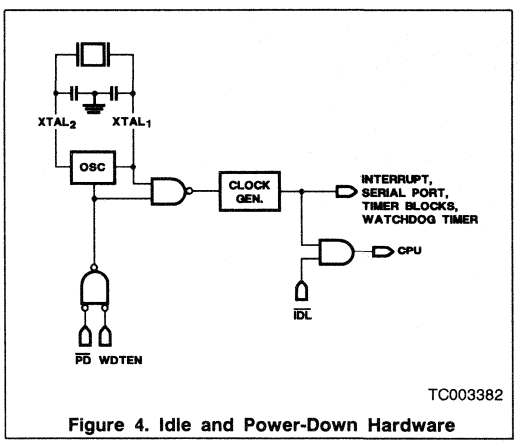


Figure 4. Idle and Power-Down Hardware

TABLE 1. PCON (Power Control Register)

(MSB)							(LSB)
SMOD	-	-	-	GF1	GF0	PD	IDL

Symbol	Position	Name and Description
SMOD	PCON.7	Double-baud-rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2, or 3.
-	PCON.6	(Reserved)
-	PCON.5	(Reserved)
-	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit
GF0	PCON.2	General-purpose flag bit
PD	PCON.1	Power-Down bit. Setting this bit activates power-down operation.
IDL	PCON.0	Idle-mode bit. Setting this bit activates idle-mode operation.

Idle Mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM, and all other registers in the 80C521 maintain their data during Idle. Table 2 describes the status of the external pins during Idle mode.

There are three possible ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following RETI, the next instruction to be executed will be the one following the instruction that wrote a 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an enabled interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with a Hardware Reset.

The third way of terminating the Idle mode is with the Watchdog Timer. If the WDT is not enabled, then it has no effect on subsequent Idle mode operations. If the WDT is enabled before Idle mode is entered, it will continue to increment in the normal fashion. If the WDT overflows, the 80C521 will experience a Watchdog Reset and Idle mode will be terminated. If Idle mode is terminated by any method other than a reset, the Watchdog Timer will continue to run.

Power-Down Mode

The instruction that sets PCON.1 is the last executed prior to going into Power-Down. Once in Power-Down, the oscillator is stopped. The contents of the on-chip RAM are preserved. The Special Function Registers are saved, until a Hardware Reset is generated. A hardware reset is the only way of exiting the Power-Down mode.

Power-Down mode cannot be entered while the Watchdog Timer is enabled. If a write of the value 1 is attempted into the PD bit of the PCON register, its value will remain 0, and no Power-Down operation will take place. To enter Power-Down mode, the Watchdog Timer must first be disabled via a Hardware Reset, Software Reset, or Watchdog Reset. After reset, the Watchdog Timer is disabled, allowing Power-Down mode to be entered.

In the Power-Down mode, V_{CC} may be lowered to minimize circuit power consumption. Care must be taken to ensure the voltage is not reduced until the Power-Down mode is entered, and that the voltage is restored before the Hardware Reset is applied, Hardware Reset frees the oscillator and should not be released until the oscillator has restarted and stabilized.

Table 2 describes the status of the external pins while in the Power-Down mode. It should be noted that if the Power-Down mode is activated while in external program memory, the port data that is held in the Special Function Register P_2 is restored to Port 2. If the data is a 1, the port pin is held HIGH during the Power-Down mode by the strong pullup, P_1, shown in Figure 5.

80C521 I/O Ports

The I/O port drive of the 80C521 is similar to the 8051. The I/O buffers for Ports 1, 2, and 3 are implemented as shown in Figure 5.

When the port latch contains a 0, all pFETS in Figure 4 are off while the nFET is turned on. When the port latch makes a 0-to-1 transition, the nFET turns off. The strong pullup pFET, P_1, turns on for two oscillator periods, pulling the output HIGH very rapidly. As the output line is drawn HIGH, pFET P_3 turns on through the inverter to supply the I_{OH} source current. This inverter and P_3 form a latch which holds the 1 and is supported by P_2.

When Port 2 is used as an address port, for access to external program or data memory, any address bit that contains a 1 will have its strong pullup turned on for the entire duration of the external memory access.

When an I/O pin on Ports 1, 2, or 3 is used as an input, the user should be aware that the external circuit must sink current during the logical 1-to-0 transition. The maximum sink current is specified as I_{TL} under the D.C. Specifications. When the input goes below approximately 2 V, P_3 turns off to save I_{CC} current. Note, when returning to a logical 1, P_2 is the only internal pullup that is on. This will result in a slow rise time if the user's circuit does not force the input line HIGH.

TABLE 2. STATUS OF THE EXTERNAL PINS DURING IDLE AND POWER-DOWN MODES

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data
Power-Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power-Down	External	0	0	Floating	Port Data	Port Data	Port Data

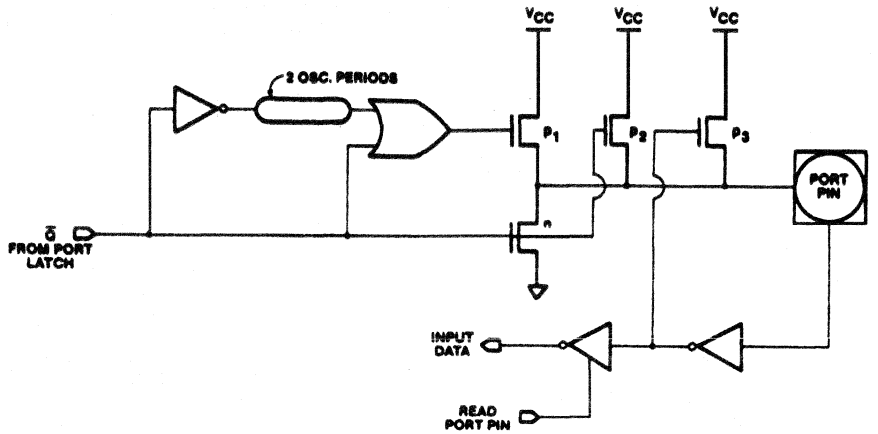
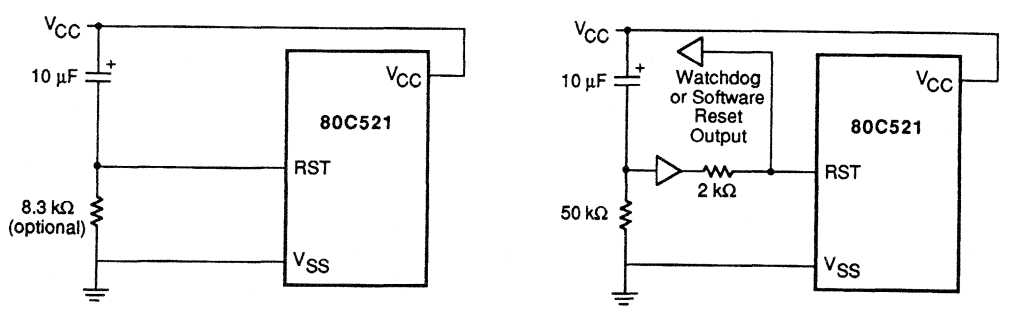


Figure 5. I/O Buffers in the 80C521 (Ports 1, 2, 3)



Standard (80C51) Reset Circuit

Neither a Watchdog nor a Software Reset will affect the "Standard" reset circuitry, nor can they be sensed by the "Standard" (80C51) reset circuitry.

Watchdog Reset Circuit

The reset circuit shown above may be used to sense a Watchdog or Software Reset. For $V_{CC} = 5$ V, the driver input must be able to source/sink 2.5 mA.

Figure 6. RESET Configurations

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any
Pin to V_{SS} ..... −0.5 V to V_{CC} +0.5 V
Voltage on V_{CC} to V_{SS} ..... −0.5 V to 6.5 V
Power Dissipation ..... 200 mW

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4 V to +6 V
Ground (V_{SS}) ..... 0 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
V_{IL}	Input LOW Voltage (Except $\overline{EA}$)		−0.5	.2 V_{CC} − .1	V
V_{IL1}	Input LOW Voltage ($\overline{EA}$)		−0.5	.2 V_{CC} − .3	V
V_{IH}	Input HIGH Voltage (Except $XTAL_1$, RST)		.2 V_{CC} + .9	V_{CC} + 0.5	V
V_{IH1}	Input HIGH Voltage ($XTAL_1$ RST)		.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output LOW Voltage (Ports 1, 2, 3)	I_{OL} = 1.6 mA (Note 1)		0.45	V
V_{OL1}	Output LOW Voltage (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA (Note 1)		0.45	V
V_{OH}	Output HIGH Voltage (Ports 1, 2, 3)	I_{OH} = −60 μA, V_{CC} = 5 V±10%	2.4		V
		I_{OH} = −25 μA	.75 V_{CC}		V
		I_{OH} = −10 μA	.9 V_{CC}		V
V_{OH1}	Output HIGH Voltage (Port 0 in External Bus Mode, ALE, $\overline{PSEN}$)	I_{OH} = −400 μA, V_{CC} = 5 V±10%	2.4		V
		I_{OH} = −150 μA	.75 V_{CC}		V
		I_{OH} = −40 μA (Note 2)	.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)	V_{IN} = 0.45 V		−50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3)	V_{IN} = 2 V		−650	μA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	μA
CIO	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{PD}	Power Down Current	V_{CC} = 2 to 6 V (Note 3)		50	μA

MAXIMUM I_{CC} (mA)

	Operating (Note 4)			Idle (Note 5)		
Freq. V_{CC}	4 V	5 V	6 V	4 V	5 V	6 V
3.5 MHz	6	8	10	1.5	2	3
8.0 MHz	11	14	18	2.5	3.5	5
12 MHz	15	20	25	3.5	5	6
16 MHz	19	25	32	4.5	6.5	8.5

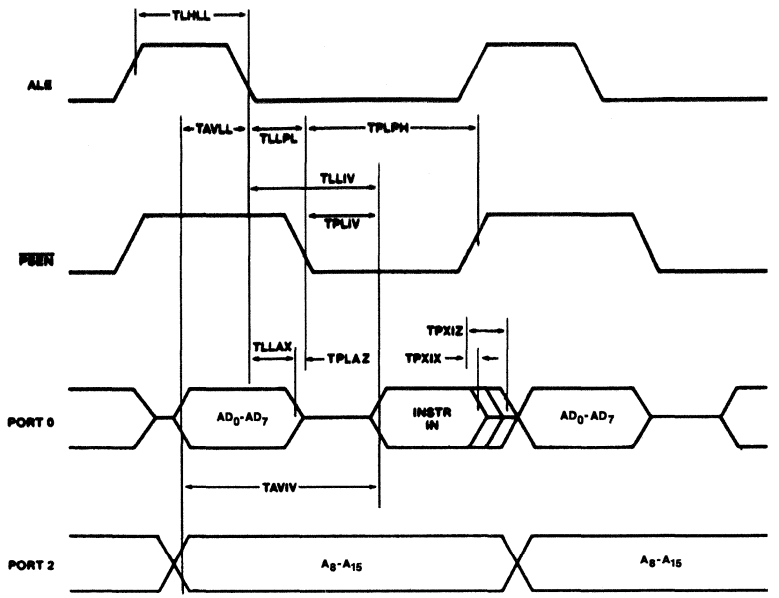
Notes:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8 V. 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 $\overline{PSEN}$ to momentarily fall below the .9 V_{CC} specification when the address bits are stabilizing.
3. Power-Down I_{CC} is measured with all outputs pins disconnected: $\overline{EA}$ = Port 0 = V_{CC}; $XTAL_2$ N.C.; RST = V_{SS}.
4. I_{CC} is measured with all output pins disconnected; $XTAL_1$ driven with TCLCH, TCHCL = 5 ns, V_{IL} = V_{SS} + .5 V, V_{IH} = V_{CC} − .5 V; $XTAL_2$ N.C.; $\overline{EA}$ = RST = Port 0 = 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; $XTAL_1$ driven with TCLCH, TCHCL = 5 ns, V_{IL} = V_{SS} + .5 V, V_{IH} = V_{CC} − .5 V; $XTAL_2$ N.C.; Port 0 = V_{CC}; $\overline{EA}$ = RST = V_{SS}, and the Watchdog Timer disabled.

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified (C_L for Port 0, ALE and $\overline{PSEN}$ Outputs = 100 pF; C_L for All Other Outputs = 80 pF)

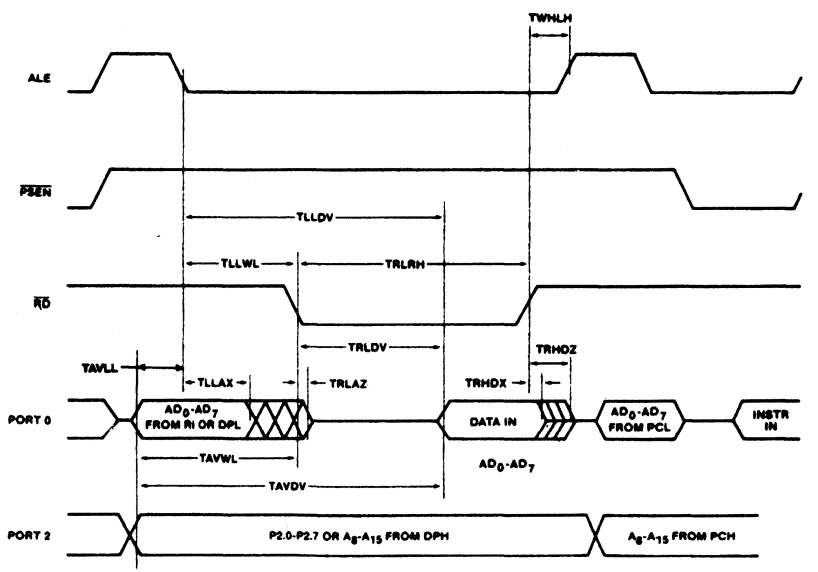
Parameter Symbol	Parameter Description	16 MHz Osc.		12 MHz Osc.		Variable Oscillator		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
External Program and Data Memory Characteristics								
1/TCLCL	Oscillator Frequency					3.5	16	MHz
TLHLL	ALE Pulse Width	85		127		2TCLCL – 40		ns
TAVLL	Address Valid to ALE LOW	7		28		TCLCL – 55		ns
TLLAX	Address Hold After ALE LOW	27		48		TCLCL – 35		ns
TLLIV	ALE LOW to Valid Instr. In		150		234		4TCLCL – 100	ns
TLLPL	ALE LOW to $\overline{PSEN}$ LOW	22		43		TCLCL – 40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	142		205		3TCLCL – 45		ns
TPLIV	$\overline{PSEN}$ LOW to Valid Instr. In		83		145		3TCLCL – 105	ns
TPXIX	Input Instr. Hold After $\overline{PSEN}$	0		0		0		ns
TPXIZ	Input Instr. Float After $\overline{PSEN}$		38		59		TCLCL – 25	ns
TAVIV	Address to Valid Instr. In		208		312		5TCLCL – 105	ns
TPLAZ	$\overline{PSEN}$ LOW to Address Float		10		10		10	ns
TRLRH	$\overline{RD}$ Pulse Width	275		400		6TCLCL – 100		ns
TWLWH	$\overline{WR}$ Pulse Width	275		400		6TCLCL – 100		ns
TRLDV	$\overline{RD}$ LOW to Valid Data In		148		252		5TCLCL – 165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		0		ns
TRHDZ	Data Float After $\overline{RD}$		55		97		2TCLCL – 70	ns
TLLDV	ALE LOW to Valid Data In		350		517		8TCLCL – 150	ns
TAVDV	Address to Valid Data In		398		585		9TCLCL – 165	ns
TLLWL	ALE LOW to $\overline{RD}$ or $\overline{WR}$ LOW	137	238	200	300	3TCLCL – 50	3TCLCL + 50	ns
TAVWL	Address Valid to Read or Write LOW	120		203		4TCLCL-130		
TQVWX	Data Valid to $\overline{WR}$ Transition	2		23		TCLCL – 60		ns
TQVWH	Valid Data to Write HIGH	287		433		7TCLCL-59		ns
TWHQX	Data Hold After $\overline{WR}$	12		33		TCLCL – 50		ns
TRLAZ	$\overline{RD}$ LOW to Address Float		0		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	22	103	43	123	TCLCL – 40	TCLCL + 40	ns

SWITCHING WAVEFORMS



WF021961

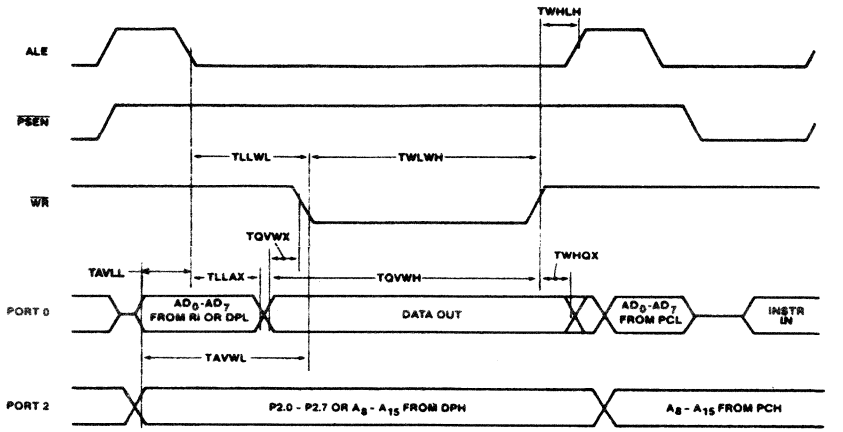
External Program Memory Read Cycle



WF020961

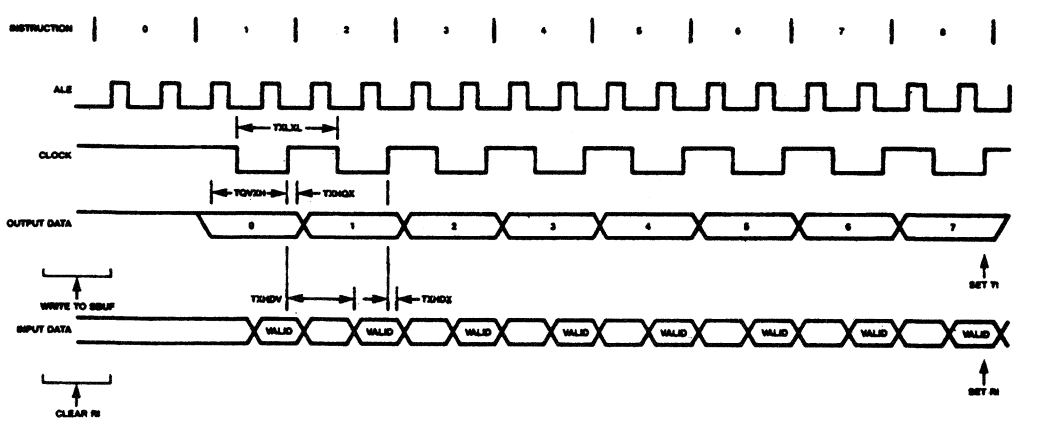
External Data Memory Read Cycle

SWITCHING WAVEFORMS (Cont'd.)



WF020931

External Data Memory Write Cycle

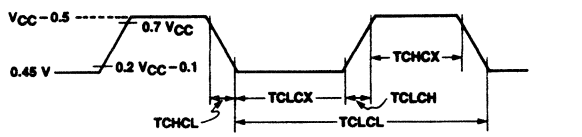


WF020950

Shift Register Timing Waveforms

EXTERNAL CLOCK DRIVE

Parameter Symbol	Parameter Description	Min.	Max.	Units
1/TCLCL	Oscillator Frequency	3.5	16	MHz
TCHCX	HIGH Time	20		ns
TCLCX	LOW Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns



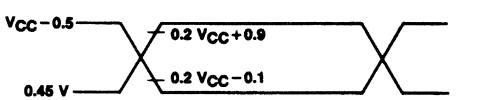
External Clock Drive Waveform

SERIAL PORT TIMING — SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to 70°C; V_{CC} = 5 V ±20%; V_{SS} = 0 V; Load Capacitance = 80 pF

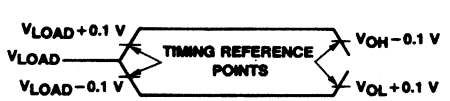
Parameter Symbol	Parameter Description	16 MHz Osc.		Variable Oscillator		Units
		Min.	Max.	Min.	Max.	
TXLXL	Serial Port Clock Cycle Time	750		12TCLCL		ns
TQVXH	Output Data Setup to Clock Rising Edge	492		10TCLCL – 133		ns
TXHQX	Output Data Hold After Clock Rising Edge	8		2TCLCL – 117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		492		10TCLCL – 133	ns

AC Testing



AC INPUTS DURING TESTING ARE DRIVEN AT V_{CC} – 0.5 FOR A LOGIC "1" AND 0.45 V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT V_{IH} MIN. FOR A LOGIC "1" AND V_{IL} MAX. FOR A LOGIC "0."

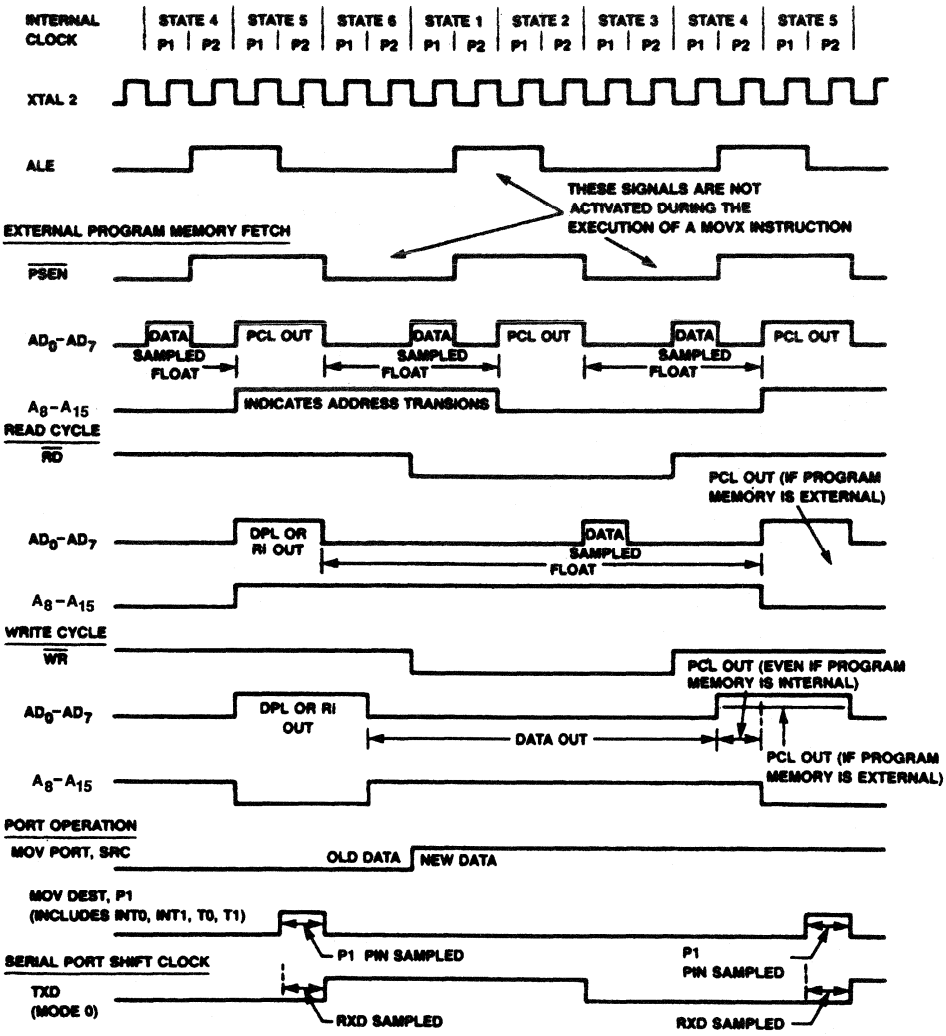
Input/Output Waveform



FOR TIMING PURPOSES A PORT PIN IS NO LONGER FLOATING WHEN A 100mV 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.

Float Waveform

CLOCK WAVEFORMS



WF020922

This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component to component. Typically though, (T_A = 25°C, fully loaded) $\overline{RD}$ and $\overline{WR}$ propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

TABLE 2. 80C521/80C321 INSTRUCTION SET

Instructions That Affect Flag Setting*

Instruction	Flag			Instruction	Flag		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	O	X		ANL C,/bit	X		
DIV	O	X		ORL C, bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C, bit	X		
RLC	X			CJNE	X		
SETB C	1						

Interrupt Response Time: To finish execution of current instruction, respond to the interrupt request and push the PC; to vector to the first instruction of the interrupt service program requires 38 to 81 oscillator periods (2.25 to 5.25 μs at 16 MHz).

*Note that operations on SFR byte address D0H or bit addresses D0 – D7H (i.e., the PSW or bits in the PSW) will also affect flag settings.

DATA TRANSFER

Mnemonic		Description	Byte	Cyc
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct byte	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Mode direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Move 16-bit constant to Data Pointer	3	2
MOVC	A,@A + DPTR	Move Code byte relative to DPTR to Accumulator	1	2
MOVC	A,@A + PC	Move Code byte relative to PC to Accumulator	1	2
MOVX	A,@Ri	Move External RAM (8-bit address) to Accumulator	1	2
MOVX	A,@DPTR	Move External RAM (16-bit address) to Accumulator	1	2
MOVX	@Ri,A	Move Accumulator to External RAM (8-bit address)	1	2
MOVX	@DPTR,A	Move Accumulator to External RAM (16-bit address)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte off of stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	1
XCHD	A,@Ri	Exchange indirect RAM's least sig nibble with A's LSN	1	1

BOOLEAN VARIABLE MANIPULATION

Mnemonic		Description	Byte	Cyc
CLR	C	Clear Carry Flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry Flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement Carry Flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry Flag	2	2
ANL	C,/bit	AND complement of direct bit to Carry	2	2
ORL	C,bit	OR direct bit to Carry Flag	2	2
ORL	C,/bit	OR complement of direct bit to Carry	2	2
MOV	C,bit	Move direct bit to Carry Flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2

LOGIC

Mnemonic		Description	Byte	Cyc
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1

LOGIC (Cont'd.)

Mnemonic		Description	Byte	Cyc
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	1
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate Accumulator Left through Carry Flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate Accumulator Right through Carry Flag	1	1
SWAP	A	Exchange nibbles within the Accumulator	1	1

ARITHMETIC

Mnemonic		Description	Byte	Cyc
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with carry	1	1
ADDC	A,direct	Add direct byte to Accumulator with Carry Flag	2	1
ADDC	A,@Ri	Add indirect RAM and Carry Flag to Accumulator	1	1
ADDC	A,#data	Add immediate data and Carry Flag to Accumulator	2	1
SUBB	A,Rn	Subtract register from Accumulator with Borrow	1	1
SUBB	A,direct	Subtract direct byte from Accumulator with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from Accumulator with Borrow	1	1
SUBB	A,#data	Subtract immediate data from Accumulator with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
MUL	AB	Multiply Accumulator times B	1	4
DIV	AB	Divide Accumulator by B	1	4
DA	A	Decimal Adjust Accumulator	1	1

OTHER

Mnemonic	Description	Byte	Cyc
NOP	No Operation	1	1

CONTOL TRANSFER (BRANCH)

Mnemonic	Description	Byte	Cyc
AJMP addr11	Absolute Jump	2	2
LJMP addr16	Long Jump	3	2
SJMP rel	Short Jump (relative addr)	2	2
JMP @A + DPTR	Jump indirect relative to the DPTR	1	2
JZ rel	Jump if Accumulator is zero	2	2
JNZ rel	Jump if Accumulator is not zero	2	2
JC rel	Jump if Carry Flag is set	2	2
JNC rel	Jump if carry is not set	2	2
JB bit,rel	Jump relative if direct bit is set	3	2
JNB bit,rel	Jump relative if direct bit is not set	3	2
JBC bit,rel	Jump relative if direct bit is set, then clear bit	3	2
CJNE A,direct,rel	Compare direct byte to Accumulator and Jump if not Equal	3	2
CJNE A,#data,rel	Compare immediate to Accumulator and Jump if not Equal	3	2
CJNE Rn,#data,rel	Compare immediate to reg and Jump if not Equal	3	2
CJNE @Ri,#data,rel	Compare immediate to indirect RAM and Jump if not Equal	3	2
DJNZ Rn,rel	Decrement register and Jump if not zero	2	2
DJNZ direct,rel	Decrement direct byte and Jump if not zero	3	2

CONTROL TRANSFER (SUBROUTINE)

Mnemonic	Description	Byte	Cyc
ACALL addr11	Absolute Subroutine Call	2	2
LCALL addr16	Long Subroutine Call	3	2
RET	Return from Subroutine Call	1	2
RETI	Return from Interrupt Call	1	2

Notes on Data Addressing Modes:

- Rn –Working register R0 – R7 of the currently selected Register bank.
- direct –128 internal RAM locations, any I/O port, control, or Special Function Registers.
- @Ri –Indirect internal RAM location addressed by register R0 or R1.
- #data –8-bit constant included in instruction.
- #data16 –16-bit constant included as bytes 2 and 3 of instruction.
- bit –128 software flags, any I/O pin, control, or status bit.

Notes on Program Addressing Modes:

- addr16 –Destination address for LCALL and LJMP may be anywhere within the 64-Kilobyte program memory address space.
- addr11 –Destination address for ACALL and AJMP will be within the same 2-Kilobyte page of program memory as the first byte of the following instruction.
- rel –SJMP and all conditional jumps include as 8-bit offset by Range is +127, –128 bytes relative to first byte of the following instruction.

TABLE 3. INSTRUCTION OPCODES IN HEXADECIMAL ORDER (Cont'd.)

Hex Code	Bytes	Mnemonic	Operands
00	1	NOP	
01	2	AJMP	Code addr
02	3	LJMP	Code addr
03	1	RR	A
04	1	INC	A
05	2	INC	Data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	Bit addr,code addr
11	2	ACALL	Code addr
12	3	LCALL	Code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	Data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	Bit addr,code addr
21	2	AJMP	Code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,#data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	Bit addr,code addr
31	2	ACALL	Code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	Code addr
41	2	AJMP	Code addr
42	2	ORL	Data addr,A
43	3	ORL	Data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	Code addr
51	2	ACALL	Code addr

Hex Code	Bytes	Mnemonic	Operands
52	2	ANL	Data addr,A
53	3	ANL	Data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	Code addr
61	2	AJMP	Code addr
62	2	XRL	Data addr,A
63	3	XRL	Data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	Code addr
71	2	ACALL	Code addr
72	2	ORL	C,bit addr
73	1	JMP	@A +DPTR
74	2	MOV	A,#data
75	3	MOV	Data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	Code addr
81	2	AJMP	Code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A +PC
84	1	DIV	AB
85	3	MOV	Data addr,data addr
86	2	MOV	Data addr,@R0
87	2	MOV	Data addr,@R1
88	2	MOV	Data addr,R0
89	2	MOV	Data addr,R1
8A	2	MOV	Data addr,R2
8B	2	MOV	Data addr,R3
8C	2	MOV	Data addr,R4
8D	2	MOV	Data addr,R5
8E	2	MOV	Data addr,R6
8F	2	MOV	Data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	Code addr
92	2	MOV	Bit addr,C
93	1	MOVC	A,@A +DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,/bit addr
A1	2	AJMP	Code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		Reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,/bit addr
B1	2	ACALL	Code addr
B2	2	CPL	Bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	Data addr
C1	2	AJMP	Code addr
C2	2	CLR	Bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	Data addr
D1	2	ACALL	Code addr
D2	2	SETB	Bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	Data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	Code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	Code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	Data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

8751H/8753H

Single-Chip 8-Bit Microcontroller with
4K/8K Bytes of EPROM

DISTINCTIVE CHARACTERISTICS

- 4K x 8 EPROM (8751H); 8K x 8 EPROM (8753H)
- 128 x 8 RAM
- Four 8-bit ports, 32 I/O lines; programmable serial port
- Two 16-bit Timer/Event counters
- 64K addressable Program and Data Memory
- Boolean processor
- Five interrupt sources/two priority levels
- 4-cycle multiply and divide
- Program memory security feature
- Fast EPROM programming: 12 sec for 4K bytes
- Supports silicon signature verification
- Pin compatible with 8051

GENERAL DESCRIPTION

The 8751H and 8753H are members of a family of advanced single-chip microcontrollers. Both the 8751H, which has 4K bytes of EPROM, and the 8753H, which has 8K bytes of EPROM, are pin-compatible EPROM versions of the 8051AH and 8053AH, respectively. Thus, the 8751H/8753H are full-speed prototyping tools which provide effective single-chip solutions for controller applications that require code modification flexibility. Refer to the block diagram of the 8051 family.

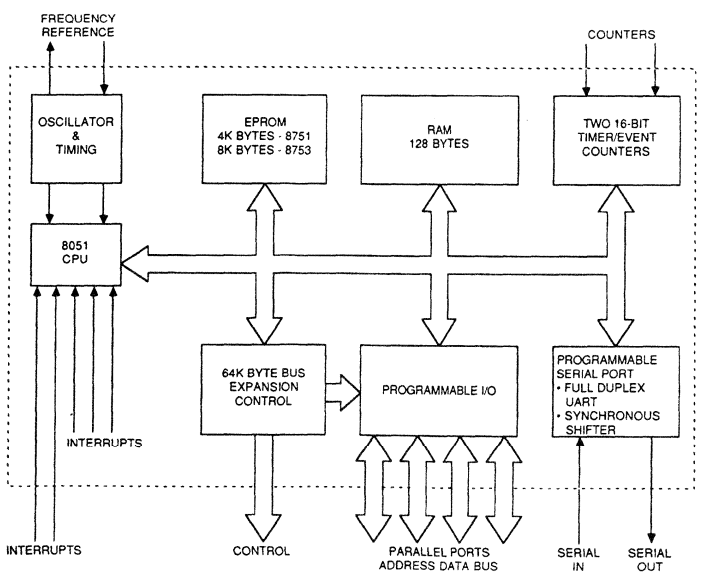
The 8751H/8753H devices feature: thirty-two I/O lines; two 16-bit timer/event counters; a Boolean processor; a 5-source, bi-level interrupt structure; a full-duplex serial channel; and on-chip oscillator and clock circuitry.

Program and Data Memory are located in independent addresses. The AMD family of microcontrollers can access up to 64K bytes of external Program Memory and up to 64K bytes of external Data Memory. The 8751H and the 8753H contain the lower 4K and 8K bytes of Program Memory, respectively, on-chip. Both parts have 128 bytes of on-chip read/write data memory.

The AMD 8051 Microcontroller Family is specifically suited for control applications. A variety of fast addressing modes, which access the internal RAM, facilitates byte processing and numerical operations on small data structures. Included in the instruction set is a menu of 8-bit arithmetic instructions, including 4-cycle multiply and divide instructions.

Extensive on-chip support enables direct bit manipulation and testing of 1-bit variables as separate data types. Thus, the device is also suited for control and logic systems that require Boolean processing.

BLOCK DIAGRAM

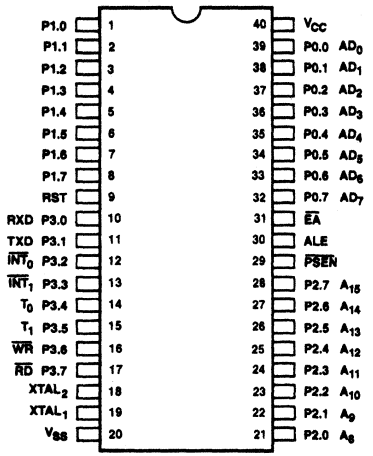


BD007250

Publication # 03896 | Rev. D | Amendment /0
Issue Date: July 1987

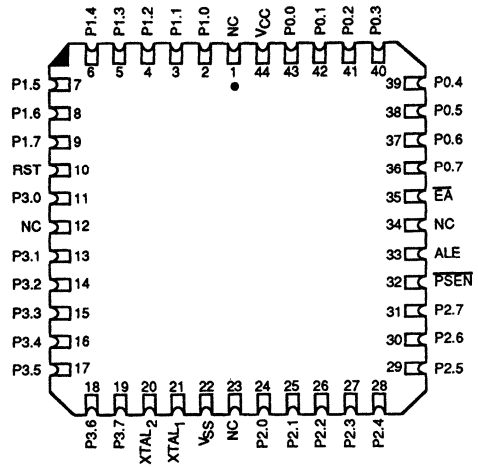
CONNECTION DIAGRAMS
Top View

DIP



CD005551

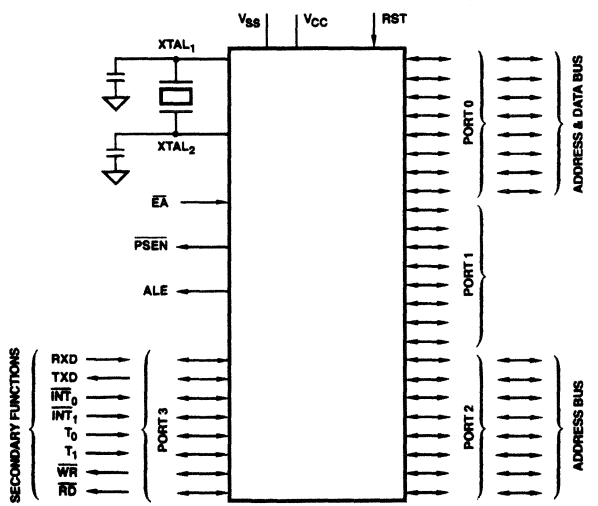
LCC



CD010870

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL

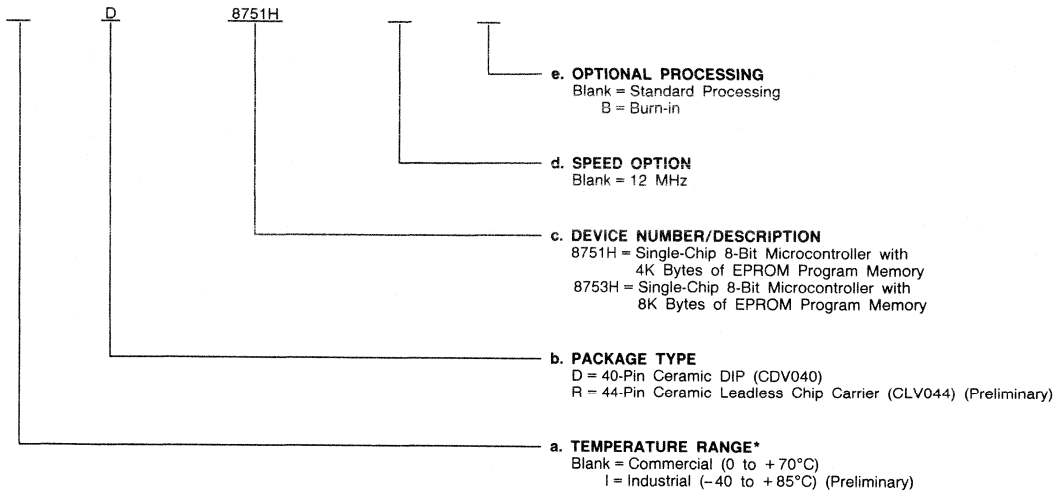


LS001325

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
D	8751H
	8753H

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

PIN DESCRIPTION

Port 0 (Bidirectional; Open Drain)

Port 0 is an open-drain I/O port. Port 0 pins that have "1"s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed LOW-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting "1"s. Port 0 also outputs the code bytes during program verification in the 8751H and 8753H. External pullups are required during program verification.

Port 1 (Bidirectional)

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four LS TTL inputs. Port 1 pins that have "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 1 pins that are externally being pulled LOW will source current (I_{IL} on the data sheet) because of the internal pullups.

Port 1 also recieves the LOW-order address bytes during program verification.

Port 2 (Bidirectional)

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four LS TTL inputs. Port 2 pins having "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 2 pins externally being pulled LOW will source current (I_{IL}) because of internal pullups.

Port 2 emits the HIGH-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting "1"s. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function register.

Port 2 also recieves the HIGH-order address bits during the programming of the EPROM and during program verification of the EPROM.

Port 3 (Bidirectional)

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four LS TTL inputs. Port 3 pins having "1"s written to them are pulled HIGH by the internal pullups and — while in this state — can be used as inputs. As inputs, Port 3 pins externally being pulled LOW will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features as listed below:

Port Pin	Alternate Function
$P_{3.0}$	RxD (Serial Input Port)
$P_{3.1}$	TxD (Serial Output Port)
$P_{3.2}$	$\overline{INT_0}$ (External Interrupt 0)
$P_{3.3}$	$\overline{INT_1}$ (External Interrupt 1)
$P_{3.4}$	T_0 (Timer 0 External Input)
$P_{3.5}$	T_1 (Timer 1 External Input)
$P_{3.6}$	$\overline{WR}$ (External Data Memory Write Strobe)
$P_{3.7}$	$\overline{RD}$ (External Data Memory Read Strobe)

RST/V_{PD} Reset (Input; Active HIGH)

This pin is used to reset the device when held HIGH for two machine cycles while the oscillator is running. If RST/V_{PD} is held within the V_{PD} spec, it will supply standby power to the RAM in the event that V_{CC} drops below its spec. When RST/V_{PD} is LOW, the RAM's bias is drawn from V_{CC}. A small internal resistor permits power-on reset using a capacitor connected to V_{CC}.

ALE/$\overline{PROG}$ Address Latch Enable/$\overline{Program\ Pulse}$ (Input/Output)

Address Latch Enable output pulse for latching the LOW byte of the address during accesses to external memory. ALE can drive eight LS TTL inputs.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, allowing use for external-timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory. This pin also accepts the program pulse input ($\overline{PROG}$) when programming the EPROM.

$\overline{PSEN}$ $\overline{Program\ Store\ Enable}$ (Output; Active LOW)

$\overline{PSEN}$ is the read strobe to external Program Memory. $\overline{PSEN}$ can drive eight LS TTL inputs. When the device is executing code from an external program memory, $\overline{PSEN}$ is activated twice each machine cycle — except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory. $\overline{PSEN}$ is not activated during fetches from internal Program Memory.

$\overline{EA}$/V_{PP} $\overline{External\ Access\ Enable}$ (Input; Active LOW)

$\overline{EA}$ must be externally held LOW to enable the device to fetch code from external Program Memory locations 0000H to 0FFFH (0000H to 1FFFH in the 8753H). If $\overline{EA}$ is held HIGH, the 8751H executes from internal Program Memory unless the program counter contains an address greater than 0FFFH (1FFFH in the 8753H).

$XTAL_1$ Crystal (Input)

Input to the inverting oscillator amplifier. When an external oscillator is used, $XTAL_1$ should be grounded.

$XTAL_2$ Crystal (Output)

Output of the inverting oscillator amplifier. $XTAL_2$ is also the input for the oscillator signal when using an external oscillator.

V_{CC} Power Supply

V_{SS} Circuit Ground

PROGRAMMING

Programming the EPROM

To program the EPROM, either the internal or external oscillator must be running at 4 to 6 MHz because the internal bus is used to transfer address and program data to the appropriate internal registers.

The 8751H and 8753H devices support an adaptive EPROM programming algorithm in addition to the conventional EPROM programming algorithm. Adaptive device programming (sometimes called interactive or intelligent programming) adapts to the actual charge storage efficiency of each byte, so that no wasted programming time occurs and minimum device programming time is realized.

The typical resulting device programming time is a mere 7% of what is required for a conventional programming algorithm. For example, to program a 4K byte EPROM using the conventional programming algorithm will require 4K x 50 ms = 200 sec. If adaptive programming is used, the theoretical programming time required will be 4K x 3 ms = 12 sec. The actual speed advantage of the adaptive programming is still very significant even allowing for the additional software overhead to implement the adaptive algorithm (2 to 8 sec depending on the brand of EPROM programmer).

To program the 8751H, pins $P_{2.4} - P_{2.6}$ and $\overline{PSEN}$ should be held LOW, and $P_{2.7}$ and RST held HIGH as shown in Table 2. The address of the location to be programmed is applied to Port 1 and $P_{2.0} - P_{2.3}$ while the code byte to be programmed is applied to Port 0 (see Figure 1).

V_{PP} should be at 21 V during device programming and the ALE/$\overline{PROG}$ pin should be pulsed LOW for 1 ms to program the code byte into the addressed EPROM location. The programmed byte is verified immediately after programming.

Figure 3 illustrates the flow of the adaptive programming algorithm. At each address, up to 15 program/verify loops are attempted to verify the programmability of the byte using 1 ms $\overline{PROG}$ pulses. After the programmability of a byte is determined, an overprogramming pulse of 2 ms is applied to $\overline{PROG}$ to guarantee data retention. (This conforms with the AMD standard of 2 ms/byte overprogramming for all N-channel EPROMs.)

The programming of 8753H is similar to the above procedures except that pin $P_{2.4}$ is the additional address pin (A_{12}) for accessing the upper 4K bytes of the EPROM (see Figure 2).

The 8751H and 8753H can also be programmed using the less efficient conventional EPROM programming algorithm. In this method, V_{PP} is held at 21 V and $\overline{PROG}$ is pulsed low for 50 ms to program each code byte into the addressed EPROM location. After the memory is programmed, all addresses would be sequenced and verified.

A Note of Caution When Programming

The maximum voltage applied to the $\overline{EA}/V_{PP}$ pin must not exceed 21.5 V at any time as specified for V_{PP}. Even a slight spike can cause permanent damage to the device. The V_{PP} source should thus be well-regulated and glitch-free.

When programming, a 0.1×10^{-6}F capacitor is required across V_{PP} and ground to suppress spurious transients which may damage the device.

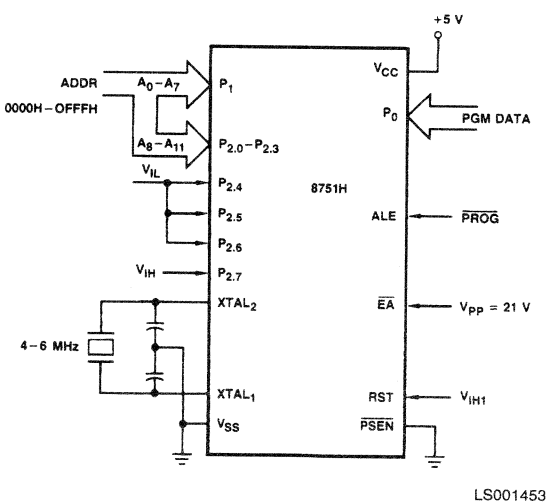


Figure 1. 8751H Programming Configuration

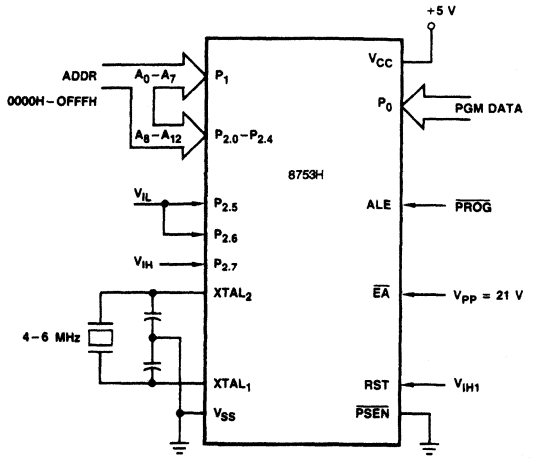


Figure 2. 8753H Programming Configuration

TABLE 1. EPROM PROGRAMMING MODES FOR THE 8751H

Mode	RST	$\overline{PSEN}$	ALE	$\overline{EA}$	$P_{2.7}$	$P_{2.6}$	$P_{2.5}$	$P_{2.4}$
Program	V_{IH1}	L	L*	V_{PP}	H	L	L	L
Inhibit	V_{IH1}	L	H	X	H	L	L	L
Verify	V_{IH1}	L	H	V_{PPX}	L	L	L	L
Security Set	V_{IH1}	L	L†	V_{PP}	H	H	L	X

Note: See notes below Table 2.

TABLE 2. EPROM PROGRAMMING MODES FOR THE 8753H

Mode	RST	$\overline{PSEN}$	ALE	$\overline{EA}$	$P_{2.7}$	$P_{2.6}$	$P_{2.5}$
Program	V_{IH1}	L	L*	V_{PP}	H	L	L
Inhibit	V_{IH1}	L	H	X	H	L	L
Verify	V_{IH1}	L	H	V_{PPX}	L	L	L
Security Set	V_{IH1}	L	L†	V_{PP}	H	H	L

Note: H = Logic HIGH for that pin
L = Logic LOW for that pin
X = Don't Care
V_{PP} = +21 V ±0.5 V
2.0 V < V_{PPX} < 21.5 V

*ALE is pulsed LOW for 1 msec in the programming loop of the adaptive programming algorithm and is pulsed LOW for 50 msec if conventional EPROM programming algorithm is used.
†ALE is pulsed LOW for 50 msec.

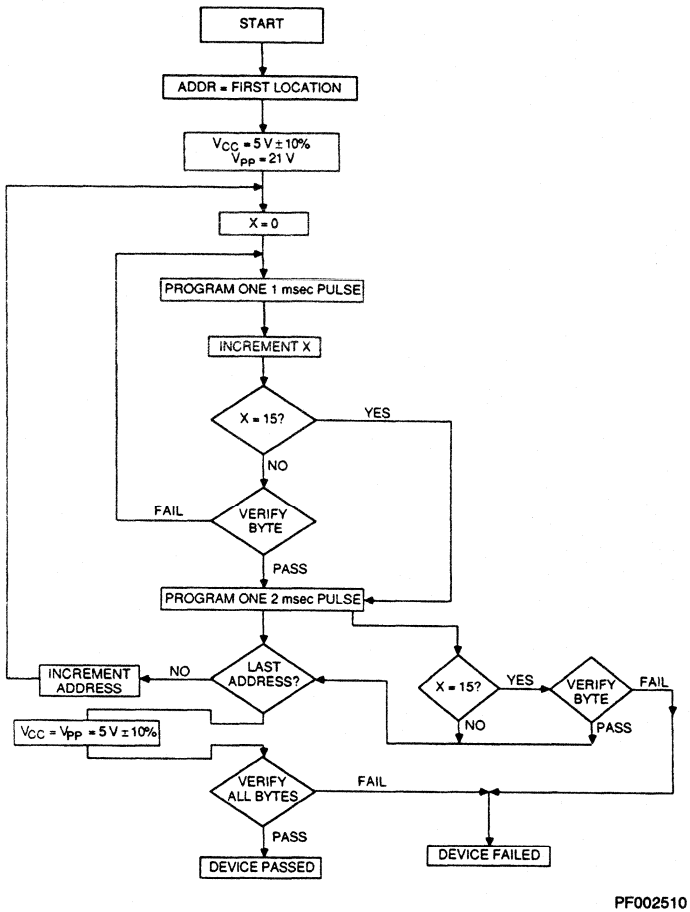


Figure 3. Adaptive Programming Algorithm for 8751H and 8753H

Program Verification

The Program Memory may be read out for verification purposes when the security bit has not been programmed. Reading the Program Memory may occur during or after programming of the EPROM. When the oscillator is running at 4 – 6 MHz, the 8751H Program Memory address location to be read is applied to Port 1 and pins $P_{2.0} - P_{2.3}$ of Port 2. Pins $P_{2.4} - P_{2.6}$ and $\overline{PSEN}$ are held at TTL LOW (see Figure 4). The 8753H utilizes Port 1 and pins $P_{2.0} - P_{2.4}$ to address the EPROM, while $P_{2.5} - P_{2.6}$ and $\overline{PSEN}$ are held LOW (see Figure 5).

The ALE/$\overline{PROG}$ and RST pins of both devices are held HIGH (RST requires only 2.5 V for HIGH) and the $\overline{EA}$/V_{PP} pin voltage can have any value from 2.0 V to 21.5 V as shown in Tables 1 and 2.

Port 0 will then output the contents of the address location. External pull-ups are needed on Port 0 when verifying the 8751H and 8753H EPROM.

Note: Since V_{PP} can be held at 21 V during program verification, the V_{PP} pin can be connected to a static 21 V power supply for device programming and verification in the adaptive device programming technique (see Figures 4 and 5).

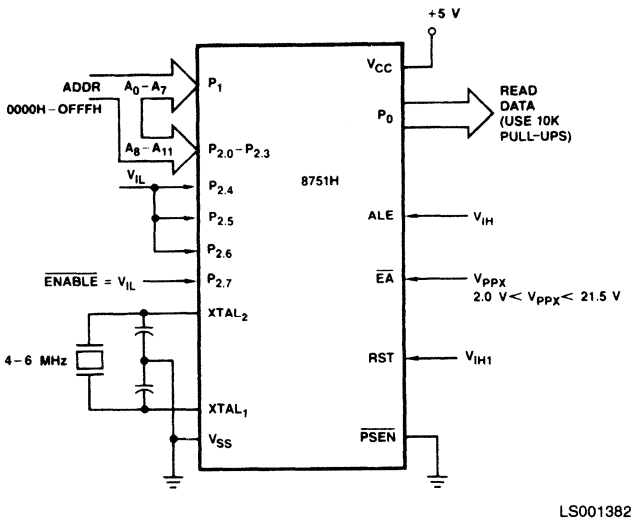


Figure 4. 8751H Program Verification

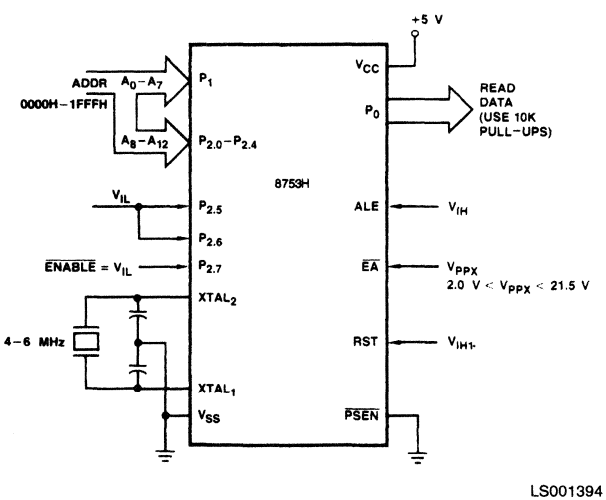


Figure 5. 8753H Program Verification

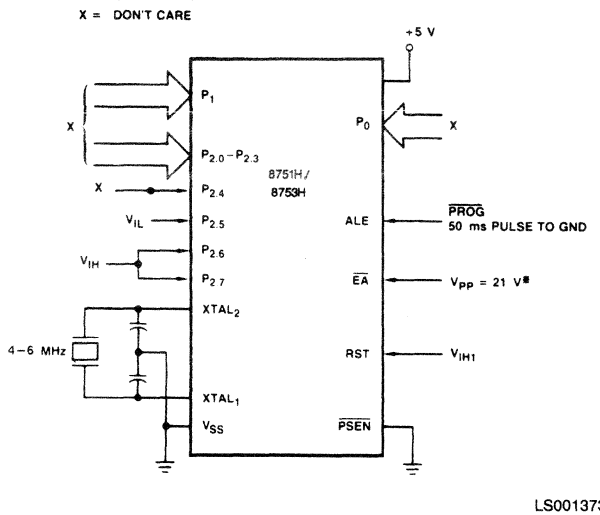
Security of the EPROM

The 8751H and 8753H incorporates a security bit, which when activated, prohibits all external readout of the on-chip EPROM contents. Figure 6 illustrates the security bit programming configuration for both the 8751H and 8753H. To activate the security bit, the same setup is used as when programming the EPROM except that $P_{2.6}$ is held HIGH. Port 0, Port 1 and pins $P_{2.0} - P_{2.3}$ may assume any state. V_{PP} should be at 21 V and the ALE/$\overline{PROG}$ pin should be pulsed LOW for 50 msec. The logic states of the other pins are detailed in Tables 1 and 2.

With the EPROM security bit programmed, retrieval of internal Program Memory cannot be achieved.

A secured Program Memory looks like a blank array of all ones, and this property can be used to verify that the EPROM is secured. The programmed security bit also prohibits further device programming and the execution of external Program Memory.

Full functionality and programmability may be restored by erasing the EPROM and thus clearing the security bit.



*When programming, a 0.1×10^{-6}F capacitor is required across V_{PP} and ground to suppress spurious transients which may damage the device.

Figure 6. Programming the Security Bit

4

Silicon Signature Verification

AMD will support silicon signature verification for the 8751H/8753H. To ensure that the device can be programmed according to the adaptive EPROM programming algorithm, the manufacturer code and part code can be read from the device before any programming is done.

To read the silicon signature, set up the conditions as specified in Figure 7. Note that $P_{2.5}$ is now required to be a TTL high level. Read the first byte of the silicon signature by applying address 0000H to the device; the byte should be a 01H, indicating AMD as the manufacturer. Then read the second byte of the silicon signature by applying address 0001H to the device; the byte should be 0DH, indicating the AMD 8751H/8753H product family.

Erasure Characteristics

Light and other forms of electromagnetic radiation can lead to erasure of the EPROM when exposed for extended periods of time.

Wavelengths of light shorter than 4000 angstroms, such as sunlight or indoor fluorescent lighting, can ultimately cause inadvertent erasure and should, therefore, not be allowed to expose the EPROM for lengthy durations (approximately one week in sunlight or three years in room-level fluorescent lighting). It is suggested that the window be covered with an opaque label if an application is likely to subject the device to this type of radiation.

It is recommended that ultraviolet light (of 2537 angstroms) be used to a dose of at least 15 W-sec/cm^2 when erasing the EPROM. An ultraviolet lamp rated at 12,000 μW/cm^2 held one inch away for 20 – 30 minutes should be sufficient.

EPROM erasure leaves the Program Memory in an "all ones" state.

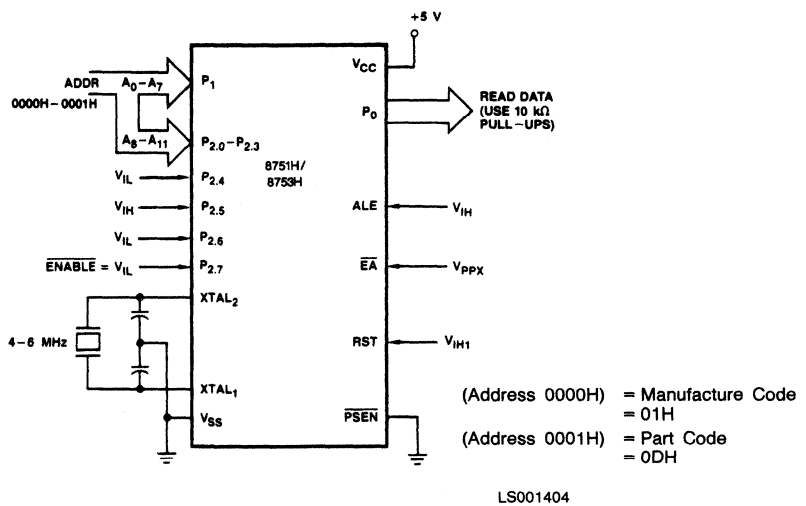


Figure 7. 8751H/8753H Silicon Signature Verification Configuration

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on $\overline{EA}$/V_{PP} Pin to V_{SS} ..... −0.5 to +21.5 V
Voltage on Any Other Pin to V_{SS} ..... −0.5 to +7 V
Power Dissipation ..... 1.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.5 to +5.5 V
Ground (V_{SS}) ..... 0 V

Industrial (I) Devices (Preliminary)
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... +4.5 to +5.5 V
Ground (V_{SS}) ..... 0 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
V_{IL}	Input LOW Voltage (Except $\overline{EA}$)		−0.5	0.8	V
V_{IL1}	Input LOW Voltage to $\overline{EA}$		0	0.7	V
V_{IH}	Input HIGH Voltage (Except $XTAL_2$, RST)		2.0	V_{CC} + 0.5	V
V_{IH1}	Input HIGH Voltage to $XTAL_2$, RST	$XTAL_1 = V_{SS}$	2.5	V_{CC} + 0.5	V
V_{OL}	Output LOW Voltage (Ports 1, 2, 3) (Note 1)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output LOW Voltage (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA I_{OL} = 2.4 mA		0.60 0.45	V
V_{OH}	Output HIGH Voltage (Ports 1, 2, 3)	I_{OH} = −80 μA	2.4		V
V_{OH1}	Output HIGH Voltage (Port 0 in External Bus Mode, ALE, $\overline{PSEN}$)	I_{OH} = −400 μA	2.4		V
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)	V_{IN} = 0.45 V		−500	μA
I_{IL1}	Logical 0 Input Current ($\overline{EA}$)			−15	mA
I_{IL2}	Logical 0 Input Current ($XTAL_2$)	V_{IN} = 0.45 V		−3.2	mA
I_{LI}	Input Leakage Current (Port 0)	$0.45 < V_{IN} < V_{CC}$		±100	μA
I_{IH}	Logical 1 Input Current ($\overline{EA}$)			500	μA
I_{IH1}	Input Current to RST to Activate Reset	$V_{IN} < (V_{CC} - 1.5\ V)$		500	μA
I_{CC}	Power Supply Current	All Outputs Disconnected; $\overline{EA} = V_{CC}$		250	mA
C_{IO}	Pin Capacitance	Test Freq = 1 MHz		10	pF
I_{PD}	Power Down Current	V_{CC} = 0 V, V_{PD} = 5 V		10	mA

Notes: 1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s 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 0.8 V. 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.

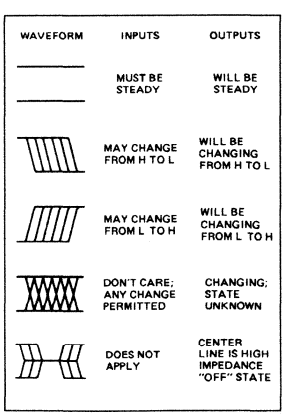
See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
(Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100 pF, Load Capacitance for All Other Outputs = 80 pF)

Parameter Symbol	Parameter Description	12 MHz Osc.		Variable Oscillator		Units
		Min.	Max.	Min.	Max.	
1/TCLCL	Oscillator Frequency			1.2	12	MHz
TLHLL	ALE Pulse Width	127		2TCLCL-40		ns
TAVLL	Address Setup to ALE	43		TCLCL-40		ns
TLLAX	Address Hold After ALE	48		TCLCL-35		ns
TLLIV	ALE to Valid Instr In		183		4TCLCL-150	ns
TLLPL	ALE to $\overline{PSEN}$	58		TCLCL-25		ns
TPLPH	$\overline{PSEN}$ Pulse Width	190		3TCLCL-60		ns
TPLIV	$\overline{PSEN}$ to Valid Instr In		100		3TCLCL-150	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		63		TCLCL-20	ns
TPXAV	Address Valid After $\overline{PSEN}$	75		TCLCL-8		ns
TAVIV	Address to Valid Instr In		267		5TCLCL-150	ns
TPLAZ	Addr Float After $\overline{PSEN}$		20		20	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL-100		ns
TRLDV	$\overline{RD}$ to Valid Data In		252		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		2TCLCL-70	ns
TLLDV	ALE to Valid Data In		517		8TCLCL-150	ns
TAVDV	Address to Valid Data In		585		9TCLCL-165	ns
TLLWL	ALE to $\overline{RD}$ or $\overline{WR}$	200	300	3TCLCL-50	3TCLCL+50	ns
TAVWL	Address to $\overline{RD}$ or $\overline{WR}$	203		4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	13		TCLCL-70		ns
TQVWH	Data Setup Before $\overline{WR}$	433		7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	33		TCLCL-50		ns
TRLAZ	Address Float After $\overline{RD}$		20		20	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	33	133	TCLCL-50	TCLCL+50	ns

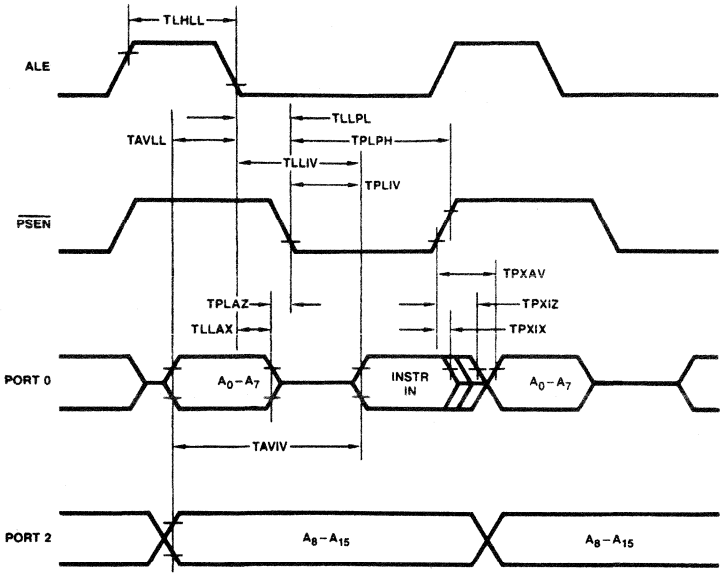
SWITCHING WAVEFORMS

KEY TO SWITCHING WAVEFORMS



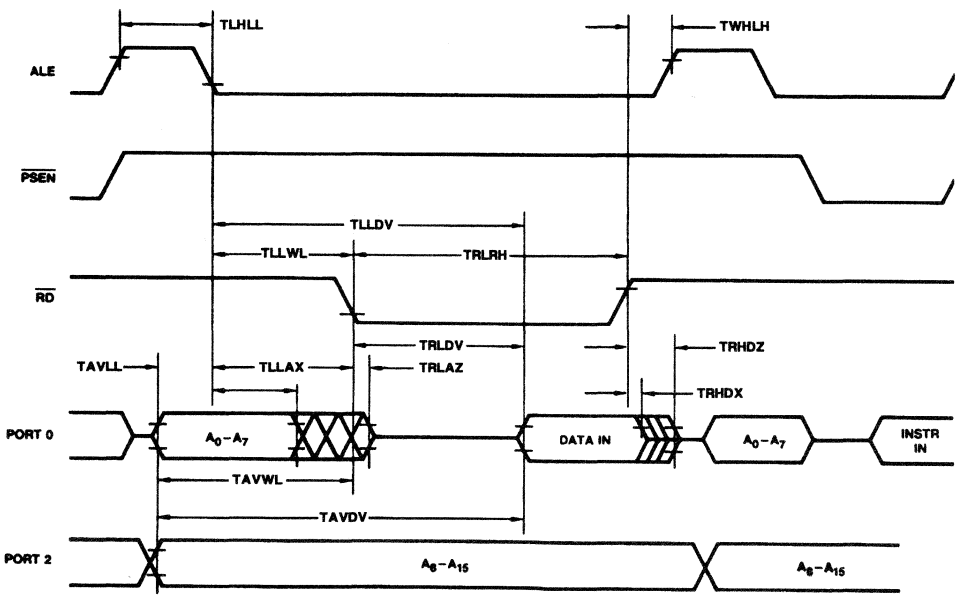
KS000010

SWITCHING WAVEFORMS



WF008744

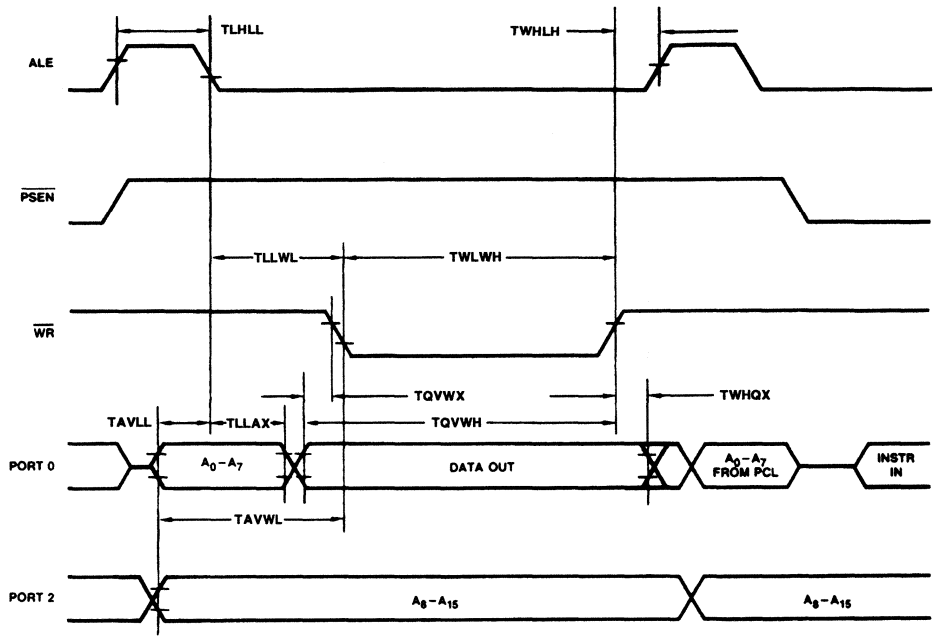
External Program Memory Read Cycle



WF008733

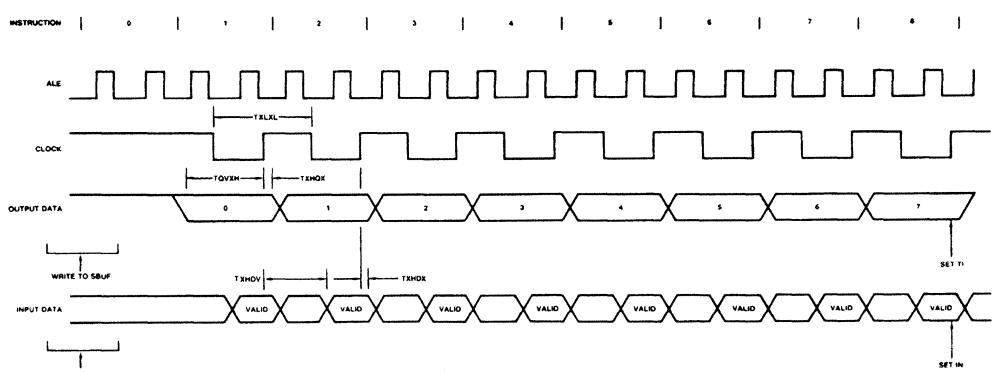
External Data Memory Read Cycle

SWITCHING WAVEFORMS (Cont'd)



WF008757

External Data Memory Write Cycle

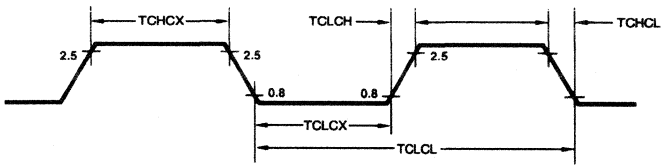


WF008723

Shift Register Timing Waveforms

EXTERNAL CLOCK DRIVE

Parameter Symbol	Parameter Description	Min.	Max.	Units
1/TCLCL	Oscillator Frequency	1.2	12	MHz
TCHCX	HIGH Time	20		ns
TCLCX	LOW Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns



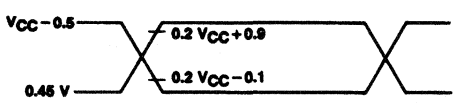
WF008762

External Clock Drive Waveforms

SERIAL PORT TIMING — SHIFT REGISTER MODE
(Load Capacitance = 80 pF)

Parameter Symbol	Parameter Description	12 MHz Osc.		Variable Oscillator		Units
		Min.	Max.	Min.	Max.	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL–133		ns
TXHQX	Output Data Hold After Clock Rising Edge	50		2TCLCL–117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL–133	ns

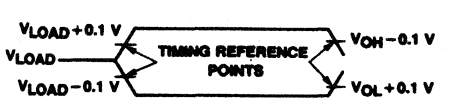
AC Testing



WF020900

AC INPUTS DURING TESTING ARE DRIVEN AT V_{CC} – 0.5 FOR A LOGIC "1" AND 0.45 V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT V_{IH} MIN. FOR A LOGIC "1" AND V_{IL} MAX. FOR A LOGIC "0."

Input/Output Waveform



WF020940

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.

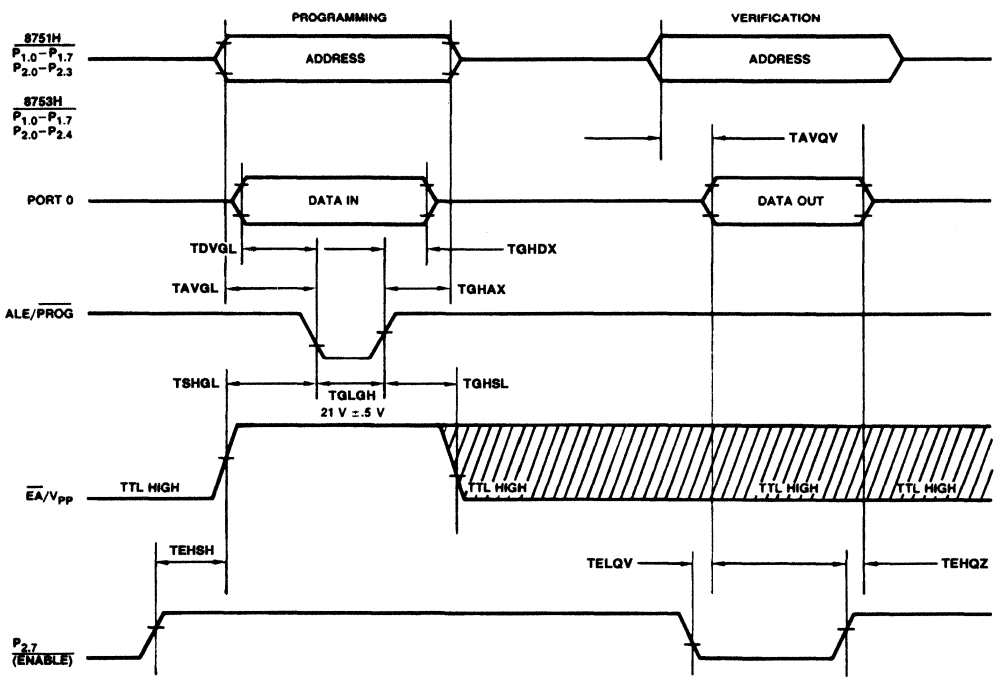
Float Waveform

EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

(T_A = +21 to +27°C, V_{CC} = +5 V ±10%, V_{SS} = 0 V)

Parameter Symbol	Parameter Description	Min.	Max.	Units
V_{PP}	Programming Supply Voltage	20.5	21.5	V
I_{PP}	Programming Supply Current		30	mA
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVGL	Address Setup to $\overline{PROG}$	48TCLCL		
TGHAX	Address Hold After $\overline{PROG}$	48TCLCL		
TDVGL	Data Setup to $\overline{PROG}$	48TCLCL		
TGHDX	Data Hold After $\overline{PROG}$	48TCLCL		
TEHSH	$P_{2.7}$ ($\overline{ENABLE}$) HIGH to V_{PP}	48TCLCL		
TSHGL	V_{PP} Setup to $\overline{PROG}$	10		µsec
TGHSL	V_{PP} Hold after $\overline{PROG}$	10		µsec
TGLGH	$\overline{PROG}$ Width	45	55	msec
TAVQV	Address to Data Valid		48TCLCL	
TELQV	$\overline{ENABLE}$ to Data Valid		48TCLCL	
TEHQZ	Data Float After $\overline{ENABLE}$	0	48TCLCL	

EPROM PROGRAMMING AND VERIFICATION WAVEFORMS



WF008713

For Programming conditions, see Figures 1, 2, and 3.
For Verification conditions, see Figures 4 and 5.
For Security Bit Programming, see Figure 6.

Z8000 FAMILY

AMD LITERATURE

To obtain literature in the U.S., write or call the AMD Literature Distribution Center, 901 Thompson Place, P.O. Box 3453 — M/S 82, Sunnyvale, CA 94088; (408) 732-2400, TOLL FREE (800) 538-8450. To obtain literature from international locations, contact the nearest AMD sales office or distributor (see listings in the back of this publication).

Z-Bus/68000 Microprogrammable Bus Translator

Anthony Dicolli
Advanced Micro Devices

5

Bus Translator

ABSTRACT

This paper describes an interface technique that permits all speed versions of the 68000 CPU to communicate with all Z8000 peripherals. Further, the microprogrammable nature of this interface allows intermixing of various speed peripherals on the same Z-Bus by dynamically modifying the bus translator's timing characteristics on a cycle-by-cycle basis. Included are a circuit description, PROM programs, PAL equations and a discussion on a typical system's architecture.

BUS TRANSLATOR CIRCUIT DESCRIPTION

(Figures 1 and 2)

The bus multiplexer contains one 8-bit bus buffer (Am29827) and two bus transceivers (Am29863). These components accept the separate address and data buses of the 68000 and, using three-state techniques, multiplex them together to form the Z-Bus address/data path (AD Bus). Only six address lines are required to directly address all control and data registers in the Z8000 peripherals. The timing sequence of this multiplexing operation is derived from the state sequencer and the translator logic.

The translator logic contains hard-wired logic elements that accept 68000 bus control signals (CLK, $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$) as inputs and, in conjunction with the state sequencer inputs (T_1, T_2, T_3, T_4, T_5), produce Z-Bus control signals ($\overline{AS}$, $\overline{DS}$, A_0, R/$\overline{W}$, B/$\overline{W}$), 68000 $\overline{DTACK}$, and the bus multiplexer control signals.

In addition, a Z8000 peripheral inhibit input (ZINH) is provided to prevent the bus translator from responding to 68000 bus operations when none of the Z8000 peripherals is being accessed (i.e., a 68000 instruction fetch). Also, a translator output enable ($\overline{TOE}$) is provided, so that control of the Z-Bus can be relinquished to another master, such as a DMA Controller, if required (see Figure 9). Both $\overline{TOE}$ and $\overline{ZINH}$ are generated by an external chip select decoder (see System Architecture section). This translator logic can easily be implemented utilizing a PAL (AmPAL16R4) as shown in Figure 3. The fuse map for this PAL is detailed in Figure 4 and the design equations are:

$\overline{AOE}$ = /T_2 * $\overline{MAS}$
$\overline{ZAS}$ = /T_1 * $\overline{MAS}$
$\overline{DOE}$: = T_3 * $\overline{MUDS}$ + T_3 * $\overline{MLDS}$
$\overline{ZDS}$: = T_4 * $\overline{MUDS}$ + T_4 * $\overline{MLDS}$
ZB/$\overline{W}$: = /$\overline{MUDS}$: + :/$\overline{MLDS}$
$\overline{MDTACK}$: = T_5 * $\overline{MUDS}$ + T_5 * $\overline{MLDS}$
/ZA_0 + $\overline{MLDS}$
;when $\overline{TOE}$ is LOW

The state sequencer, shown in Figure 2, is a registered PROM (Am27S35) which provides microprogramming ability for the bus translator. The registered PROM contains a fusible-link PROM memory array, an output register, and an (user-programmable) initialize word (Figure 5). When $\overline{INIT}$ is pulled LOW, the contents of the initialize word are sent to the output, regardless of the state of the clock or the address inputs. In this application, the initialize word is programmed to OOH. Outputs O_0 through O_4 generate timing signals T_1 through T_5, which are used as gating inputs by the translator logic. These gating signals along with the 68000 control signals determine when the Z-Bus control signals will be activated and the duration of these signals. The program sequences stored in this PROM are user-definable and are a function of the CPU and peripheral types and speeds implemented in the system.

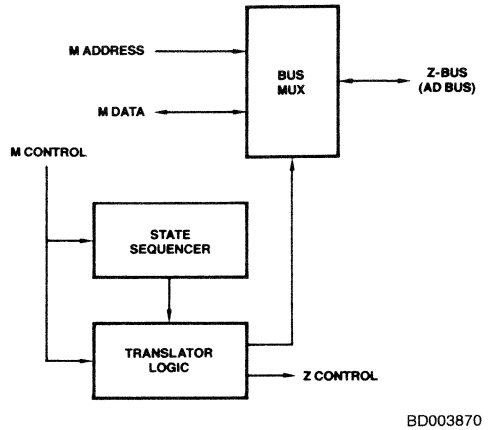


Figure 1.

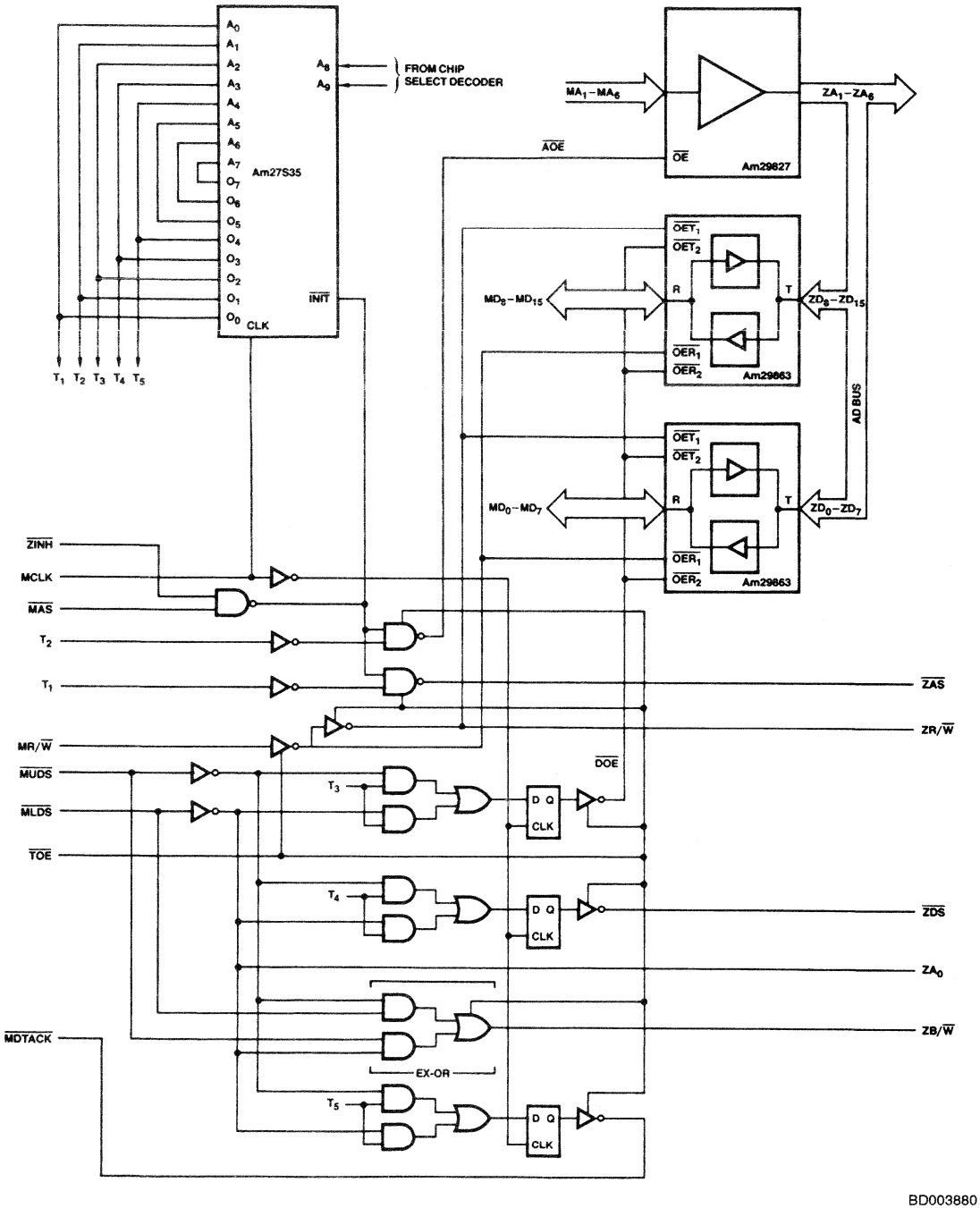
The next address to be accessed is determined by the clock strobed data outputs, O_0 through O_7. For this to work properly, a unique and non-redundant output word must exist for every clock cycle. Since there will be instances where T_1 through T_5 will not change for many clock cycles, a "counter" function must be included via O_5, O_6 and O_7. This counter function allows up to 8 idle states (83.3ns per idle state) to be inserted between T_1 through T_5 transitions. Address lines A_8 and A_9 provide the ability to dynamically select up to 4 individual sequences. Figure 6 is a PROM program required to generate the read and write cycles shown in Figure 8. When A_8 is LOW, a read cycle is selected; with A_8 in the HIGH state, a write cycle is selected; with A_8 LOW and A_9 HIGH, a "read interrupt" cycle is selected. Other possible sequences would support intermixing of 4 and 6MHz peripherals on a cycle-by-cycle basis.

This state sequencer design assumes that a minimum chip count is preferred and that PROM space is inexpensive. In the example of Figure 6, only 29 locations out of 1,000 are used. If smaller memories are preferred and chip count is not critical, then the state sequencer in Figure 7 can be substituted. In this example, a hardware counter is provided so that no "counter" function is required in the PROM. Also, the PROM array need only be 64 words deep. This approach also allows more sequences to be added without drastically increasing the PROM array size. For instance, a separate read and write cycle for 4 and 6MHz peripherals, a separate read interrupt vector for 4 and 6MHz peripherals, and a two speed CPU cycle would require only 256 words of PROM space.

It should be noted that the address inputs, used to select the sequence to be enabled, are generated by the same external chip select decoder that generates $\overline{ZINH}$.

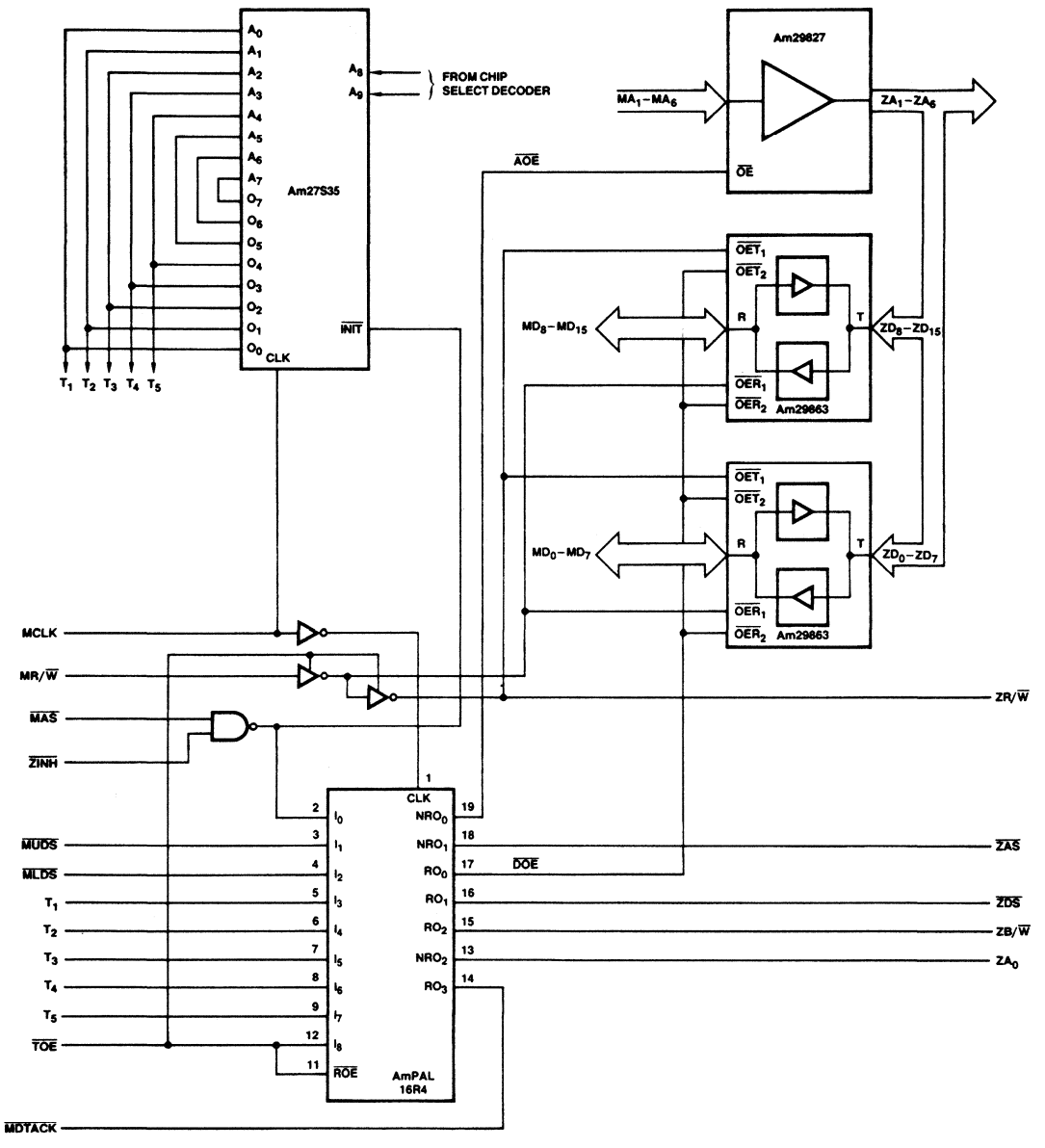
BUS TRANSLATOR TIMING ANALYSIS

Figure 8 illustrates the timing relationships between a 12MHz 68000 and all 4MHz Z8000 peripherals. The signals shown are the 68000 control, address, and data inputs to the bus translator of Figure 2, and the corresponding Z-Bus control, address, and data outputs. The following discussion assumes that the 68000 CPU, bus translator and Z8000 peripherals are physically located on the same circuit board. If a backplane is used, its propagation delays must also be considered; however, the set-up and hold times in this illustration will be sufficient for most system architectures.



BD003880

Figure 2.



BD003890

Figure 3.

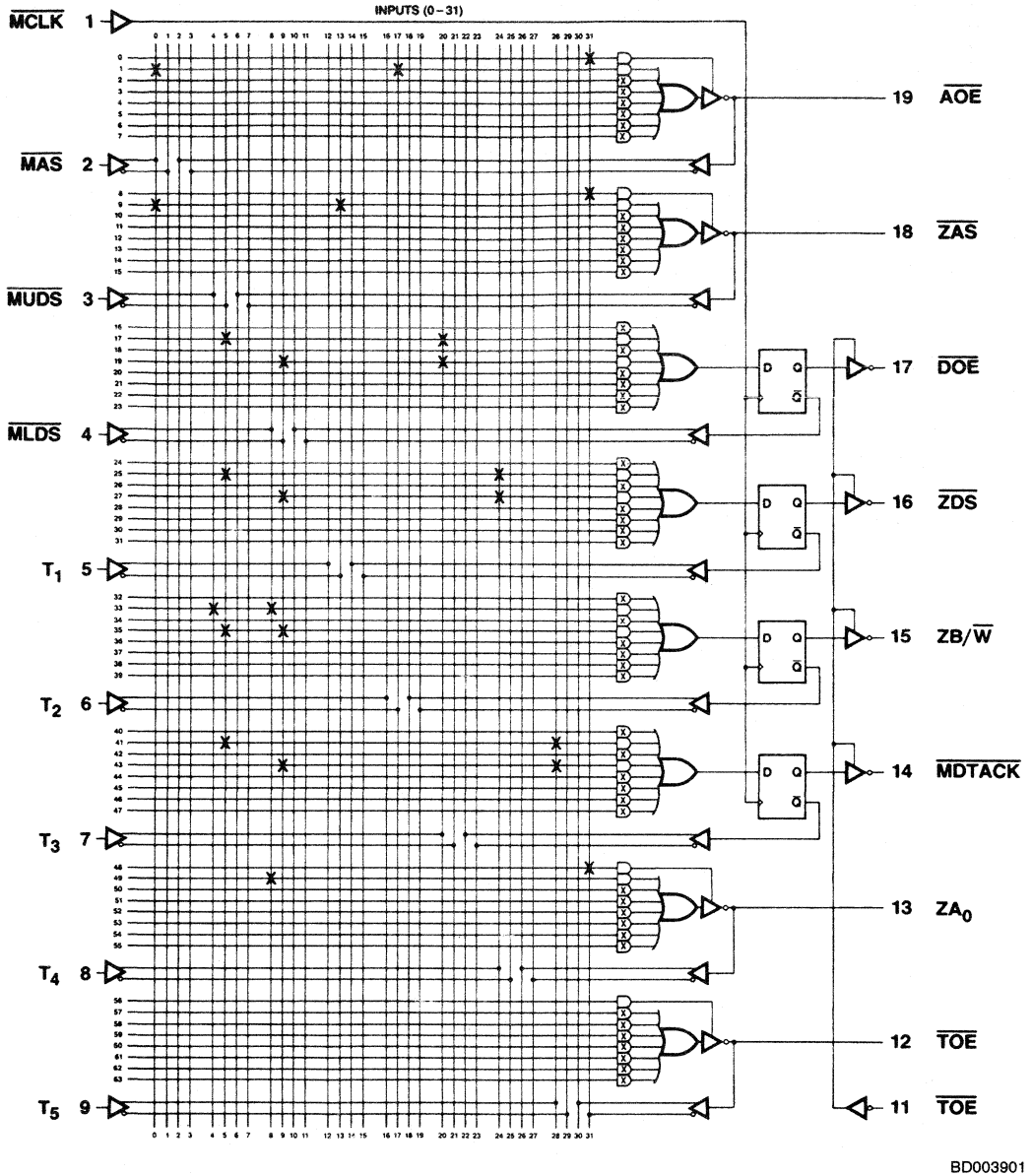


Figure 4. Logic Diagram PAL16R4

Bus Translator

During S_0 the read/write line from the 68K will be set HIGH to indicate a read operation. MR/$\overline{W}$ puts the transceivers in the receive mode and is used by the Z-Bus as ZR/$\overline{W}$. In a read operation, the 68K address will become valid sometime during S_1, and the external chip select decoder will read this address and determine whether a Z8000 I/O operation is being started. If it is a Z8000 I/O operation, $\overline{ZINH}$, A_8, and A_9 will be appropriately set. $\overline{MAS}$ will be asserted during S_2. This action: 1) releases the state sequencer via $\overline{INIT}$ and begins the Tx state on the next positive edge of $\overline{MCLK}$ (shown as 0_1), 2) causes $\overline{AOE}$ to go active, thus driving the AD bus with the address, and 3) asserts $\overline{ZAS}$. $\overline{MUDS}$ and/or $\overline{MLDS}$ are also asserted in S_2 and cause ZA_0 and ZB/$\overline{W}$ to be appropriately set. At 0_2 the state sequencer asserts T_1, causing $\overline{ZAS}$ to be negated. This phase was chosen to meet the required 70ns (min) $\overline{ZAS}$ pulse width (TwAS), and 30ns (min) address to $\overline{ZAS}$ set-up time (TsA (AS)). At 0_3 the state sequencer asserts T_2 and T_4. T_2 causes the $\overline{AOE}$ line to negate which removes the

address and three-states the AD bus. This meets the required 50ns (min) address to $\overline{ZAS}$ hold time (ThA (AS)). T_4 causes the $\overline{ZDS}$ to be asserted on the negative edge of 0_3. This meets the $\overline{ZAS}$ to $\overline{ZDS}$ delay of 60ns (min).

On the falling edge of $\overline{ZDS}$, the peripheral will drive the AD bus, and data on the AD bus will be valid 250ns later (an exception to this occurs in the Z8030 which requires 520ns from rising edge of $\overline{ZAS}$ to valid read data). At 0_5 the sequencer sets T_5, which in turn causes $\overline{MDTACK}$ to be asserted on the negative edge of 0_5. The 68K samples this line on the negative edge of 0_6 and accepts the input data on the negative edge of 0_7. At 0_6 the T_3 line is asserted, which activates $\overline{DOE}$ on the negative edge of 0_6. This enables AD bus data to the 68K data bus. T_3 may be set to occur anytime before the negative edge of 0_7. During S_7 the 68K will negate $\overline{MAS}$, $\overline{MUDS}$, and $\overline{MLDS}$. This action causes the sequencer to reset (via $\overline{INIT}$), and on the negative edge of 0_8, $\overline{ZDS}$, $\overline{MDTACK}$, and $\overline{DOE}$ will be negated to end the read cycle. Note that the peripheral provides the zero data hold time required by the 68000. It should also be noted that $\overline{DTACK}$ is used to insert wait states (2 wait states = 1 MCLK). The assertion of $\overline{DTACK}$ via T_5 is a function of the minimum required $\overline{ZDS}$ pulse width (in this case 390ns), and the minimum time required for the peripheral to provide valid data to the AD bus. Therefore, in using the Z8030, T_5 would not occur (this is accomplished by inserting 2 idle states in the sequencer) until 0_7, and the rest of the read cycle would be proportionately extended.

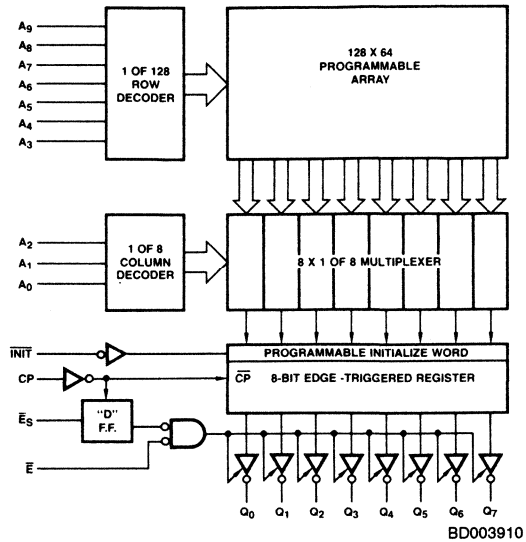


Figure 5. Am27S35 Block Diagram

		ADDR INPUT											PROM OUTPUT									REG. OUTPUT								
$\overline{INIT}$	CLOCK		A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	HEX	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	HEX	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7	HEX
0	X		0	0	0	0	0	0	0	0	0	0	000	0	0	0	0	0	1	0	0	20	0	0	0	0	0	0	0	0	00
1	0_1		0	0	0	0	0	1	0	0	0	0	020	1	0	0	0	0	0	0	0	01	0	0	0	0	0	1	0	0	20
1	0_2		1	0	0	0	0	0	0	0	0	0	001	1	1	0	1	0	0	0	0	0B	1	0	0	0	0	0	0	0	01
1	0_3	READ CYCLE	1	1	0	1	0	0	0	0	0	0	00B	1	1	0	1	0	1	0	0	2B	1	1	0	1	0	0	0	0	0B
1	0_4		1	1	0	1	0	1	0	0	0	0	02B	1	1	0	1	1	0	0	0	1B	1	1	0	1	0	1	0	0	2B
1	0_5		1	1	0	1	1	0	0	0	0	0	01B	1	1	1	1	1	0	0	0	1F	1	1	0	1	1	0	0	0	1B
1	0_6		1	1	1	1	1	0	0	0	0	0	01F	1	1	1	1	1	0	0	0	1F	1	1	1	1	1	0	0	0	1F
0	X		0	0	0	0	0	0	0	0	1	0	100	0	0	0	0	0	1	0	0	20	0	0	0	0	0	0	0	0	00
1	0_1		0	0	0	0	0	1	0	0	1	0	120	1	0	0	0	0	0	0	0	01	0	0	0	0	0	1	0	0	20
1	0_2		1	0	0	0	0	0	0	0	1	0	101	1	1	1	0	0	0	0	0	07	1	0	0	0	0	0	0	0	01
1	0_3		1	1	1	0	0	0	0	0	1	0	107	1	1	1	1	0	0	0	0	0F	1	1	1	0	0	0	0	0	07
1	0_4		1	1	1	1	0	0	0	0	1	0	10F	1	1	1	1	0	1	0	0	2F	1	1	1	1	0	0	0	0	0F
1	0_5		1	1	1	1	0	1	0	0	1	0	12F	1	1	1	1	0	0	1	0	4F	1	1	1	1	0	1	0	0	2F
1	0_6		1	1	1	1	0	0	1	0	1	0	14F	1	1	1	1	0	1	1	0	6F	1	1	1	1	0	0	1	0	4F
1	0_7	WRITE CYCLE	1	1	1	1	0	1	1	0	1	0	16F	1	1	1	1	1	0	0	0	1F	1	1	1	1	0	1	1	0	6F
1	0_8		1	1	1	1	1	0	0	0	1	0	11F	1	1	1	0	1	0	0	0	17	1	1	1	1	1	0	0	0	1F
1	0_9		1	1	1	0	1	0	0	0	1	0	117	1	1	1	0	1	0	0	0	17	1	1	1	0	1	0	0	0	17
0	X		0	0	0	0	0	0	0	0	0	1	200	0	0	0	0	0	1	0	0	20	0	0	0	0	0	0	0	0	00
1	0_1		0	0	0	0	0	1	0	0	0	1	220	1	0	0	0	0	0	0	0	01	0	0	0	0	0	1	0	0	20
1	0_2		1	0	0	0	0	0	0	0	0	1	201	1	1	0	0	0	0	0	0	03	1	0	0	0	0	0	0	0	01
1	0_3		1	1	0	0	0	0	0	0	0	1	203	1	1	0	0	0	1	0	0	23	1	1	0	0	0	0	0	0	03
1	0_4		1	1	0	0	0	1	0	0	0	1	223	1	1	0	0	0	0	1	0	43	1	1	0	0	0	1	0	0	23
1	0_5		1	1	0	0	0	0	1	0	0	1	243	1	1	0	0	0	1	1	0	63	1	1	0	0	0	0	1	0	43
1	0_6		1	1	0	0	0	1	1	0	0	1	263	1	1	0	0	0	0	0	1	83	1	1	0	0	0	1	1	0	63
1	0_7		1	1	0	0	0	0	0	1	0	1	283	1	1	0	0	0	1	0	1	A3	1	1	0	0	0	0	0	1	83
1	0_8		1	1	0	0	0	1	0	1	0	1	2A3	1	1	0	0	0	0	1	1	C3	1	1	0	0	0	1	0	1	A3
1	0_9		1	1	0	0	0	0	1	1	0	1	2C3	1	1	0	0	0	1	1	1	E3	1	1	0	0	0	0	1	1	C3
1	0_{10}	READ INTERRUPT VECTOR CYCLE	1	1	0	0	0	1	1	1	0	1	2E3	1	1	0	1	0	0	0	0	0B	1	1	0	0	0	1	1	1	E3
1	0_{11}		1	1	0	1	0	0	0	0	0	1	20B	1	1	0	1	0	1	0	0	2B	1	1	0	1	0	0	0	0	0B
1	0_{12}		1	1	0	1	0	1	0	0	0	1	22B	1	1	0	1	1	0	0	0	1B	1	1	0	1	0	1	0	0	2B
1	0_{13}		1	1	0	1	1	0	0	0	0	1	21B	1	1	1	1	1	0	0	0	1F	1	1	0	1	1	0	0	0	1B

Figure 6.

The write cycle operates in a similar manner with two exceptions. First, there is a required data set-up time of 30ns with respect to the falling edge of $\overline{ZDS}$ (TsDW(DSF)). Therefore, one sequencer idle state is required between T_2 and T_4. Secondly, there is a required data hold time of 30ns with respect to the rising edge of $\overline{ZDS}$ (ThDW(DS)). Therefore, T_5 is asserted at 0_8, while T_4 is negated at 0_9. If bidirectional registers are substituted for the data transceivers, four wait states can be eliminated from the write cycle.

TIMING PARAMETERS

During a read interrupt vector cycle, timing parameters are the same as a normal read operation with the exception of the $\overline{ZAS}$ to $\overline{ZDS}$ delay. In a normal read, this parameter is 60ns. However, during an interrupt, this parameter (TdDCST) must be equal to or greater than the interrupt daisy chain settling time. When five Z8000 peripherals are in this chain, this parameter is approximately 710ns. To accommodate this requirement, 8 idle states must be inserted between T_2 and T_4 by the state sequencer. This state sequencer detects an interrupt operation via A_9 (reference Figure 6).

SYSTEM ARCHITECTURE

Figure 9 illustrates an architecture that allows the 68K to perform other tasks while an I/O operation is in progress. The ZI/O chip select decoder accepts:

1) CPU Status — indicates an interrupt read cycle is required. This activates RIV (Read Interrupt Vector).

2) R/W — indicates that the current cycle is a read or a write. This activates I/OR/$\overline{W}$.

3) MAn — A user-definable number of address lines (6 lines minimum) that identifies which peripheral is being addressed. This activates the appropriate $\overline{CS}$ line, ZI/O REQ, $\overline{ZINH}$ and 4/6MHz.

The ZI/O bus arbiter is used to determine who has I/O bus control and is only required if a DMA controller is part of the Z peripheral chain.

It accepts:

1) ZI/O REQ — indicates the CPU is requesting the I/O bus.

2) DMA BUS REQ — indicates the DMA controller is requesting the I/O bus.

3) M BUS REQ — allows the 68K to request the I/O bus before the I/O cycle is started.

and generates:

1) ZBUS ERR — flags the 68K when the CPU starts a ZI/O operation and the I/O bus is busy.

2) ZI/O BUS BUSY — A Z-bus status line that can be polled by the CPU.

3) DMA BUS ACQ — grants control of the ZI/O bus to the DMA controller.

4) $\overline{TOE}$ — grants control of the ZI/O bus to the Bus Translator.

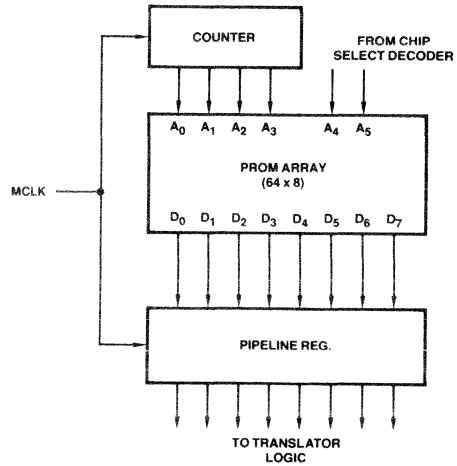


Figure 7.

This type of architecture is useful in I/O intensive applications where the data must be operated on concurrently by the CPU (i.e., PBX systems, disk controllers, etc.). Data is received and transmitted by this system via the DMA controller. Transmitted data is loaded into the Buffer Memory by the CPU and is then transmitted via a DMA operation. Received data is loaded into the Buffer Memory via a DMA operation and is then read by the CPU. In extreme cases this Buffer Memory can be dual ported so that true concurrent operation between CPU and I/O is achieved. In single-ported Buffer Memory architectures, a Bus Arbiter is required to grant ZI/O bus control to CPU or DMA controller. In this example, the CPU can poll the arbiter for ZI/O bus control and lock out the DMA controller until the CPU has completed its I/O operation. Alternately, the CPU can start a ZI/O bus operation, and if the bus is busy, a bus error is generated. The operation of this arbiter is straightforward and can be constructed using a minimal number of gates and flip-flops.

The I/O chip select decoder is a combination chip selector and memory mapper. Note that the 4MHz/6MHz signal is used by the bus translator to allow intermixing of slow and fast peripheral chips.

The DMA controller is an AmZ8016, which utilizes "link-lists" to update its control registers. This further reduces the need for CPU intervention, thus increasing system performance.

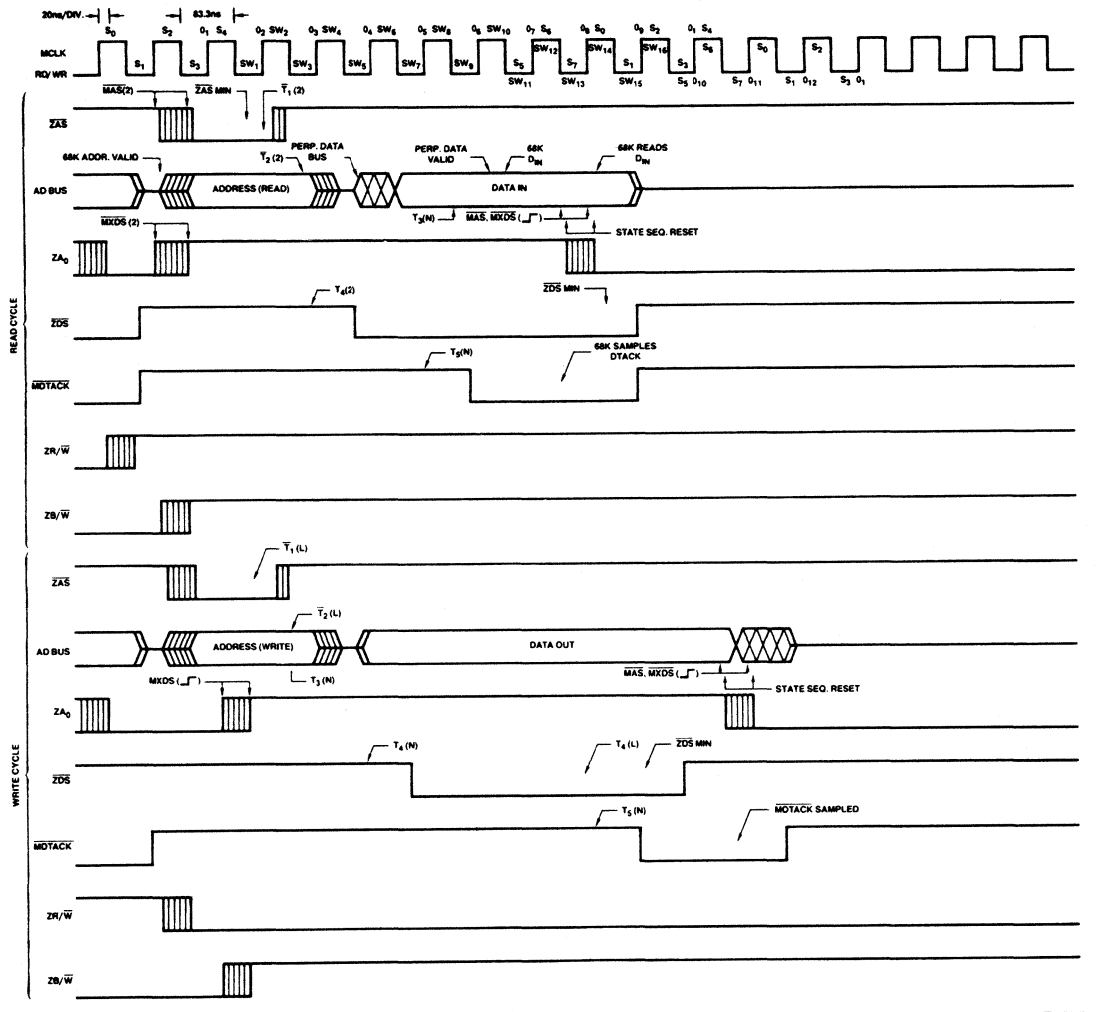


Figure 8. 12MHz 68000 to 4MHz Peripherals Timing Relationships

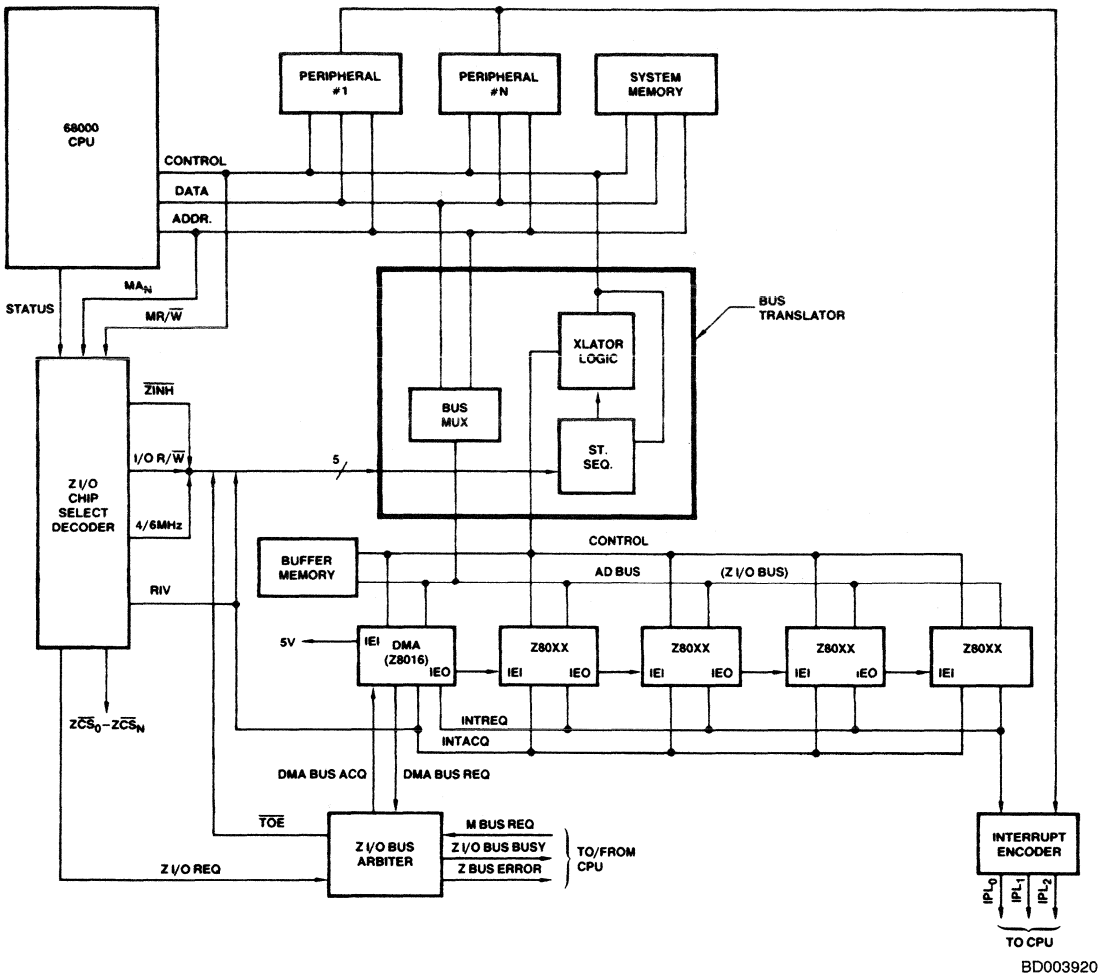


Figure 9.

CONCLUSIONS

This circuit is a highly efficient and flexible bus translator that allows Z8000 peripherals and a 68000 CPU to intercommunicate in a uniform manner; the same technique is also applicable to other CPU's, such as iAPX86, LSI-11 and N16000.

5

Z8001*/Z8002*

16-Bit Microprocessors

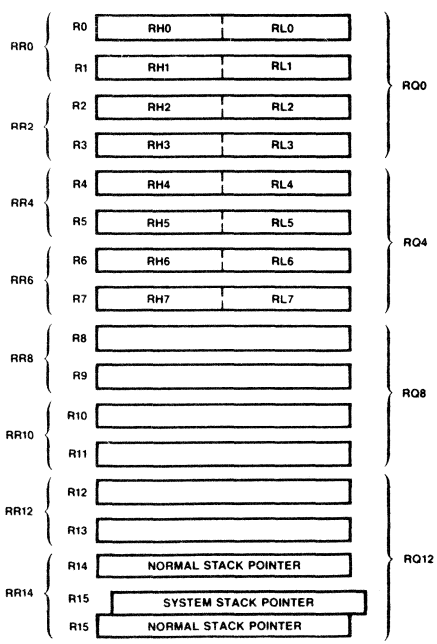
DISTINCTIVE CHARACTERISTICS

- **4, 6, 8 MHz CPU Clock**
  High throughput with low system clock rate for easier system design
- **Powerful General Register Architecture**
  16 general registers provide high throughput in all types of applications.
- **Wide Variety of Data Types**
  Instructions operate on bits, bytes, 16- and 32-bit words for efficient programming of a wide variety of functions.
- **Partitioned for Operating System Protection**
  Hardware bit protects privileged instructions from execution except by operating system.
- **Supports 3 Types of Interrupts**
  Separate pins provided for vectored, non-vectored and non-maskable interrupts
- **Two Compatible CPUs**
  Compact 40-pin Z8002 supports 64KB memory; larger 48-pin Z8001 supports 8MB memory.

GENERAL DESCRIPTION

The Z8001* is a general-purpose 16-bit CPU belonging to the Z8000 family of microprocessors. Its architecture is centered around sixteen 16-bit general registers. The CPU deals with 23-bit address spaces and hence can address directly 8MB of memory. The 23-bit address consists of two components: 7-bit segment number and 16-bit offset. Facilities are provided to maintain three distinct address spaces — code, data and stack. The Z8001 implements a powerful instruction set with flexible addressing modes. These instructions operate on several data types — bit, byte, word (16-bit), long word (32-bit), byte string and word string. The CPU can execute instructions in one of two modes — System and Normal. Sometimes these modes are also known as Privileged and Non-Privileged, respectively. The CPU also contains an on-chip memory refresh facility. The Z8001 is software compatible with the Z8002 microprocessor. The Z8001 is fabricated using silicon-gate N-MOS technology and is packaged in a 48-pin DIP. The Z8001 requires a single +5 power supply and a single phase clock for its operation.

GENERAL REGISTERS

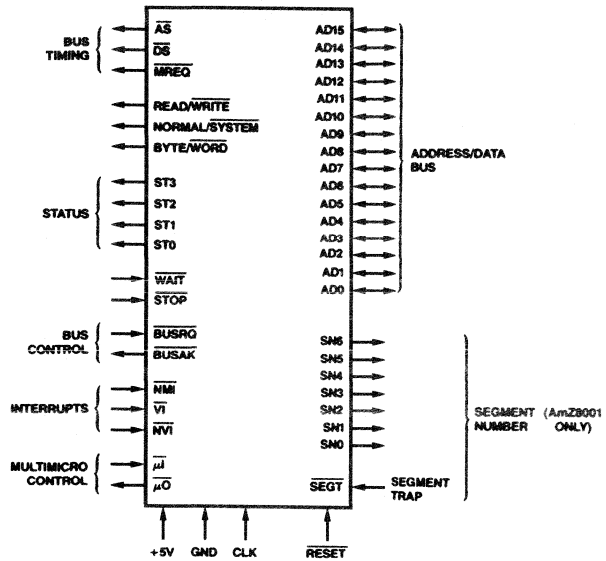


AF002630

*Z8000, Z8001, and Z8002 are trademarks of Zilog, Inc.

Publication # 00971 Rev. C Amendment /0
Issue Date: May 1987

LOGIC SYMBOL



LS001272

ORDERING INFORMATION – Z8001

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option
c. Package Type
d. Temperature Range
e. Optional Processing

Z8001 A D C B

e. OPTIONAL PROCESSING
Blank = Standard processing
B = Burn-in

d. TEMPERATURE RANGE*
C = Commercial (0 to +70°C)

c. PACKAGE TYPE
P = 48-Pin Plastic DIP (PD 048)
D = 48-Pin Sidebrazed Ceramic DIP (SD 048)

b. SPEED OPTION
Blank = 4 MHz
A = 6 MHz
A-8 = 8 MHz

a. DEVICE NUMBER/DESCRIPTION
Z8001
16-Bit Segmented Microprocessor

Valid Combinations	
Z8001	PC, DC, DCB
Z8001A	
Z8001A-8	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

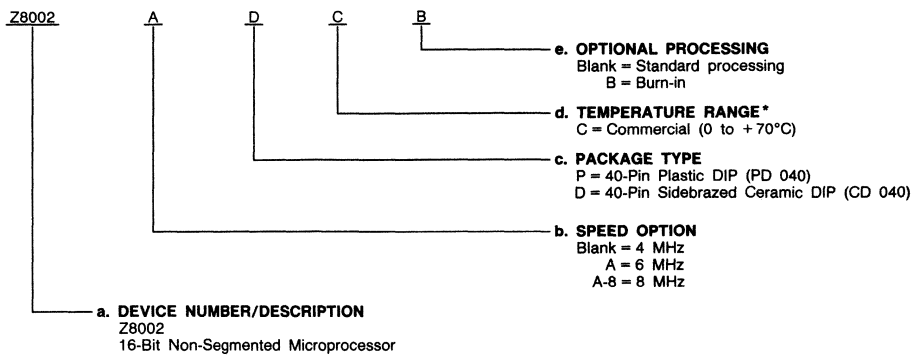
*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

5

ORDERING INFORMATION – Z8002

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
Z8002	PC, DC, DCB
Z8002A	
Z8002A-8	

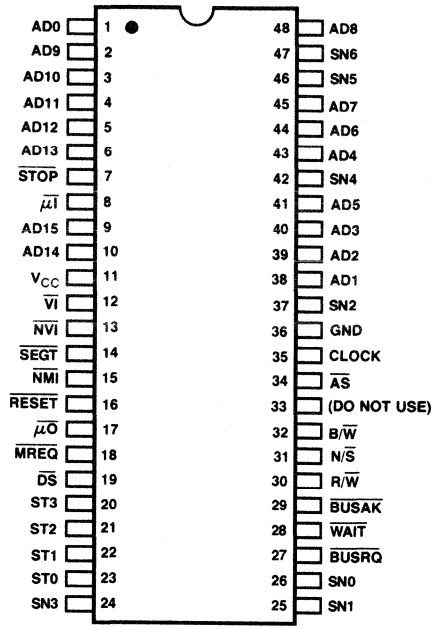
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

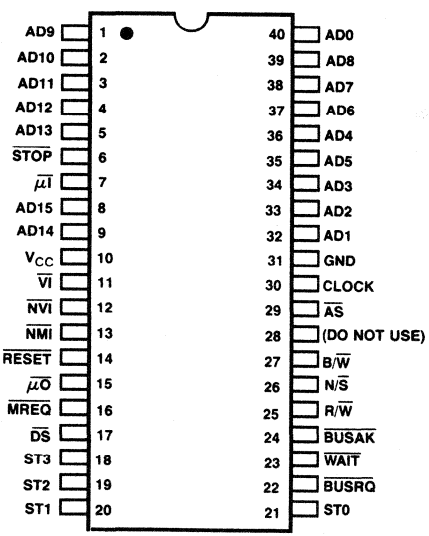
CONNECTION DIAGRAMS
Top View

Z8001



CD005262

Z8002



CD005272

Note: Pin 1 is marked for orientation.

Z8001 PIN DESCRIPTION

Pin No.	Name	I/O	Description
11	V_{CC}		+5V Power Supply.
36	V_{SS}		Ground.
1, 38-40, 43, 41, 44, 45, 48, 2-6, 10, 9	AD0-AD15	I/O	Bidirectional, 3-State. Address/Data Bus. This 16-bit multiplexed address/data bus is used for all I/O and memory transactions. HIGH on the bus corresponds to 1 and LOW corresponds to 0. AD0 is the least significant bit position with AD15 is most significant. The $\overline{AS}$ output and $\overline{DS}$ output will indicate whether the bus is used for address offset or data. The status output lines ST0-ST3 will indicate the type of transaction: memory or I/O.
34	$\overline{AS}$	O	3-State. Address Strobe. LOW on this output indicates that the AD0-AD15 bus contains address information. The address information is stable by the time of the LOW-to-HIGH transition of the $\overline{AS}$ output (see timing diagram). The status outputs ST0-ST3 indicate whether the bus contains a memory address or I/O address.
19	$\overline{DS}$	O	3-State. Data Strobe. LOW on this output indicates that the AD0-AD15 bus is being used for data transfer. The R/$\overline{W}$ output indicates the direction of data transfer – read (or in) means data into the CPU and write (or out) means data from the CPU. During a read operation, data can be gated on to the bus where $\overline{DS}$ goes LOW. A LOW-to-HIGH transition on the $\overline{DS}$ output indicates that the CPU has accepted the data (see timing diagram). During a write operation, LOW on the $\overline{DS}$ output indicates that data is set up on the bus. Data will be removed sometime after the LOW-to-HIGH transition of the $\overline{DS}$ output (see timing diagram).
30	R/$\overline{W}$	O	3-State. Read/Write. This output indicates the direction of data flow on the AD0-AD15 bus. HIGH indicates a read operation, i.e., data into the CPU, and LOW indicates a write operation, i.e., data from the CPU. This output is activated at the same time as $\overline{AS}$ going LOW and remains stable for the duration of the whole transaction (see timing diagram).
32	B/$\overline{W}$	O	3-State. Byte/Word. This output indicates the type of data transferred on the AD0-AD15 bus. HIGH indicates byte (8-bit) and LOW indicates word (16-bit) transfer. This output is activated at the same stage as $\overline{AS}$ going LOW and remains valid for the duration of the whole transaction (see timing diagram). The address generated by the CPU is always a byte address. However, the memory is organized as 16-bit words. All instructions and word operands are word aligned and are addressed by even addresses. Thus, for all word transactions with the memory, the least significant address bit will be zero. When addressing the memory for byte transactions, the least significant address bit determines which byte of the memory word is needed; even address specifies the most significant byte, and odd address specifies the least significant byte. In the case of I/O transactions, the address information on the AD0-AD15 bus refers to an I/O port, and B/$\overline{W}$ determines whether a data word or data byte will be transacted. During I/O byte transactions, the least significant address bit A0 determines which half of the AD0-AD15 bus will be used for the I/O transactions. The ST0-ST3 outputs will indicate whether the current transaction is for memory, normal I/O or special I/O.
23-20	ST0-ST3	O	3-State. Status. These four outputs contain information regarding the current transaction in a coded form. The status line codes are shown in the table on the following page.
28	$\overline{WAIT}$	I	Wait. LOW on this input indicates to the CPU that memory or I/O is not ready for the data transfer, and hence, the current transaction should be stretched. The $\overline{WAIT}$ input is sampled by the CPU at certain instances during the transaction (see timing diagram). If $\overline{WAIT}$ input is LOW at these instances, the CPU will go into wait state to prolong the transaction. The wait state will repeat until the $\overline{WAIT}$ input is HIGH at the sampling instant.
31	N/$\overline{S}$	O	3-State. Normal/System Mode. HIGH on this output indicates that the CPU is operating in Normal Mode and LOW indicates operation in System Mode. This output is derived from the Flag Control Word (FCW) register. The FCW register is described under the processor status information section of this document.
18	$\overline{MREQ}$	O	3-State. Memory Request. LOW on this output indicates that a CPU transaction with memory is taking place.
27	$\overline{BUSRQ}$	I	Bus Request. LOW on this input indicates to the CPU that another device (such as DMA) is requesting to take control of the bus. The $\overline{BUSRQ}$ input can be driven LOW anytime. The CPU synchronizes this input internally. The CPU responds by activating $\overline{BUSAK}$ output LOW to indicate that the bus has been relinquished. Relinquishing the bus means that the AD0-AD15, $\overline{AS}$, $\overline{DS}$, B/$\overline{W}$, R/$\overline{W}$, N/$\overline{S}$, ST0-ST3, SN0-SN6 and $\overline{MREQ}$ outputs will be in the high impedance state. The requesting device should control these lines in an identical fashion to the CPU to accomplish transactions. The $\overline{BUSRQ}$ input must remain LOW as long as needed to perform all the transactions and the CPU will keep the $\overline{BUSAK}$ output LOW. After completing the transactions, the device must disable the AD0-AD15, $\overline{AS}$, $\overline{DS}$, B/$\overline{W}$, R/$\overline{W}$, N/$\overline{S}$, ST0-ST3, SN0-SN6 and $\overline{MREQ}$ into the high impedance state and stop driving the $\overline{BUSRQ}$ input LOW. The CPU will make $\overline{BUSAK}$ output HIGH sometime later and take back the bus control.
29	$\overline{BUSAK}$	O	Bus Acknowledge. LOW on this output indicates that the CPU has relinquished the bus in response to a bus request.
15	$\overline{NMI}$	I	Non-Maskable Interrupt. A HIGH-to-LOW transition on this input constitutes non-maskable interrupt request. The CPU will respond with the Non-maskable Interrupt Acknowledge on the ST0-ST3 outputs and will enter an interrupt sequence. The transition on the $\overline{NMI}$ can occur anytime. Of the three kinds of interrupts available, the non-maskable interrupt has the highest priority.
12	$\overline{VI}$	I	Vectored Interrupt. LOW on this input constitutes vectored interrupt request. Vectored interrupt is next lower to the non-maskable interrupt in priority. The NVIE bit in the Flag and Control Word register must be 1 for the vectored interrupt to be honored. The CPU will respond with Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The $\overline{VI}$ input can be driven LOW anytime and should be held LOW until acknowledged.

Z8001 PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
13	$\overline{NVI}$	I	Non-Vectored Interrupt. LOW on this input constitutes non-vectored interrupt request. Non-vectored has the lowest priority of the three types of interrupts. The NVIE bit in the Flag and Control Word register must be 1 for this request to be honored. The CPU will respond with Non-Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The $\overline{NVI}$ input can be driven LOW anytime and should be held LOW until acknowledged.
8	$\overline{\mu I}$	I	Micro-In. This input participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.
17	$\overline{\mu O}$	O	Micro-Out. This output participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.
16	$\overline{RESET}$	I	Reset. LOW on this input initiates a reset sequence in the CPU. See the section on Initialization for details on reset sequence.
35	CLK	I	Clock. All CPU operations are controlled from the signal fed into this input. See DC Characteristics for clock voltage level requirements.
7	$\overline{STOP}$	I	Stop. This active LOW input facilitates one instruction at a time operation. See the section on single stepping.
26, 25, 37, 24, 42, 46, 47	SN0-SN6	O	3-State. Segment Number. These seven outputs contain the segment number part of a memory address. A HIGH on the output corresponds to 1 and a LOW corresponds to 0. SN0 is the least significant bit position and SN6 is the most significant bit position.
14	$\overline{SEGT}$	I	Segment Trap. LOW on this input constitutes a segment trap request. If the line is driven LOW, the CPU will respond with the Segment Trap Acknowledge code on the Status lines and commence a trap sequence. The $\overline{SEGT}$ input may be driven LOW at any time and is customarily held LOW until acknowledged. This input has priority over the interrupts.

Status Line Codes

ST3	ST2	ST1	ST0	
L	L	L	L	Internal Operation
L	L	L	H	Memory Refresh
L	L	H	L	Normal I/O Transaction
L	L	H	H	Special I/O Transaction
L	H	L	L	Segment Trap Acknowledge
L	H	L	H	Non-Maskable Interrupt Acknowledge
L	H	H	L	Non-Vectored Interrupt Acknowledge
L	H	H	H	Vectored Interrupt Acknowledge
H	L	L	L	Memory Transaction for Operand
H	L	L	H	Memory Transaction for Stack
H	L	H	L	Reserved
H	L	H	H	Reserved
H	H	L	L	Memory Transaction for Instruction Fetch (Subsequent Word)
H	H	L	H	Memory Transaction for Instruction Fetch (First Word)
H	H	H	L	Reserved
H	H	H	H	Reserved

Z8002 PIN DESCRIPTION

Pin No.	Name	I/O	Description
10	V_{CC}		+5V Power Supply.
31	V_{SS}		Ground.
40, 32-39, 1-5, 9, 8	AD0-AD15	I/O	Bidirectional, 3-State. Address/Data Bus. This 16-bit multiplexed address/data bus is used for all I/O and memory transactions. HIGH on the bus corresponds to 1 and LOW corresponds to 0. AD0 is the least significant bit position with AD15 is most significant. The $\overline{AS}$ output and $\overline{DS}$ output will indicate whether the bus is used for address offset or data. The status output lines ST0-ST3 will indicate the type of transaction: memory or I/O.
29	$\overline{AS}$	O	3-State. Address Strobe. LOW on this output indicates that the AD0-AD15 bus contains address information. The address information is stable by the time of the LOW-to-HIGH transition of the $\overline{AS}$ output (see timing diagram). The status outputs ST0-ST3 indicate whether the bus contains a memory address or I/O address.
17	$\overline{DS}$	O	3-State. Data Strobe. LOW on this output indicates that the AD0-AD15 bus is being used for data transfer. The R/$\overline{W}$ output indicates the direction of data transfer – read (or in) means data into the CPU and write (or out) means data from the CPU. During a read operation, data can be gated on to the bus where $\overline{DS}$ goes LOW. A LOW-to-HIGH transition on the $\overline{DS}$ output indicates that the CPU has accepted the data (see timing diagram). During a write operation, LOW on the $\overline{DS}$ output indicates that data is set up on the bus. Data will be removed sometime after the LOW-to-HIGH transition of the $\overline{DS}$ output (see timing diagram).
25	R/$\overline{W}$	O	3-State. Read/Write. This output indicates the direction of data flow on the AD0-AD15 bus. HIGH indicates a read operation, i.e., data into the CPU, and LOW indicates a write operation, i.e., data from the CPU. This output is activated at the same time as $\overline{AS}$ going LOW and remains stable for the duration of the whole transaction (see timing diagram).
27	B/$\overline{W}$	O	3-State. Byte/Word. This output indicates the type of data transferred on the AD0-AD15 bus. HIGH indicates byte (8-bit) and LOW indicates word (16-bit) transfer. This output is activated at the same stage as $\overline{AS}$ going LOW and remains valid for the duration of the whole transaction (see timing diagram). The address generated by the CPU is always a byte address. However, the memory is organized as 16-bit words. All instructions and word operands are word aligned and are addressed by even addresses. Thus, for all word transactions with the memory, the least significant address bit will be zero. When addressing the memory for byte transactions, the least significant address bit determines which byte of the memory word is needed; even address specifies the most significant byte, and odd address specifies the least significant byte. In the case of I/O transactions, the address information on the AD0-AD15 bus refers to an I/O port, and B/$\overline{W}$ determines whether a data word or data byte will be transacted. During I/O byte transactions, the least significant address bit A0 determines which half of the AD0-AD15 bus will be used for the I/O transactions. The ST0-ST3 outputs will indicate whether the current transaction is for memory, normal I/O or special I/O.
21-18	ST0-ST3	O	3-State. Status. These four outputs contain information regarding the current transaction in a coded form (see table on previous page).
23	$\overline{WAIT}$	I	Wait. LOW on this input indicates to the CPU that memory or I/O is not ready for the data transfer, and hence, the current transaction should be stretched. The $\overline{WAIT}$ input is sampled by the CPU at certain instances during the transaction (see timing diagram). If $\overline{WAIT}$ input is LOW at these instances, the CPU will go into wait state to prolong the transaction. The wait state will repeat until the $\overline{WAIT}$ input is HIGH at the sampling instant.
26	N/$\overline{S}$	O	3-State. Normal/System Mode. HIGH on this output indicates that the CPU is operating in Normal Mode and LOW indicates operation in System Mode. This output is derived from the Flag Control Word (FCW) register. The FCW register is described under the processor status information section of this document.
16	$\overline{MREQ}$	O	3-State. Memory Request. LOW on this output indicates that a CPU transaction with memory is taking place.
22	$\overline{BUSRQ}$	I	Bus Request. LOW on this input indicates to the CPU that another device (such as DMA) is requesting to take control of the bus. The $\overline{BUSRQ}$ input can be driven LOW anytime. The CPU synchronizes this input internally. The CPU responds by activating $\overline{BUSAK}$ output LOW to indicate that the bus has been relinquished. Relinquishing the bus means that the AD0-AD15, $\overline{AS}$, $\overline{DS}$, B/$\overline{W}$, R/$\overline{W}$, N/$\overline{S}$, ST0-ST3, SN0-SN6 and $\overline{MREQ}$ outputs will be in the high impedance state. The requesting device should control these lines in an identical fashion to the CPU to accomplish transactions. The $\overline{BUSRQ}$ input must remain LOW as long as needed to perform all the transactions and the CPU will keep the $\overline{BUSAK}$ output LOW. After completing the transactions, the device must disable the AD0-AD15, $\overline{AS}$, $\overline{DS}$, B/$\overline{W}$, R/$\overline{W}$, N/$\overline{S}$, ST0-ST3, SN0-SN6 and $\overline{MREQ}$ into the high impedance state and stop driving the $\overline{BUSRQ}$ input LOW. The CPU will make $\overline{BUSAK}$ output HIGH sometime later and take back the bus control.
24	$\overline{BUSAK}$	O	Bus Acknowledge. LOW on this output indicates that the CPU has relinquished the bus in response to a bus request.
13	$\overline{NMI}$	I	Non-Maskable Interrupt. A HIGH-to-LOW transition on this input constitutes non-maskable interrupt request. The CPU will respond with the Non-maskable Interrupt Acknowledge on the ST0-ST3 outputs and will enter an interrupt sequence. The transition on the $\overline{NMI}$ can occur anytime. Of the three kinds of interrupts available, the non-maskable interrupt has the highest priority.
11	$\overline{VI}$	I	Vectored Interrupt. LOW on this input constitutes vectored interrupt request. Vectored interrupt is next lower to the non-maskable interrupt in priority. The VIE bit in the Flag and Control Word register must be 1 for the vectored interrupt to be honored. The CPU will respond with Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The $\overline{VI}$ input can be driven LOW anytime and should be held LOW until acknowledged.
12	$\overline{NVI}$	I	Non-Vectored Interrupt. LOW on this input constitutes non-vectored interrupt request. Non-vectored has the lowest priority of the three types of interrupts. The NVIE bit in the Flag and Control Word register must be 1 for this request to be honored. The CPU will respond with Non-Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The $\overline{NVI}$ input can be driven LOW anytime and should be held LOW until acknowledged.
7	$\overline{\mu I}$	I	Micro-In. This input participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.
15	$\overline{\mu O}$	O	Micro-Out. This output participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.
14	$\overline{RESET}$	I	Reset. LOW on this input initiates a reset sequence in the CPU. See the section on Initialization for details on reset sequence.
30	CLK	I	Clock. All CPU operations are controlled from the signal fed into this input. See DC Characteristics for clock voltage level requirements.
6	$\overline{STOP}$	I	Stop. This active LOW input facilitates one instruction at a time operation. See the section on single stepping.

DETAILED DESCRIPTION

The following is a description of the Z8001 and Z8002 CPUs.

General Purpose Registers

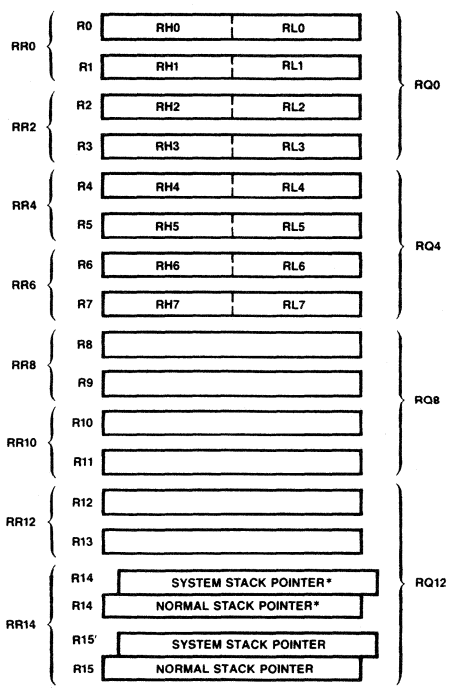
The CPU is organized around sixteen 16-bit general purpose registers R0 through R15 as shown in Figure 1. For byte operations, the first eight registers (R0 through R7) can also be addressed as sixteen 8-bit registers designated as RL0, RH0 and so on to RL7 and RH7. The sixteen registers can also be grouped in pairs RR0, RR2 and so on to RR14 to form eight long word (32-bit) registers. Similarly, the sixteen registers can be grouped in quadruples RQ0, RQ4, RQ8 and RQ12 to form four 64-bit registers.

Stack Pointer (Z8001)

The Z8001 architecture allows stacks to be maintained in memory. Any general-purpose register pair except RR0 can be used as a stack pointer in stack manipulating instructions, such as PUSH and POP. The designated register pair holds a 23-bit segmented address. Certain instructions (such as subroutine call and return) make implicit use of the register pair RR14 as the stack pointer. Two implicit stacks are allowed — normal stack using RR14 as the stack pointer and system stack using RR14′ as the system stack pointer (see Figure 1). If the CPU is operating in the Normal Mode, RR14 is active, and if the CPU is in System Mode, RR14′ will be used instead of RR14. The implied stack pointer is a part of the general registers and hence can be manipulated using the instructions available for register operations.

Stack Pointer (Z8002)

The Z8002 architecture allows stacks to be maintained in the memory. Any general purpose register except R0 can be used as a stack pointer in stack manipulating instructions, such as PUSH and POP. However, certain instructions such as subroutine call and return make implicit use of the register R15 as the stack pointer. Two implicit stacks are maintained — normal stack using R15 as the stack pointer and system stack using R15′ as the system stack pointer (see Figure 1). If the CPU is operating in the Normal Mode, R15 is active, and if the CPU is in System Mode, R15′ will be used instead of R15. The implied stack pointer is a part of the general registers and hence can be manipulated using the instructions available for register operations.



*AmZ8001 only

AF002640

Figure 1. CPU General Registers

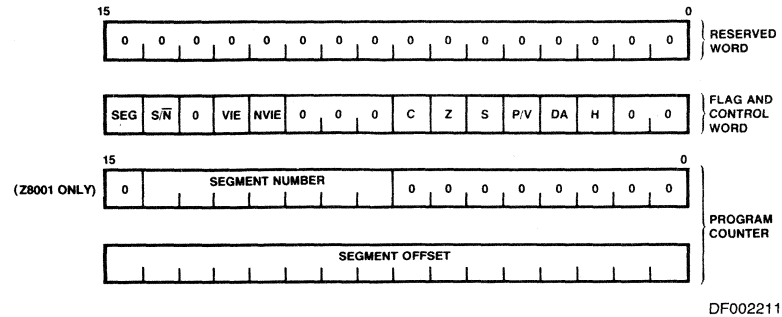


Figure 2. CPU Processor Status

Processor Status

The CPU status consists of the 16-bit flag and control word (FCW) register, and the 16- or 23-bit program counter (see Figure 2). A reserved word is also included for future expansion. The following is a brief description of the FCW bits.

SEG: Segmented/Non-Segmented Bit. Indicates whether the Z8001 is running in segmented or non-segmented mode. 1 indicates segmented; 0 indicates non-segmented. See the section on non-segmented mode, elsewhere in this document. This bit is always 0 in the Z8002.

S/$\overline{N}$: System/Normal – 1 indicates System Mode, and 0 indicates Normal Mode.

VIE: Vectored Interrupt Enable – 1 indicates that Vectored Interrupt requests will be honored.

NVIE: Non-Vectored Interrupt Enable – 1 indicates that Non-vectored interrupt requests will be honored.

C: Carry – 1 indicates that a carry has occurred from the most significant bit position when performing arithmetic operations.

Z: Zero – 1 indicates that the result of an operation is zero.

S: Sign – 1 indicates that the result of an operation is negative; i.e., most significant bit is one.

P/V: Parity/Overflow – 1 indicates that there was an overflow during arithmetic operations. For byte logical operations, this bit indicates parity of the result.

DA: Decimal Adjust – Records byte arithmetic operations.

H: Half Carry – 1 indicates that there was a carry from the most significant bit of the lower digit during byte arithmetic.

Data Types

The CPU instructions operate on bits, digits (4 bits), bytes (8 bits), words (16 bits), long words (32 bits), byte strings and word strings type operands. Bits can be set, reset or tested. Digits are used to facilitate BCD arithmetic operations. Bytes are used for characters and small integers. Words are used for integer values and addresses while long words are used for large integer values and addresses. All operands except strings can reside either in memory or general registers. Strings can reside in memory only.

Interrupt and Trap Structure

Interrupt is defined as an external asynchronous event requiring program interruption. For example, interruption is caused by a peripheral needing service. Traps are synchronous events resulting from execution of certain instructions under some defined circumstances. Both interrupts and traps are handled in a similar manner.

The CPU supports three types of interrupts in order of descending priority — non-maskable, vectored and non-vectored. The vectored and non-vectored interrupts can be disabled by appropriate control bits in the FCW. The CPU has four traps — system call, segment trap, unimplemented opcode and privileged instruction. The traps have higher priority than interrupts.

When an interrupt or trap occurs, the current program status is automatically pushed onto the system stack. The program status consists of processor status (i.e., PC and FCW) plus a 16-bit identifier. The identifier contains the reason, source and other coded information relating to the interrupt or trap.

After saving the current program status, the new processor status is automatically loaded from the new program status area located in the memory. This area is designated by the New Program Status Area Pointer (NPSAP) register.

Segmented Addressing (Z8001 Only)

The Z8001 can directly address up to 8MB of memory space, using a 23-bit segmented address. The memory space is divided up into 128 segments, each up to 64kB in size. The upper seven bits of address designate the segment number and are available on the SN0-SN6 outputs during a memory transaction. See the section on memory transactions for details.

The lower sixteen bits of address designate an offset within the segment, relative to the start of the segment, and are available on AD0-AD15 during part of the memory transaction. See the section on memory transactions for details.

The segmented address may be stored as a long word in memory or in a register pair. The segment number and offset can be manipulated separately or together by suitable use of the instruction set.

When the segmented address is contained in code space, a short offset format may be adopted. The segmented address is stored as one word, seven bits of segment number and eight bits of offset. Figure 3 shows the format for segmented addresses.

Addressing Modes

Information contained in the CPU instructions consists of the operation to be performed, the operand type and the location of the operands. Operand locations are designated by general register addresses, memory addresses or I/O addresses. The addressing mode of a given instruction defines the address space referenced and the method to compute the operand address. Addressing modes are explicitly specified or implied in an instruction. Figure 4 illustrates the eight explicit address-

ing modes: Register (R), Immediate (IM), Indirect Register (IR), Direct Address (DA), Indexed (X), Relative Address (RA), Base Address (BA) and Base Indexed (BX).

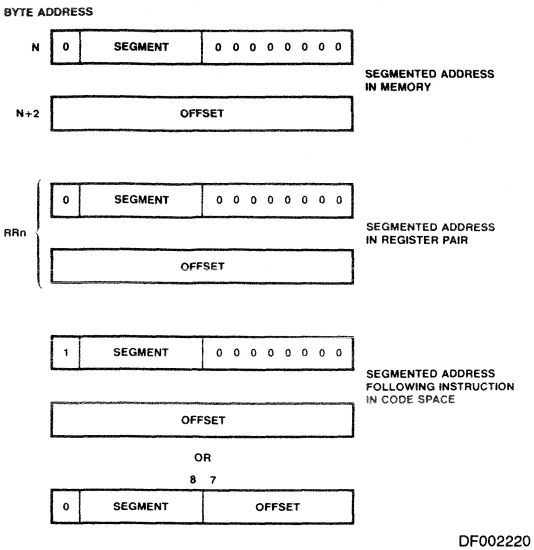


Figure 3. Segmented Address Formats

When an effective segmented address is being computed according to the designated addressing mode, the segment number is not affected by any carry from the 16-bit offset.

Non-Segmented Mode on the Z8001

The Z8001 can execute code designed to run on the non-segmented Z8002. This is achieved by changing the mode of execution of the Z8001 from segmented to non-segmented by writing a 0 to the SEG bit in the FCW. (See the section on processor status.) The change to non-segmented mode sets up a suitable environment for running non-segmented code. However, the environment only exists within the code segment that caused the change of mode from segmented to non-segmented.

SN0-SN6 will continue to indicate the code segment until a reset, interruption or return to segmented mode is encountered.

The effects of the non-segmented mode of operation on the Z8001 are described below:

a) The Z8001 will interpret instruction length like it was a non-segmented Z8002.

b) The Z8001 will implement address computation in an identical manner to the Z8002.

Other CPU functions, such as interrupt and trap handling, reset and stack pointer manipulation, are unaltered. These functions are characterized by the type of CPU, not by the state of the SEG bit in the FCW.

Input/Output

A set of I/O instructions are provided to accomplish byte or word transfers between the CPU and I/O devices. I/O devices are addressed using 16-bit I/O port addresses, and I/O address space is not a part of the memory address space. Two types of I/O instructions are provided, each with its own 16-bit address space. I/O instructions include a comprehensive set of In, Out and Block transfers.

CPU Timing

The CPU accomplishes instruction execution by stepping through a pre-determined sequence of machine cycles, such as memory read, memory write, etc. Each machine cycle requires between three and ten clock cycles. But Requests by DMA devices are granted at machine cycle boundaries. No machine cycle is longer than ten clock cycles, thus assuring fast response to a Bus Request (assuming no extra wait states). The start of a machine cycle is always marked by a LOW pulse on the $\overline{AS}$ output. The status output lines ST0-ST3 indicate the nature of the current cycle in a coded form.

Status Line Codes

Status line coding was listed in the table shown under Pin Description. The following is a detailed description of the status codes.

Internal Operation:

This status code indicates that the CPU is going through a machine cycle for its internal operation. Figure 5 depicts an internal operation cycle. It consists of three clock periods identified as T1, T2 and T3. The $\overline{AS}$ output will be activated with a LOW pulse by the Z8001 to mark the start of a machine cycle. The ST0-ST3 will reflect the code for the internal operation. The $\overline{MREQ}$, $\overline{DS}$ and R/$\overline{W}$ outputs will be HIGH. The N/$\overline{S}$ and SN0-SN6 outputs will remain at the same level as in the previous machine cycle. The CPU will ignore the $\overline{WAIT}$ input during the internal operation cycle. The CPU will drive the AD0-AD15 bus with unspecified information during T1. However, the bus will go into high-impedance during T2 and remain in that state for the remainder of the cycle. The B/$\overline{W}$ output is also activated by the CPU with unspecified information.

Memory Refresh:

This status code indicates that CPU is accessing the memory to refresh. The refresh cycle consists of three clock periods as depicted in Figure 6. The CPU will activate the $\overline{AS}$ output with a LOW pulse to mark the beginning of a machine cycle, and ST0-ST3 outputs will reflect the refresh cycle code. The least significant 9 lines of the AD0-AD15 bus contain the refresh address. Because the memory is word organized, the AD0 will always be LOW. The most significant 7 bus lines are not specified. The $\overline{DS}$ output will remain HIGH for the entire cycle, while R/$\overline{W}$, B/$\overline{W}$, SN0-SN6 and N/$\overline{S}$ outputs will remain at the same level as in the machine cycle prior to refresh. The AD0-AD15 bus will go into high-impedance state during T2 period and remain there for the remainder of the cycle. The CPU will activate the $\overline{MREQ}$ output LOW during the refresh cycle. It should be noted that $\overline{WAIT}$ input is ignored by the CPU for refresh operations.

I/O Transactions:

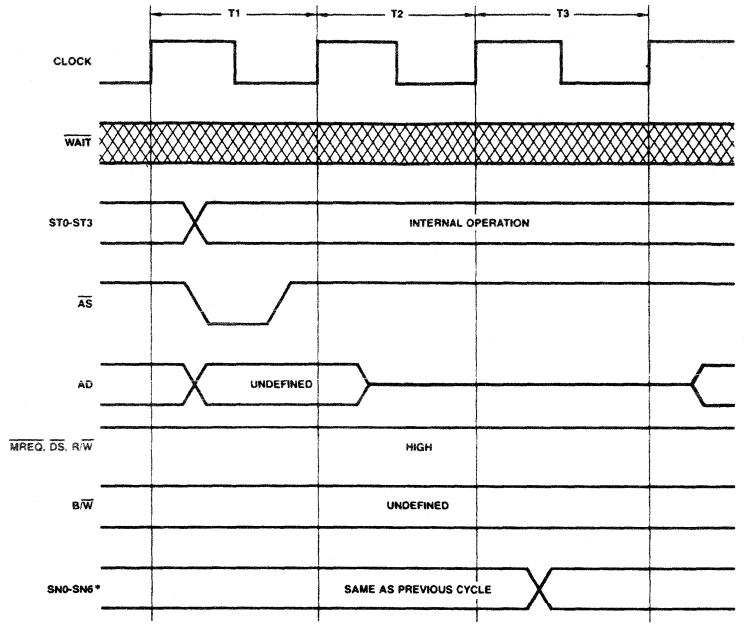
There are two status line codes used for I/O transaction cycles. The CPU provides two separate I/O spaces and two types of instructions called Normal I/O and Special I/O. Each I/O space is addressed by a 16-bit address called port address. The timing for both types of I/O transactions is essentially identical. A typical I/O cycle consists of four clock periods T1, T2, TWA and T3 as shown in Figure 7. The TWA is the wait state; insertion of one wait state for an I/O cycle is always automatic. Additional wait cycles can be inserted by LOW on the $\overline{WAIT}$ input. The $\overline{WAIT}$ input is sampled during every TW state. If this input is LOW, one more wait state will be inserted. Insertion of wait states continues until $\overline{WAIT}$ input is HIGH. T3 state will follow the last wait state to complete the I/O cycle.

Mode	Operand Addressing			Operand Value
	In the Instruction	In a Register	In Memory	
Register		REGISTER ADDRESS → OPERAND		The content of the register.
Immediate	OPERAND			In the instruction
Indirect Register		REGISTER ADDRESS → ADDRESS	→ OPERAND	The content of the location whose address is in the register.
Direct Address			ADDRESS → OPERAND	The content of the locations whose address is in the instruction.
Index		REGISTER ADDRESS, BASE ADDRESS; DISPLACEMENT → ⊕	→ OPERAND	The content of the location whose address is the address in the instruction, offset by the content of the working register.
Relative Address		DISPLACEMENT; PC VALUE → ⊕	→ OPERAND	The content of the locations whose address is the content of the program counter, offset by the displacement in the instruction.
Base Address		REGISTER ADDRESS, DISPLACEMENT; BASE ADDRESS → ⊕	→ OPERAND	The content of the location whose address is the address in the register, offset by the displacement in the instruction.
Base Index		REGISTER ADDRESS, REGISTER ADDRESS; BASE ADDRESS, DISPLACEMENT → ⊕	→ OPERAND	The content of the location whose address is the address in the register, offset by the displacement in the register.

Figure 4. Addressing Modes

During I/O cycles the ST0-ST3 outputs will reflect the appropriate code depending on the type of instruction being executed (Normal I/O or Special I/O). $\overline{AS}$ output will be pulsed LOW to mark the beginning of the cycle. The CPU drives the AD0-AD15 bus with the 16-bit port address specified by the current instruction. The N/$\overline{S}$ output will be LOW indicating that the CPU is operating in the system mode. It should be recalled that the N/$\overline{S}$ output is derived from the appropriate bit in the FCW register. All I/O instructions are privileged instructions and will be allowed to execute only if the FCW specifies system mode operation. The $\overline{MREQ}$ output will be HIGH. The I/O instructions provide both word or byte transactions. The B/$\overline{W}$ output will be HIGH or LOW depending whether the instruction specifies a byte or word transfer. The SN0-SN6 output will remain at the same level as in the machine cycle prior to the I/O cycle.

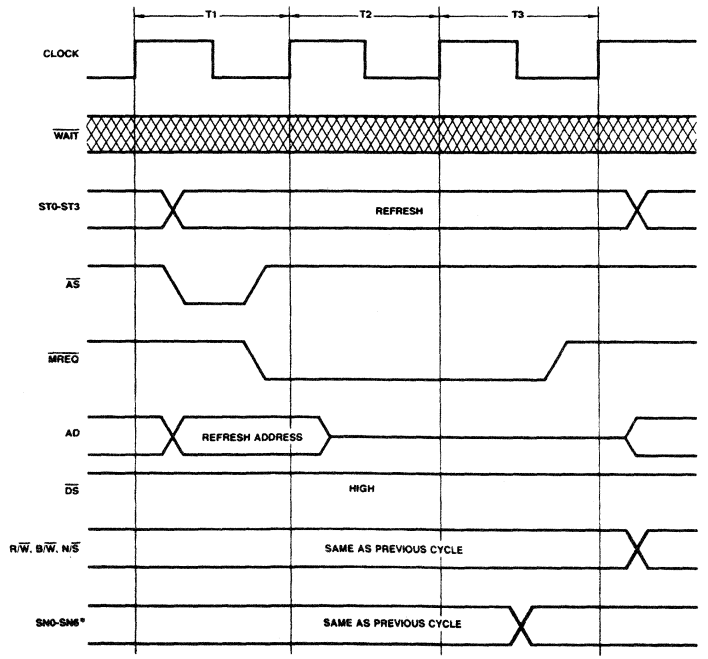
Two kinds of I/O transfers should be considered: Data In means reading from the device, and Data Out means writing into the device. For In operations, the R/$\overline{W}$ output will be HIGH. The AD0-AD15 bus will go into high-impedance state during T2. During byte input instructions, the CPU reads either the even or odd half of the Data Bus dependent upon the port address. If the port address is even, the most significant half of the Data Bus is read. If the port address is odd, the least significant half of the Data Bus is read. During word input instructions, the CPU reads all 16 bits of the Data Bus. The CPU will drive the $\overline{DS}$ output LOW to signal to the device that data can be gated onto the bus. The CPU will accept the data during T3, and $\overline{DS}$ output will go HIGH signaling the end of an I/O transaction.



WF005210

*Z8001 only

Figure 5. Internal Operation Cycle

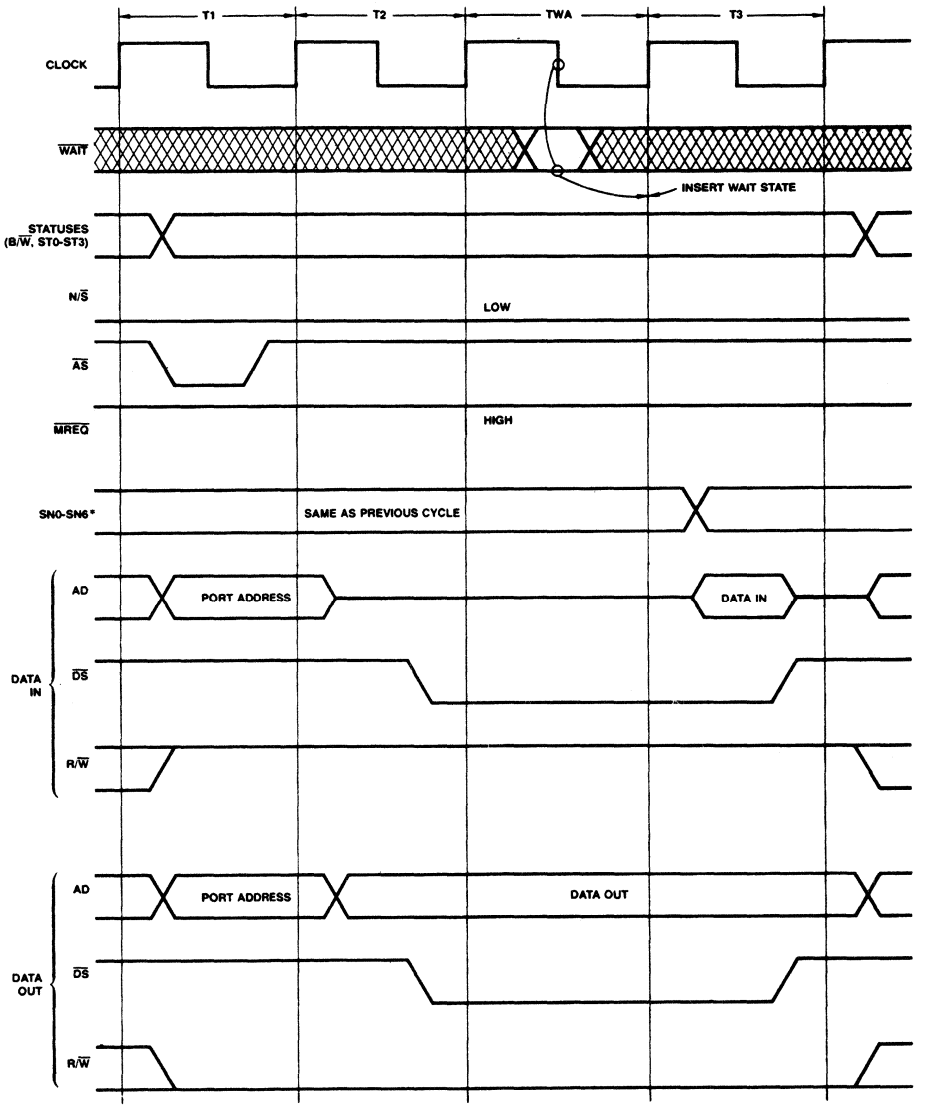


WF005220

*Z8001 only

Figure 6. Refresh Cycle

5



*Z8001 only

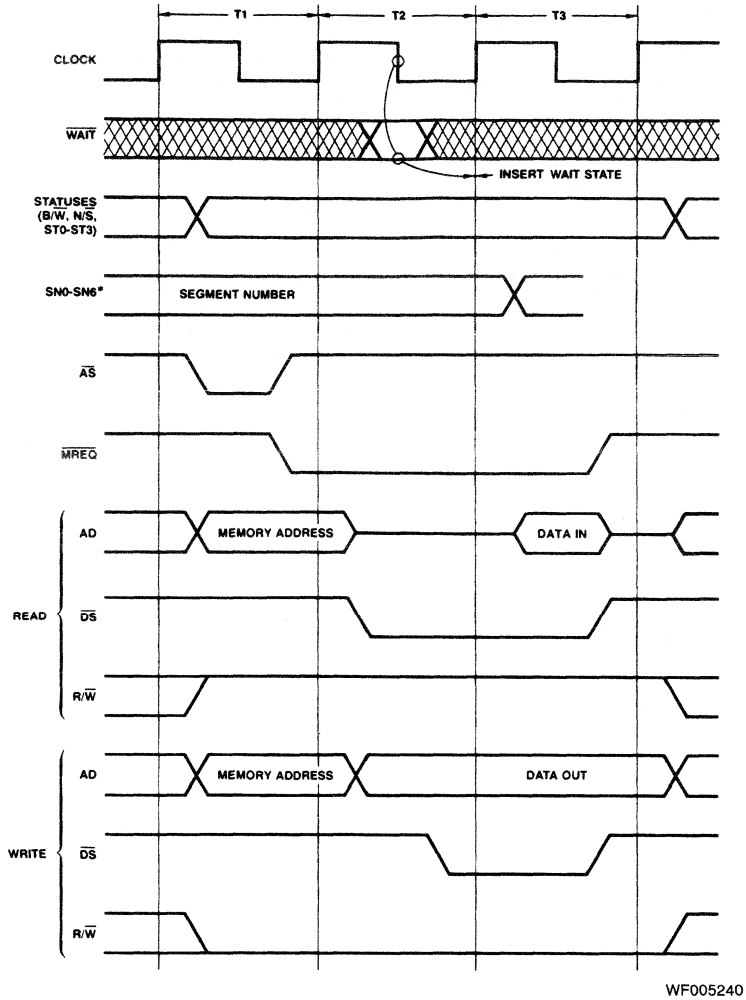
Figure 7. Z8001 I/O Cycle

For Data Out, the R/$\overline{W}$ output will be LOW. The CPU will provide data on the AD0-AD15 bus and activate the $\overline{DS}$ output LOW during T2. During byte output instructions, the CPU duplicates the byte data onto both the high and low halves of the Data Bus, and external logic, using A0, enables the appropriate byte port. During word output instructions the CPU outputs data onto all 16 bits of the Data Bus. The $\overline{DS}$ output goes HIGH during T3 and the cycle is complete.

Memory Transactions:

There are four status line codes that indicate a memory transaction:

a) Memory transaction to read or write an operand;
b) Memory transaction to read from or write into the stack;
c) Memory transaction to fetch the first word of an instruction (sometimes called IF1); and
d) Memory transaction to fetch the subsequent word of an instruction (sometimes called IFN).



*Z8001 only

Figure 8. Memory Transactions

It can be appreciated that all the above transactions essentially fall into two categories: memory read and memory write. In the case of IF1 and IFN cycles, the memory will be read at the address supplied by the program counter. All instructions are multiples of 16-bit words. Words are always addressed by an even address. Thus IF1 and IFN cycles involve performing a memory read for words. On the other hand, a memory transaction for operand and stack operation could be a read or write. Moreover, an operand could be a word or a byte. For stack operation involving the implied stack pointer, the address will be in the appropriate stack pointer register (R15, R15', RR14 or RR14'). For operand transactions, the memory address will come from several sources depending on the instruction and the addressing mode. Memory transaction cycle timing is shown in Figure 8. It typically consists of three clock periods: T1, T2 and T3. Wait states (TW) can be inserted between T2 and T3 by activating the $\overline{WAIT}$ input LOW. The $\overline{WAIT}$ input will be sampled during T2 and during every subsequent TW. The ST0-ST3 outputs will reflect the appropriate code for the current cycle early in T1, and the $\overline{AS}$ output will be pulsed LOW to mark the beginning of the cycle. The N/$\overline{S}$ output will indicate whether the normal or system address space will be used for the current cycle. As shown in the figure, the $\overline{MREQ}$ output will go LOW during T1 to indicate a memory operation.

The segment number becomes valid on the segment lines one clock period before the state of the memory operation and remains valid until the state of T3.

Consider a read operation first. The R/$\overline{W}$ output will be HIGH. The CPU will drive the AD0-AD15 with the appropriate address early in T1. During T2, the bus will go into high-impedance state, and $\overline{DS}$ output will be activated LOW by the CPU. The data can be gated onto the bus when $\overline{DS}$ is LOW. During T1 the B/$\overline{W}$ will also be activated to indicate which byte or word will be transacted. The memory is word organized, and words

are addressed by even addresses. However, when addressing bytes, the memory address may be odd or even, an even address for the most significant byte of a word and the next odd address for the least significant byte of that word. When reading a byte from the memory, the least significant address bit can be ignored and the whole word containing the desired byte is gated onto the bus. The CPU will pick the appropriate byte automatically and will drive the $\overline{DS}$ output HIGH indicating data acceptance.

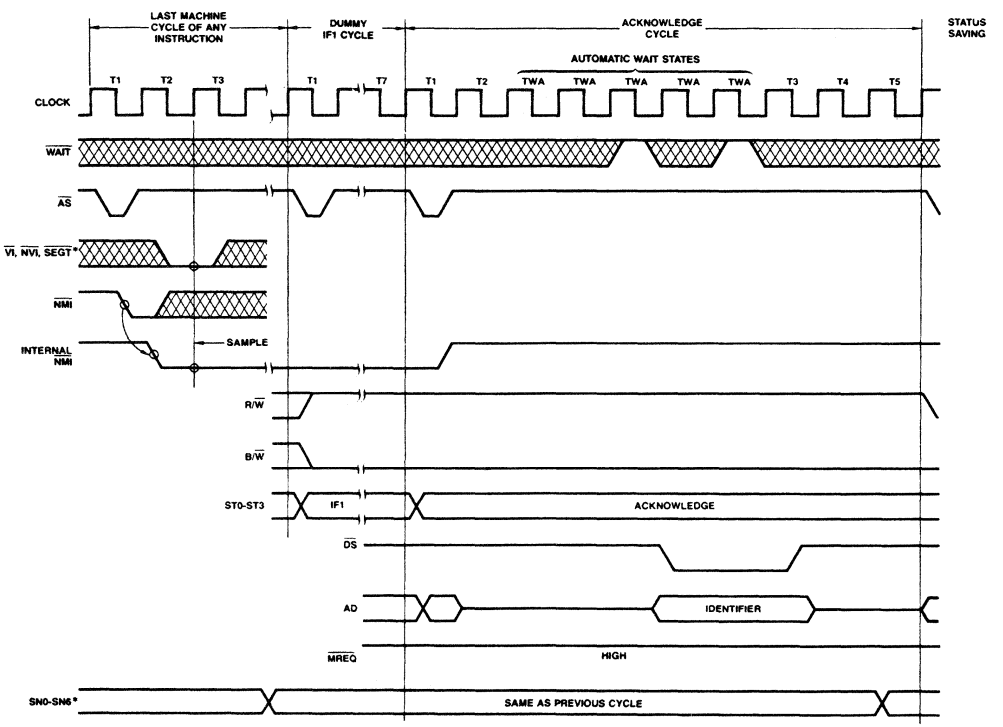
Consider the write operation next. The R/$\overline{W}$ output will be LOW. The CPU removes the address and gates out the data to be written on the bus and activates the $\overline{DS}$ output LOW during T2. If the data to be written is a byte, then the same byte will be on both halves of the bus. The $\overline{DS}$ output will go HIGH during T3, signifying completion of the cycle.

Interrupt and Segment Trap Acknowledge:

There are four status line codes devoted to interrupt and trap acknowledgement. These correspond to non-maskable, vectored and non-vectored interrupts, as well as segment trap. The Interrupt Acknowledge cycle is illustrated in Figure 9. The $\overline{NMI}$ input of the Z8001 is edge detected; i.e., a HIGH-to-LOW input level change is stored in an internal latch. Similar internal storage is not provided for the $\overline{VI}$, $\overline{NVI}$, and $\overline{SEGT}$ inputs. For $\overline{VI}$ and $\overline{NVI}$ inputs to cause an interruption, the corresponding interrupt enable bits in the FCW must be 1. For the following discussion, both the NVIE and VIE bits in the FCW are assumed to be 1.

As shown in the figure, the $\overline{VI}$, $\overline{NVI}$ and $\overline{SEGT}$ input and the internal $\overline{NMI}$ latch output are sampled during T3 of the last machine cycle of an instruction.

A LOW on these signals triggers the corresponding interrupt acknowledge sequence described below. The CPU executes a dummy IF1 cycle prior to entering the actual acknowledge cycle (see memory transactions for IF1 cycle description).



WF005250

*Z8001 only

Figure 9. Interrupt Acknowledge Cycle

During this dummy IF1 cycle, the program counter is not updated; instead, the implied-system stack pointer (RR14') will be decremented. Following the dummy IF1 cycle is the actual interrupt/trap acknowledge cycle.

The interrupt acknowledge cycle typically consists of 10 clock periods: T1 through T5 and five automatic TW (wait states). As usual, the $\overline{AS}$ output will be pulsed LOW during T1 to mark the beginning of a cycle. The ST0-ST3 outputs will reflect the appropriate interrupt acknowledge code; the $\overline{MREQ}$ output will be HIGH; the N/$\overline{S}$ output remains the same as in the preceding cycle; the R/$\overline{W}$ output will be HIGH; and the B/$\overline{W}$ output will be LOW. The CPU will drive the AD0-AD15 bus with

unspecified information during T1, and the bus will go into the high-impedance state during T2. Three TWA states will automatically follow T2. The WAIT input will be sampled during the third TWA state.

If LOW, an extra TW state will be inserted and the WAIT will be sampled again during TW. Such insertion of TW states continues until the WAIT input is HIGH. After the last TW state, the DS output will go LOW and two more automatic wait states (TWA) follow. The interrupting device can gate up to a 16-bit identifier onto the bus when the DS output is LOW. The WAIT input will be sampled again during the last TWA state. If the WAIT input is LOW, one TW state will be inserted, and the WAIT will be sampled during TW. Such TW insertion continues until the WAIT input is HIGH. After completing the last TW state, T3 will be entered, and the DS output will go HIGH. The interrupting device should remove the identifier and cease driving the bus. T4 and T5 states will follow T3 to complete the cycle. Following the interrupt acknowledge cycle will be memory transaction cycles to save the status on the stack. Note that the N/S output will be automatically LOW during status saving. The SN0-SN6 outputs are undefined during the acknowledge cycle.

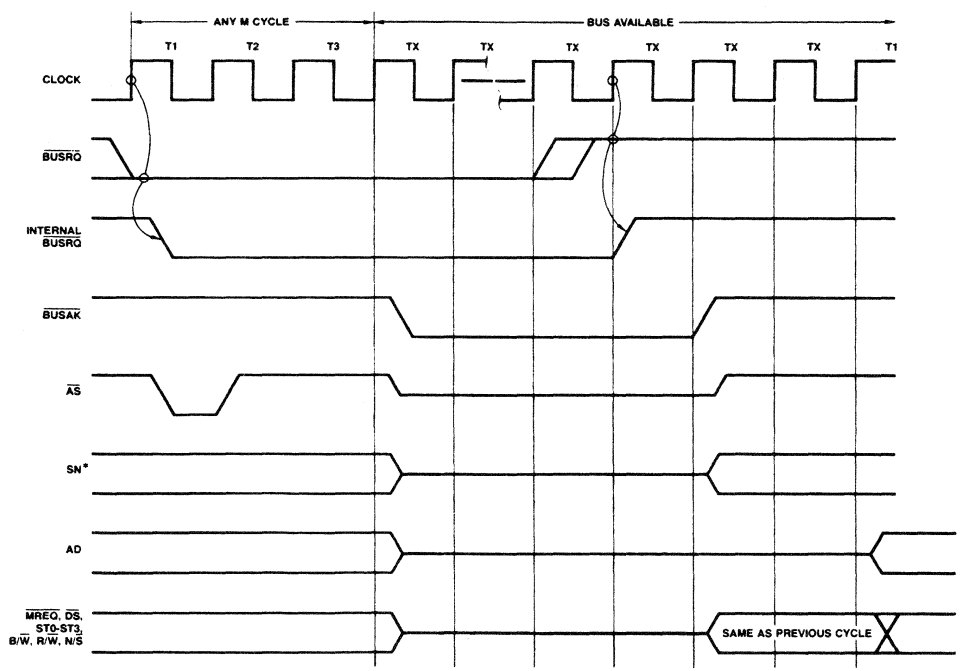
The internal NMI latch will be reset to the initial state as AS going HIGH in the interrupt acknowledge cycle. The VI, NVI and SEGT input should be kept LOW until this time also.

Status Saving Sequence:

The machine cycles following the interrupt acknowledge cycle push the old status information on the system stack in the following order: program counter, the flag and control word, and the interrupt/trap identifier. Subsequent machine cycles fetch the new program status from the new program status area and then branch to the interrupt/service routine.

Bus Request/Bus Acknowledge Timing:

A LOW on the BUSRQ input is an indication to the CPU that another device (such as DMA) is requesting control of the bus. The BUSRQ input is synchronized internally at T1 of any machine cycle. (See next paragraph for exception.) The BUSAK will go LOW after the last clock period of the machine cycle. The LOW on the BUSAK output indicates acknowledgement. When BUSAK is LOW, the following outputs will go into the high-impedance state: AD0-AD15, AS, DS, MREQ, ST0-ST3, B/W, R/W, SN0-SN6 and N/S. The BUSRQ must be held LOW until all transactions are completed. When BUSRQ goes HIGH, it is synchronized internally; the BUSAK output will go HIGH, and normal CPU operation will resume. Figure 10 illustrates the BUSRQ/BUSAK timing.



*Z8001 only

Figure 10. Bus Request/Acknowledge Cycle

It was mentioned that BUSRQ will be honored during any machine cycle with one exception. This exception is during the execution of TSET/TSETB instructions. BUSRQ will not be honored once execution of these instructions has started.

Single Stepping

The $\overline{STOP}$ input of the CPU facilitates one instruction at a time or single step operation. Figure 11 illustrates $\overline{STOP}$ input timing. The $\overline{STOP}$ input is sampled on the HIGH-to-LOW transition of the clock input that immediately precedes an IF1 cycle. If the $\overline{STOP}$ is found LOW, Z8001 introduces a memory refresh cycle after T3. Moreover, $\overline{STOP}$ input will be sampled again at T3 in the refresh cycle. If $\overline{STOP}$ is LOW, one more refresh cycle will follow the previous refresh cycle. The $\overline{STOP}$ will be sampled during T3 of the refresh cycle, also. One additional refresh cycle will be added every time $\overline{STOP}$ input is sampled LOW. After completing the last refresh cycle which will occur after $\overline{STOP}$ is HIGH, the CPU will insert two dummy states, T4 and T5, to complete the IF1 cycle and resume its normal operations for executing the instruction. See appropriate sections on memory transactions and memory refresh. It should be noted that refresh cycles will occur even if the refresh facility is disabled during single stepping.

Multimicroprocessor Facilities

The CPU is provided with hardware and software facilities to support multiple microprocessor systems. The $\overline{\mu O}$ and $\overline{\mu I}$ signals of the CPU are used in conjunction with the MBIT, $\overline{MREQ}$, MRES and MSET instructions for this purpose. The $\overline{\mu O}$ output can be activated LOW by using appropriate instruction to signal a request from the CPU for a resource. The $\overline{\mu I}$ input is tested by the CPU before activating the $\overline{\mu O}$ output. LOW at the $\overline{\mu I}$ input indicates that the resource is busy. The CPU can examine the $\overline{\mu I}$ input after activating the $\overline{\mu O}$ output LOW. The $\overline{\mu I}$ will be tested again to see if the requested resource became available.

Initialization

A LOW on the $\overline{Reset}$ input starts the CPU initialization. The initialization sequence is shown in Figure 12. Within five clock periods after the HIGH-to-LOW level change of the $\overline{Reset}$ input, the following will occur:

a) AD0-AD15 bus will be in the high-impedance state.

b) $\overline{AS}$, $\overline{DS}$, $\overline{MREQ}$, $\overline{BUSAK}$ and $\overline{\mu O}$ outputs will be HIGH.

c) ST0-ST3 outputs will be LOW.

d) Refresh will be disabled.

e) R/$\overline{W}$, B/$\overline{W}$ and N/$\overline{S}$ outputs are not affected. For a power-on reset, the state of these outputs is not specified.

f) SN0-SN6 outputs will be LOW.

After the $\overline{Reset}$ input returns HIGH and remains HIGH for three clock periods, two (three for the Z8001) 16-bit memory read operations will be performed as follows. Note that the N/$\overline{S}$ output will be LOW, and ST0-ST3 outputs will reflect IFN code.

a) The contents of the memory location 0002 (segment 0) will be read. This information will be loaded into the FCW of the CPU.
b) The contents of the memory location 0004 (segment 0) will be read. This information will be loaded into the program counter segment number.
c) (Z8001 only.) The contents of the memory location 0006 (segment 0) will be read. This information will be loaded into the program counter offset.

This completes initialization sequence, and an IF1 cycle will follow to fetch the first instruction to begin program execution. See the section on memory transactions for timing.

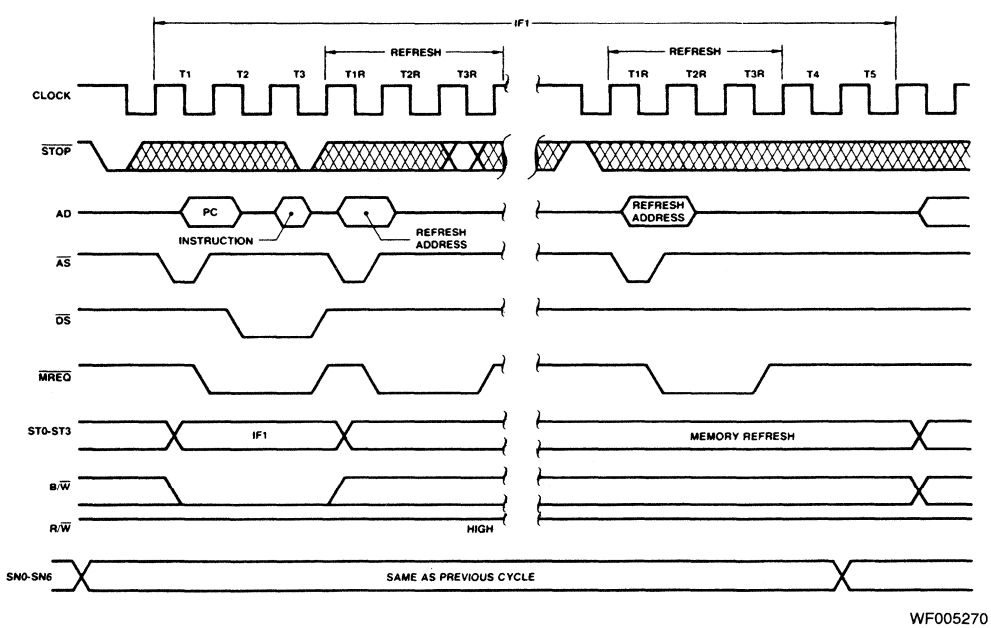
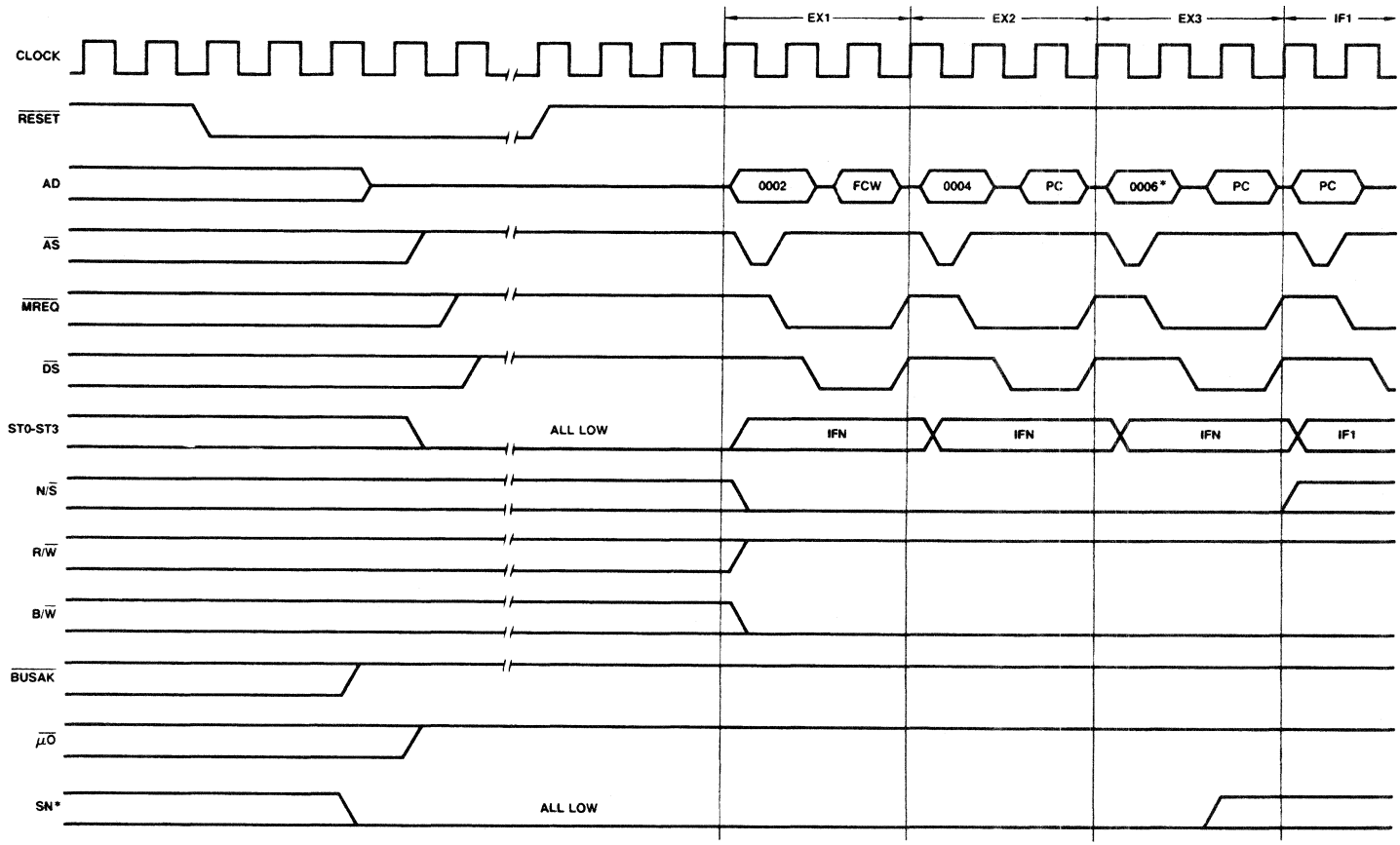


Figure 11. Single Step Timing



WF005280

*Z8001 only

Z8001/2 CPU INSTRUCTION SET

ARITHMETIC

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Add	ADC ADCB	R, src	R	5	Add with Carry R←R + src + carry
	ADD ADDB ADDL	R, src	R IM IR DA X	4 7 7 9 10	Add R←R + src
Compare	CP CPB CPL	R, src	R IM IR X	4 7 7 10	Compare with Register R – src
	CP CPB	dst, IM	IR DA X	11 14 15	Compare with Immediate dst – IM
	DAB	dst	R	5	Decimal Adjust
Decrement	DEC DECB	dst, n	R IR DA X	4 11 13 14	Decrement by n dst←dst – n (n = 1 . . . 16)
Divide	DIV DIVL	R, src	R IM IR DA X	107 107 107 108 109	Divide (signed) Word: $R_n + 1 \leftarrow R_{n,n} + 1 \div src$ $R_n \leftarrow$ remainder Long Word: $R_n + 2,n + 3$ $\leftarrow R_{n...n} + 3 \div src$ $R_{n,n} + 1$ ←remainder
Extend	EXTS EXTSB EXTSL	dst	R	11	Extend Sign Extend sign of low order half of st through high order half of dst
Increment	INC INCB	dst, n	R IR DA X	4 11 13 14	Increment by n dst←dst + n (n = 1 . . . 16)
Multiply	MULT MULTL	R, src	R IM IR DA X	70 70 70 71 72	Multiply (signed) Word: $R_{n,n} + 1 \leftarrow R_n + 1 \bullet src$ Long Word: $R_{n...n} + 3$ $\leftarrow R_n + 2, + 3 \bullet src$ *Plus seven cycles for each 1 in the multiplicand
Negate	NEG NEGB	dst	R R DA X	7 12 15 16	Negate dst←0 – dst
Subtract	SBC SBCB	R, src	R	5	Subtract with Carry R←R – src – carry
	SUB SUBB SUBL	R, src	R IM IR DA X	4 7 7 9 10	Subtract R←R – src

LOGICAL

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
AND	AND ANDB	R, scr	R IM IR DA X	4 7 7 9 10	AND R←R AND src
Complement	COM COMB	dst	R IR DA X	7 12 15 16	Complement dst←NOT dst
OR	OR ORB	R, src	R IM IR DA X	4 7 7 9 10	OR R←R OR src
Test	TEST TESTB TESTL	dst	R IR DA X	7 8 11 12	TEST dst OR 0
	TCC TCCB	cc, dst	R	5	Test Condition Code Set LSB if cc is true
XOR	XOR XORB	R, src	R IM IR DA X	4 7 7 9 10	Exclusive OR R←R XOR src

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

LOAD AND EXCHANGE

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Clear	CLR CLRB	dst	R IR DA X	7 8 11 12	Clear dst←0
Exchange	EX EXB	R, src	R IR DA X	6 12 15 16	Exchange R←src
Load	LD LDB LDL	R, src	R IM IR DA X RA BA BX	3 7 7 9 10 14 14 14	Load into Register R←src
	LD LDB LDL	dst, R	IR DA X RA BA BX	8 11 12 14 14 14	Load into Memory (Store) dst←R
	LD LDB	dst, IM	R IR DA X	5 11 14 15	Load Immediate into Memory dst←IM
	LDA	R, src	DA X RA BA BX	12 13 15 15 15	Load Address R←source address
	LDAR	R, src	RA	15	Load Address Relative R←source address
	LDK	R, src	IM	5	Load Constant R←n (n = 0...15)
	LDM	R, src, n	IR DA X	11 + 3n 14 + 3n 15 + 3n	Load Multiple R←src (n consecutive words) (n = 1...16)
	LDM	dst, R, n	IR DA X	11 + 3n 14 + 3n 15 + 3n	Load Multiple (Store Multiple) dst←R (n consecutive words) (n = 1...16)
	LDR LDRB LDRL	R, src	RA	14	Load Relative R←src (range – 32768... + 32767)
	LDR LDRB LDRL	dst, R	RA	14	Load Relative (Store Relative) dst←R (range – 32768... + 32767)
Pop	POP POPL	dst, R	R IR DA X	8 12 16 16	Pop dst←IR Autoincrement contents of R
Push	PUSH PUSHL	IR, src	R IM IR DA X	9 12 13 14 14	Push Autodecrement contents of R IR←src

BIT MANIPULATION

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Test	BIT BITB	dst, b	R IR DA X	4 8 10 11	Test Bit Static Z flag←NOT dst bit specified by b
	BIT BITB	dst, R	R	10	Test Bit Dynamic Z flag←NOT dst bit specified by contents of R
Reset	RES RESB	dst, b	R IR DA X	4 11 13 14	Reset Bit Static Reset dst bit specified by b
	RES RESB	dst, R	R	10	Reset Bit Dynamic Reset dst bit specified by contents of R
Set	SET SETB	dst, b	R IR DA X	4 11 13 14	Set Bit Static Set dst bit specified by b
	SET SETB	dst, R	R	10	Set Bit Dynamic Set dst bit specified by contents of R
Test and Set	TSET TSETB	dst	R IR DA X	7 11 14 15	Test and Set S flag←MSB of dst dst←all 1s

ROTATE AND SHIFT

	Mnemonics	Operand	Addr. Modes	Clock Cycles†	Operation
Rotate	RLDB	R, src	R	9	Rotate Digit Left
	RRDB	R, src	R	9	Rotate Digit Right
	RL RLB	dst, n	R R	6	Rotate Left by n bits (n = 1, 2)
	RLC RLCB	dst, n	R R	6	Rotate Left through Carry by n bits (n = 1, 2)
	RR RRB	dst, n	R R	6	Rotate Right by n bits (n = 1, 2)
	RRC RRCB	dst, n	R R	6	Rotate Right through Carry by n bits (n = 1, 2)
Shift	SDA SDAB SDAL	dst, R	R	15 + 3n	Shift Dynamic Arithmetic Shift dst left or right by contents of R
	SDL SDLB SDLL	dst, R	R	15 + 3n	Shift Dynamic Logical Shift dst left or right by contents of R
	SLA SLAB SLAL	dst, n	R	13 + 3n	Shift Left Arithmetic by n bits
	SLL SLLB SLLL	dst, n	R	13 + 3n	Shift Left Logical by n bits
	SRA SRAB SRAL	dst, n	R	13 + 3n	Shift Right Arithmetic by n bits
	SRL SRLB SRLL	dst, n	R	13 + 3n	Shift Right Logical by n bits

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

BLOCK TRANSFER AND STRING MANIPULATION (AUTO INCREMENT/DECREMENT AND REPEAT)

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Compare	CPD CPDB	R_X, src, R_Y, cc	IR	20	Compare and Decrement R_X - src Autodecrement src address $R_Y \leftarrow R_Y - 1$
	CPDR CPDRB	R_X, src, R_Y, cc	IR	11 + 9n	Compare, Decrement and Repeat R_X - src Autodecrement src address $R_Y \leftarrow R_Y - 1$ Repeat until cc is true or $R_Y = 0$
	CPI CPIB	R_X, src, R_Y, cc	IR	20	Compare and Increment R_X - src Autoincrement src address $R_Y \leftarrow R_Y - 1$
	CPIR CPIRB	R_X, src, R_Y, cc	IR	11 + 9n	Compare, Increment and Repeat R_X - src Autoincrement src address $R_Y \leftarrow R_Y - 1$ Repeat until cc is true or $R_Y = 0$
	CPSD CPSDB	dst, src, R, cc	IR	25	Compare String and Decrement dst - src Autodecrement dst and src addresses R ← R - 1
	CPSDR CPSDRB	dst, src, R, cc	IR	11 + 14n	Compare String, Decr. and Repeat dst - src Autodecrement dst and src addresses R ← R - 1 Repeat until cc is true or R = 0
	CPSI CPSIB	dst, src, R, cc	IR	25	Compare String and Increment dst - src Autoincrement dst and src addresses R ← R - 1
	CPSIR CPSIRB	dst, src, R, cc	IR	11 + 14n	Compare String, incr. and Repeat dst - src Autoincrement dst and src addresses R ← R - 1 Repeat until cc is true or R = 0
Load	LDD LDDB	dst, src, R	IR	20	Load and Decrement dst ← src Autodecrement dst and src addresses R ← R - 1
	LDDR LDDRB	dst, src, R	IR	11 + 9n	Load, Decrement and Repeat dst ← src Autodecrement dst and src addresses R ← R - 1 Repeat until R = 0

BLOCK TRANSFER AND STRING MANIPULATION (Cont.)

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Load	LDI LDIB	dst, src, R	IR	20	Load and Increment dst ← src Autoincrement dst and src addresses R ← R - 1
	LDIR LDIRB	dst, src, R	IR	11 + 9n	Load, Increment and Repeat dst ← src Autoincrement dst and src addresses R ← R - 1 Repeat until R = 0
Translate	TRDB	dst, src, R	IR	25	Translate and Decrement dst ← src (dst) Autodecrement dst address R ← R - 1
	TRDRB	dst, src, R	IR	11 + 14n	Translate, Decrement and Repeat dst ← src (dst) Autodecrement dst address R ← R - 1 Repeat until R = 0
	TRIB	dst, src, R	IR	25	Translate and Increment dst ← src (dst) Autoincrement dst address R ← R - 1 0
	TRIRB	dst, src, R	IR	11 + 14n	Translate, Increment and Repeat dst ← src (dst) Autoincrement dst address R ← R - 1 Repeat until R = 0
Translate and Test	TRTDB	src 1, src 2, R	IR	25	Translate and Test, Decrement RH1 ← src 2 (src 1) Autodecrement src 1 address R ← R - 1
	TRTDRB	src 1, src 2, R	IR	11 + 14n	Translate and Test, Decrement and Repeat RH1 ← src 2 (src 1) Autodecrement src 1 address R ← R - 1 Repeat until R = 0 or RH1 = 0
	TRTIB	src 1, src 2, R	IR	25	Translate and Test, Increment RH1 ← src 2 (src 1) Autoincrement src 1 address R ← R - 1
	TRTIRB	src 1, src 2, R	IR	11 + 14n	Translate and Test, Increment and Repeat RH1 ← src 2 (src 1) Autoincrement src 1 address R ← R - 1 Repeat until R = 0 or RH1 = 0

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

INPUT/OUTPUT

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Input	IN* INB*	R, src	IR DA	10 12	Input R ← src
	IND* INDB*	dst, src, R	IR	21	Input and Decrement dst ← src Autodecrement dst address R ← R − 1
	INDR* INDRB*	dst, src, R	IR	11 + 10n	Input, Decrement and Repeat dst ← src Autodecrement dst address R ← R − 1 Repeat until R = 0
	INI* INIB*	dst, src, R	IR	21	Input and Increment dst ← src Autoincrement dst address R ← R − 1
	INIR* INIRB*	dst, src, R	IR	11 + 10n	Input, Increment and Repeat dst ← src Autoincrement dst address R ← R − 1 Repeat until R = 0
Output	OUT* OUTB*	dst, R	IR DA	10 12	Output dst ← R
	OUTD* OUTDB*	dst, src, R	IR	21	Output and Decrement dst ← src Autodecrement src address R ← R − 1
	OTDR* OTDRB*	dst, src, R	IR	11 + 10n	Output and Decrement dst ← src Autodecrement src address R ← R − 1 Repeat until R = 0
	OUTI* OUTIB*	dst, src, R	IR	21	Output and Increment dst ← src Autoincrement src address R ← R − 1
	OTIR* OTIRB*	dst, src, R	IR	11 + 10n	Output, Increment and Repeat dst ← src Autoincrement src address R ← R − 1 Repeat until R = 0
Special input (identical to input but different status code)	SIN* SINB*	R, src	DA	12	Special Input R ← src
	SIND* SINDB*	dst, src, R	IR	21	Special Input and Decrement dst ← src Autodecrement dst address R ← R − 1
	SINDR* SINDRB*	dst, src, R	IR	11 + 10n	Special Input, Decr. and Repeat dst ← src Autodecrement dst address R ← R − 1 Repeat until R = 0
	SINI* SINIB*	dst, src, R	IR	21	Special Input and Increment dst ← src Autoincrement dst address R ← R − 1
	SINIR* SINIRB*	dst, src, R	IR	11 + 10n	Special Input, Incr. and Repeat dst ← src Autoincrement dst address R ← R − 1 Repeat until R = 0

INPUT/OUTPUT (Cont.)

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Special Output (identical to output, but different status code)	SOUT* SOUTB*	dst, src	DA	12	Special Output dst ← src
	SOUTD* SOUTDB*	dst, src R	IR	21	Special Output and Decrement dst ← src Autodecrement src address R ← R − 1
	SOTDR* SOTDRB*	dst, src, R	IR	11 + 10n	Special Output, Decr. and Repeat dst ← src Autodecrement src address R ← R − 1 Repeat until R = 0
	SOUTI* SOUTIB*	dst, src R	IR	21	Special Output and Increment dst ← src Autoincrement src address R ← R − 1
	SOTIR* SOTIRB*	dst, src R	R	11 + 10n	Special Output, Incr. and Repeat dst ← src Autoincrement src address R ← R − 1 Repeat until R = 0

CPU CONTROL

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Interrupts	DI*	int	–	7	Disable Interrupt (Any combination of NVI, VI)
	EI*	int	–	7	Enable Interrupt (Any combination of NVI, VI)
Halt	HALT*	–	–	8 + 3n	HALT
Control Words	LDCTL*	CTLR, src	R	7	Load into Control Register CTLR ← src
	LDCTL*	dst, CTLR	R	7	Load into Control Register dst ← CTLR
	LDCTLB	dst, FLGR	R	7	Load into Flag Byte Register FLGR ← src
	LDPS*	src	IR DA X	12 16 17	Load Program Status PS ← src
Multi Micro	MBIT*	–	–	7	Test Multi-Micro Bit Set S if μI is HIGH; reset S if μI is LOW
	MREQ*	dst	R	12 + 7n	Multi-Micro Request
	MRES*	–	–	5	Multi-Micro Reset
	MSET*	–		5	Multi-Micro Set
NOP	NOP			7	No Operation
Flags	RESFLG	flag		7 7	Reset Flag (Any combination of C, Z, S, P/V)
	SETFLG	flag		7	Set Flag (Any combination of C, Z, S, P/V)
	COMFLG	flags	–	7	Complement Flag (Any combination of C, Z, S, P/V)

*Privileged instructions. Executed in system mode only.

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

PROGRAM CONTROL

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Call	CALL	dst	IR DA X	10 12 13	Call Subroutine Autodecrement SP @ SP←PC PC←dst
	CALR	dst	RA	10	Call Relative Autodecrement SP @SP←PC PC←PC + dst (range −4094 to +4096)
	SC	src	IM	33	System Call Autodecrement SP @ SP←old PS Push instruction PS←System Call PS
Jump	DJNZ DBJNZ	R, dst	RA	11	Decrement and Jump if Non-Zero R←R−1 IF R = 0: PC←PC + dst (range −254 to 0)
	IRET*	–	–	13	Interrupt Return PS←@SP Autoincrement SP
	JP	cc, dst	IR DA X	7 – 7 8	Jump Conditional If cc is true: PC←dst
	JR	cc, dst	RA	6	Jump Conditional Relative If cc is true: PC←PC + dst (range −256 to +254)
	RET	cc	–	10	Return Conditional If cc is true: PC←@SP Autodecrement SP
Return	IRET*	–	–	13	Interrupt Return PS←@SP Autoincrement SP

*Privileged instructions. Executed in system mode only.

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage at any Pin
Relative to V_{SS} ..... −0.3 to +7.0V
Power Dissipation 2.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ± 5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

4, 6MHz Devices
DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator	V_{CC}−0.4	V_{CC}+0.3	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator	−0.3	0.45	Volts
V_{IH}	Input High Voltage		2.0	V_{CC}+0.3	Volts
V_{IH} $\overline{NMI}$, Reset	Input High Voltage		2.4	V_{CC}+0.3	Volts
V_{IL}	Input Low Voltage		−0.3	0.8	Volts
V_{OH}	Output High Voltage	I_{OH} = −250μA	2.4		Volts
V_{OL}	Output Low Voltage	I_{OL} = +2.0mA		0.4	Volts
I_{IL}	Input Leakage Except $\overline{SEGT}$ Pin	0.4 ≤ V_{IN} ≤ +2.4V		±10	μA
I_{IL} on $\overline{SEGT}$	Input Leakage on $\overline{SEGT}$ Pin		−100	100	μA
I_{OL}	Output Leakage	0.4 ≤ V_{OUT} ≤ +2.4V		±10	μA
ICC	V_{CC} Supply Current	Commercial		300	mA

8MHz Devices
DC CHARACTERISTICS over operating range unless otherwise specified

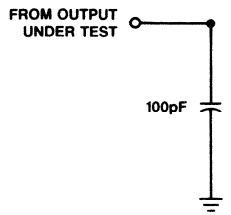
Parameters	Description	Test Conditions	Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator	V_{CC}−0.4	V_{CC}+0.3	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator	−0.3	0.45	Volts
V_{IH}	Input High Voltage		2.0	V_{CC}+0.3	Volts
V_{IH} $\overline{NMI}$, Reset	Input High Voltage		2.4	V_{CC}+0.3	Volts
V_{IL}	Input Low Voltage		−0.3	0.8	Volts
V_{OH}	Output High Voltage	I_{OH} = −250μA	2.4		Volts
V_{OL}	Output Low Voltage	I_{OL} = +2.0mA		0.4	Volts
I_{IL}	Input Leakage Except $\overline{SEGT}$ Pin	0.4 ≤ V_{IN} ≤ +2.4V		±10	μA
I_{IL} on $\overline{SEGT}$	I_{IL} on $\overline{SEGT}$ Pin	0.4 ≤ V_{IN} ≤ 2.4 V	−100	100	μA
I_{OL}	Output Leakage	0.4 ≤ V_{OUT} ≤ +2.4V		±10	μA
I_{CC}	V_{CC} Supply Current	Commercial		400	mA

See section 6 for Thermal Characteristics Information.

Standard Test Conditions

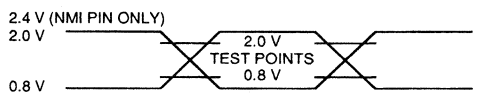
The characteristics below apply for the following test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

SWITCHING TEST CIRCUIT



TC001702

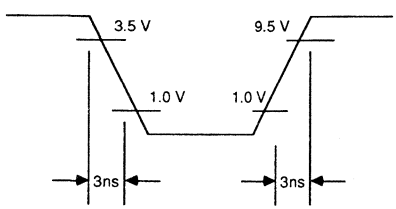
SWITCHING TEST WAVEFORMS
Input/Output



LS003010

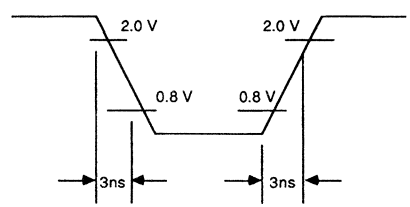
AC testing outputs are driven at 2.0 V for a logical 1 and 0.5 V for a logical 0. The clock is driven at V_{CC} −0.4 V and 0.45 V. Timing measurements are made at 2.0 V for a logical 1 and 0.5 V for a logical 0.

AC Clock Input



WF024270

AC Input (Except Clock)



WF024280

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified

Number	Parameters	Description	4MHz Devices		6MHz Devices		Units
			Min	Max	Min	Max	
1	TcC	Clock Cycle Time	250	2000	165	2000	ns
2	TwCh	Clock Width (HIGH)	105	1895	70	1930	ns
3	TwCl	Clock Width (LOW)	105	1895	70	1930	ns
4	TfC	Clock Fall Time		20		10	ns
5	TrC	Clock Rise Time		20		15	ns
6†	TdC(SNv)	Clock ↑ to Segment Number Valid (50pF Load)		130		110	ns
7†	TdC(SNn)	Clock ↑ to Segment Number Not Valid	20		10		ns
8	TdC(Bz)	Clock ↑ to Bus Float		65		55	ns
9	TdC(A)	Clock ↑ to Address Valid		100		75	ns
10	TdC(Az)	Clock ↑ to Address Float		65		55	ns
11	TdA(DR)*	Address Valid to Read Data Required Valid		475		305	ns
12	TsDI(C)	Data In to Clock ↓ Set-up Time	30		20		ns
13	TdDS(A)*	$\overline{DS}$ ↑ to Address Active	80		45		ns
14	TdC(DW)	Clock ↑ to Write Data Valid		100		75	ns
15	ThDI(DS)	Data In to $\overline{DS}$ ↑ Hold Time	0		0		ns
16	TdDO(DS)*	Data Out Valid to $\overline{DS}$ ↑ Delay	295		195		ns
17	TdA(MR)*	Address Valid to $\overline{MREQ}$ ↓ Delay	55		35		ns
18	TdC(MR)	Clock ↓ to $\overline{MREQ}$ ↓ Delay		80		70	ns
19	TwMRh*	$\overline{MREQ}$ Width (HIGH)	210		135		ns
20	TdMR(A)*	$\overline{MREQ}$ ↓ to Address Not Active	70		35		ns
21	TdDO(DSW)*	Data Out Valid to $\overline{DS}$ ↓ (Write) Delay	55		35		ns
22	TdMR(DR)*	$\overline{MREQ}$ ↓ to Read Data Required Valid		370		230	ns
23	TdC(MR)	Clock ↓ to $\overline{MREQ}$ ↑ Delay		80		60	ns
24	TdC(ASf)	Clock ↑ to $\overline{AS}$ ↓ Delay		80		60	ns
25	TdA(AS)*	Address Valid to $\overline{AS}$ ↑ Delay	55		35		ns
26	TdC(ASr)	Clock ↓ to $\overline{AS}$ ↑ Delay		90		80	ns
27	TdAS(DR)*	$\overline{AS}$ ↑ to Read Data Required Valid		360		220	ns
28	TdDS(AS)*	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	70		35		ns
29	TwAS*	$\overline{AS}$ Width (LOW)	85		55		ns
30	TdAS(A)*	$\overline{AS}$ ↑ to Address Not Active Delay	70		45		ns
31	TdAz(DSR)	Address Float to $\overline{DS}$ (Read) ↓ Delay	0		0		ns
32	TdAS(DSR)*	$\overline{AS}$ ↑ to $\overline{DS}$ (Read) ↓ Delay	80		55		ns
33	TdDSR(DR)*	$\overline{DS}$ (Read) ↓ to Read Data Required Valid		205		130	ns
34	TdC(DSr)	Clock ↓ to $\overline{DS}$ ↑ Delay		70		65	ns
35	TdDS(DW)*	$\overline{DS}$ ↑ to Write Data and STATUS Not Valid	75		45		ns
36	TdA(DSR)*	Address Valid to $\overline{DS}$ (Read) ↓ Delay	180		110		ns
37	TdC(DSR)	Clock ↑ to $\overline{DS}$ (Read) ↓ Delay		120		85	ns
38	TwDSR*	$\overline{DS}$ (Read) Width (LOW)	275		185		ns
39	TdC(DSW)	Clock ↓ to $\overline{DS}$ (Write) ↓ Delay		95		80	ns
40	TwDSW*	$\overline{DS}$ (Write) Width (LOW)	185		110		ns
41	TdDSI(DR)*	$\overline{DS}$ (Input) ↓ to Read Data Required Valid		330		210	ns

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Number	Parameters	Description	4MHz Devices		6MHz Devices		Units
			Min	Max	Min	Max	
42	TdC(DSI)	Clock ↓ to $\overline{DS}$ (I/O) ↓ Delay		120		90	ns
43	TwDS*	$\overline{DS}$ (I/O) Width (LOW)	410		255		ns
44	TdAS(DSA)*	$\overline{AS}$ ↑ to $\overline{DS}$ (Acknowledge) ↓ Delay	1065		690		ns
45	TdC(DSA)	Clock ↑ to $\overline{DS}$ (Acknowledge) ↓ Delay		120		85	ns
46	TdDSA(DR)*	$\overline{DS}$ (Acknowledge) ↓ to Read Data Required Delay		455		295	ns
47	TdC(S)	Clock ↑ to Status Valid Delay		110		85	ns
48	TdS(AS)*	Status Valid to AS ↑ Delay	50		30		ns
49	TsR(C)	$\overline{RESET}$ to Clock ↑ Set-up Time	180		70		ns
50	ThR(C)	$\overline{RESET}$ to Clock ↑ Hold Time	0		0		ns
51	TwNMI	$\overline{NMI}$ Width (LOW)	100		70		ns
52	TsNMI(C)	$\overline{NMI}$ to Clock ↑ Set-up Time	140		70		ns
53	TsVI(C)	$\overline{VI}$, $\overline{NVI}$ to Clock ↑ Set-up Time	110		50		ns
54	ThVI(C)	$\overline{VI}$, $\overline{NVI}$ to Clock ↑ Hold Time	20		20		ns
55†	TsSGT(C)	$\overline{SEGT}$ to Clock ↑ Set-up Time	70		55		ns
56†	ThSGT(C)	$\overline{SEGT}$ to Clock ↑ Hold Time	0		0		ns
57	TsMI(C)	$\overline{MI}$ to Clock ↑ Set-up Time	180		140		ns
58	ThMI(C)	$\overline{MI}$ to Clock ↑ Hold Time	0		0		ns
59	TdC(MO)	Clock ↑ to $\overline{MO}$ Delay		120		85	ns
60	TsSTP(C)	$\overline{STOP}$ to Clock ↓ Set-up Time	140		100		ns
61	ThSTP(C)	$\overline{STOP}$ to Clock ↓ Hold Time	0		0		ns
62	TsWT(C)	$\overline{WAIT}$ to Clock ↓ Set-up Time	50		30		ns
63	ThWT(C)	$\overline{WAIT}$ to Clock ↓ Hold Time	10		10		ns
64	TsBRQ(C)	$\overline{BUSRQ}$ to Clock ↑ Set-up Time	90		80		ns
65	ThBRQ(C)	$\overline{BUSRQ}$ to Clock ↑ Hold Time	10		10		ns
66	TdC(BAKr)	Clock ↑ to $\overline{BUSAK}$ ↑ Delay		100		75	ns
67	TdC(BAKf)	Clock ↑ to $\overline{BUSAK}$ ↓ Delay		100		75	ns
68	TwA	Address Valid Width	150		95		ns
69	TdDS(S)	$\overline{DS}$ ↑ to STATUS Not Valid	80		55		ns

*Clock-cycle-time-dependent characteristics. These numbers are computed assuming the clock characteristics are at the limits given in parameters 1 through 5. For other clock frequencies, these parameters can be derived from other specs and the clock characteristics. See tables on following pages.

†Z8001 and Z8001A only.

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified (Cont'd.)

Number	Parameters	Description	8MHz Devices		Units
			Min	Max	
1	TcC	Clock Cycle Time	125	2000	ns
2	TwCh	Clock Width (HIGH)	55	2000	ns
3	TwCl	Clock Width (LOW)	55	2000	ns
4	TfC	Clock Fall Time		10	ns
5	TrC	Clock Rise Time		10	ns
6†	TdC(SNv)	Clock ↑ to Segment Number Valid (50pF Load)		100	ns
7†	TdC(SNn)	Clock ↑ to Segment Number Not Valid	10		ns
8	TdC(Bz)	Clock ↑ to Bus Float		50	ns
9	TdC(A)	Clock ↑ to Address Valid		65	ns
10	TdC(Az)	Clock ↑ to Address Float		45	ns
11	TdA(DR)*	Address Valid to Read Data Required Valid		225	ns
12	TsDI(C)	Data In to Clock ↓ Set-up Time	15		ns
13	TdDS(A)*	$\overline{DS}$ ↑ to Address Active	40		ns
14	TdC(DW)	Clock ↑ to Write Data Valid		65	ns
15	ThDI(DS)	Data In to $\overline{DS}$ ↑ Hold Time	0		ns
16	TdDO(DS)*	Data Out Valid to $\overline{DS}$ ↑ Delay	150		ns
17	TdA(MR)*	Address Valid to $\overline{MREQ}$ ↓ Delay	30		ns
18	TdC(MR)	Clock ↓ to $\overline{MREQ}$ ↓ Delay		55	ns
19	TwMRh*	$\overline{MREQ}$ Width (HIGH)	105		ns
20	TdMR(A)*	$\overline{MREQ}$ ↓ to Address Not Active	35		ns
21	TdDO(DSW)*	Data Out Valid to $\overline{DS}$ ↓ (Write) Delay	30		ns
22	TdMR(DR)*	$\overline{MREQ}$ ↓ to Read Data Required Valid		175	ns
23	TdC(MR)	Clock ↓ to $\overline{MREQ}$ ↑ Delay		55	ns
24	TdC(ASf)	Clock ↑ to $\overline{AS}$ ↓ Delay		55	ns
25	TdA(AS)*	Address Valid to $\overline{AS}$ ↑ Delay	30		ns
26	TdC(ASr)	Clock ↓ to $\overline{AS}$ ↑ Delay		65	ns
27	TdAS(DR)*	$\overline{AS}$ ↑ to Read Data Required Valid		170	ns
28	TdDS(AS)*	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	35		ns
29	TwAS*	$\overline{AS}$ Width (LOW)	45		ns
30	TdAS(A)*	$\overline{AS}$ ↑ to Address Not Active Delay	30		ns
31	TdAz(DSR)	Address Float to $\overline{DS}$ (Read) ↓ Delay	0		ns
32	TdAS(DSR)*	$\overline{AS}$ ↑ to $\overline{DS}$ (Read) ↓ Delay	30		ns
33	TdDSR(DR)*	$\overline{DS}$ (Read) ↓ to Read Data Required Valid		115	ns
34	TdC(DSr)	Clock ↓ to $\overline{DS}$ ↑ Delay		65	ns
35	TdDS(DW)*	$\overline{DS}$ ↑ to Write Data and STATUS Not Valid	40		ns
36	TdA(DSR)*	Address Valid to $\overline{DS}$ (Read) ↓ Delay	85		ns
37	TdC(DSR)	Clock ↑ to $\overline{DS}$ (Read) ↓ Delay		70	ns
38	TwDSR*	$\overline{DS}$ (Read) Width (LOW)	140		ns
39	TdC(DSW)	Clock ↓ to $\overline{DS}$ (Write) ↓ Delay		65	ns
40	TwDSW*	$\overline{DS}$ (Write) Width (LOW)	85		ns
41	TdDSI(DR)*	$\overline{DS}$ (Input) ↓ to Read Data Required Valid		135	ns
42	TdC(DSf)	Clock ↓ to $\overline{DS}$ (I/O) ↓ Delay		85	ns
43	TwDS*	$\overline{DS}$ (I/O) Width (LOW)	200		ns
44	TdAS(DSA)*	$\overline{AS}$ ↑ to $\overline{DS}$ (Acknowledge) ↓ Delay	520		ns
45	TdC(DSA)	Clock ↑ to $\overline{DS}$ (Acknowledge) ↓ Delay		65	ns

†Z8001A-8 only.

SWITCHING CHARACTERISTICS (Cont'd.)

Number	Parameters	Description	8MHz Devices		Units
			Min	Max	
46	TdDSA(DR)*	$\overline{DS}$ (Acknowledge) ↓ to Read Data Required Delay		235	ns
47	TdC(S)	Clock ↑ to Status Valid Delay		75	ns
48	TdS(AS)*	Status Valid to $\overline{AS}$ ↑ Delay	25		ns
49	TsR(C)	$\overline{RESET}$ to Clock ↑ Set-up Time	70		ns
50	ThR(C)	$\overline{RESET}$ to Clock ↑ Hold Time	0		ns
51	TwNMI	$\overline{NMI}$ Width (LOW)	50		ns
52	TsNMI(C)	$\overline{NMI}$ to Clock ↑ Set-up Time	70		ns
53	TsVI(C)	$\overline{VI}$,$\overline{NV1}$, to Clock ↑ Set-up Time	50		ns
54	ThVI(C)	$\overline{VI}$,$\overline{NV1}$ to Clock ↑ Hold Time	20		ns
55†	TsSGT(C)	$\overline{SEGT}$ to Clock ↑ Set-up Time	45		ns
56†	ThSGT(C)	$\overline{SEGT}$ to Clock ↑ Hold Time	0		ns
57	TsMI(C)	$\overline{MI}$ to Clock ↑ Set-up Time	90		ns
58	ThMI(C)	$\overline{MI}$ to Clock ↑ Hold Time	0		ns
59	TdC(MO)	Clock ↑ to $\overline{MO}$ Delay		65	ns
60	TsSTP(C)	$\overline{STOP}$ to Clock ↓ Set-up Time	75		ns
61	ThSTP(C)	$\overline{STOP}$ to Clock ↓ Hold Time	0		ns
62	TsWT(C)	$\overline{WAIT}$ to Clock ↓ Set-up Time	25		ns
63	ThWT(C)	$\overline{WAIT}$ to Clock ↓ Hold Time	10		ns
64	TsBRQ(C)	$\overline{BUSRQ}$ to Clock ↑ Set-up Time	60		ns
65	ThBRQ(C)	$\overline{BUSRQ}$ to Clock ↑ Hold Time	10		ns
66	TdC(BAKr)	Clock ↑ to $\overline{BUSAK}$ ↑ Delay		60	ns
67	TdC(BAKf)	Clock ↑ to $\overline{BUSAK}$ ↓ Delay		60	ns
68	TwA*	Address Valid Width	90		ns
69	TdDS(S)*	$\overline{DS}$ ↓ to STATUS Not Valid	45		ns

*Clock-cycle-time-dependent characteristics. These numbers are computed assuming the clock characteristics are at the limits given in parameters 1 through 5. For other clock frequencies, three parameters can be derived from other specs and the clock characteristics. See following table.

†Z8001A-8 only.

CLOCK-CYCLE-TIME-DEPENDENT CHARACTERISTICS

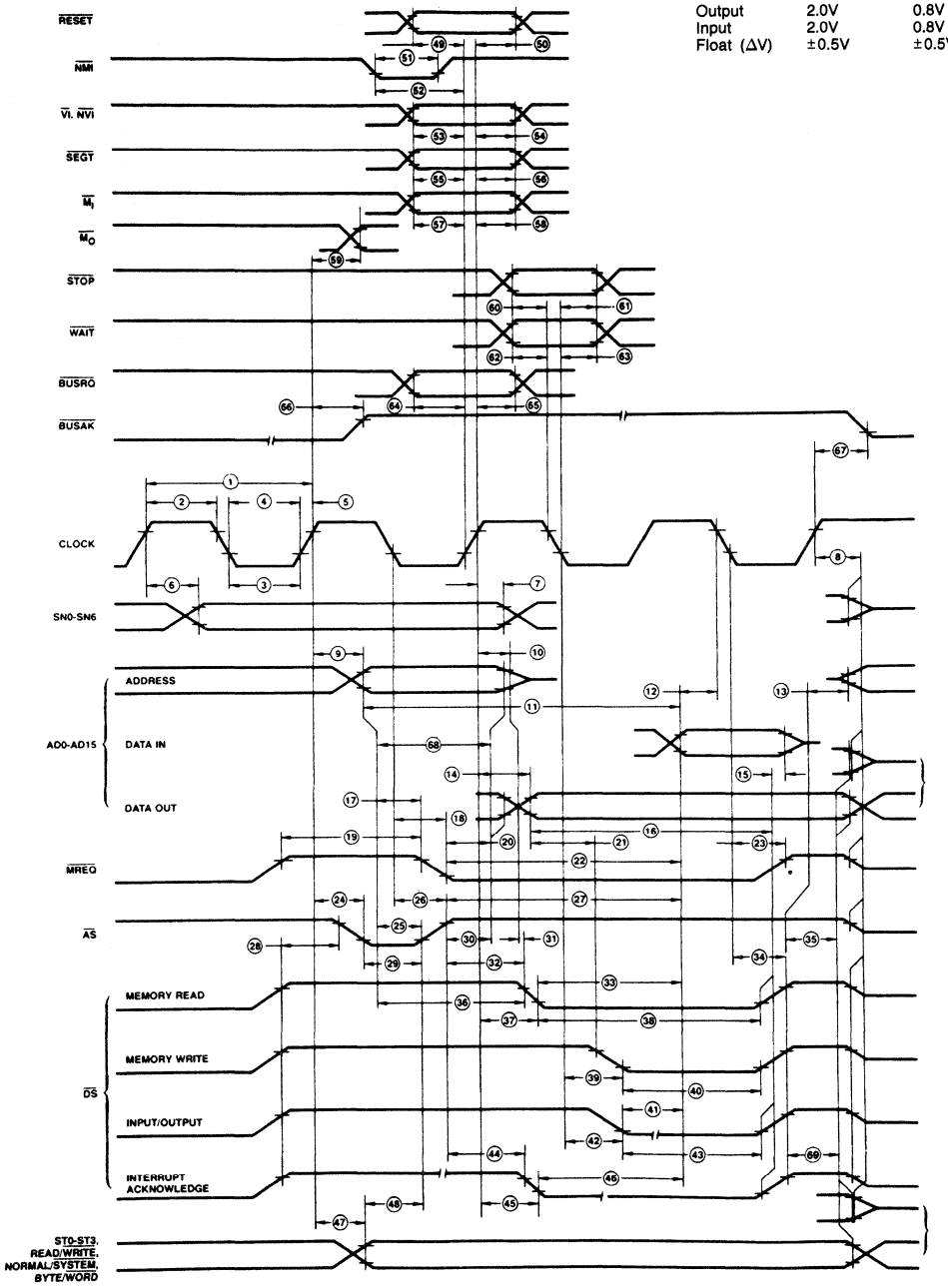
The parameters listed below are also shown in the switching specification. However, they are dependent on the actual values of the clock periods. The equations below define that dependence, so the correct limit for these parameters may be determined for any system, regardless of the actual clock characteristics.

No.	Parameters	4MHz Devices	6MHz Devices	8MHz
11	TdA(DR)	2TcC + TwCH – 125ns	2TcC + TwCh – 95ns	2TcC + TwCh – 80ns
13	TdDS(C)	TwCl – 25ns	TWCl – 30ns	TwCl – 15ns
16	TdDO(DS)	TcC + TwCh – 60ns	TcC+ TwCh – 40ns	TcC + TwCh – 30ns
17	TdA(MR)	TwCh – 50ns	TwCh – 35ns	TwCh – 25ns
19	TwMRh	TcC – 40ns	TcC – 30ns	TcC – 20ns
20	TdMR(A)	TwCl – 35ns	TwCl – 35ns	TwCl – 20ns
21	TdDO(DSW)	TwCh – 50ns	TwCh – 35ns	TwCh – 25ns
22	TdMR(DR)	2TcC – 125ns	2TcC – 105ns	2TcC – 75ns
25	TdA(AS)	TwCh – 50ns	TwCh – 35ns	TwCh – 25ns
27	TdAS(Dl)	2TcC – 140ns	2TcC – 115ns	2TcC – 80ns
28	TdDS(AS)	TwCl – 35ns	TwCl – 35ns	TwCl – 20ns
29	TwAS	TwCh – 20ns	TwCh – 15ns	TwCh – 10ns
30	TdAS(A)	TwCl – 35ns	TwCl – 40ns	TwCl – 25ns
32	TdAS(DSR)	TwCl – 25ns	TwCl – 35ns	TwCl – 25ns
33	TdDSR(DR)	TcC + TwCh – 150ns	TcC+ TwCh – 105ns	TcC + TwCh – 65ns
35	TdDS(DW)	TwCl – 30ns	TwCl – 25ns	TwCl – 15ns
36	TdA(DSR)	TcC – 70ns	TcC – 55ns	TcC – 40ns
38	TwDSR	TcC + TwCh – 80ns	TcC+ TwCh – 50ns	TcC + TwCh – 40ns
40	TwDSW	TcC – 65ns	TcC – 55ns	TcC – 40ns
41	TdDSl(DR)	2TcC – 170ns	2TcC – 130ns	2TcC – 115ns
43	TwDS	2TcC – 90ns	2TcC – 75ns	2TcC – 50ns
44	TdAS(DSA)	4TcC + TwCh – 40ns	4TcC – TwCl – 40ns	4TcC + TwCl – 35ns
46	TdDSA(DR)	2TcC + TwCh – 150ns	2TcC+ TwCh – 105ns	2TcC + TwCh – 70ns
48	TdS(AS)	TwCh – 55ns	TwCh – 40ns	TwCh – 30ns
68	TwA	TcC – 90ns	TcC – 70ns	Tcc – 35ns
69	TdDS(S)	TwCl – 25ns	TwCl – 15ns	TwCl – 10ns

Z8001 TIMING DIAGRAM

Timing Measurements are made at the following voltages.

	HIGH	LOW
Clock	4.0V	0.8V
Output	2.0V	0.8V
Input	2.0V	0.8V
Float (ΔV)	±0.5V	±0.5V



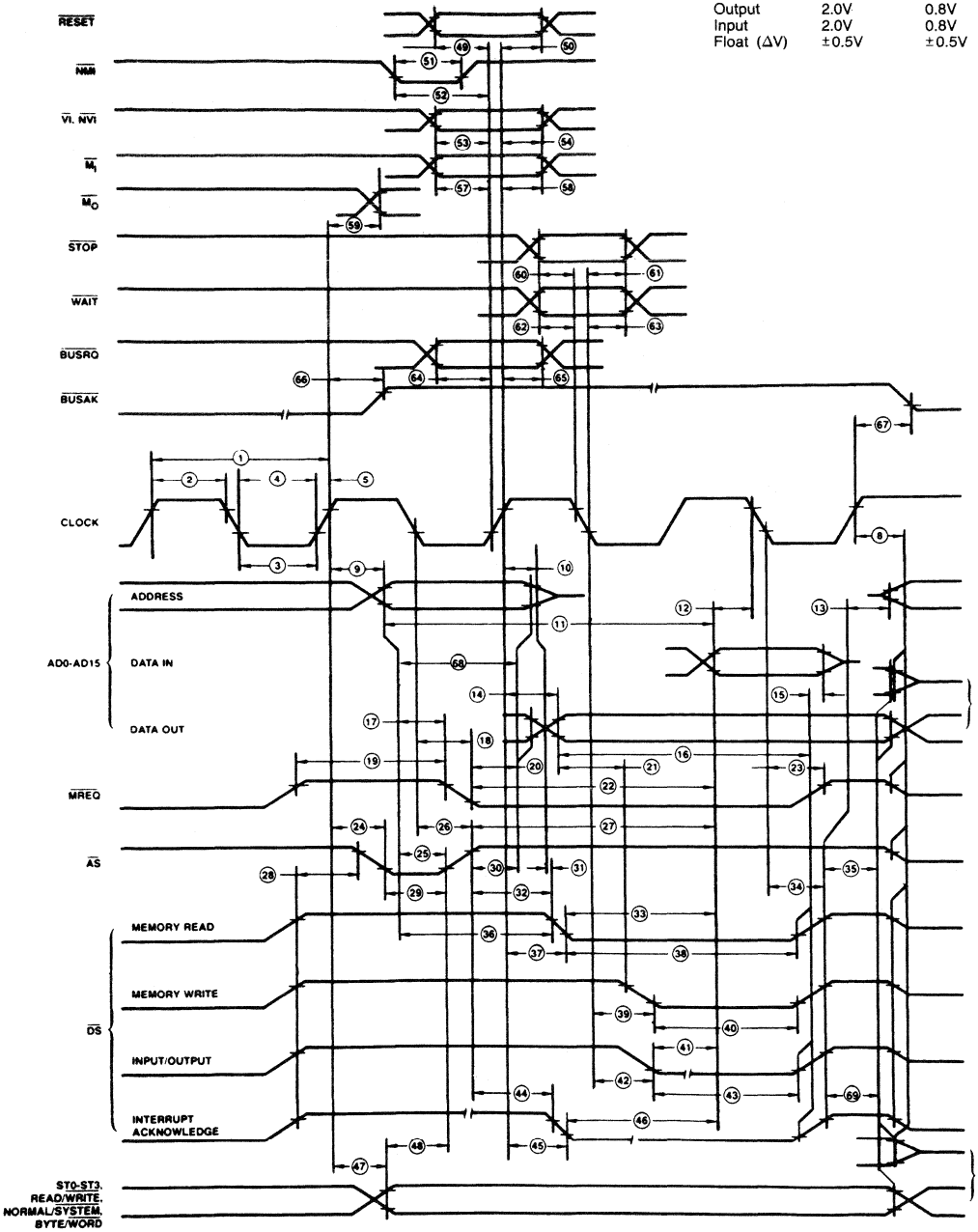
WF005290

This composite timing diagram does not show actual timing sequences. Refer to this diagram only for the detailed timing relationships of individual edges. Use the preceding illustrations as an explanation of the various timing sequences.

Z8002 TIMING DIAGRAM

Timing Measurements are made at the following voltages.

	HIGH	LOW
Clock	4.0V	0.8V
Output	2.0V	0.8V
Input	2.0V	0.8V
Float (ΔV)	±0.5V	±0.5V



WF005300

This composite timing diagram does not show actual timing sequences. Refer to this diagram only for the detailed timing relationships of individual edges. Use the preceding illustrations as an explanation of the various timing sequences.

5

Z8030/Z8530(H)

Serial Communications Controller

DISTINCTIVE CHARACTERISTICS

- **Two 0 to 2 Mbps full duplex serial channels**
  Each channel has independent oscillator, baud-rate generator, and PLL for clock recovery, dramatically reducing external components.
- **Programmable protocols**
  NRZ, NRZI, and FM data encoding supported under program control.
- **Programmable Asynchronous Modes**
  5- to 8-bit characters with programmable stop bits, clock, break detect, and error conditions.
- **Z8000* compatible**
  The Z8030 interfaces directly with the Z8000 CPU bus and to the Z8000 interrupt structure.
- **Programmable Synchronous Modes**
  SDLC and HDLC and SDLC loop supported with frame control, zero insertion and deletion, abort, and residue handling. CRC-16 and CCITT generators and checkers.
- **Compatible with non-multiplexed bus**
  The Z8530(H) interfaces easily to most other CPUs.
- **Enhanced Version**
  The Z8530(H) is an enhanced version whose features include 8-MHz operation and an improved Valid Access Recovery Time (t_{RC}) specification.

GENERAL DESCRIPTION

The SCC Serial Communications Controller is a dual-channel, multi-protocol data communications peripheral designed for use with 8- and 16-bit microprocessors. The SCC functions as a serial-to-parallel, parallel-to-serial converter/controller. The SCC can be software-configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions, including on-chip baud rate generators, digital phase-locked loops, and crystal oscillators, which dramatically reduce the need for external logic.

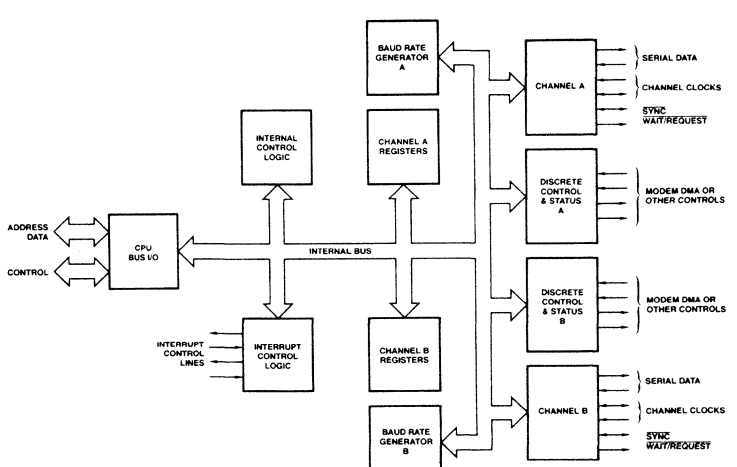
The SCC handles asynchronous formats, synchronous byte-oriented protocols, such as IBM Bisync, and synchronous bit-oriented protocols, such as HDLC and IBM SDLC.

This versatile device supports virtually any serial data transfer application (cassette, diskette, tape drivers, etc.).

The device can generate and check CRC codes in any synchronous mode and can be programmed to check data integrity in various modes. The SCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

The SCC is offered in two versions. The Z8030 is directly compatible with the Z8000 and 8086 CPUs. The Z8530(H) is designed for non-multiplexed buses and is easily interfaced with most other CPUs, such as 8080, Z80, 6800, 68000, and MULTIBUS.†

BLOCK DIAGRAM



BD003520

Figure 1.

* Z8000, Z8030 and Z8530 are trademarks of Zilog, Inc.
†MULTIBUS is a trademark of Intel, Corp.

Publication # 00970 | Rev. E | Amendment /0
Issue Date: August 1987

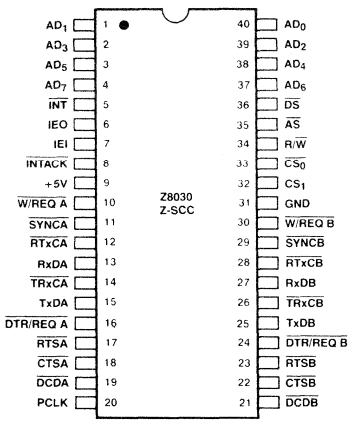
RELATED AMD PRODUCTS

Part No.	Description
Am79C12	Full Duplex 1200 bps Modem
Am7960	Coded Data Transceiver
80186	Highly Integrated 16-Bit Microprocessor
80286	High-Performance 16-Bit Microprocessor
8080A	8-Bit Microprocessor
Am9517A	DMA Controller

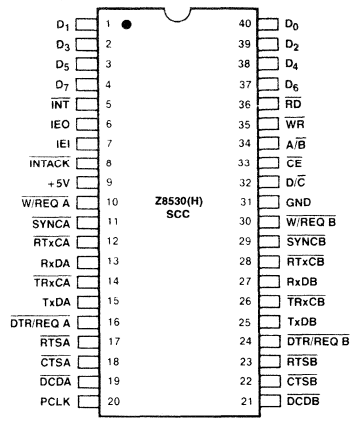
CONNECTION DIAGRAMS

Top View

DIPs



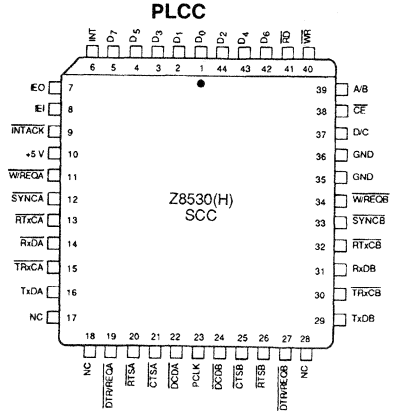
CD005350



CD005361

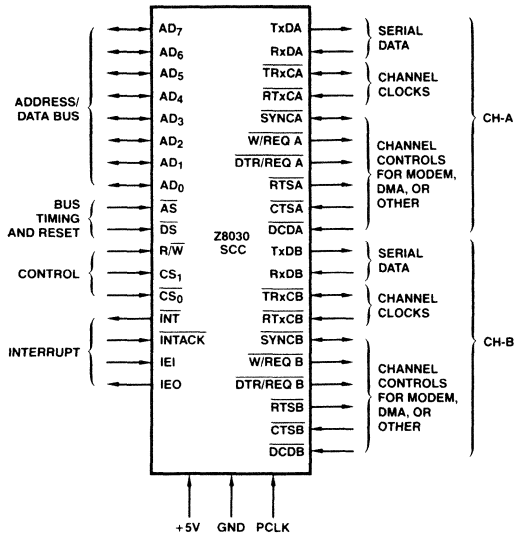
Note: Pin 1 is marked for orientation.

PLCC

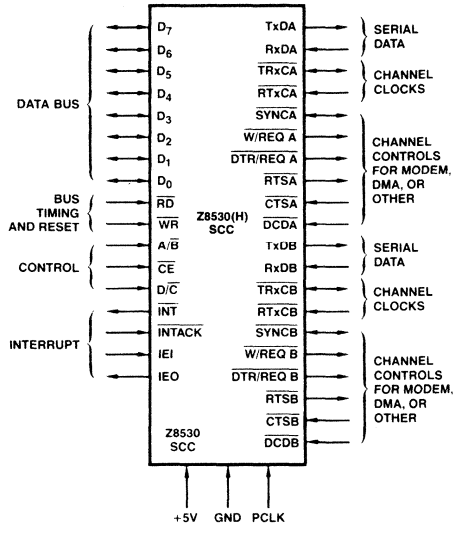


CD010931

LOGIC SYMBOL



LS001300



LS001312

Summary of Difference Between Z8530 and Z8530H

No.	Parameter Symbol		Z8530		Z8530H	
			Min.	Max	Min.	Max.
3	TsRxC (PC)	4 MHz	80	TWPCL	80	TWPCL
		6 MHz	70	TWPCL	70	TWPCL
27	Tda (DR)	4 MHz		400		300
		6 MHz		350		280
49	Trc (Note 1)	4 MHz	6 TcPC + 200 ns		4 TcPC	
		6 MHz	6 TcPC + 130 ns		4 TcPC	

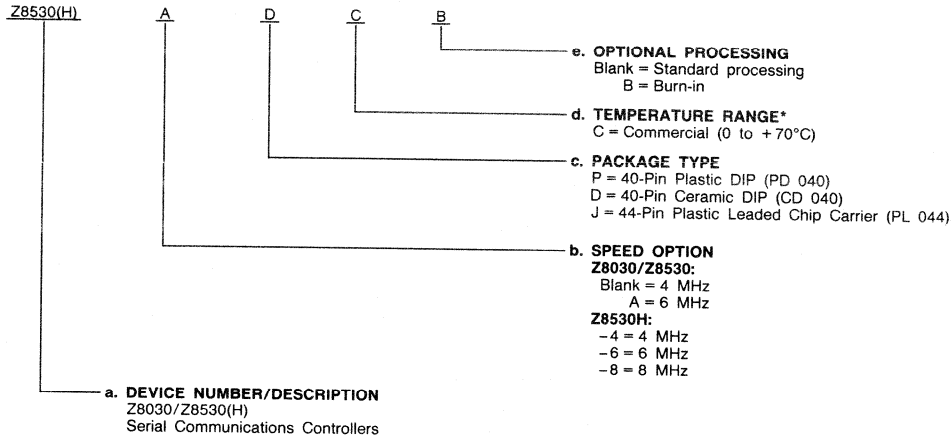
Notes: 1. Z8530 is measured from Rising Edge to Falling Edge:
Z8530H is measured from Falling Edge to Falling Edge.

*The Z8530H is available in an 8-MHz version; the Z8530 is not.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
Z8030	PC, DC, DCB
Z8030A	
Z8530	PC, DC, DCB, JC
Z8530A	
Z8530H-4	
Z8530H-6	
Z8530H-8	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

Z8030 PIN DESCRIPTION

Pin No.	Name	I/O	Description
9	V_{CC}		+5V Power Supply.
31	GND		Ground.
40, 1, 39, 2, 38, 3, 37, 4	AD_0-AD_7	I/O	Address/Data Bus (bidirectional, active High, 3-state). These multiplexed lines carry register addresses to the SCC as well as data or control information to and from the SCC.
35	$\overline{AS}$	I	Address Strobe (active Low). Addresses on AD_0-AD_7 are latched by the rising edge of this signal.
33	$\overline{CS_0}$	I	Chip Select 0 (active Low). This signal is latched concurrently with the addresses on AD_0-AD_7 and must be active for the intended bus transaction to occur.
32	CS_1	I	Chip Select 1 (active High). This second select signal must also be active before the intended bus transaction can occur. CS_1 must remain active throughout the transaction.
18, 22	$\overline{CTSA}$, $\overline{CTSB}$	I	Clear to Send (active Low). If these pins are programmed as Auto Enables, a LOW on these inputs enables their respective transmitter. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.
19, 21	$\overline{DCDA}$, $\overline{DCDB}$	I	Data Carrier Detect (active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise, they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.
36	$\overline{DS}$	I	Data Strobe (active Low). This signal provides timing for the transfer of data into and out of the SCC. If $\overline{AS}$ and $\overline{DS}$ coincide, this is interpreted as a reset.
16, 24	$\overline{DTR/REQA}$, $\overline{DTR/REQB}$	O	Data Terminal Ready/Request (active Low). These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request Lines for a DMA controller.
7	IEI	I	Interrupt Enable In (active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.
6	IEO	O	Interrupt Enable Out (active High). IEO is HIGH only if IEI is HIGH and the CPU is not servicing a SCC interrupt or the SCC is not requesting an interrupt (interrupt acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.
5	$\overline{INT}$	O	Interrupt Request (open-drain, active Low). This signal is activated when the SCC requests an interrupt.
8	$\overline{INTACK}$	I	Interrupt Acknowledge (active Low). This signal indicates an active interrupt acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When $\overline{DS}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is High). $\overline{INTACK}$ is latched by the rising edge of $\overline{AS}$.
20	PCLK	I	Clock. This is the master SCC clock used to synchronize internal signals. PCLK is not required to have any phase relationship with the master system clock, although the frequency of this clock must be at least 90% of the CPU clock frequency for a Z8000. PCLK is a TTL level signal. Maximum transmit rate is 1/4 PCLK.
13, 27	RxDA, RxDB	I	Receive Data (active High). These input signals receive serial data at standard TTL levels.
12, 28	$\overline{RTxCA}$, $\overline{RTxCB}$	I	Receive/Transmit Clocks (active Low). These pins can be programmed in several different modes of operation. In each channel, RTxC may supply the receive clock, the transmit clock, the clock for the baud-rate generator, or the clock of the digital phase-locked loop. These pins can also be programmed for use with the respective SYNC pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.
17, 23	$\overline{RTSA}$, $\overline{RTSB}$	O	Request to Send (active Low). When the Request to Send RTS bit in Write Register 5 is set, the $\overline{RTS}$ signal goes LOW. When the RTS bit is reset in the asynchronous mode and Auto Enable is on, the signal goes HIGH after the transmitter is empty. In synchronous mode or in asynchronous mode with Auto Enable off, the $\overline{RTS}$ pins strictly follow the state of the RTS bit. Both pins can be used as general-purpose outputs.
34	R/$\overline{W}$	I	Read/Write. This signal specifies whether the operation to be performed is read or a write.
11, 29	$\overline{SYNCA}$, $\overline{SYNCB}$	I or O	Synchronization (active Low). These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to $\overline{CTS}$ and $\overline{DCD}$. In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 but have no other function. In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, $\overline{SYNC}$ must be driven LOW two receive clock cycles after the last bit in the synchronous character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of $\overline{SYNC}$. In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which synchronous characters are recognized. The synchronous condition is not latched, so these outputs are active each time a synchronous pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.
15, 25	TxDA, TxDB	O	Transmit Data (active High). These output signals transmit serial data at standard TTL levels.
14, 26	$\overline{TRxCA}$, $\overline{TRxCB}$	I or O	Transmit/Receive Clocks (active Low). These pins can be programmed in several different modes of operation. $\overline{TRxC}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the digital phase-locked loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.
10, 30	$\overline{W/REQA}$, $\overline{W/REQB}$	O	Wait/Request (open-drain when programmed for a Wait function, driven HIGH or LOW when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait.

Z8530(H) PIN DESCRIPTION

Pin No.*	Name	I/O	Description
9	V_{CC}		+5V Power Supply.
31	GND		Ground.
34	A/$\overline{B}$	I	Channel A/Channel B Select. This signal selects the channel in which the read or write operation occurs.
33	$\overline{CE}$	I	Chip Enable (active Low). This signal selects the SCC for a read or write operation.
18, 22	$\overline{CTSA}$, $\overline{CTSB}$	I	Clear To Send (active Low). If these pins are programmed as Auto Enables, a LOW on the inputs enables the respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.
32	D/$\overline{C}$	I	Data/Control Select. This signal defines the type of information transferred to or from the SCC. A HIGH means data is transferred; a LOW indicates a command.
19, 21	$\overline{DCDA}$, $\overline{DCDB}$	I	Data Carrier Detect (active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise, they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.
40, 1, 39, 2, 38, 3, 37, 4	D_0-D_7	I/O	Data Bus (3-state). These lines carry data and commands to and from the SCC.
16, 24	$\overline{DTR}$/$\overline{REQA}$, $\overline{DTR}$/$\overline{REQB}$	O	Data Terminal Ready/Request (active Low). These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request lines for a DMA controller.
7	IEI	I	Interrupt Enable In (active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.
6	IEO	O	Interrupt Enable Out (active High). IEO is HIGH only if IEI is HIGH and the CPU is not servicing an SCC interrupt or the SCC is not requesting an interrupt (Interrupt Acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.
5	$\overline{INT}$	O	Interrupt Request (open-drain, active Low). This signal is activated when the SCC requests an interrupt.
8	$\overline{INTACK}$	I	Interrupt Acknowledge (active Low). This signal indicates an active Interrupt Acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When $\overline{RD}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is HIGH). $\overline{INTACK}$ is latched by the rising edge of PCLK.
20	PCLK	I	Clock. This is the master SCC clock used to synchronize internal signals; PCLK is a TTL level signal.
36	$\overline{RD}$	I	Read (active Low). This signal indicates a read operation and, when the SCC is selected, enables the SCC's bus drivers. During the Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.
13, 27	RxDA, RxDB	I	Receive Data (active High). These input signals receive serial data at standard TTL levels.
12, 28	$\overline{RTxCA}$, $\overline{RTxCB}$	I	Receive/Transmit Clocks (active Low). These pins can be programmed in several different modes of operation. In each channel, $\overline{RTxC}$ may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed for use with the respective $\overline{SYNC}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.
17, 23	$\overline{RTSA}$, $\overline{RTSB}$	O	Request To Send (active Low). When the Request to Send (RTS) bit in Write Register 5 is set, the $\overline{RTS}$ signal goes LOW. When the RTS bit is reset in the asynchronous mode and Auto Enable is on, the signal goes HIGH after the transmitter is empty. In synchronous mode or in asynchronous mode with Auto Enable off, the $\overline{RTS}$ pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.
11, 29	$\overline{SYNCA}$, $\overline{SYNCB}$	I or O	Synchronization (active Low). These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to $\overline{CTS}$ and $\overline{DCD}$. In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 but have no other function. In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, $\overline{SYNC}$ must be driven LOW two receive clock cycles after the last bit in the synchronous character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of $\overline{SYNC}$. In the Internal Synchronizaton mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which synchronous characters are recognized. The synchronous condition is not latched, so these outputs are active each time a synchronization pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.
15, 25	TxDA, TxDB	O	Transmit Data (active High). These output signals transmit serial data at standard TTL levels.
14, 26	$\overline{TRxCA}$, $\overline{TRxCB}$	I or O	Transmit/Receive Clocks (active Low). These pins can be programmed in several different modes of operation. $\overline{TRxC}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the Digital Phase-Locked Loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.
35	$\overline{WR}$	I	Write (active Low). When the SCC is selected, this signal indicates a write operation. The coincidence of $\overline{RD}$ and $\overline{WR}$ is interpreted as a reset.
10, 30	$\overline{W}$/$\overline{REQA}$, $\overline{W}$/$\overline{REQB}$	O	Wait/Request (open-drain when programmed for a Wait function, driven HIGH or LOW when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait.

*Pin numbers correspond to DIPs only.

5

ARCHITECTURE

The SCC internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to the Z8000 CPU (Z8030) or to a non-multiplexed CPU bus (Z8530(H)). Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface with modems or other external devices (Figure 1).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modem control.

The register set for each channel includes ten control (write) registers, two synchronous character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel, a write-only Master Interrupt Control register and three read registers: one containing the vector with status information (Channel B only), one containing the vector without status (A only), and one containing the Interrupt Pending bits (A only).

The registers for each channel are designated as follows:

WR0 – WR15 – Write Registers 0 through 15.
RR0 – RR3, RR10, RR12, RR13, RR15 – Read Registers 0 through 3, 10, 12, 13, 15.

The following table lists the functions assigned to each read or write register. The SCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

Data Path

The transmit and receive data path illustrated in Figure 2 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high-speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode (the character length in asynchronous modes also determines the data path).

The transmitter has an 8-bit transmit data buffer register loaded from the internal data bus and a 20-bit transmit shift register that can be loaded either from the sync-character registers or from the transmit data register. Depending on the operational mode, outgoing data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD).

TABLE 1. READ AND WRITE REGISTER FUNCTIONS

READ REGISTER FUNCTIONS

RR0 Transmit/Receive buffer status and External status

RR1 Special Receive Condition status

RR2 Modified interrupt vector (Channel B only)
Unmodified interrupt vector (Channel A only)

RR3 Interrupt Pending bits (Channel A only)

RR8 Receive buffer

RR10 Miscellaneous status

RR12 Lower byte of baud rate generator time constant

RR13 Upper byte of baud rate generator time constant

RR15 External/Status interrupt information

WRITE REGISTER FUNCTIONS

WR0 CRC initialize, initialization commands for the various modes, shift right/shift left command

WR1 Transmit/Receive interrupt and data transfer mode definition

WR2 Interrupt vector (accessed through either channel)

WR3 Receive parameters and control

WR4 Transmit/Receive miscellaneous parameters and modes

WR5 Transmit parameters and controls

WR6 Sync characters or SDLC address field

WR7 Sync character or SDLC flag

WR8 Transmit buffer

WR9 Master interrupt control and reset (accessed through either channel)

WR10 Miscellaneous transmitter/receiver control bits

WR11 Clock mode control

WR12 Lower byte of baud rate generator time constant

WR13 Upper byte of baud rate generator time constant

WR14 Miscellaneous control bits

WR15 External/Status interrupt control

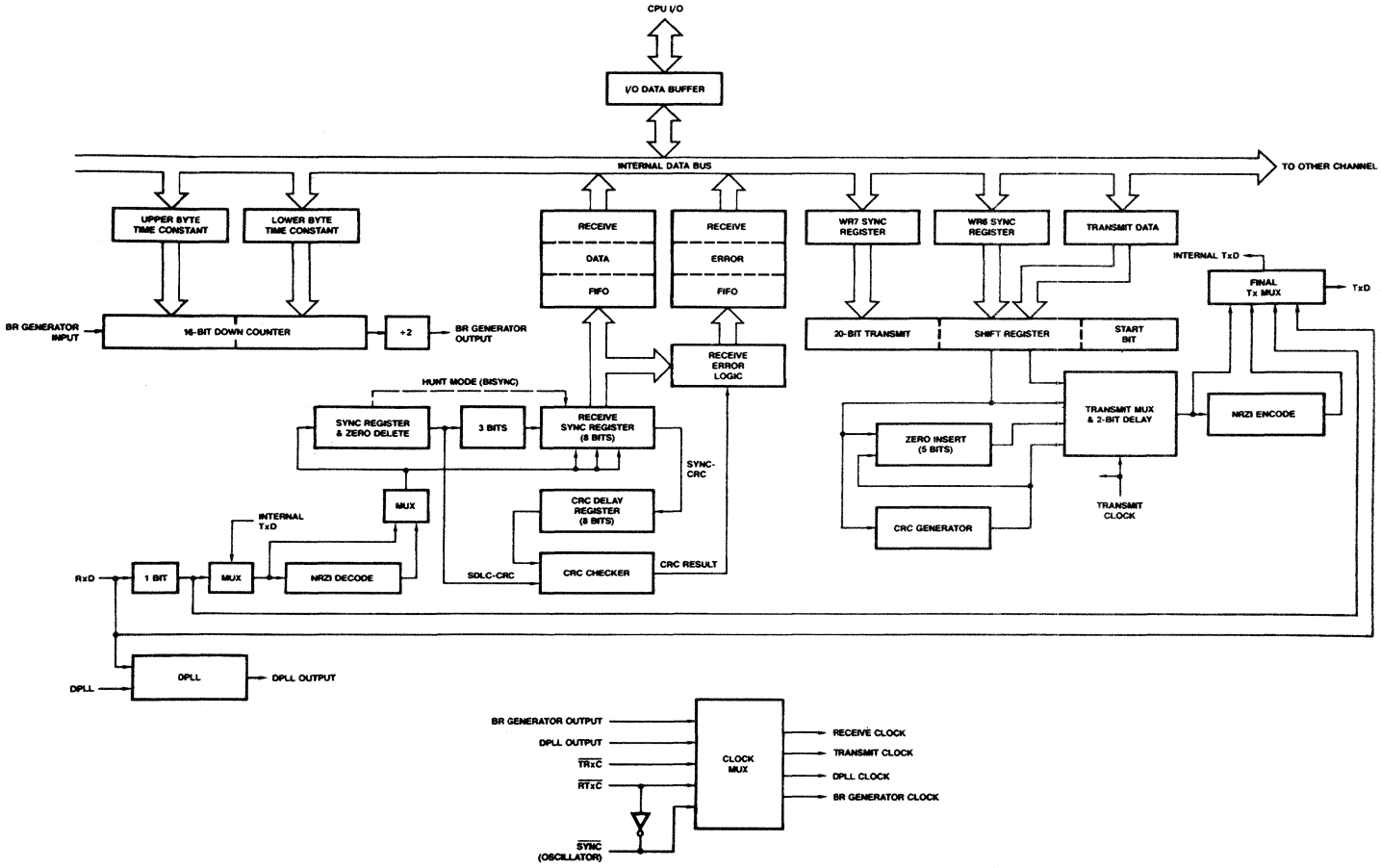


Figure 2. Data Path

DETAILED DESCRIPTION

The functional capabilities of the SCC can be described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, it interacts with the CPU and provides vectored interrupts and handshaking signals.

Data Communications Capabilities

The SCC provides two independent full-duplex channels programmable for use in any common asynchronous or synchronous data-communication protocol. Figure 3 and the following description briefly detail these protocols.

Asynchronous Modes

Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in the Z8530(H) Logic Symbol). If the LOW does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing or error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The SCC does not require symmetric transmit and receive clock signals – a feature allowing use of the wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs. In asynchronous modes, the SYNC pin may be programmed as an input used for functions, such as monitoring a ring indicator.

Synchronous Modes

The SCC supports both byte-oriented and bit-oriented synchronous communication. Synchronous byte-oriented protocols can be handled in several modes, allowing character synchronization with a 6-bit or 8-bit synchronous character (Monosync), any 12-bit synchronous pattern (Bisync), or with an external synchronous signal. Leading synchronous characters can be removed without interrupting the CPU.

5- or 7-bit synchronous characters are detected with 8- or 16-bit patterns in the SCC by overlapping the larger pattern across multiple incoming synchronous characters as shown in Figure 4.

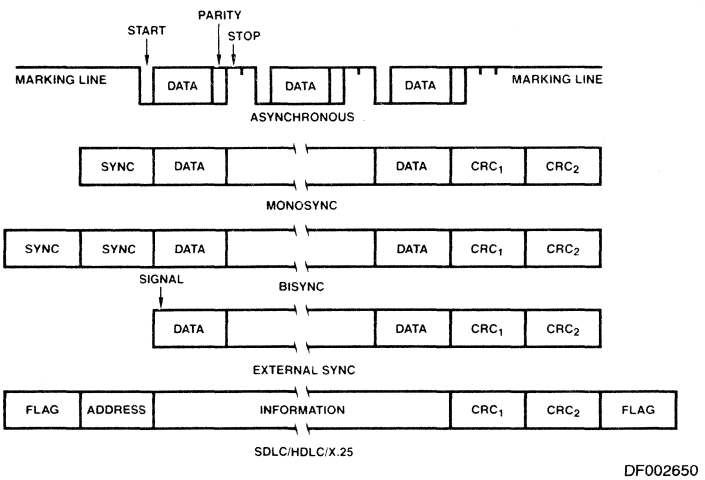


Figure 3. SCC Protocols

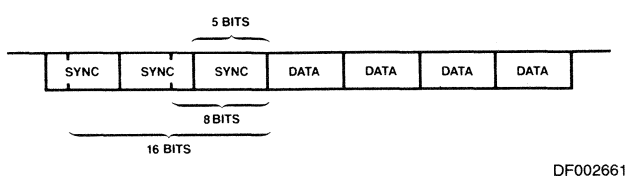


Figure 4. Detecting 5- or 7-Bit Synchronous Characters

CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols, such as IBM Bisync.

Both CRC-16 ($X^{16} + X^{15} + X^2 + 1$) and CCITT ($X^{16} + X^{12} + X^5 + 1$) error checking polynomials are supported. Either polynomial may be selected in all synchronous modes. Users may preset the CRC generator and checker to all 1s or all 0s. The SCC also provides a feature that automatically transmits CRC data when no other data is available for transmission. This allows for high-speed transmissions under DMA control with no need for CPU intervention at the end of a message. When there is no data or CRC to send in synchronous modes, the transmitter inserts 6-, 8-, or 16-bit synchronous characters, regardless of the programmed character length.

The SCC supports synchronous bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message, the SCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. The SCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the $\overline{\text{SYNC}}$ pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all 0s inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the SCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all 1s or all 0s. The CRC is inverted before transmission and the receiver checks against the bit pattern 0001110100001111.

NRZ, NRZI or FM coding may be used in any 1X mode. The parity options available in asynchronous modes are available in synchronous modes.

The SCC can be conveniently used under DMA control to provide high-speed reception or transmission. In reception, for example, the SCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The SCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received. The CPU may also enable the DMA first and have the SCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

SDLC LOOP MODE

The SCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow and any number of secondary stations. In SDLC Loop mode, the SCC performs the functions of a secondary station while an SCC operating in regular SDLC mode can act as a controller (Figure 5).

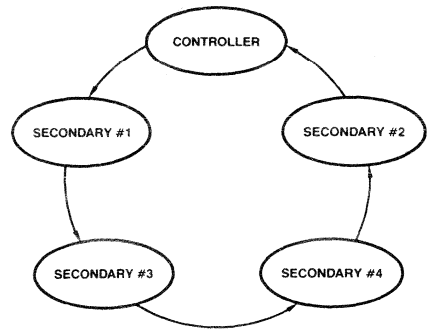


Figure 5. An SDLC Loop

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop and, in fact, must pass these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary one of the EOP to a zero before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop (except upon recognizing an EOP).

SDLC Loop mode is a programmable option in the SCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

Baud Rate Generator

Each channel in the SCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On start-up, the flip-flop on the output is set in a High state; the value in the time constant register is loaded into the counter; and the counter starts counting down. The output of the baud rate generator toggles upon reaching zero; the value in the time constant register is loaded into the counter; and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the digital phase-locked loop (see next section).

If the receive clock or transmit clock is not programmed to come from the $\overline{TRxC}$ pin, the output of the baud rate generator may be echoed out via the $\overline{TRxC}$ pin.

The following formula relates the time constant to the baud rate. (The baud rate is in bits/second and the BR clock period is in seconds.)

$$\text{baud rate} = \frac{1}{2\ (\text{time constant} + 2) \times (\text{BR clock period})}$$

Time-Constant Values for Standard Baud Rates at BR Clock = 3.9936MHz

Rate (Baud)	Time Constant (decimal notation)	Error
19200	102	–
9600	206	–
7200	275	0.12%
4800	414	–
3600	553	0.06%
2400	830	–
2000	996	0.04%
1800	1107	0.03%
1200	1662	–
600	3326	–
300	6654	–
150	13310	–
134.5	14844	0.0007%
110	18151	0.0015%
75	26622	–
50	39934	–

Digital Phase-Locked Loop

The SCC contains a digital phase-locked loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the SCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32X clock to create nominal bit times. As the 32X clock is counted, the DPLL is searching the incoming data stream for edges (either 1/0 or 0/1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15/16 counting transition.

The 32X clock for the DPLL can be programmed to come from either the RTxC input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the $\overline{TRxC}$ pin (if this pin is not being used as an input).

Data Encoding

The SCC may be programmed to encode and decode the serial data in four different ways (Figure 6). In NRZ encoding, a 1 is represented by a High level, and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level, and a 0 is represented by a change in level. In FM_1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell, and a 0 is represented by no additional transition at the center of the bit cell. In FM_0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the SCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0/1, the bit is a 0. If the transition is 1/0, the bit is a 1.

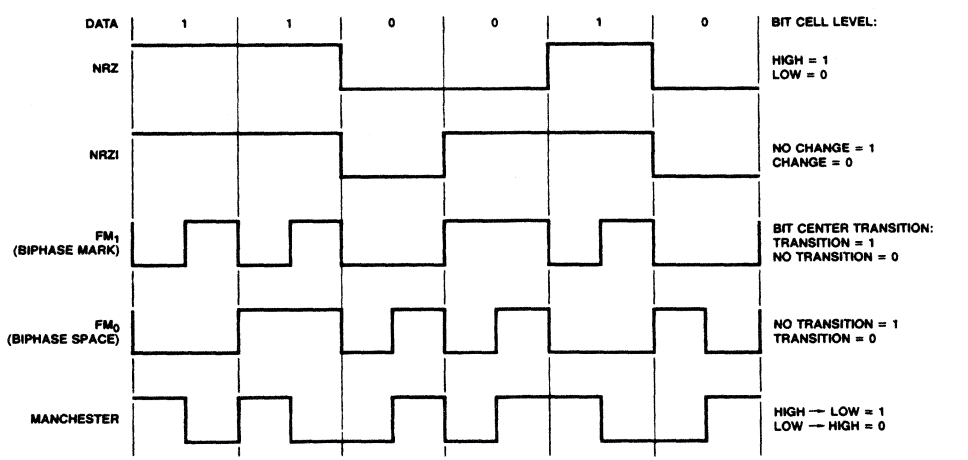


Figure 6. Data Encoding Methods

Auto Echo and Local Loopback

The SCC is capable of automatically echoing everything it receives. This feature is useful mainly in asynchronous modes but works in synchronous and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In Auto Echo mode, the $\overline{CTS}$ input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed, and the programmer is responsible for disabling transmitter interrupts and $\overline{WAIT/REQUEST}$ on transmit.

The SCC is also capable of local loopback. In this mode, TxD is RxD just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data, and RxD is ignored (except to be echoed out via TxD). The CTS and DCD inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in asynchronous, synchronous and SDLC modes with NRZ, NRZI or FM coding of the data stream.

I/O Interface Capabilities

The SCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

Polling

All interrupts are disabled. Three status registers in the SCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

Interrupts

When a SCC responds to an Interrupt Acknowledge signal ($\overline{INTACK}$) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures 8 and 9).

To speed interrupt response time, the SCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the SCC (Transmit, Receive and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write-only.

The other two bits are related to the Z-Bus interrupt priority chain (Figure 7). As a Z-Bus peripheral, the SCC may request an interrupt only when no higher priority device is requesting one; e.g., when IEI is HIGH. If the device in question requests an interrupt, it pulls down $\overline{INT}$. The CPU then responds with $\overline{INTACK}$, and the interrupting device places the vector on the A/D bus.

In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is HIGH, the INT output is pulled LOW, requesting an interrupt. In the SCC, if the IE bit is not set by enabling interrupts, then the IP for that source can never be set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the SCC and external to the SCC are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the SCC being pulled LOW and propagated to subsequent peripherals. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts.

There are three types of interrupts: Transmit, Receive and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receive, Transmit and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the Receive can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive condition.
- Interrupt on all Receive Characters or Special Receive condition.
- Interrupt on Special Receive condition only.

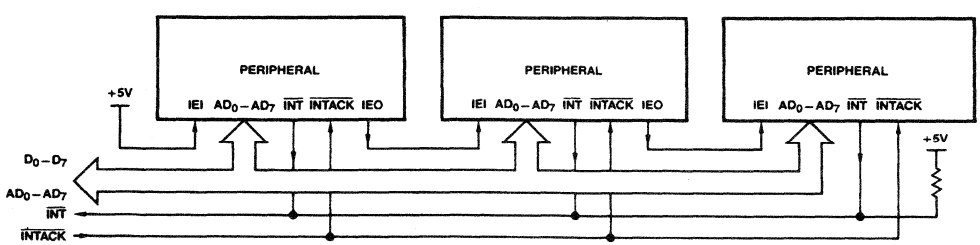


Figure 7. Z-Bus Interrupt Schedule

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in asynchronous mode, End-of-Frame in SDLC mode, and optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only in the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt can occur from Special Receive conditions any time after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{CTS}$, $\overline{DCD}$, and $\overline{SYNC}$ pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition, a zero count in the baud rate generator, the detection of a Break (asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the SCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the wishes of the primary station to regain control of the loop during a poll sequence.

CPU/DMA Block Transfer

The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{WAIT}/\overline{Request}$ output in conjunction with the Wait/Request bits in WR1. The $\overline{WAIT}/\overline{REQUEST}$ output can be defined under software control as a $\overline{WAIT}$ line in the CPU Block Transfer mode or as a $\overline{REQUEST}$ line in the DMA Block Transfer mode.

To a DMA controller, the SCC $\overline{REQUEST}$ output indicates that the SCC is ready to transfer data to or from memory. To the CPU, the $\overline{WAIT}$ line indicates that the SCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The $\overline{DTR}/\overline{REQUEST}$ line allows full-duplex operation under DMA control.

PROGRAMMING INFORMATION

Each channel has fifteen Write registers that are individually programmed from the system bus to configure the functional personality of each channel. Each channel also has eight Read registers from which the system can read Status, Baud rate, or Interrupt information.

The Z8030 and Z8530(H) differ in the way the system accesses these registers:

In the Z8030 all registers are directly addressable from the multiplexed Address Data bus. See Figure 10 and Figure 11 for timing. The Z8030 can operate in either of two modes: when bit 0 in Write Register 0 is reset (or after initialization with a hardware reset), Address lines AD_1 through AD_5 select the register to be read from or written into during Data Strobe $\overline{DS}$. (This is called left shift and is the natural Z8000 mode.) When bit 0 in Write Register 0 is set, Address lines AD_0 through AD_4 select the register to be read from or written into. (This is called right shift and is more natural for interfacing with other microprocessors.)

Table 2 describes the register addressing for both modes.

Channel A/Channel B selection is made either by AD_0 or by AD_5.

If Bit D_0 in WR0 is reset (or after hardware reset):
AD_5 selects the channel (0 = B, 1 = A)
(this is called "Select Shift Left Mode").

If Bits D_0 and D_1 in WR0 are set:
AD_0 selects the channel (0 = B, 1 = A)
(this is called "Select Shift Right Mode).

In the Z8530(H) only the four data registers (Read and Write for Channels A and B) are directly selected by a HIGH on the $D/\overline{C}$ input and the appropriate levels on the $\overline{RD}$, $\overline{WR}$ and $A/\overline{B}$ pins. All other registers are addressed indirectly by the content of Write Register 0 in conjunction with a LOW on the $D/\overline{C}$ input and the appropriate levels on the $\overline{RD}$, $\overline{WR}$ and $A/\overline{B}$ pins. If bit D3 in WR0 is 1 and bits 5 and 6 are 0, then bits 0, 1, 2 address the higher registers 8 through 15. If bits 4, 5, 6 contain a different code, bits 0, 1, 2 address the lower registers 0 through 7 as shown in Table 3.

TABLE 2. REGISTER ADDRESSING (Z8030 ONLY)

AD_4	AD_3	AD_2	AD_1	Write Register	Read Register
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	(0)
0	1	0	1	5	(1)
0	1	1	0	6	(2)
0	1	1	1	7	(3)
1	0	0	0	Data	Data
1	0	0	1	9	–
1	0	1	0	10	10
1	0	1	1	11	(15)
1	1	0	0	12	12
1	1	0	1	13	13
1	1	1	0	14	(10)
1	1	1	1	15	15

Writing to or reading from any register except RR0, WR0 and the Data Registers thus involves two operations:

First, write the appropriate code into WR0, then follow this by a write or read operation on the register thus specified. Bits 0 through 4 in WW0 are automatically cleared after this operation, so that WW0 then points to WR0 or RR0 again.

Channel A/Channel B selection is made by the $A/\overline{B}$ input (HIGH = A, LOW = B)

In both Z8030 and Z8530(H), the system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, the asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set and, finally, receiver or transmitter enable.

TABLE 3. REGISTER ADDRESSING (Z8530(H) ONLY)

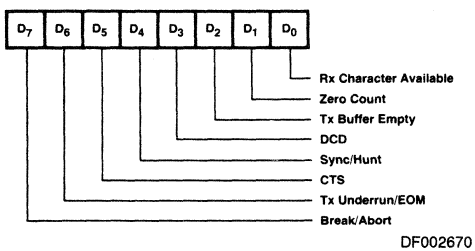
D/C̄ "Point High" Code in WR0:		D_2 In WR0:	D_1 In WR0:	D_0 In WR0:	Write Register	Read Register
HIGH	Either way	X	X	X	Data	Data
LOW	Not true	0	0	0	0	0
LOW	Not true	0	0	1	1	1
LOW	Not true	0	1	0	2	2
LOW	Not true	0	1	1	3	3
LOW	Not true	1	0	0	4	(0)
LOW	Not true	1	0	1	5	(1)
LOW	Not true	1	1	0	6	(2)
LOW	Not true	1	1	1	7	(3)
LOW	True	0	0	0	Data	Data
LOW	True	0	0	1	9	–
LOW	True	0	1	0	10	10
LOW	True	0	1	1	11	(15)
LOW	True	1	0	0	12	12
LOW	True	1	0	1	13	13
LOW	True	1	1	0	14	(10)
LOW	True	1	1	1	15	15

Read Registers

The SCC contains eight read registers (actually nine, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). Figure 8 shows the formats for each read register.

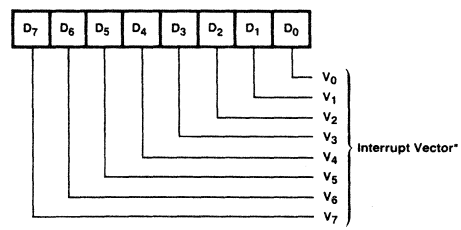
The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring; e.g., when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).

Read Register 0



DF002670

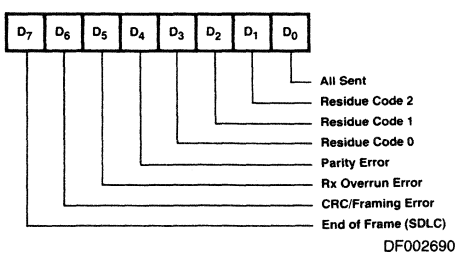
Read Register 2



*Modified in B Channel

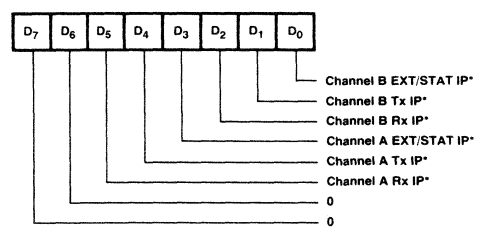
DF002680

Read Register 1



DF002690

Read Register 3



*Always 0 in B Channel

DF002700

Figure 8. Read Register Bit Functions

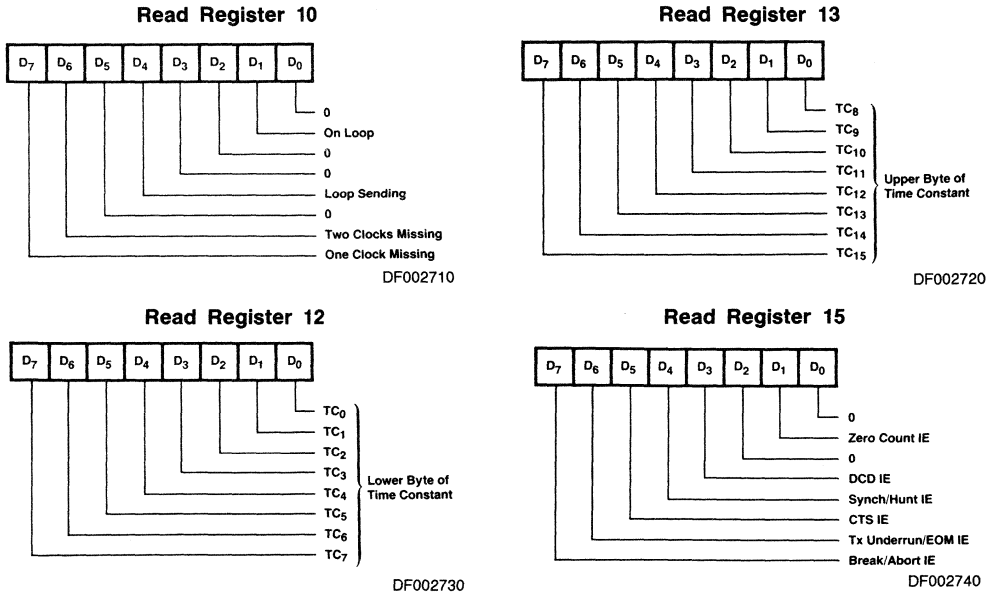


Figure 8. Read Register Bit Functions (Cont.)

Write Registers

The SCC contains 15 write registers (16 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. In addition, there are two registers (WR2 and WR9) shared by the two channels that may be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Figure 9 shows the format of each write register.

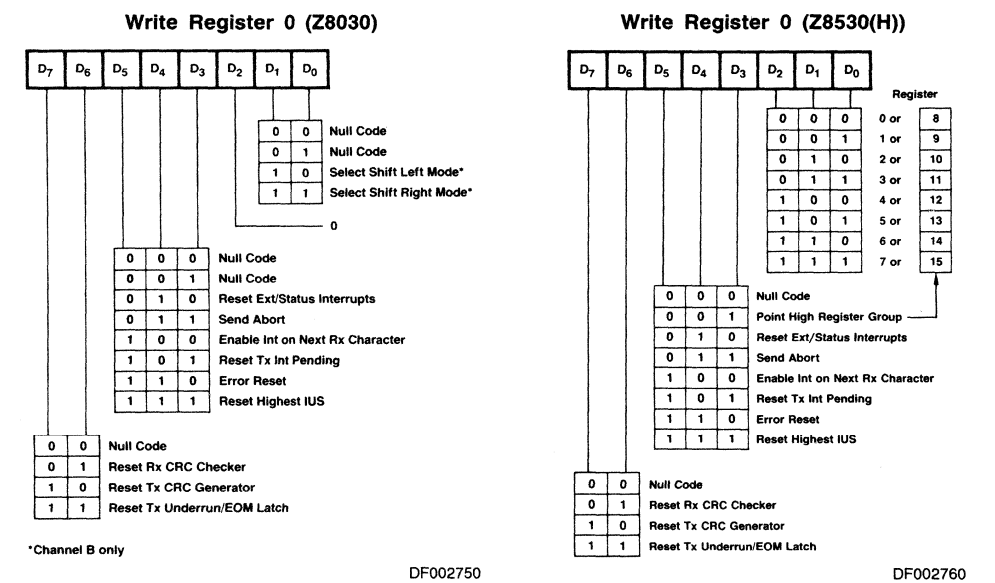
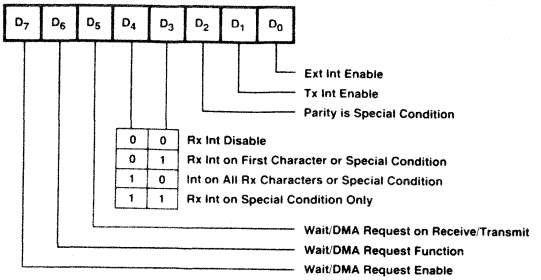


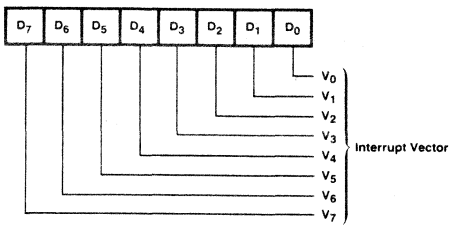
Figure 9. Write Register Bit Functions

Write Register 1



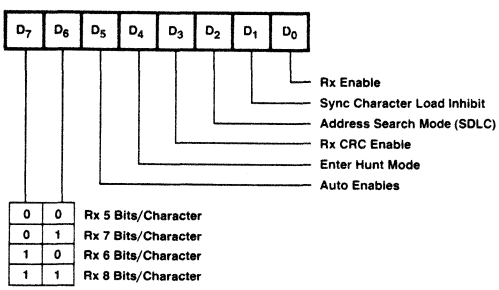
DF002770

Write Register 2



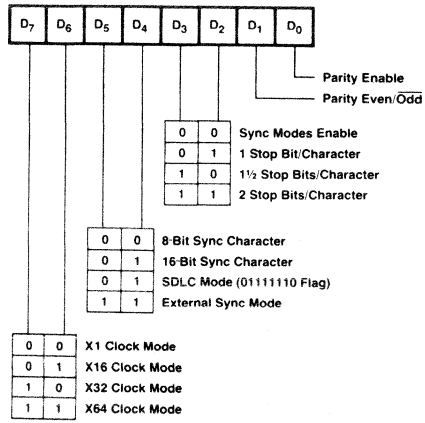
DF002780

Write Register 3



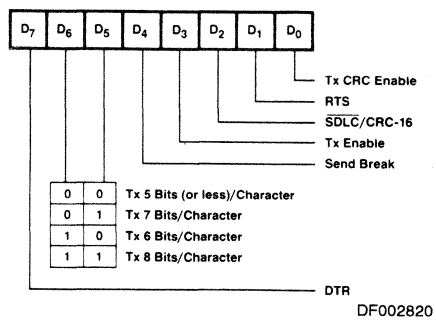
DF002790

Write Register 4



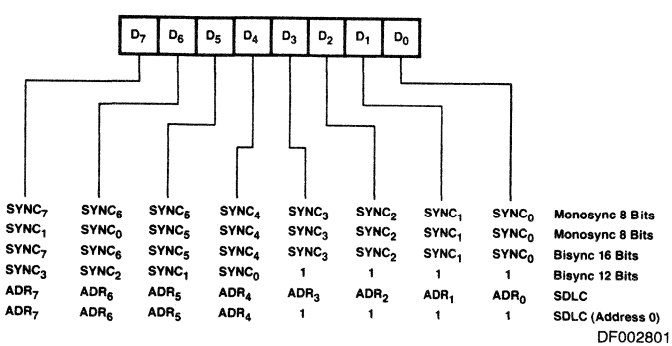
DF002811

Write Register 5



DF002820

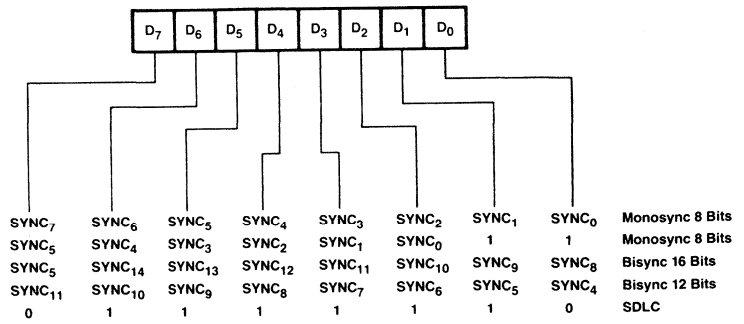
Write Register 6



DF002801

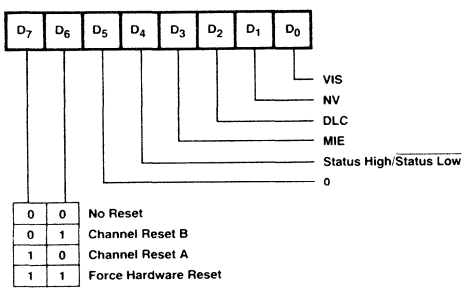
Figure 9. Write Register Bit Functions (Cont.)

Write Register 7



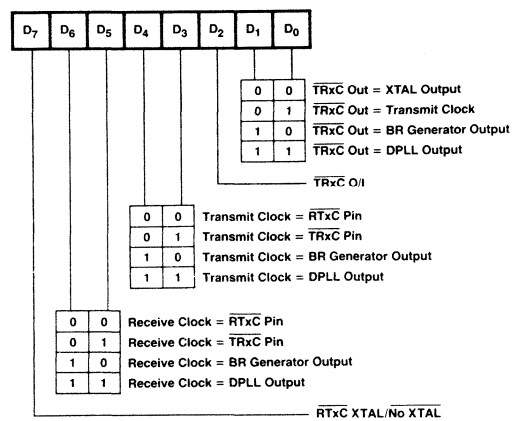
DF002831

Write Register 9



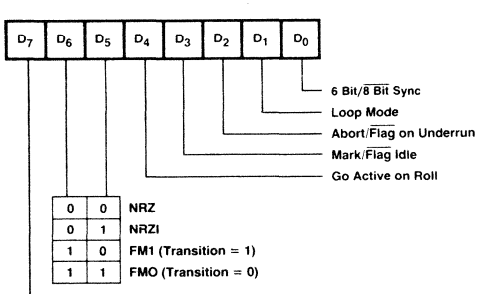
DF002840

Write Register 11



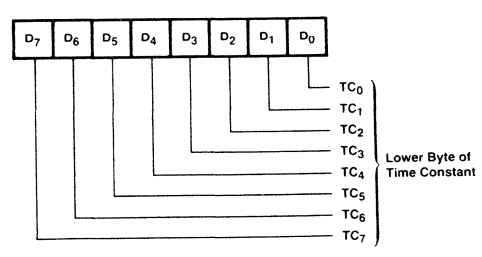
DF002850

Write Register 10



DF002860

Write Register 12



DF002870

Figure 9. Write Register Bit Functions (Cont.)

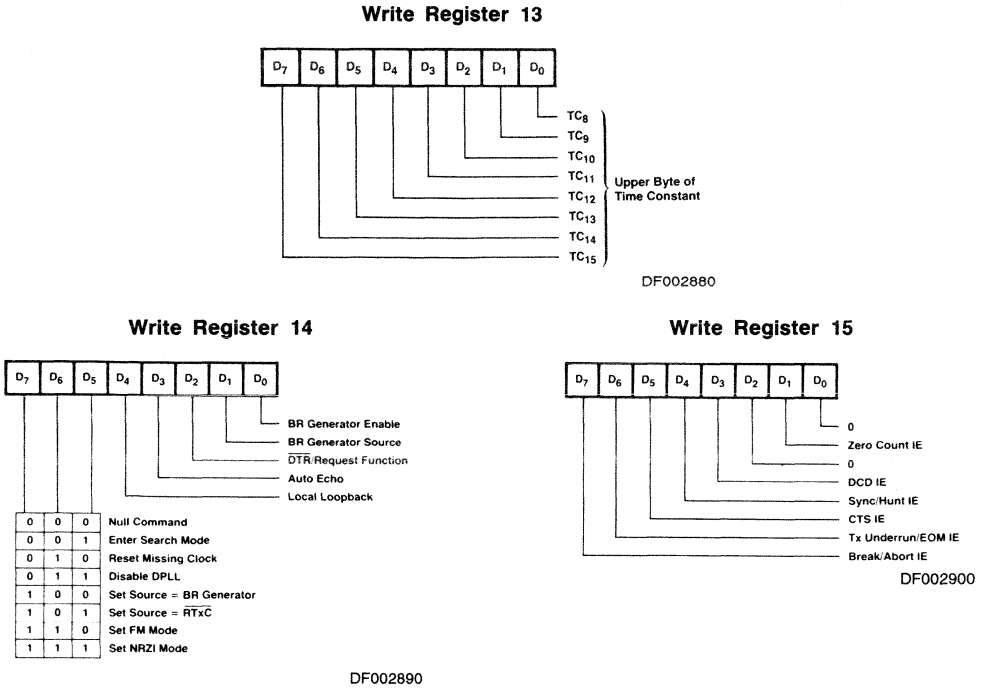


Figure 9. Write Register Bit Functions (Cont.)

Z8030 Timing

The SCC generates internal control signals from $\overline{AS}$ and $\overline{DS}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{AS}$ and $\overline{DS}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC to the falling edge of $\overline{DS}$ in the second transaction involving the SCC. This time must be at least 6 PCLK cycles plus 200ns.

Read Cycle Timing

Figure 10 illustrates read cycle timing. The address on $AD_0 - AD_7$ and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. R/$\overline{W}$ must be HIGH to indicate a read cycle. CS_1 must also be HIGH for the read cycle to occur. The data bus drivers in the SCC are then enabled while $\overline{DS}$ is LOW.

Write Cycle Timing

Figure 11 illustrates write cycle timing. The address on $AD_0 - AD_7$ and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. R/$\overline{W}$ must be LOW to indicate a write cycle. CS_1 must be HIGH for the write cycle to occur. $\overline{DS}$ Low strobes the data into the SCC.

Interrupt Acknowledge Cycle Timing

Figure 12 illustrates interrupt acknowledge cycle timing. The address on $AD_0 - AD_7$ and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. However, if $\overline{INTACK}$ is LOW, the address and $\overline{CS_0}$ are ignored. The state of R/$\overline{W}$ and CS_1 are also ignored for the duration of the interrupt acknowledge cycle. Between the rising edge of $\overline{AS}$ and the falling edge of $\overline{DS}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEI is HIGH when $\overline{DS}$ falls, the acknowledge cycle was intended for the SCC. In this case, the SCC may be programmed to respond to $\overline{DS}$ Low by placing its interrupt vector on $AD_0 - AD_7$. It then sets the appropriate interrupt-under-service latch internally.

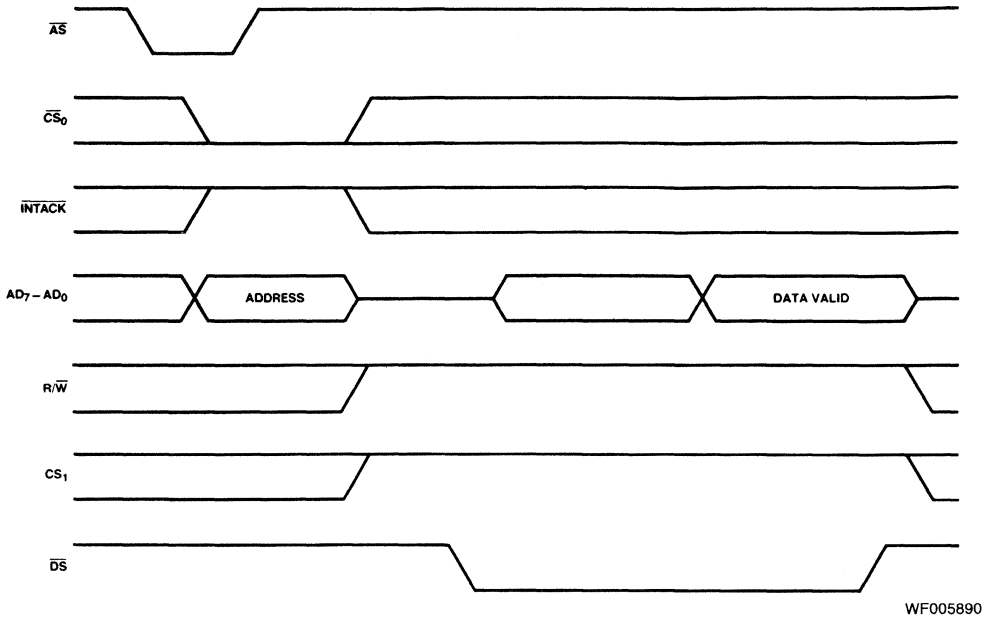


Figure 10. Read Cycle Timing

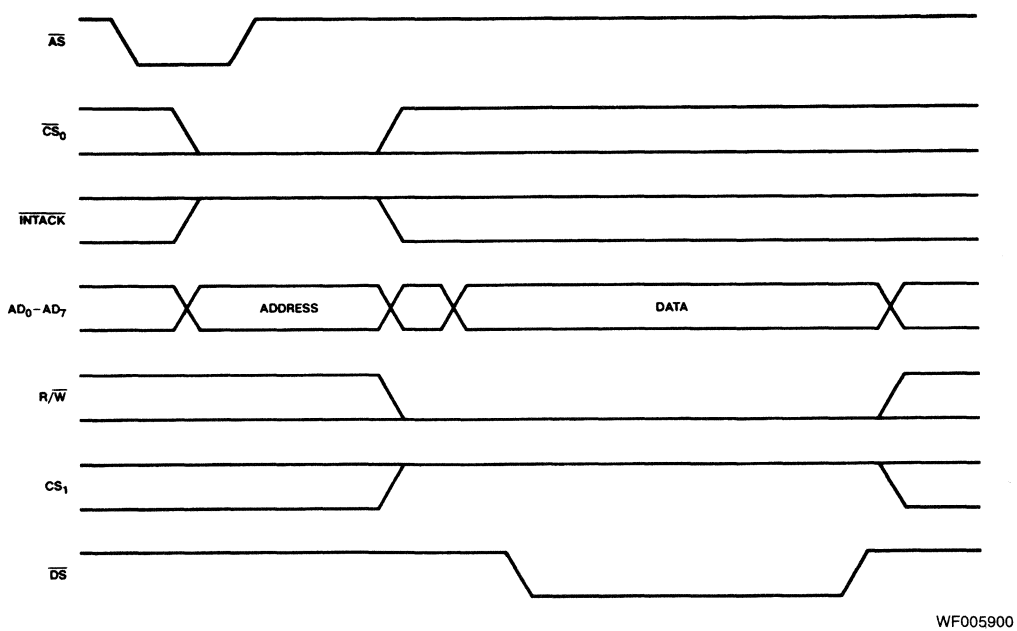


Figure 11. Write Cycle Timing

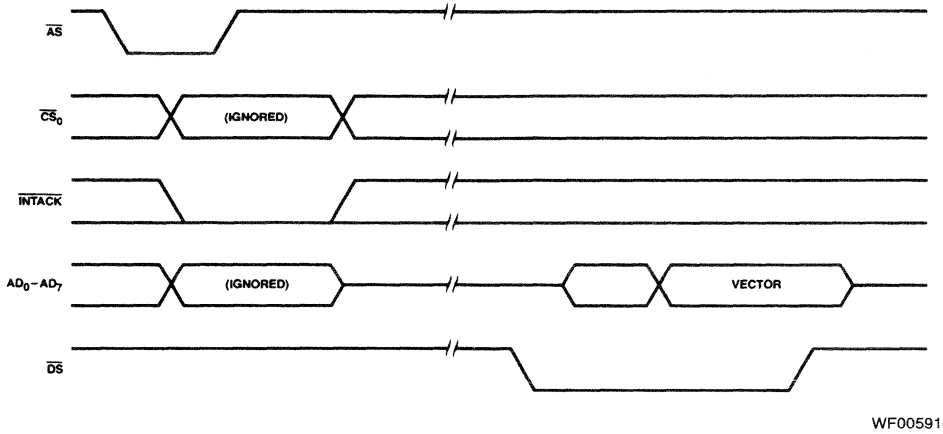


Figure 12. Interrupt Acknowledge Cycle Timing

Z8530(H) Timing

The SCC generates internal control signals from $\overline{WR}$ and $\overline{RD}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{WR}$ and $\overline{RD}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC. The recovery time required for proper operation is specified from the rising edge of $\overline{WR}$ or $\overline{RD}$ in the first transaction involving the SCC to the falling edge of $\overline{WR}$ or $\overline{RD}$ in the second transaction involving the SCC. This time must be at least 6 PCLK cycles plus 200ns.

Read Cycle Timing

Figure 13 illustrates Read cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{RD}$ falls or if it rises before $\overline{RD}$ rises, the effective $\overline{RD}$ is shortened.

Write Cycle Timing

Figure 14 illustrates Write cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{WR}$ falls or if it rises before $\overline{WR}$ rises, the effective $\overline{WR}$ is shortened

Interrupt Acknowledge Cycle Timing

Figure 15 illustrates Interrupt Acknowledge cycle timing. Between the time $\overline{INTACK}$ goes LOW and the falling edge of $\overline{RD}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEI is HIGH when $\overline{RD}$ falls, the Acknowledge cycle is intended for the SCC. In this case, the SCC may be programmed to respond to $\overline{RD}$ Low by placing its interrupt vector on $D_0 - D_7$, and it then sets the appropriate Interrupt-Under-Service internally.

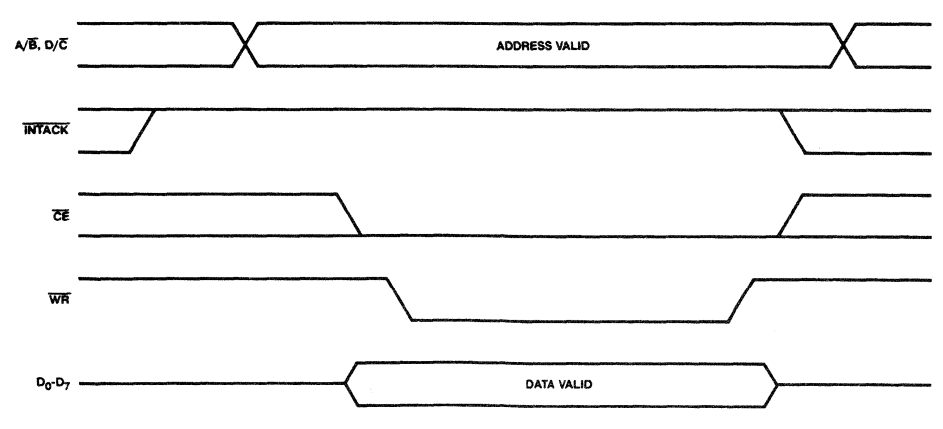


Figure 13. Read Cycle Timing

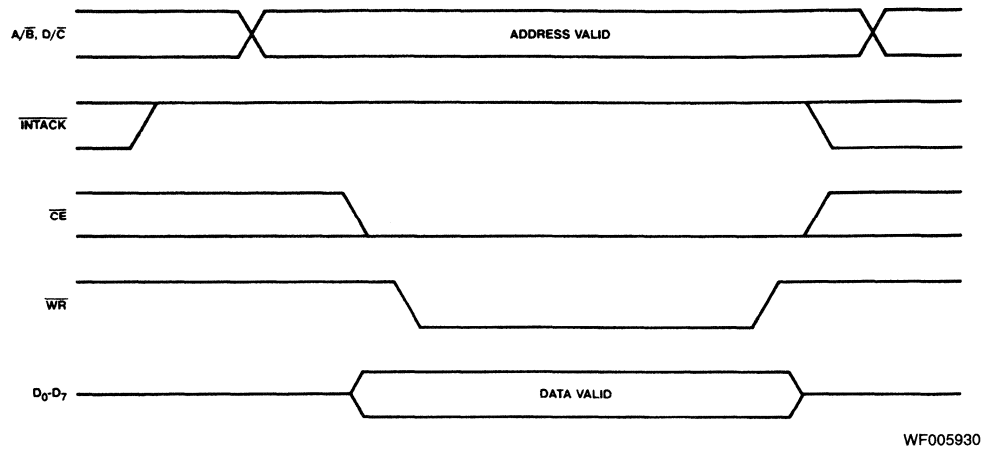


Figure 14. Write Cycle Timing

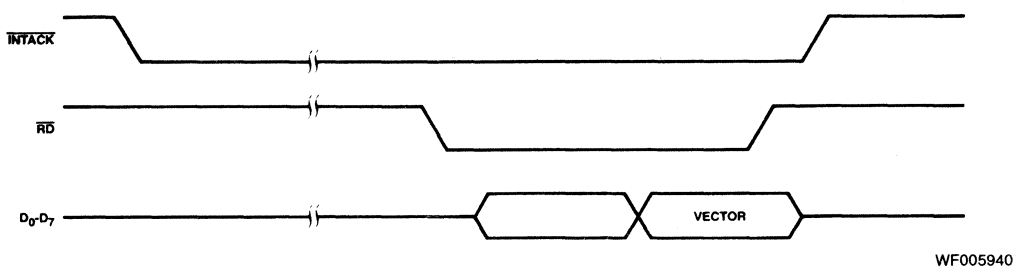


Figure 15. Interrupt Acknowledge Cycle Timing

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage at any Pin
Relative to V_{SS} ..... −0.5 to +7.0V
Power Dissipation 1.8W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ± 5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range

Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{IH}	Input HIGH Voltage	Commercial	2.0		V_{CC}+0.3	V
V_{IL}	Input LOW Voltage		−0.3		0.8	V
V_{OH}	Output HIGH Voltage	I_{OH} = −250μA	2.4			V
V_{OL}	Output LOW Voltage	I_{OL} = +2.0mA			0.4	V
I_{IL}	Input Leakage	0.4 V ≤ V_{IN} ≤ 2.4 V			±10.0	μA
I_{OL}	Output Leakage	0.4 V ≤ V_{OUT} ≤ 2.4 V			±10.0	μA
I_{CC}	V_{CC} Supply Current				250	mA
C_{IN}	Input Capacitance	Unmeasured pins returned to ground. f = 1MHz at T_A = 25°C.			10	pF
C_{OUT}	Output Capacitance				15	pF
$C_{I/O}$	Bidirectional Capacitance				20	pF

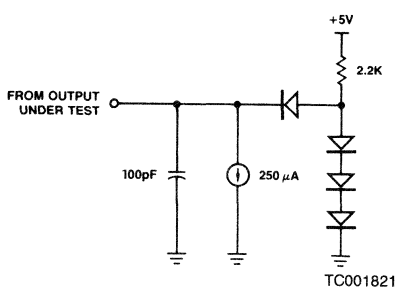
Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

+4.75V ≤ V_{CC} ≤ +5.25V
GND = 0V
0°C ≤ T_A ≤ 70°C

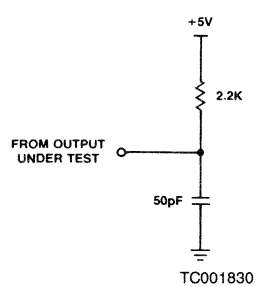
SWITCHING TEST CIRCUITS

Standard Test Load



TC001821

Open Drain Test Load



TC001830

SWITCHING TEST INPUT/OUTPUT WAVEFORM

WF006352

AC testing: Inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0".
Timing measurements are made at 2.0 V for a logic "1" and 0.8 V for logic "0".

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating range

GENERAL TIMING

Number	Parameters	Description		4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Z8530H Only Min	8 MHz Z8530H Only Max	Units
1	TdPC(REQ)	PCLK ↓ to W/REQ Valid Delay			250		250		250	ns
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay			350		350		350	ns
3	TsRXC(PC)	RxC ↑ to PCLK ↑ Set-up Time (Notes 1, 4)	Z8530	80	T_{WPCL}	70	T_{WPCL}	NA	NA	ns
			Z8530H	80	T_{WPCL}	70	T_{WPCL}	60	T_{WPCL}	
4	TsRXD(RXCr)	RxD to RxC ↑ Set-up Time (XI Mode) (Note 1)		0		0		0		ns
5	ThRXD(RXCr)	RxD to RxC ↑ Hold Time (XI Mode) (Note 1)		150		150		150		ns
6	TsRXD(RXCf)	RxD to RxC ↓ Set-up Time (XI Mode) (Notes 1, 5)		0		0		0		ns
7	ThRXD(RXCf)	RxD to RxC ↓ Hold Time (XI Mode) (Notes 1, 5)		150		150		150		ns
8	TsSY(RXC)	SYNC to RxC ↑ Set-up Time (Note 1)		−200		−200		−200		ns
9	ThSY(RXC)	SYNC to RxC ↑ Hold Time (Note 1)		3TcPC +400		3TcPC +320		3TcPC +250		ns
10	TsTXC(PC)	TxC ↓ to PCLK ↑ Set-up Time (Notes 2, 4)		0		0		0		ns
11	TdTXCf(TXD)	TxC ↓ to TxD Delay (XI Mode) (Note 2)			300		230		200	ns
12	TdTXCr(TXD)	TxC ↑ to TxD Delay (XI Mode) (Notes 2, 5)			300		230		200	ns
13	TdTXD(TRX)	TxD to TRxC Delay (Send Clock Echo)			200		200		200	ns
14	TwRTXh	RTxC High Width (Note 6)		180		180		150		ns
15	TwRTXl	RTxC Low Width (Note 6)		180		180		150		ns
16	TcRTX	RTxC Cycle Time (Notes 6, 7)		1000		660		500		ns
17	TcRTXX	Crystal Oscillator Period (Note 3)		250	1000	165	1000	125	1000	ns
18	TwTRXh	TRxC High Width (Note 6)		180		180		150		ns
19	TwTRXl	TRxC Low Width (Note 6)		180		180		150		ns
20	TcTRX	TRxC Cycle Time (Notes 6, 7)		1000		660		500		ns
21	TwEXT	DCD or CTS Pulse Width		200		200		200		ns
22	TwSY	SYNC Pulse Width		200		200		200		ns

Notes:
1. RxC is RTxC or TRxC, whichever is supplying the receive clock.
2. TxC is TRxC or RTxC, whichever is supplying the transmit clock.
3. Both RTxC and SYNC have 18 pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between RxC and PCLK or TxC and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.

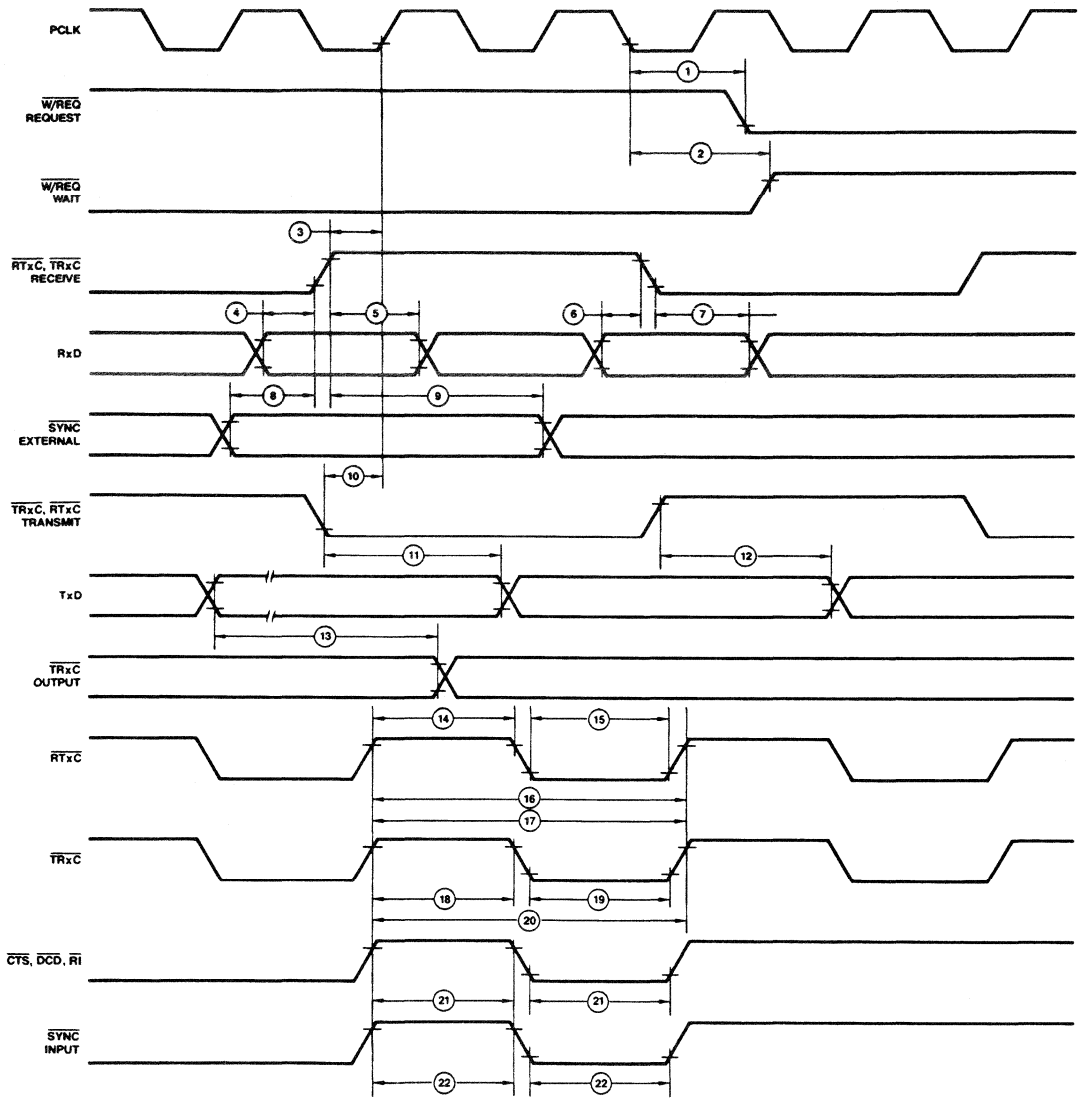


Figure 16. General Timing

SWITCHING CHARACTERISTICS over operating range

SYSTEM TIMING (Z8030)

Number	Parameters	Description	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	Units
1	TdRXC(REQ)	$\overline{RxC}$ ↑ to $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	TcPC
2	TdRXC(W)	$\overline{RxC}$ ↑ Wait Inactive Delay (Notes 1, 2)	8	14	8	14	TcPC
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	TcPC
4	TdRXC(INT)	$\overline{RxC}$ ↑ $\overline{INT}$ Valid Delay (Notes 1, 2)	8 + 2	12 + 3	8 + 2	12 + 3	TcPC / $\overline{AS}$
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	TcPC
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	TcPC
7	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	TcPC
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	4 + 2	6 + 3	4 + 2	6 + 3	TcPC / $\overline{AS}$
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	3	2	3	TcPC
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	3	2	3	TcPC

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.

SWITCHING CHARACTERISTICS over operating range

SYSTEM TIMING (Z8530, Z8530H)

Number	Parameters	Description	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Z8530H Only Min	8 MHz Z8530H Only Max	Units
1	TdRXC(REQ)	$\overline{RxC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	8	12	TcPC
2	TdRXC(W)	$\overline{RxC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	8	14	8	14	TcPC
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	4	7	TcPC
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	10	16	TcPC
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	5	8	TcPC
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	5	11	TcPC
7	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	4	7	TcPC
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	6	10	TcPC
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	2	6	TcPC
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	2	6	TcPC

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.

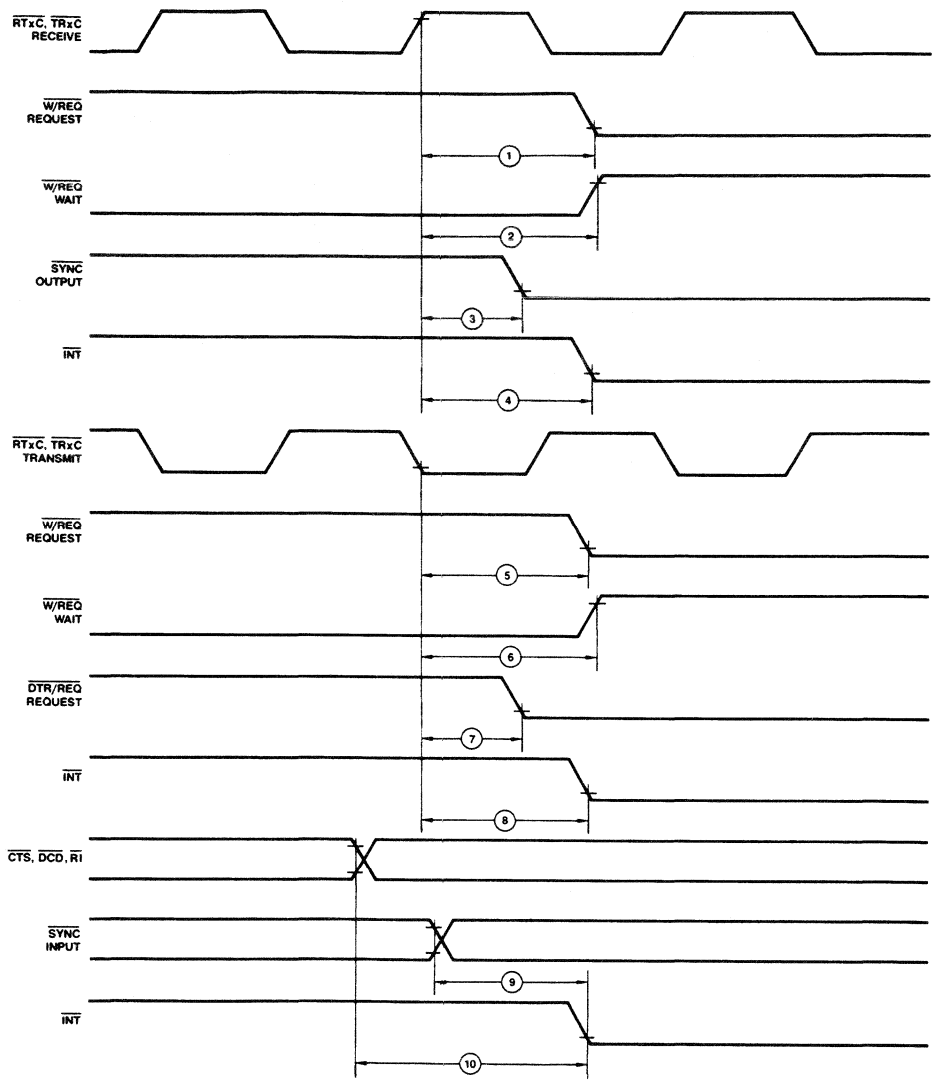


Figure 17. System Timing

SWITCHING CHARACTERISTICS over operating range

READ AND WRITE TIMING (Z8030)

Number	Parameters	Description	4 MHz		6 MHz		Units
			Min	Max	Min	Max	
1	TwAS	$\overline{AS}$ LOW Width	70		50		ns
2	TdDS(AS)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	50		25		ns
3	TsCS0(AS)	$\overline{CS_0}$ to $\overline{AS}$ ↑ Set-up Time (Note 1)	0		0		ns
4	ThCS0(AS)	$\overline{CS_0}$ to $\overline{AS}$ ↑ Hold Time (Note 1)	60		40		ns
5	TsCS1(DS)	CS_1 to $\overline{DS}$ ↓ Set-up Time (Note 1)	100		80		ns
6	ThCS1(DS)	CS_1 to $\overline{DS}$ ↑ Hold Time (Note 1)	55		40		ns
7	TsIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Set-up Time	10		10		ns
8	ThIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Hold Time	250		200		ns
9	TsRWR(DS)	R/$\overline{W}$ (Read) to $\overline{DS}$ ↓ Set-up Time	100		80		ns
10	ThRW(DS)	R/$\overline{W}$ to $\overline{DS}$ ↑ Hold Time	55		40		ns
11	TsRWW(DS)	R/$\overline{W}$ (Write) to $\overline{DS}$ ↓ Set-up Time	0		0		ns
12	TdAS(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay	60		40		ns
13	TwDSI	$\overline{DS}$ LOW Width	390		250		ns
14	TrC	Valid Access Recovery Time (Note 2)	6TcPC + 200		6TcPC + 130		ns
15	TsA(AS)	Address to $\overline{AS}$ ↑ Set-up Time (Note 1)	30		10		ns
16	ThA(AS)	Address to $\overline{AS}$ ↑ Hold Time (Note 1)	50		30		ns
17	TsDW(DS)	Write Data to $\overline{DS}$ ↓ Set-up Time	30		20		ns
18	ThDW(DS)	Write Date to $\overline{DS}$ ↑ Hold Time	30		20		ns
19	TdDS(DA)	$\overline{DS}$ ↓ to Data Active Delay	0		0		ns
20	TdDSr(DR)	$\overline{DS}$ ↑ to Read Data Not Valid Delay	0		0		ns
21	TdDSf(DR)	$\overline{DS}$ ↓ to Read Data Valid Delay		250		180	ns
22	TdAS(DR)	$\overline{AS}$ ↑ to Read Data Valid Delay		520		335	ns

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Parameter applies only between transactions involving the S_{CC}.

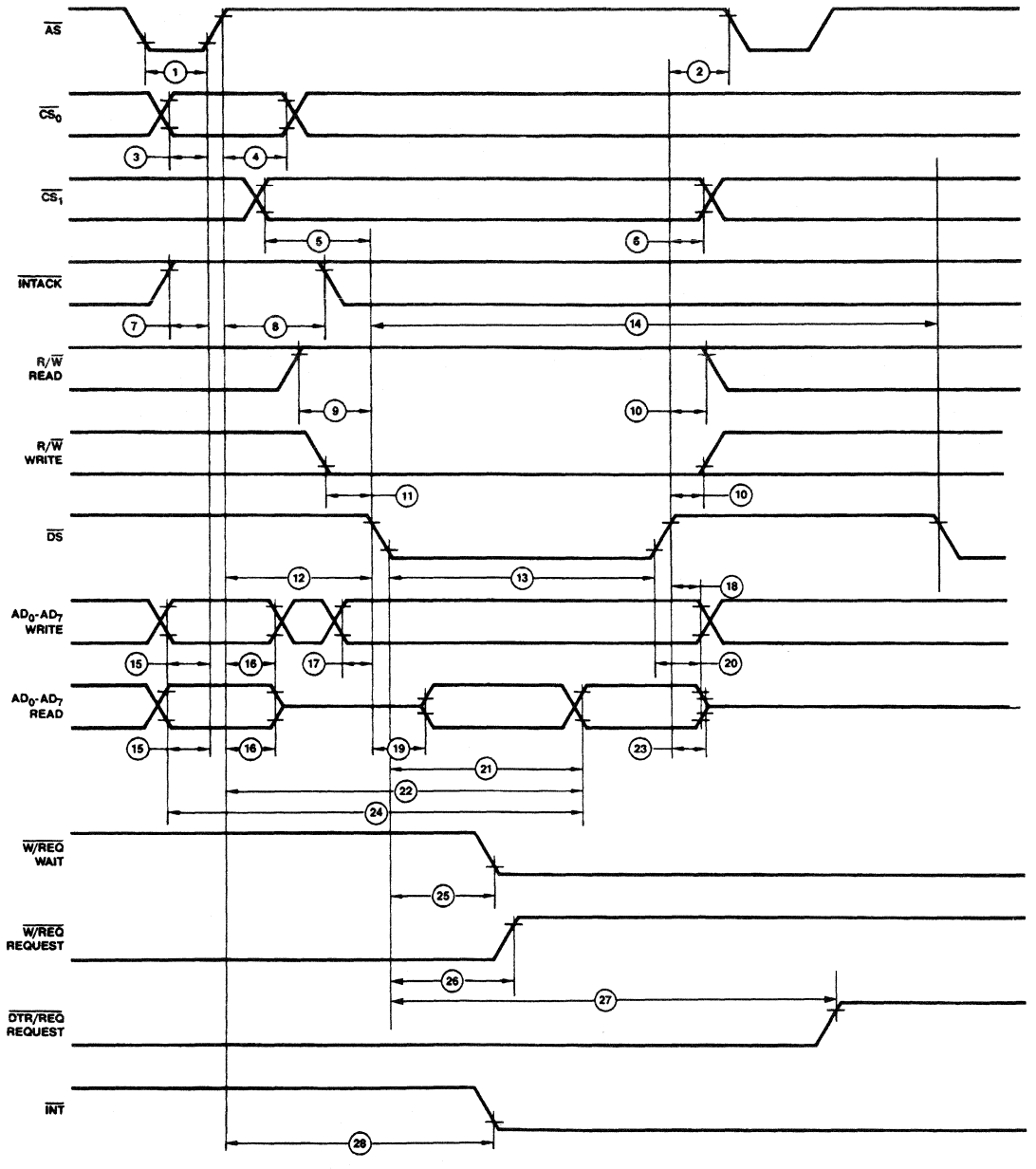


Figure 18. Read and Write Timing (Z8030)

SWITCHING CHARACTERISTICS over operating range
READ AND WRITE TIMING (Z8530, Z8530H)

Number	Parameters	Description	4 MHz		6 MHz		8 MHz Z8530H Only		Units
			Min	Max	Min	Max	Min	Max	
1	TwPCl	PCLK Low Width	105	2000	70	1000	50	1000	ns
2	TwPCh	PCLK High Width	105	2000	70	1000	50	1000	ns
3	TfPC	PCLK Fall Time		20		10		10	ns
4	TrPC	PCLK Rise Time		20		10		10	ns
5	TcPC	PCLK Cycle Time	250	4000	165	2000	125	2000	ns
6	TsA(WR)	Address to $\overline{WR}$ ↓ Set-up Time	80		80		70		ns
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		0		ns
8	TsA(RD)	Address to $\overline{RD}$ ↓ Set-up Time	80		80		70		ns
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		0		ns
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Set-up Time	10		10		10		ns
11	TsIAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Set-up Time (Note 1)	200		160		145		ns
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		0		ns
13	TsIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Set-up Time (Note 1)	200		160		145		ns
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		0		ns
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	100		100		85		ns
16	TsCEl(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Set-up Time	0		0		0		ns
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		0		ns
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Set-up Time	100		70		60		ns
19	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Set-up Time (Note 1)	0		0		0		ns
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note 1)	0		0		0		ns
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Set-up Time (Note 1)	100		70		60		ns
22	TwRDl	$\overline{RD}$ Low Width (Note 1)	240		200		150		ns
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		0		ns
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		0		ns
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		250		180		140	ns
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		70		45		140	ns

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time required for a ±0.5V change in the output with a maximum DC load and minimum AC load.

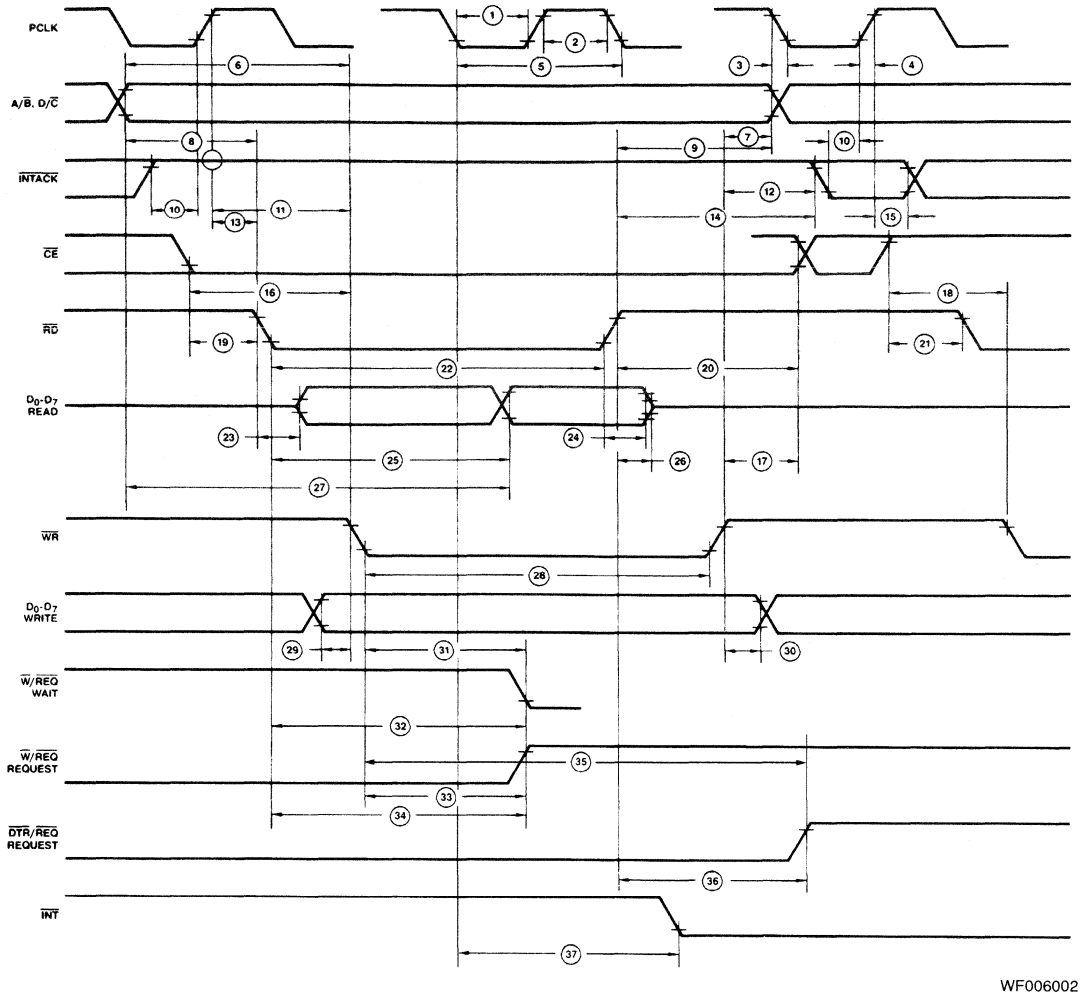


Figure 19. Read and Write Timing (Z8530, Z8530,H)

SWITCHING CHARACTERISTICS over operating range
INTERRUPT ACKNOWLEDGE TIMING, RESET TIMING, CYCLE TIMING (Z8030)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
23	TdDS(DRz)	$\overline{DS}$ ↑ to Read Data Float Delay (Note 3)		70		45	ns
24	TdA(DR)	Address Required Valid to Read Data Valid Delay		570		310	ns
25	TdDS(W)	$\overline{DS}$ ↓ to Wait Valid Delay (Note 4)		240		200	ns
26	TdDSf(REQ)	$\overline{DS}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200	ns
27	TdDSr(REQ)	$\overline{DS}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		5TcPC +300		5TcPC +250	ns
28	TdAS(INT)	$\overline{AS}$ ↑ to $\overline{INT}$ Valid Delay (Note 4)		500		500	ns
29	TdAS(DSA)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ (Acknowledge) Delay (Note 5)	250		250		ns
30	TwDSA	$\overline{DS}$ (Acknowledge) Low Width	390		250		ns
31	TdDSA(DR)	$\overline{DS}$ ↓ (Acknowledge) to Read Data Valid Delay		250		180	ns
32	TsIEI(DSA)	IEI to $\overline{DS}$ ↓ (Acknowledge) Set-up Time	120		100		ns
33	ThIEI(DSA)	IEI to $\overline{DS}$ ↑ (Acknowledge) Hold Time	0		0		ns
34	TdIEI(IEO)	IEI to IEO Delay		120		100	ns
35	TdAS(IEO)	$\overline{AS}$ ↑ to IEO Delay (Note 6)		250		250	ns
36	TdDSA(INT)	$\overline{DS}$ ↓ (Acknowledge) to $\overline{INT}$ Inactive Delay (Note 4)		500		500	ns
37	TdDS(ASQ)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay for No Reset	30		15		ns
38	TdASQ(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay for No Reset	30		30		ns
39	TwRES	$\overline{AS}$ and $\overline{DS}$ Coincident Low for Reset (Note 7)	250		200		ns
40	TwPCl	PCLK Low Width	105	2000	70	1000	ns
41	TwPCh	PCLK High Width	105	2000	70	1000	ns
42	TcPC	PCLK Cycle Time	250	4000	165	2000	ns
43	TrPC	PCLK Rise Time		20		10	ns
44	TfPC	PCLK Fall Time		20		10	ns

Notes: 3. Float delay is defined as the time required for a ±0.5V change in the output with a maximum DC load and minimum AC load.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any Z-SCC in the daisy chain, TdAS(DSA) must be greater than the sum of TdAS(IEO) for the highest priority device in the daisy chain, TsIEI(DSA) for the Z-SCC, and TdIEIf(IEO) for each device separating them in the daisy chain.
6. Parameter applies only to a Z-SCC pulling $\overline{INT}$ LOW at the beginning of the Interrupt Acknowledge transaction.
7. Internal circuitry allows for the reset provided by the Z8 to be recognized as a reset by the Z-SCC.

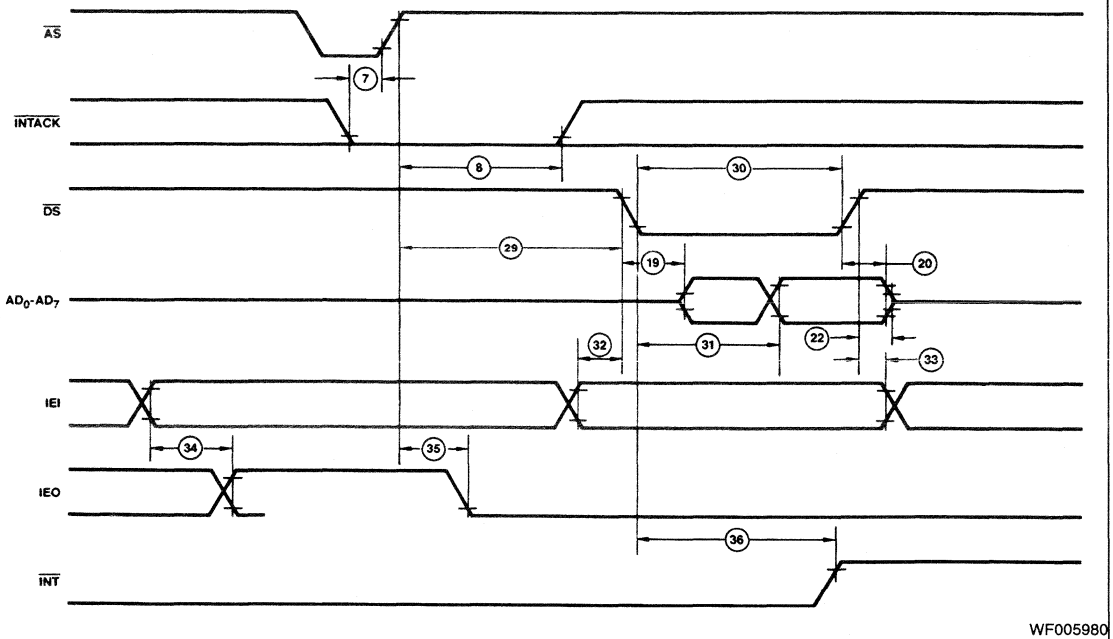


Figure 20. Interrupt Acknowledge Timing (Z8030)

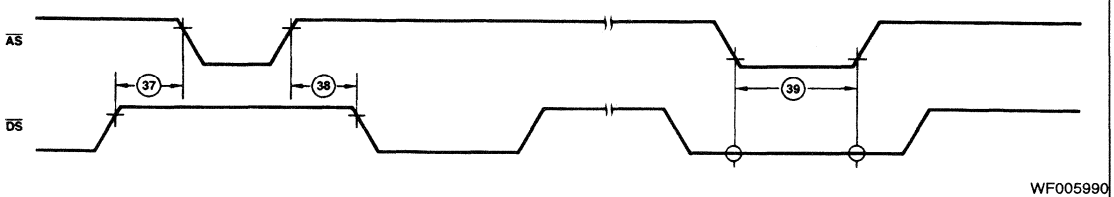


Figure 21. Reset Timing (Z8030)

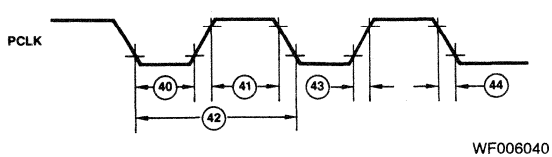


Figure 22. Cycle Timing (Z8030)

SWITCHING CHARACTERISTICS over operating range
INTERRUPT ACKNOWLEDGE TIMING, RESET TIMING, CYCLE TIMING (Z8530, Z8530H)

Number	Parameters	Description		4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Z8530H Only Min	8 MHz Z8530H Only Max	Units
27	TdA(DR)	Address Required Valid to Read Data Valid Delay	Z8530		400		350	NA	NA	ns
			Z8530H		300		280		220	
28	TwWRI	$\overline{WR}$ Low Width		240		200		150		ns
29	TsDW(WR)	Write Data to $\overline{WR}$ ↓ Set-up Time		10		10		10		ns
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time		0		0		0	170	ns
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 4)			240		200		170	ns
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 4)			240		200		170	ns
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay			240		200		170	ns
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay			240		200		170	ns
35	TdWRr(REQ)	$\overline{WR}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay			4TcPC		4TcPC		4TcPC	ns
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay			4TcPC		4TcPC		4TcPC	ns
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 4)			500		500		500	ns
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 5)		250		200		150		ns
39	TwRDA	$\overline{RD}$ (Acknowledge) Width		250		200		150		ns
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay			250		180		140	ns
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Set-up Time		120		100		95		ns
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time		0		0		0		ns
43	TdIEI(IEO)	IEI to IEO Delay Time			120		100		95	ns
44	TdPC(IEO)	PCLK ↑ to IEO Delay			250		250		200	ns
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 4)			500		500		450	ns
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset		30		15		15		ns
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset		30		30		20		ns
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset		250		200		150		ns
49	Trc	Valid Access Recovery Time (Note 3)	Z8530	6TcPC + 200		6TcPC + 130		NA	NA	ns
			Z8530H	4TcPC		4TcPC		4TcPC		

Notes: 3. Parameter applies only between transactions involving the SCC.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any SCC in the daisy chain, TdIAI(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEIf(IEO) for each device separating them in the daisy chain.

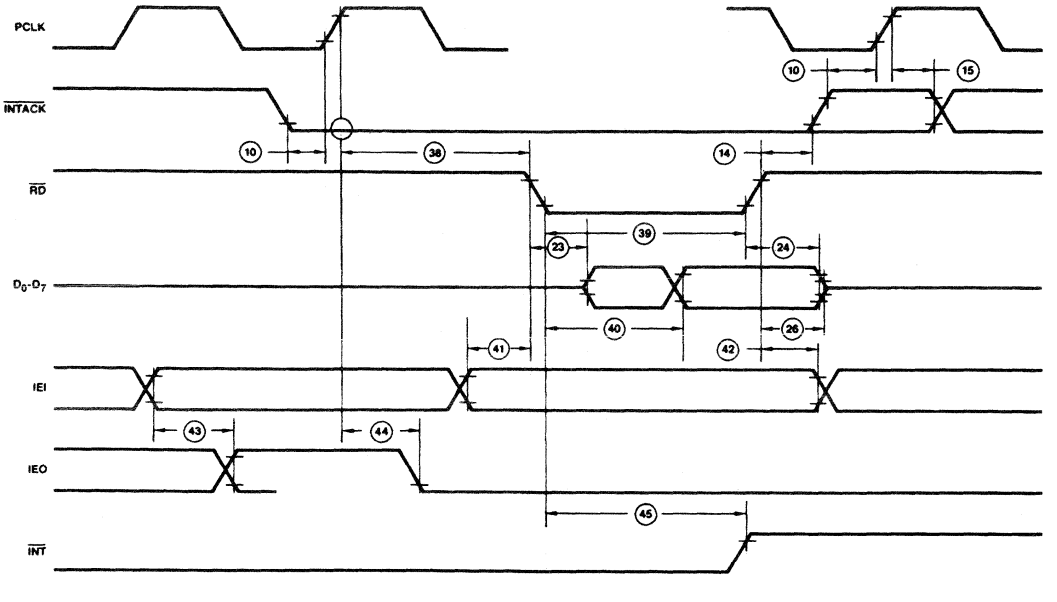


Figure 23. Interrupt Acknowledge Timing (Z8530)

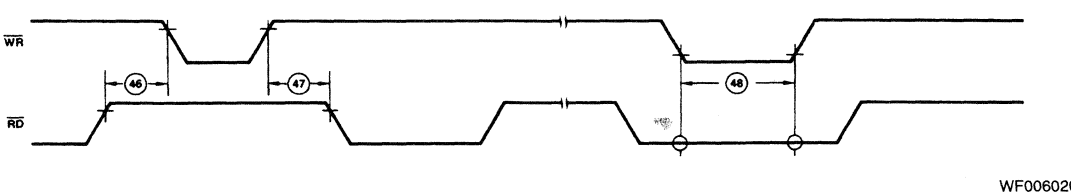


Figure 24. Reset Timing (Z8530)

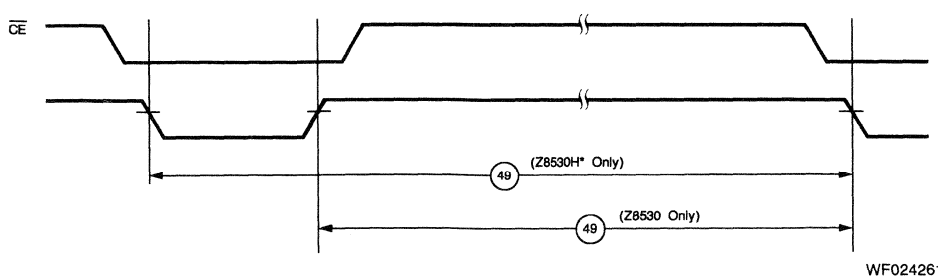


Figure 25. Cycle Timing (Z8530, Z8530H)

*Timings are preliminary and subject to change.

Z8031/Z8531

Asynchronous Serial Communications Controller (ASCC)

DISTINCTIVE CHARACTERISTICS

- **Two 0 to 2Mbps full duplex serial channels –** Each channel has independent oscillator, band-rate generator, and PLL for clock recovery, dramatically reducing the need for external components.
- **Programmable protocols –** NRZ, NRZI, and FM data encoding supported under program control.
- **Programmable Asynchronous Modes –** 5 to 8 bit characters with programmable stop bits, clock break detect, and error conditions.
- **Z8000* compatible –** The Z8031 interfaces directly to the Z8000 CPU bus and to the Z8000 interrupt structure.
- **Compatible with non-multiplexed bus –** The Z8531 interfaces easily to most other CPUs.

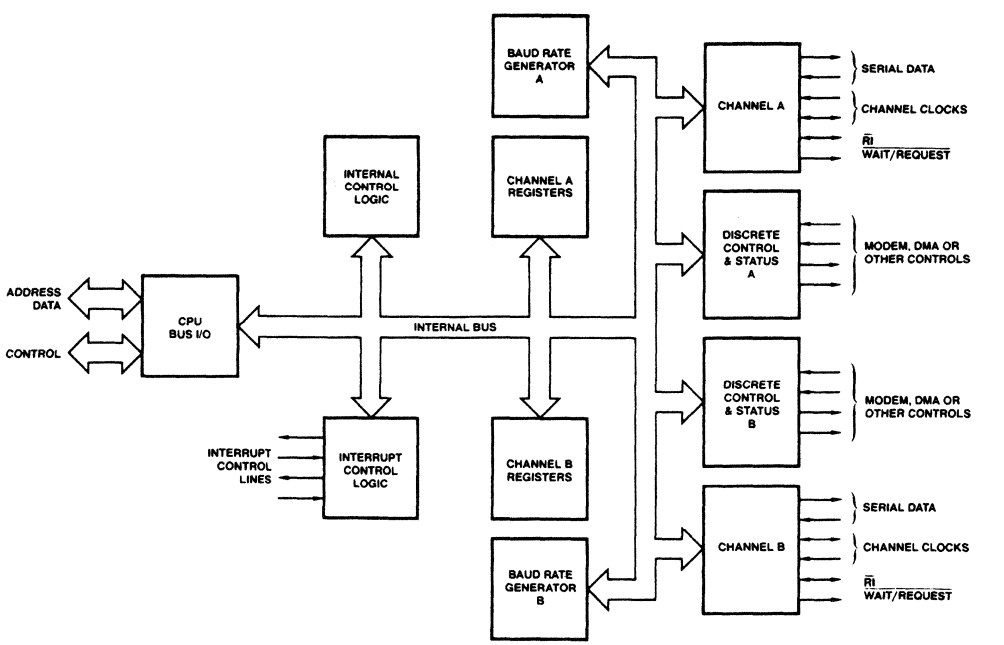
GENERAL DESCRIPTION

Asynchronous Serial Communications Controllers are dual-channel communications peripherals designed for use with 8- and 16-bit microprocessors. They function as serial-to-parallel, and parallel-to-serial converter/controllers, and contain a variety of new, sophisticated internal functions, including on-chip baud rate generators, digital phase-locked loops and crystal oscillators, to dramatically reduce the need for external circuitry.

Both channels have facilities for modem control; in cases where these controls aren't needed, they can be used for general purpose I/O.

The Z8031 is directly compatible with the Z8000 and 8086 CPUs, while the Z8531 is designed for non-multiplexed buses, and is easily interfaced with most other CPUs such as 8080, Z80, 6800, 68000 and MULTIBUS.**

BLOCK DIAGRAM



BD003260

Figure 1.

*Z8000 is a trademark of Zilog, Inc.
**MULTIBUS is a trademark of Intel Corporation.

Publication # 03818 | Rev. D | Amendment /0
Issue Date: April 1987

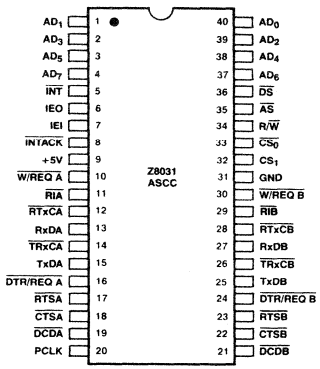
RELATED AMD PRODUCTS

Part No.	Description
Am79C12	212A Modem (1200BPS)
Am7960	Coded Data Transceiver
Am80186	Highly Integrated 16-Bit Microprocessor
Am80286	High Performance 16-Bit Microprocessor
Am8080A	8-Bit Microprocessor
Am9517A	DMA Controller

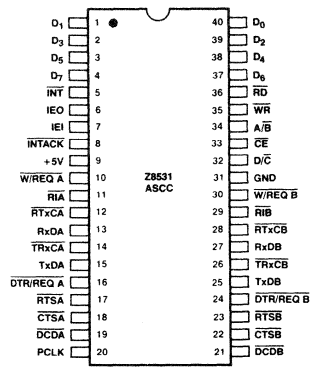
CONNECTION DIAGRAMS

Top View

DIPs



CD005080

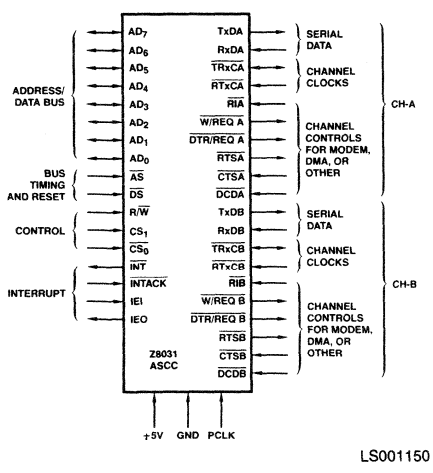


CD005090

Note: Pin 1 is marked for orientation.

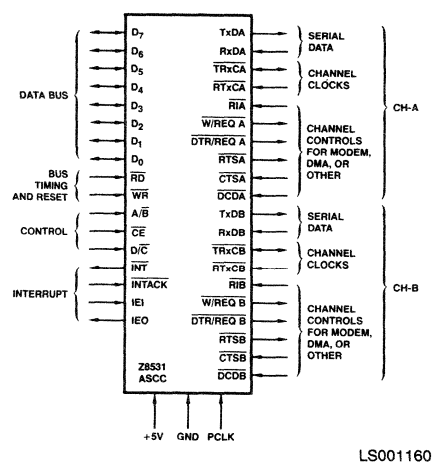
LOGIC SYMBOLS

Z8031



LS001150

Z8531



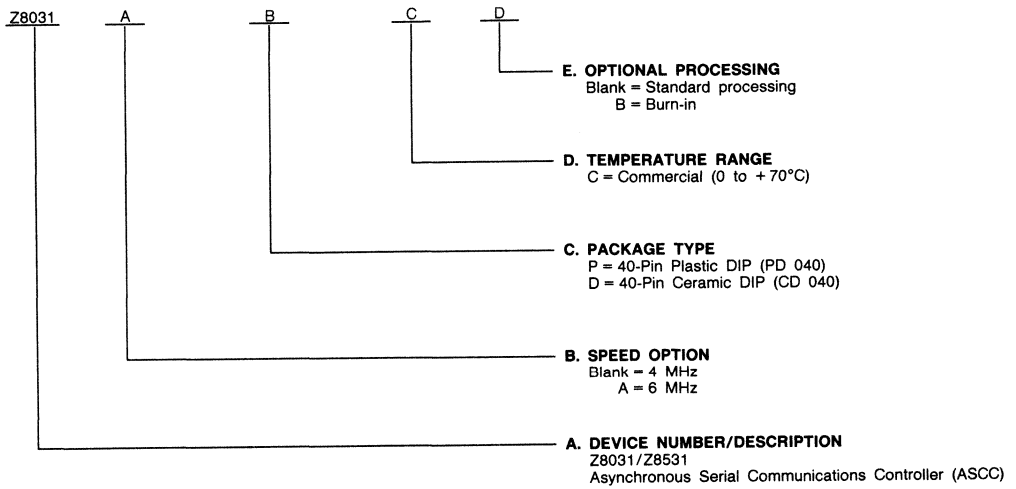
LS001160

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

A. Device Number
B. Speed Option
C. Package Type
D. Temperature Range
E. Optional Processing



Valid Combinations	
Z8031	PC, DC, DCB
Z8031A	
Z8531	
Z8531A	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Z8031

Pin No.	Name	I/O	Description
9	V_{CC}		+ 5V Power Supply
31	GND		Ground
1-4, 37-40	$AD_0 - AD_7$	I/O	Address/Data Bus (bidirectional, active High, three-state). These multiplexed lines carry register addresses to the ASCC as well as data or control information to and from the ASCC.
35	$\overline{AS}$	I	Address Strobe (active Low). Addresses on $AD_0 - AD_7$ are latched by the rising edge of this signal.
33	$\overline{CS_0}$	I	Chip Select 0 (active Low). This signal is latched concurrently with the addresses on $AD_0 - AD_7$ and must be active for the intended bus transaction to occur.
32	CS_1	I	Chip Select 1 (active High). This second select signal must also be active before the intended bus transaction can occur. CS_1 must remain active throughout the transaction.
18, 22	$\overline{CTSA}$, $\overline{CTSB}$	I	Clear to Send (active Low). If these pins are programmed as Auto Enables, a Low on these inputs enables their respective transmitter. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The ASCC detects puises on these inputs and can interrupt the CPU on both logic level transitions.
19, 21	$\overline{DCDA}$, $\overline{DCDB}$	I	Data Carrier Detect (active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise they may be used as general-purpose inputs pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The ASCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.
36	$\overline{DS}$	I	Data Strobe (active Low). This signal provides timing for the transfer of data into and out of the ASCC. If AS and DS coincide, this is interpreted as a reset.
16, 24	$\overline{DTR}/\overline{REQA}$, $\overline{DTR}/\overline{REQB}$	O	Data Terminal Ready/Request (active Low).These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request Lines for a DMA controller.
7	IEI	I	Interrupt Enable In (active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.
6	IEO	O	Interrupt Enable Out (active High). IEO is High only if IEI is High and the CPU is not servicing an ASCC interrupt or the ASCC is not requesting an interrupt (interrupt acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.
5	$\overline{INT}$	O	Interrupt Request (open-drain, active Low). This signal is activated when the ASCC requests an interrupt.
8	$\overline{INTACK}$	I	Interrupt Acknowledge (active Low). This signal indicates an active interrupt acknowledge cycle. During this cycle, the ASCC interrupt daisy chain settles. When $\overline{DS}$ becomes active, the ASCC places an interrupt vector on the data bus (if IEI is High). $\overline{INTACK}$ is latched by the rising edge of $\overline{AS}$.
20	PCLK	I	Clock. This is the master ASCC clock used to synchronize internal signals. PCLK is not required to have any phase relationship with the master system clock, although the frequency of this clock must be at least 90% of the CPU clock frequency for a Z8000. PCLK is a TTL level signal.
13, 27	RxDA, RxDB	I	Receive Data (active High). These input signals receive serial data at standard TTL levels.
12, 28	$\overline{RTxCA}$, $\overline{RTxCB}$	I	Receive/Transmit Clocks (active Low). These pins can be programmed in several different modes of operation. In each channel, RTxC may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock of the digital phase-locked loop. These pins can also be programmed for use with the respective $\overline{RI}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.
17, 23	$\overline{RTSA}$, $\overline{RTSB}$	O	Request to Send (active Low). When the Request to Send (RTS) bit in Write Register 5 (Figure 7) is set, the RTS signal goes Low. When the $\overline{RTS}$ bit is reset and Auto Enable is on, the signal goes High after the transmitter is empty. With Auto Enable off, the $\overline{RTS}$ pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.
34	$R/\overline{W}$	I	Read/Write. This signal specifies whether the operation to be performed is read or a write.
11, 29	$\overline{RIA}$, $\overline{RIB}$	I	Ring Indicator (active Low). These pins can act either as inputs or as part of the crystal oscillator circuit. In normal operation (crystal oscillator option not selected), these pins are inputs similar to $\overline{CTS}$ and $\overline{DCD}$. In this mode, transitions on these lines affect the state of the Ring Indicator status bits in Read Register 0 (Figure 6) but have no other function.
15, 25	TxDA, TxDB	O	Transmit/Receive Clocks (active Low). These output signals transmit serial data at standard TTL levels.
14, 26	$\overline{TRxCA}$, $\overline{TRxCB}$	I	Transmit/Receive Clocks (active Low). These pins can be programmed in several different modes of operation. $\overline{TRxC}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the digital phase-locked loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.
10, 30	$\overline{W}/\overline{REQA}$, $\overline{W}/\overline{REQB}$	O	Wait/Request (open-drain when programmed for a Wait function, driven High or Low when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the ASCC data rate. The reset state is Wait.

PIN DESCRIPTION (Cont'd.)

Z8531

Pin No.	Name	I/O	Description
9	V_{CC}		+ 5V Power Supply
31	GND		Ground
34	A/$\overline{B}$	I	Channel A/Channel B Select. This signal selects the channel in which the read or write operation occurs.
33	$\overline{CE}$	I	Chip Enable (active Low). This signals selects the ASCC for a read or write operation.
18, 22	$\overline{CTSA}$, $\overline{CTSB}$	I	Clear to Send (active Low). If these pins are programmed as Auto Enables, a Low on the inputs enables the respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The ASCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.
32	D/$\overline{C}$	I	Data/Control Select. This signal defines the type of information transferred to or from the ASCC. A High means data is transferred; a Low indicates a command.
19, 21	$\overline{DCDA}$, $\overline{DCDB}$	I	Data Carrier Detect (active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise they may be used as general-purpose inputs pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The ASCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.
1-4, 37-40	$D_0 - D_7$	I/O	Data Bus (bidirectional, three-state). These lines carry data and commands to and from the ASCC.
16, 24	$\overline{DTR}$/$\overline{REQA}$, $\overline{DTR}$/$\overline{REQB}$	O	Data Terminal Ready/Request (active Low). These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request Lines for a DMA controller.
7	IEI	I	Interrupt Enable In (active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.
6	IEO	O	Interrupt Enable Out (active High). IEO is High only if IEI is High and the CPU is not servicing an ASCC interrupt or the ASCC is not requesting an interrupt (interrupt acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.
5	$\overline{INT}$	O	Interrupt Request (open-drain, active Low). This signal is activated when the ASCC requests an interrupt.
8	$\overline{INTACK}$	I	Interrupt Acknowledge (active Low). This signal indicates an active interrupt acknowledge cycle. During this cycle, the ASCC interrupt daisy chain settles. When $\overline{RD}$ becomes active, the ASCC places an interrupt vector on the data bus (if IEI is High). $\overline{INTACK}$ is latched by the rising edge of PCLK.
20	PCLK	I	Clock. This is the master ASCC clock used to synchronize internal signals. PCLK is a TTL level signal.
36	$\overline{RD}$	I	Read (active Low). This signal indicates a read operation and when the ASCC is selected, enables the ASCC's bus drivers. During the interrupt acknowledge cycle, this signal gates the interrupt vector onto the bus if the ASCC is the highest priority device requesting an interrupt.
13, 27	RxDA, RxDB	I	Receive Data (active High). These input signals receive serial data at standard TTL levels.
12, 28	$\overline{RTxCA}$, $\overline{RTxCB}$	I	Receive/Transmit Clocks (active Low). These pins can be programmed in several different modes of operation. In each channel $\overline{RTxC}$ may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock of the digital phase-locked loop. These pins can also be programmed for use with the respective $\overline{RI}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.
17, 23	$\overline{RTSA}$, $\overline{RTSB}$	O	Request to Send (active Low). When the Request to Send (RTS) bit in Write Register 5 (Figure 7) is set, the $\overline{RTS}$ signal goes Low. When the RTS bit is reset in the asynchronous mode and Auto Enable is on, the signal goes High after the transmitter is empty. With Auto Enable off, the $\overline{RTS}$ pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.
15, 25	TxDA, TxDB	O	Transmit Data (active High). These output signals transmit serial data at standard TTL levels.
14, 26	$\overline{TRxCA}$, $\overline{TRxCB}$	I/O	Transmit/Receive Clocks (active Low). These pins can be programmed in several different modes of operation. $\overline{TRxC}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the digital phase-locked loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.
35	$\overline{WR}$	I	Write (active Low). When the ASCC is selected, this signal indicates a write operation. The coincidence of $\overline{RD}$ and $\overline{WR}$ is interpreted as a reset.
10, 30	$\overline{W}$/$\overline{REQA}$, $\overline{W}$/$\overline{REQB}$	O	Wait/Request (open-drain when programmed for a Wait function, driven High or Low when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the ASCC data rate. The reset state is Wait.

ARCHITECTURE

The ASCC internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to the Z8000 CPU (Z8031) or to a non-multiplexed CPU bus (Z8531). Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface to modems or other external devices (Figure 1).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modern control.

The register set for each channel includes ten control (write) registers, two synchronous character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel, a write-only Master Interrupt Control register and three read registers: one containing the vector with status information (Channel B only), one containing the vector without status (A only), and one containing the Interrupt Pending bits (A only).

The registers for each channel are designated as follows: WR0–WR15 – Write Registers 0 through 15. RR0–RR3, RR10, RR12, RR13, RR15 – Read Registers 0 through 3, 10, 12, 13, 15.

The following table lists the functions assigned to each read or write register. The ASCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

DATA PATH

The transmit and receive data path illustrated in Figure 2 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high-speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode (the character length in asynchronous modes also determines the data path).

The transmitter has an 8-bit transmit data buffer register loaded from the internal data bus and an 11-bit transmit shift register that can be loaded from the transmit data register. Depending on the operational mode, outgoing data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD).

TABLE 1. READ AND WRITE REGISTER FUNCTIONS

READ REGISTER FUNCTIONS	
RR0	Transmit/Receive buffer status and External status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only) Unmodified interrupt vector (Channel A only)
RR3	Interrupt Pending bits (Channel A only)
RR8	Receive buffer
RR10	Miscellaneous status
RR12	Lower byte of baud rate generator time constant
RR13	Upper byte of baud rate generator time constant
RR15	External/Status interrupt information

WRITE REGISTER FUNCTIONS	
WR0	CRC initialize, initialization commands for the various modes, shift right/shift left command
WR1	Transmit/Receive interrupt and data transfer mode definition
WR2	Interrupt vector (accessed through either channel)
WR3	Receive parameters and control
WR4	Transmit/Receive miscellaneous parameters and modes
WR5	Transmit parameters and controls
WR6	Sync characters or SDLC address field
WR7	Sync character or SDLC flag
WR8	Transmit buffer
WR9	Master interrupt control and reset (accessed through either channel)
WR10	Miscellaneous transmitter/receiver control bits
WR11	Clock mode control
WR12	Lower byte of baud rate generator time constant
WR13	Upper byte of baud rate generator time constant
WR14	Miscellaneous control bits
WR15	External/Status interrupt control

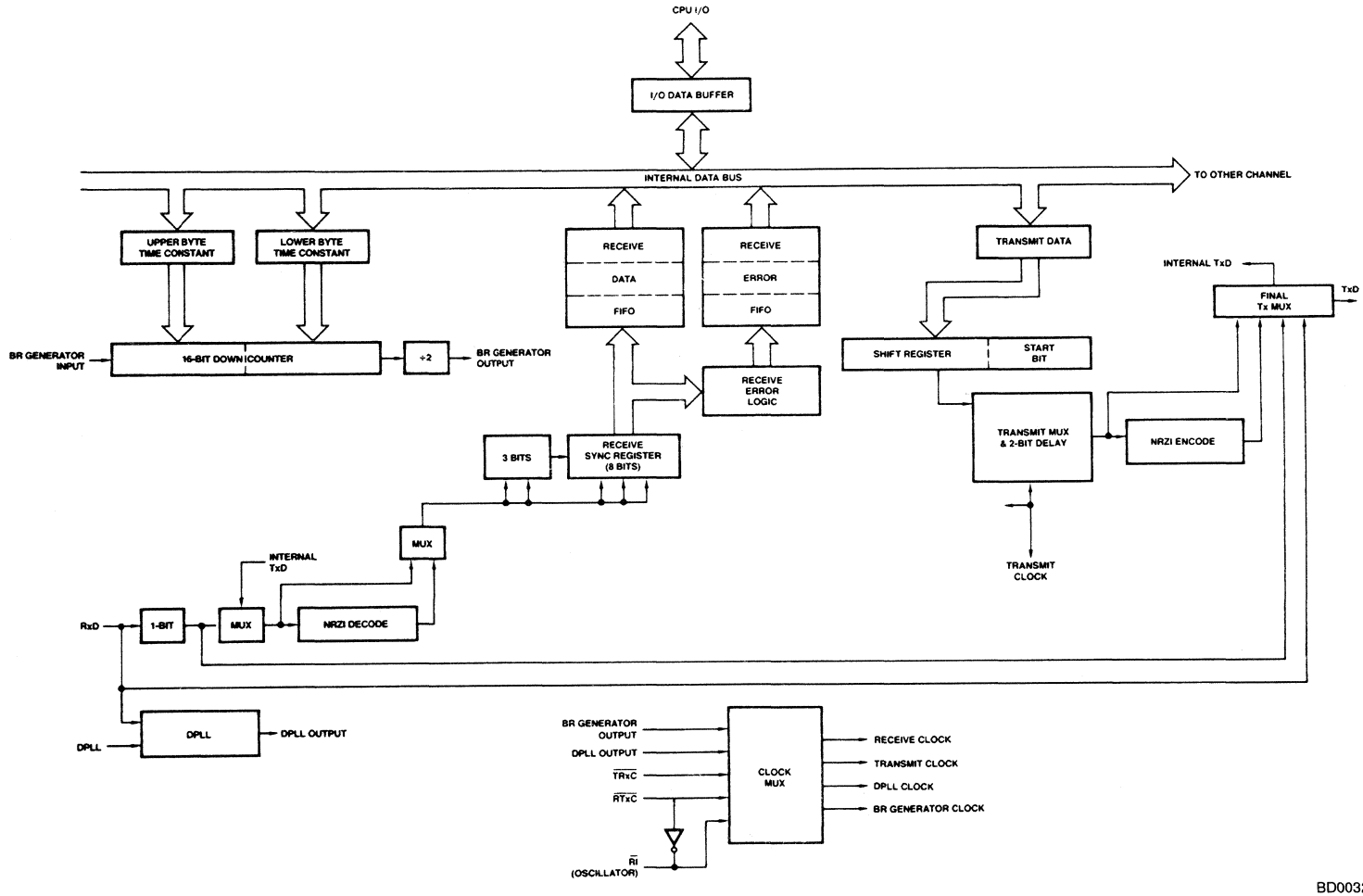


Figure 2. Data Path

DETAILED DESCRIPTION

The functional capabilities of the SCC can be described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, it interacts with the CPU and provides vectored interrupts and handshaking signals.

DATA COMMUNICATIONS CAPABILITIES

The ASCC provides two independent full-duplex channels programmable for use in any common asynchronous data-communication protocol. Figure 3 and the following description briefly detail this protocol.

Asynchronous Modes

Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in Figure 14). If the Low does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing or error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The ASCC does not require symmetric transmit and receive clock signals – a feature allowing use of the wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs.

BAUD RATE GENERATOR

Each channel in the ASCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On startup, the flip-flop on the output is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down. The output of the baud rate generator toggles upon reaching zero, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the digital phase-locked loop (see next section).

If the receive clock or transmit clock is not programmed to come from the $\overline{TRxC}$ pin, the output of the baud rate generator may be echoed out via the $\overline{TRxC}$ pin.

The following formula relates the time constant to the baud rate. (The baud rate is in bits/second and the BR clock period is in seconds.)

$$\text{baud rate} = \frac{1}{2\ (\text{time constant} + 2) \times (\text{BR clock period})}$$

Time-Constant Values for Standard Baud Rates at BR Clock = 3.9936MHz

Rate (Baud)	Time Constant (decimal notation)	Error
19200	102	–
9600	206	–
7200	275	0.12%
4800	414	–
3600	553	0.06%
2400	830	–
2000	996	0.04%
1800	1107	0.03%
1200	1662	–
600	3326	–
300	6654	–
150	13310	–
134.5	14844	0.0007%
110	18151	0.0015%
75	26622	–
50	39934	–

DIGITAL PHASE-LOCKED LOOP

The ASCC contains a digital phase-locked loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the ASCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32X clock to create nominal bit times. As the 32X clock is counted, the DPLL is searching the incoming data stream for edges (either 1/0 or 0/1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15/16 counting transition.

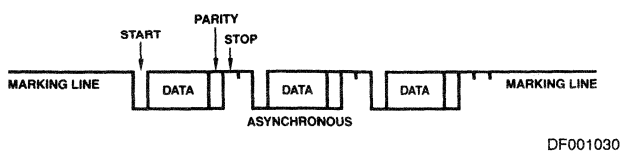


Figure 3. ASCC Protocols

The 32X clock for the DPLL can be programmed to come from either the $\overline{RTxC}$ input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the ASCC via the $\overline{TRxC}$ pin (if this pin is not being used as an input).

DATA ENCODING

The ASCC may be programmed to encode and decode the serial data in four different ways. In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. In FM_1 (more properly, bi-phase mark) a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by no additional transition at the center of the bit cell. In FM_0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the ASCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0/1, the bit is a 0. If the transition is 1/0, the bit is a 1.

AUTO ECHO AND LOCAL LOOPBACK

The ASCC is capable of automatically echoing everything it receives. This feature is useful mainly in asynchronous modes, but works in synchronous and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In Auto Echo mode, the $\overline{CTS}$ input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed and the programmer is responsible for disabling transmitter interrupts and $\overline{WAIT/REQUEST}$ on transmit.

The ASCC is also capable of local loopback. In this mode, TxD is RxD just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data and RxD is ignored (except to be echoed out via TxD). The CTS and DCD inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works with NRZ, NRZI or FM coding of the data stream.

I/O INTERFACE CAPABILITIES

The ASCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

POLLING

All interrupts are disabled. Three status registers in the ASCC are automatically updated whenever any function is performed. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

INTERRUPTS

When an ASCC responds to an Interrupt Acknowledge signal ($\overline{INTACK}$) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures 6 and 7).

To speed interrupt response time, the ASCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the ASCC (Transmit, Receive and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write-only.

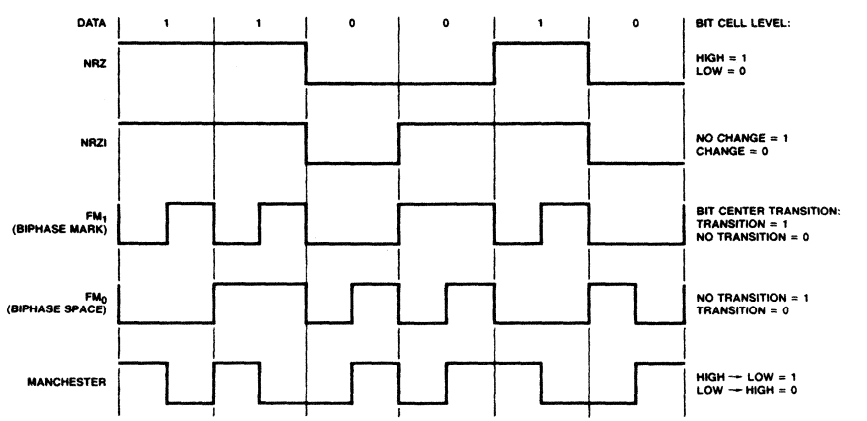


Figure 4. Data Encoding Methods

The other two bits are related to the Z-Bus interrupt priority chain (Figure 5). As a Z-Bus peripheral, the ASCC may request an interrupt only when no higher-priority device is requesting one, e.g., when IEI is High. If the device in question requests an interrupt, it pulls down $\overline{INT}$. The CPU then responds with $\overline{INTACK}$, and the interrupting device places the vector on the A/D bus.

In the ASCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is High, the INT output is pulled Low, requesting an interrupt. In the ASCC, if the IE bit is not set by enabling interrupts, then the IP for that source can never be set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the ASCC and external to the ASCC are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the ASCC being pulled Low and propagated to subsequent peripherals. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts.

There are three types of interrupts: Transmit, Receive and External/Status interrupts. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receiver, Transmit and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the receiver can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive condition.
- Interrupt on all Receive Characters or Special Receive condition.
- Interrupt on Special Receive condition only.

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in asynchronous mode and, optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only in the status placed in the vector during the Interrupt-Acknowledge cycle. In Interrupt on first Receive Character, an interrupt can occur from Special Receive conditions any time after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{CTS}$, $\overline{DCD}$, and $\overline{RI}$ pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition, or a zero count in the baud rate generator, or by the detection of a Break (asynchronous mode).

CPU/DMA BLOCK TRANSFER

The ASCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{WAIT/REQUEST}$ output in conjunction with the Wait/Request bits in WR1. The $\overline{WAIT/REQUEST}$ output can be defined under software control as a $\overline{WAIT}$ line in the CPU Block Transfer mode or as a $\overline{REQUEST}$ line in the DMA Block Transfer mode.

To a DMA controller, the ASCC $\overline{REQUEST}$ output indicates that the ASCC is ready to transfer data to or from memory. To the CPU, the $\overline{WAIT}$ line indicates that the ASCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The $\overline{DTR/REQUEST}$ line allows full-duplex operation under DMA control.

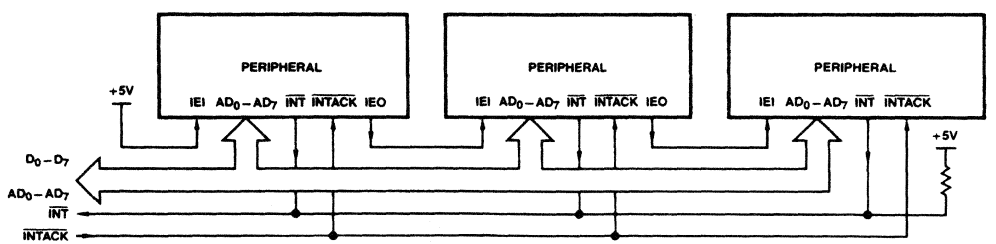


Figure 5. Z-Bus Interrupt Schedule

PROGRAMMING INFORMATION (Z8031)

The Z8031 contains 11 write registers in each channel that are programmed by the system separately to configure the functional personality of the channels.

All of the registers in the Z8031 are directly addressable. How the Z8031 decodes the address placed on the address/data bus at the beginning of a Read or Write cycle is controlled by a command issued in WR0B. In the shift right mode, the channel select $\overline{A/B}$ is taken from AD_0 and the state of AD_5 is ignored. In the shift left mode, $\overline{A/B}$ is taken from AD_5 and the state of AD_0 is ignored. AD_7 and AD_6 are always ignored as address bits and the register address itself occupies AD_4–AD_1.

The system program first issues a series of commands to initialize the basic mode of operation. For example, the character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set, and finally, receiver or transmitter enable.

PROGRAMMING INFORMATION (Z8531)

The Z8531, register addressing is direct for the data registers only, which are selected by a High on the D/$\overline{C}$ pin. In all other cases (with the exception of WR0 and RR0), programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to WR0 and contains three bits that point to the selected register. The second write is the actual control word

for the selected register, and if the second operation is read, the selected read register is accessed. All of the registers in the Z8531, including the data registers, may be accessed in this fashion. The pointer bits are automatically cleared after the read or write operation so that WR0 (or RR0) is addressed again.

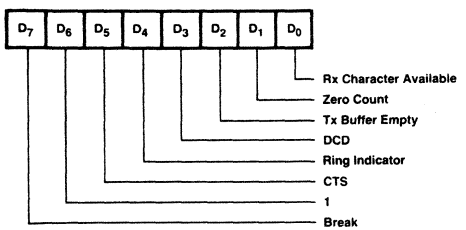
The system program first issues a series of commands to initialize the basic mode of operation. For example, the character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set, and finally, receiver or transmitter enable.

READ REGISTERS

The ASCC contains 8 read registers (actually 9, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10 and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). Figure 6 shows the formats for each read register.

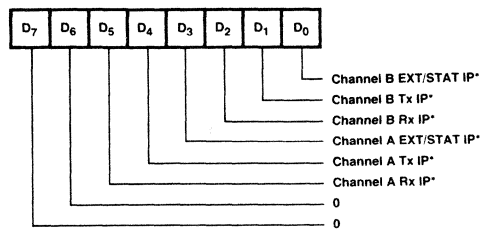
The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring; e.g., when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).

Read Register 0



DF001040

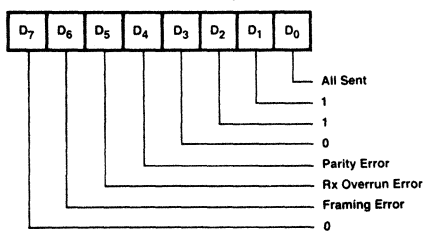
Read Register 3



DF001070

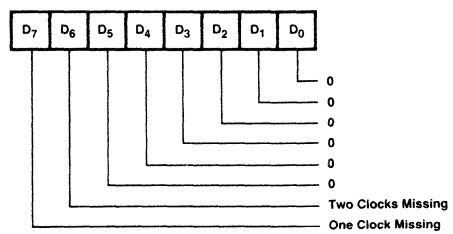
*Always 0 in B Channel

Read Register 1



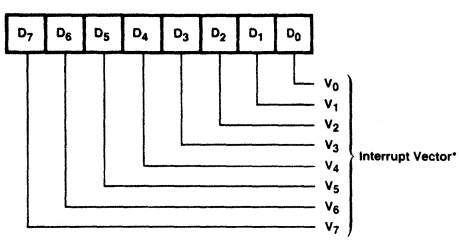
DF001050

Read Register 10



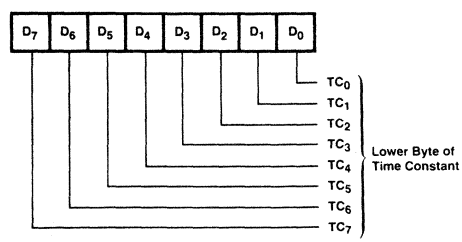
DF001080

Read Register 2



DF001060

Read Register 12

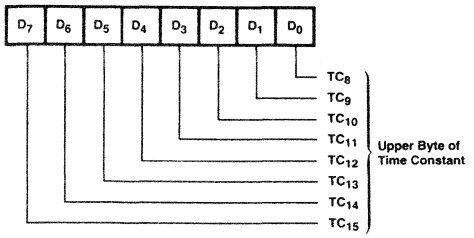


DF001090

*Modified in B Channel

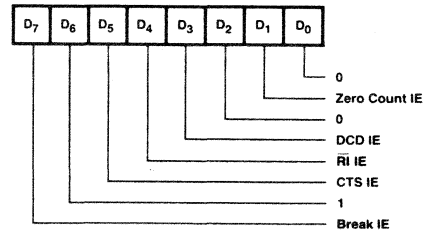
Figure 6. Read Register Bit Functions

Read Register 13



DF003380

Read Register 15



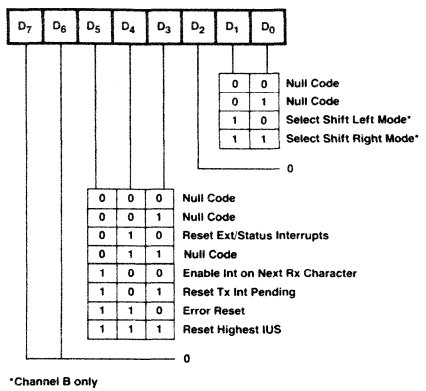
DF001100

Figure 6. Read Register Bit Functions (Cont.)

WRITE REGISTERS

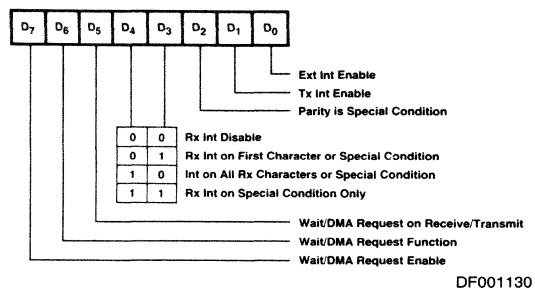
The ASCC contains 11 write registers (12 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. In addition, there are two registers (WR2 and WR9) shared by the two channels that may be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Figure 7 shows the format of each write register.

Write Register 0 (Z8031)



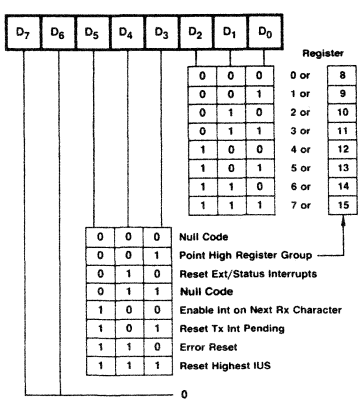
DF001110

Write Register 1



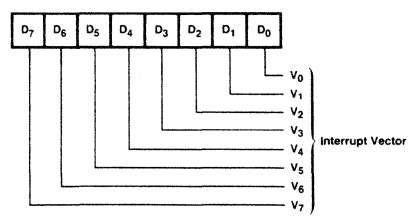
DF001130

Write Register 0 (Z8531)



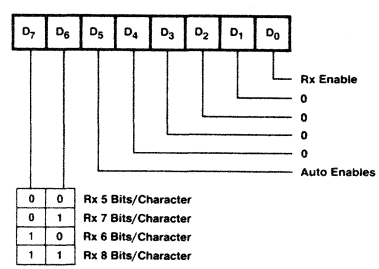
DF001120

Write Register 2



DF001140

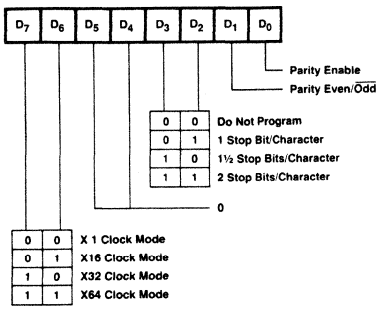
Write Register 3



DF001150

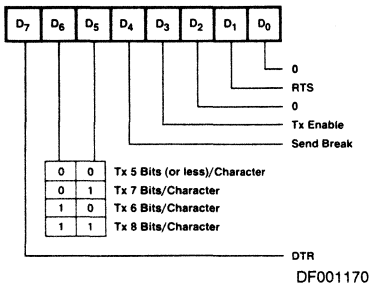
Figure 7. Write Register Bit Functions

Write Register 4



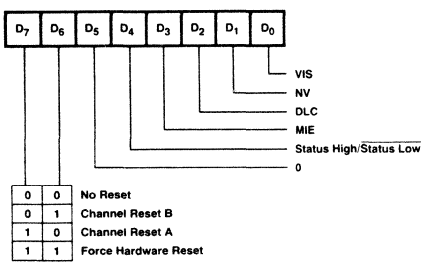
DF001160

Write Register 5



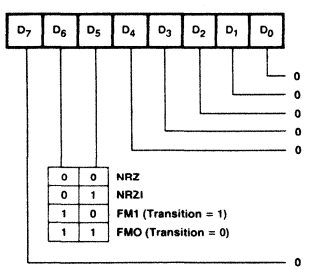
DF001170

Write Register 9



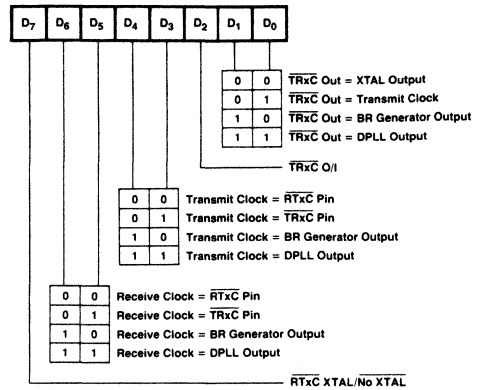
DF001180

Write Register 10



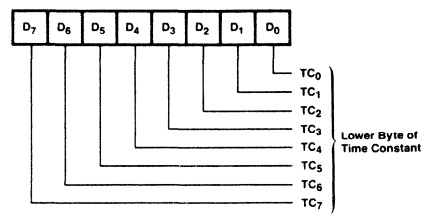
DF001190

Write Register 11



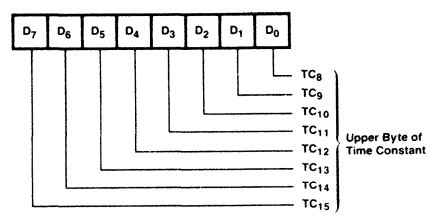
DF001200

Write Register 12



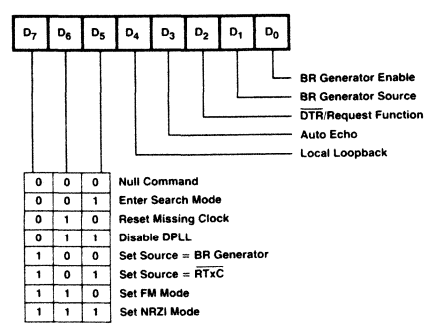
DF001210

Write Register 13



DF001220

Write Register 14



DF001230

Figure 7. Write Register Bit Functions (Cont.)

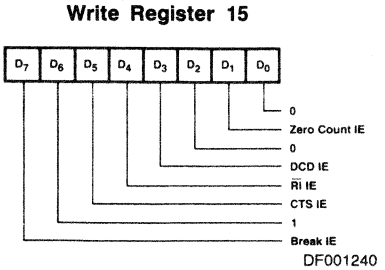


Figure 7. Write Register Bit Functions (Cont.)

Z8031 TIMING

The ASCC generates internal control signals from $\overline{AS}$ and $\overline{DS}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{AS}$ and $\overline{DS}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the ASCC to the falling edge of $\overline{DS}$ in the second transaction involving the ASCC. This time must be at least 6 PCLK cycles plus 200ns.

READ CYCLE TIMING

Figure 8 illustrates read cycle timing. The address on AD_0–AD_7 and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. R/W must be High to indicate a read cycle. CS_1 must also be High for the read cycle to occur. The data bus drivers in the ASCC are then enabled while $\overline{DS}$ is Low.

WRITE CYCLE TIMING

Figure 9 illustrates write cycle timing. The address on AD_0–AD_7 and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. R/$\overline{W}$ must be Low to indicate a write cycle. CS_1 must be High for the write cycle to occur. $\overline{DS}$ Low strobes the data into the ASCC.

INTERRUPT ACKNOWLEDGE CYCLE TIMING

Figure 10 illustrates interrupt acknowledge cycle timing. The address on AD_0–AD_7 and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. However, if $\overline{INTACK}$ is Low, the address and $\overline{CS_0}$ are ignored. The state of R/$\overline{W}$ and CS_1 are also ignored for the duration of the interrupt acknowledge cycle. Between the rising edge of $\overline{AS}$ and the falling edge of $\overline{DS}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the ASCC and IEI is High when DS falls, the acknowledge cycle was intended for the ASCC. In this case, the ASCC may be programmed to respond to $\overline{DS}$ Low by placing its interrupt vector on AD_0–AD_7. It then sets the appropriate interrupt-under-service latch internally.

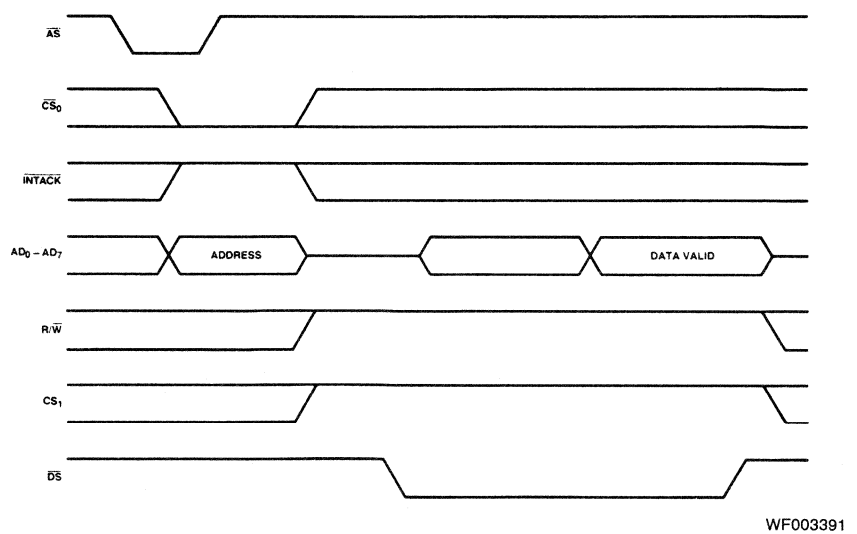


Figure 8. Z8031 Read Cycle Timing

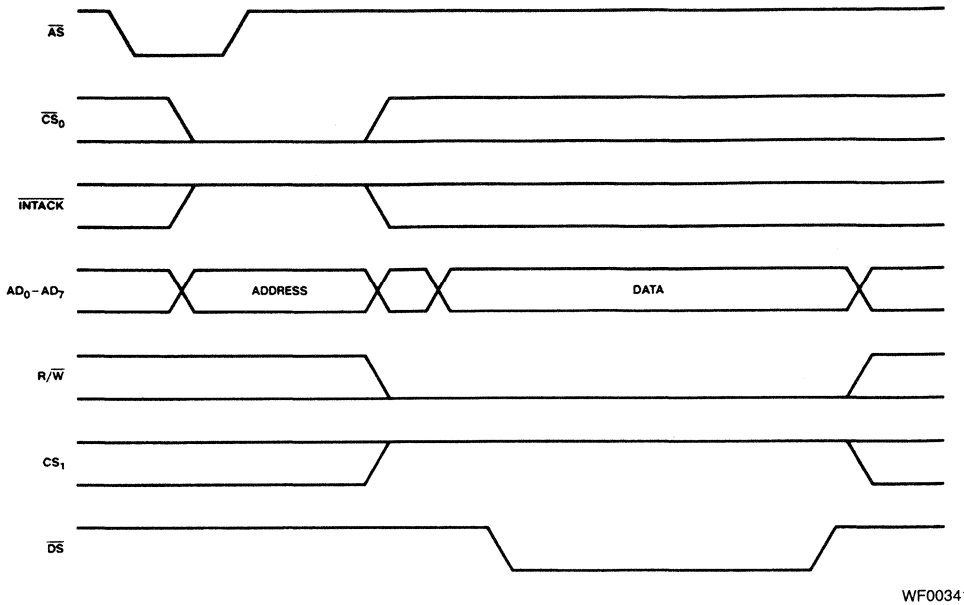


Figure 9. Z8031 Write Cycle Timing

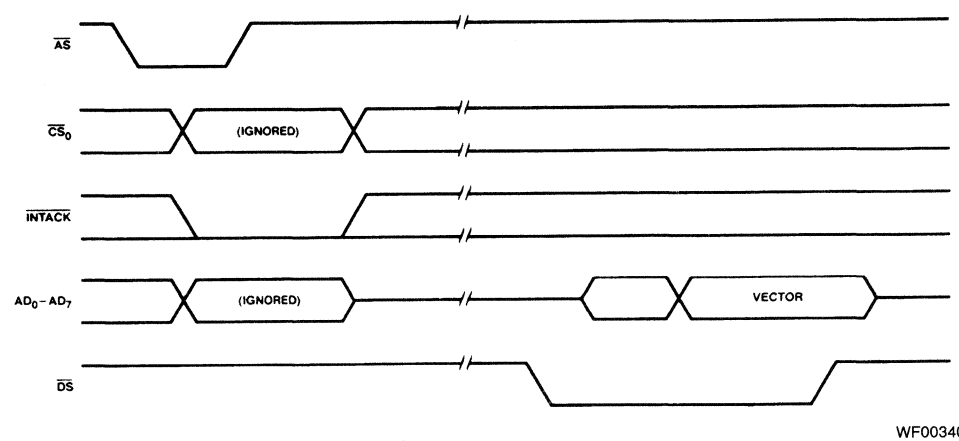


Figure 10. Z8031 Interrupt Acknowledge Cycle Timing

Z8531 TIMING

The ASCC generates internal control signals from $\overline{WR}$ and $\overline{RD}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{WR}$ and $\overline{RD}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the ASCC. The recovery time required for proper operation is specified from the rising edge of $\overline{WR}$ or $\overline{RD}$ in the first transaction involving the ASCC to the falling edge of $\overline{WR}$ or $\overline{RD}$ in the second transaction involving the ASCC. This time must be at least 6 PCLK cycles plus 200ns.

READ CYCLE TIMING

Figure 11 illustrates read cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{RD}$ falls or if it rises before $\overline{RD}$ rises, the effective $\overline{RD}$ is shortened.

WRITE CYCLE TIMING

Figure 12 illustrates write cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{WR}$ falls or if it rises before $\overline{WR}$ rises, the effective $\overline{WR}$ is shortened.

INTERRUPT ACKNOWLEDGE CYCLE TIMING

Figure 13 illustrates interrupt acknowledge cycle timing. Between the time $\overline{INTACK}$ goes Low and the falling edge of $\overline{RD}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the ASCC and IEI is High when $\overline{RD}$ falls, the acknowledge cycle is intended for the ASCC. In this case, the ASCC may be programmed to respond to $\overline{RD}$ Low by placing its interrupt vector on D_0–D_7; it then sets the appropriate interrupt-under-service latch internally.

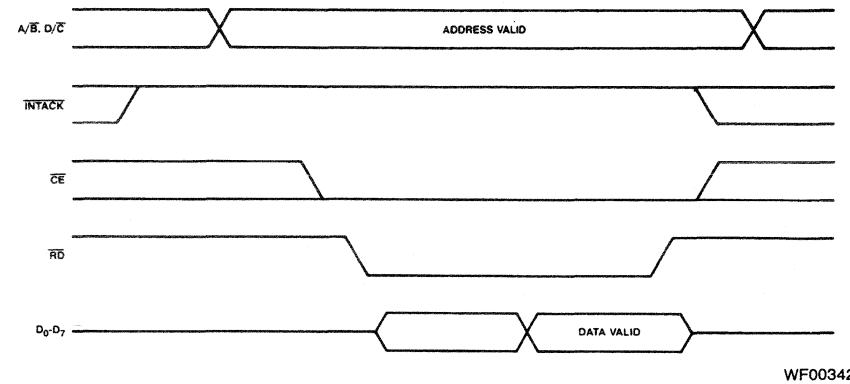


Figure 11. Z8531 Read Cycle Timing

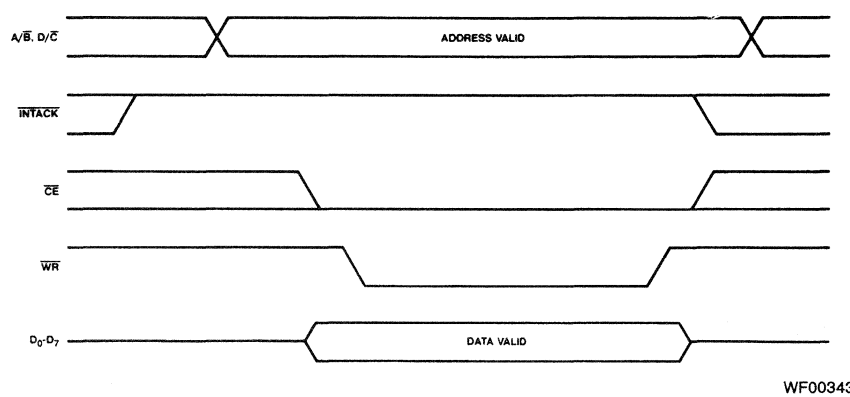


Figure 12. Z8531 Write Cycle Timing

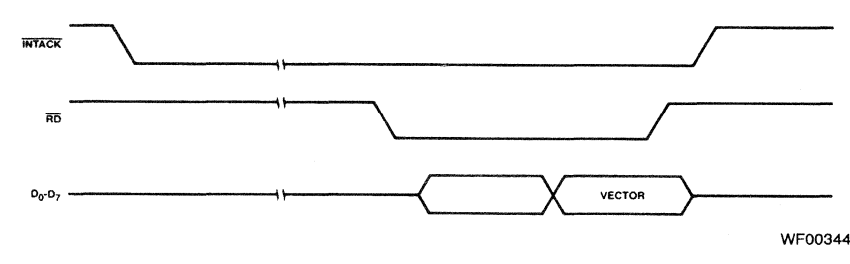


Figure 13. Z8531 Interrupt Acknowledge Cycle Timing

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage at any Pin Relative to V_{SS} ..... −0.5V to +7.0V
Power Dissipation 1.8W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to 70°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

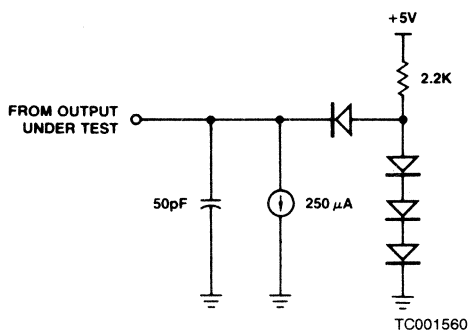
Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

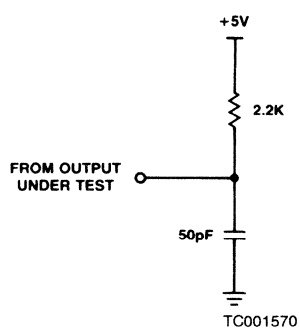
Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{IH}	Input HIGH Voltage		2.0		V_{CC} + 0.3	V
V_{IL}	Input LOW Voltage		−0.3		0.8	V
V_{OH}	Output HIGH Voltage	I_{OH} = −250μA	2.4			V
V_{OL}	Output LOW Voltage	I_{OL} = +2.0mA			0.4	V
I_{IL}	Input Leakage	$V_{SS} \leq V_{IN} + V_{CC}$			±10.0	μA
I_{OL}	Output Leakage	$V_{SS} \leq V_{OUT} + V_{CC}$			±10.0	μA
I_{CC}	V_{CC} Supply Current				250	mA
C_{IN}	Input Capacitance	Unmeasured pins returned to ground. f = 1 MHz at T_A = 25°C.			10	pF
C_{OUT}	Output Capacitance				15	pF
$C_{I/O}$	Bidirectional Capacitance				20	pF

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

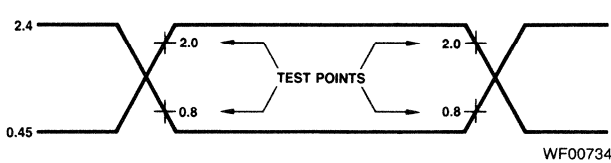


A. Standard Test Load



B. Open-Drain Test Load

SWITCHING TEST WAVEFORM



AC testing inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0." Timing measurements are made at 2.0 V for a logic "1" and 0.8 V for a logic "0."

See Section 6 for Thermal Characteristics Information.

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified

Z8531 System Timing

Number	Parameters	Description	4MHz Min	4MHz Max	6MHz Min	6MHz Max	Units
1	TdRXC(REQ)	$\overline{RxC}$ ↑ to $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	TcPC
2	TdRXC(W)	$\overline{RxC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	8	14	TcPC
3	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	TcPC
4	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	TcPC
5	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	TcPC
6	TdTXC(DRQ)	$\overline{TxC}$ ↓ $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	TcPC
7	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	TcPC
8	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPC

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.

Z8031 System Timing

Number	Parameters	Description	4MHz Min	4MHz Max	6MHz Min	6MHz Max	Units
1	TdRXC(REQ)	$\overline{RxC}$ ↑ to $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	TcPC
2	TdRXC(W)	$\overline{RxC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	8	14	TcPC
3	TdRXC(INT)	$\overline{RxC}$ ↑ $\overline{INT}$ Valid Delay (Notes 1, 2)	8 +2	12 +3	8 +2	12 +3	TcPC $\overline{AS}$
4	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	TcPC
5	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	TcPC
6	TdTXC(DRQ)	$\overline{TxC}$ ↓ $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	TcPC
7	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	4 +2	6 +3	4 +2	6 +3	TcPC $\overline{AS}$
8	TdEXT(INT)	$\overline{DCD}$, $\overline{RI}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	3	2	3	$\overline{AS}$

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.

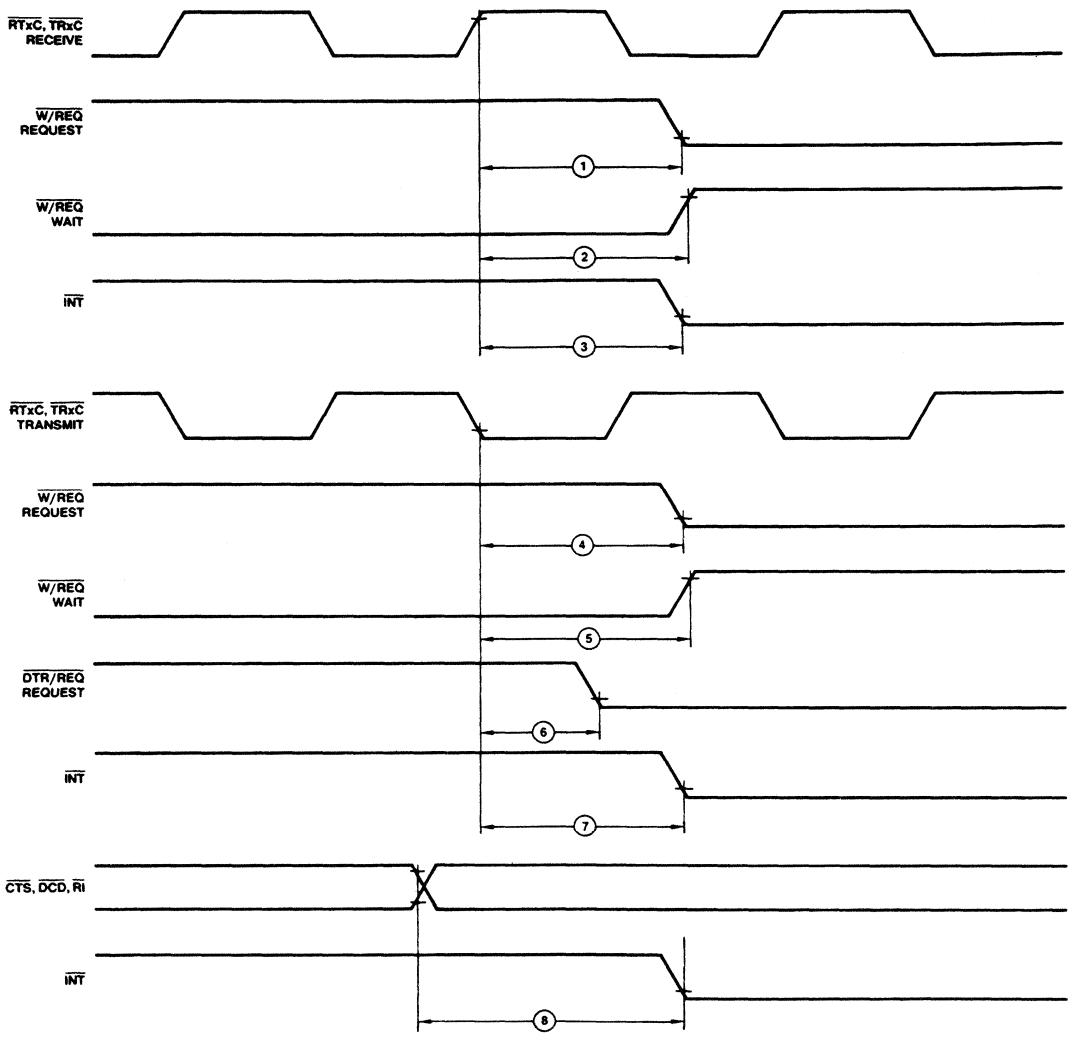


Figure 14. System Timing

SWITCHING CHARACTERISTICS (Cont'd.)

General Timing (See Figure 15)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TdPC(REQ)	PCLK ↓ to W/$\overline{REQ}$ Valid Delay		250		250	ns
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		350		350	ns
3	TsRXC(PC)	$\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4)	80	TwPCL	70	TwPCL	ns
4	TsRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Setup Time (X1 Mode) (Note 1)	0		0		ns
5	ThRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Hold Time (X1 Mode) (Note 1)	150		150		ns
6	TsRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Setup Time (X1 Mode) (Notes 1, 5)	0		0		ns
7	ThRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Hold Time (X1 Mode) (Notes 1, 5)	150		150		ns
8	TsTXC(PC)	$\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4)	0		0		ns
9	TdTXCf(TXD)	$\overline{TxC}$ ↓ to TxD Delay (X1 Mode) (Note 2)		300		230	ns
10	TdTXCr(TXD)	$\overline{TxC}$ ↑ to TxD Delay (X1 Mode) (Notes 2, 5)		300		230	ns
11	TdTXD(TRX)	TxD to $\overline{TRxC}$ Delay (Send Clock Echo)		200		200	ns
12	TwRTXh	$\overline{RTxC}$ High Width (Note 6)	180		180		ns
13	TwRTXl	$\overline{RTxC}$ Low Width (Note 6)	180		180		ns
14	TcRTX	$\overline{RTxC}$ Cycle Time (Notes 6, 7)	1000		660		ns
15	TcRTXX	Crystal Oscillator Period (Note 3)	250	1000	165	1000	ns
16	TwTRXh	$\overline{TRxC}$ High Width (Note 6)	180		180		ns
17	TwTRXl	$\overline{TRxC}$ Low Width (Note 6)	180		180		ns
18	TcTRX	$\overline{TRxC}$ Cycle Time (Notes 6, 7)	1000		660		ns
19	TwEXT	$\overline{DCD}$ or $\overline{CTS}$ Pulse Width	200		200		ns

Notes:
1. RxC is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. TxC is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{RI}$ have 30pF capacitors to the ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{RxC}$ and $\overline{PCLK}$ or $\overline{TxC}$ and $\overline{PCLK}$ is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only to transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.

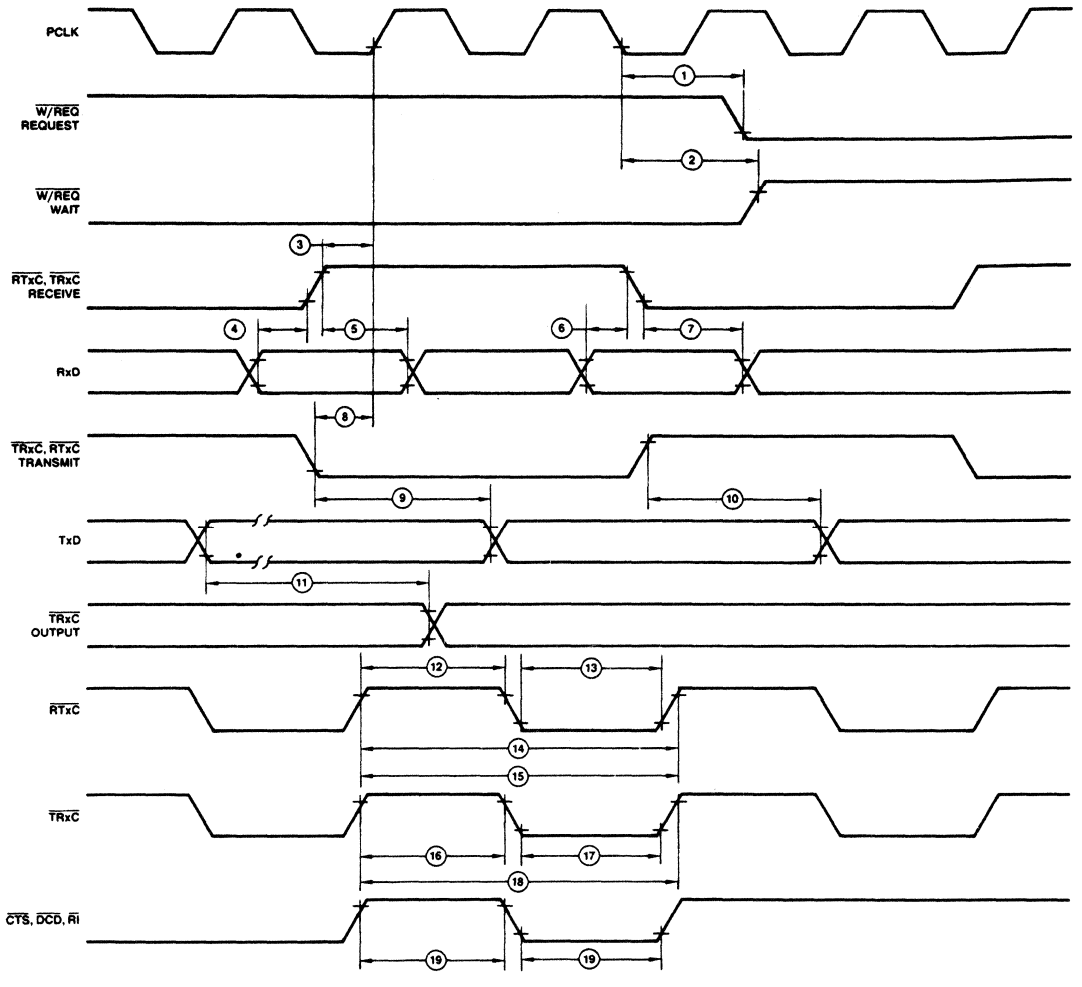


Figure 15. General Timing

SWITCHING CHARACTERISTICS (Cont'd.)

Z8031 Read and Write Timing (see Figure 16)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TwAS	$\overline{AS}$ LOW Width	70		50		ns
2	TdDS(AS)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	50		25		ns
3	TsCS0(AS)	$\overline{CS_0}$ to $\overline{AS}$ ↑ Setup Time (Note 1)	0		0		ns
4	ThCS0(AS)	$\overline{CS_0}$ to $\overline{AS}$ ↑ Hold Time (Note 1)	60		40		ns
5	TsCS1(DS)	CS_1 to $\overline{DS}$ ↓ Setup Time (Note 1)	100		80		ns
6	ThCS1(DS)	CS_1 to $\overline{DS}$ ↑ Hold Time (Note 1)	55		40		ns
7	TsIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Setup Time	10		10		ns
8	ThIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Hold Time	250		200		ns
9	TsRWR(DS)	R/$\overline{W}$ (Read) to $\overline{DS}$ ↓ Setup Time	100		80		ns
10	ThRW(DS)	R/$\overline{W}$ to $\overline{DS}$ ↑ Hold Time	55		40		ns
11	TsRWW(DS)	R/$\overline{W}$ (Write) to $\overline{DS}$ ↓ Setup Time	0		0		ns
12	TdAS(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay	60		40		ns
13	TwDSl	$\overline{DS}$ LOW Width	390		250		ns
14	TrC	Valid Access Recovery Time (Note 2)	6TcPC + 200		6TcPC + 130		ns
15	TsA(AS)	Address to $\overline{AS}$ ↑ Setup Time (Note 1)	30		10		ns
16	ThA(AS)	Address to $\overline{AS}$ ↑ Hold Time (Note 1)	50		30		ns
17	TsDW(DS)	Write Data to $\overline{DS}$ ↓ Setup Time	30		20		ns
18	ThDW(DS)	Write Data to $\overline{DS}$ ↑ Hold Time	30		20		ns
19	TdDS(DA)	$\overline{DS}$ ↓ to Data Active Delay	0		0		ns
20	TdDSr(DR)	$\overline{DS}$ ↑ to Read Data Not Valid Delay	0		0		ns
21	TdDSf(DR)	$\overline{DS}$ ↓ to Read Data Valid Delay		250		180	
22	TdAS(DR)	$\overline{AS}$ ↑ to Read Data Valid Delay		520		335	ns

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Parameter applies only between transactions involving the SCC.

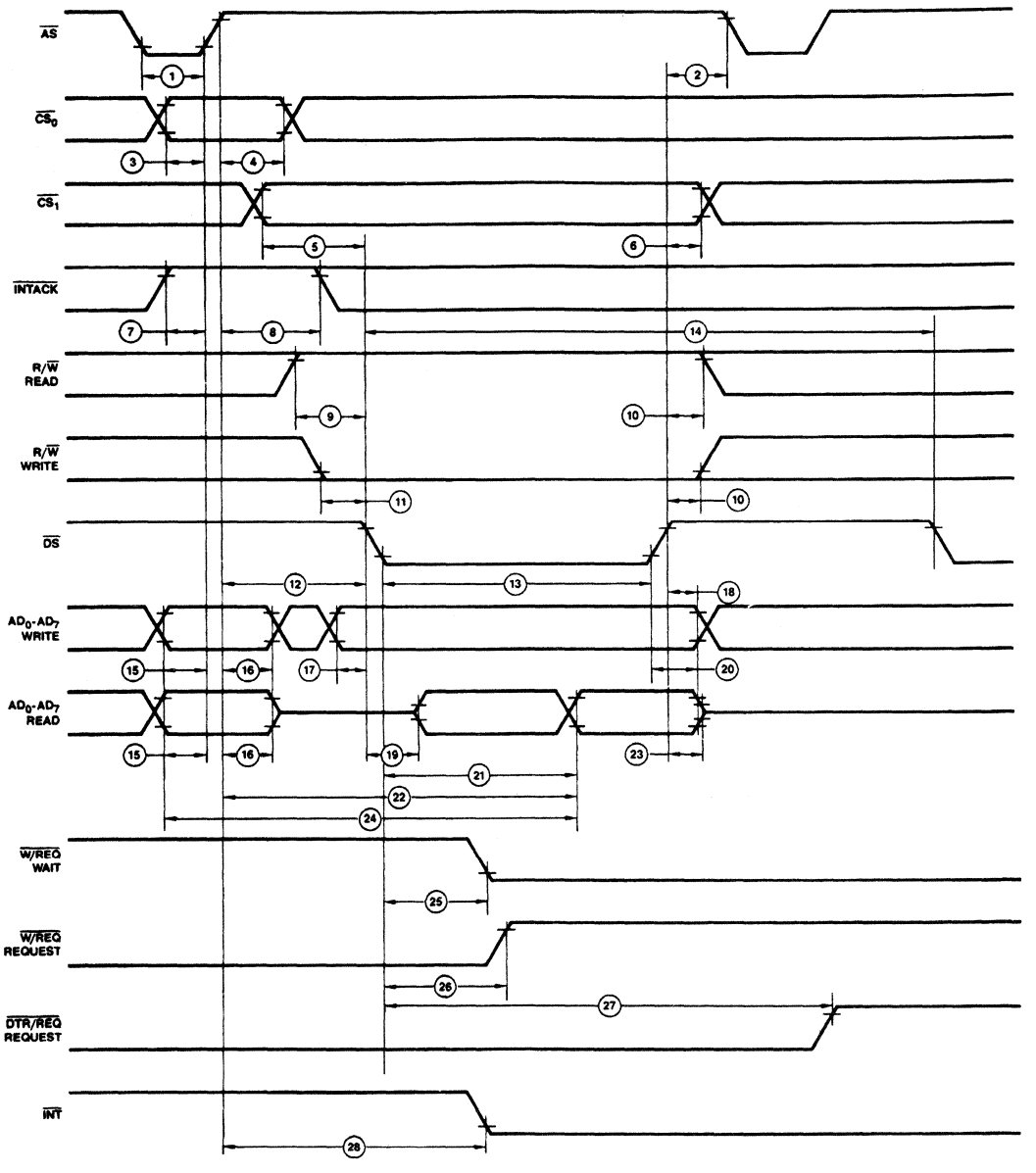


Figure 16. Z8031 Read and Write Timing

SWITCHING CHARACTERISTICS (Cont'd.)

Z8531 Read and Write Timing (see Figure 20)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TwPCl	PCLK Low Width	105	2000	70	1000	ns
2	TwPCh	PCLK High Width	105	2000	70	1000	ns
3	TfPC	PCLK Fall Time		20		10	ns
4	TrPC	PCLK Rise Time		20		10	ns
5	TcPC	PCLK Cycle Time	250	4000	165	2000	ns
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	80		80		ns
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		ns
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	80		80		ns
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		ns
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	10		10		ns
11	TsIAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1)	200		160		ns
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
13	TsIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time (Note 1)	200		160		ns
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		ns
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	100		100		ns
16	TsCEl(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time	0		0		ns
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time	100		70		ns
19	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time (Note 1)	0		0		ns
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note 1)	0		0		ns
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time (Note 1)	100		70		ns
22	TwRDl	$\overline{RD}$ Low Width (Note 1)	240		200		ns
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		ns
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		ns
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		250		180	ns
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		70		45	ns

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time required for a ±0.5V change in the output with a maximum D.C. load and minimum A.C. load.

5

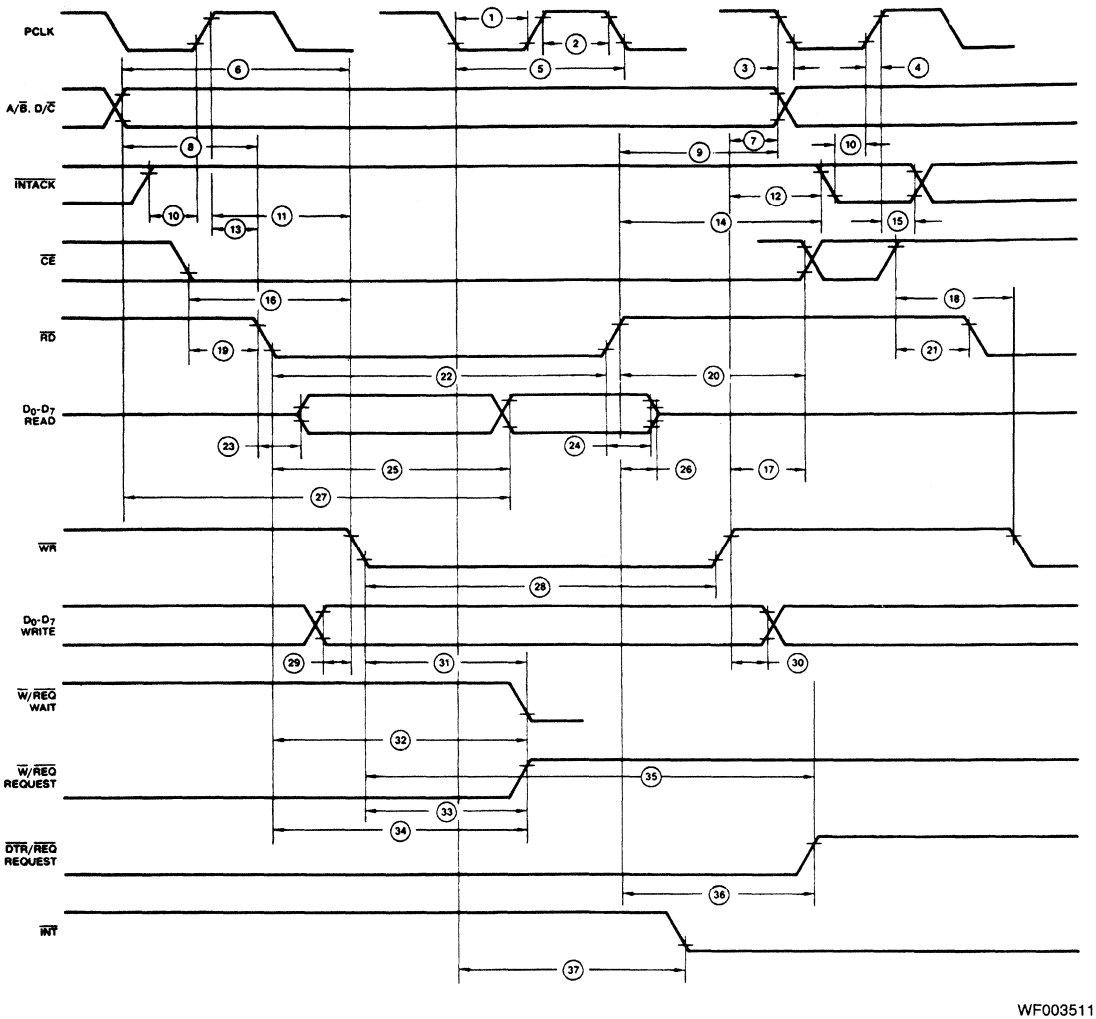


Figure 20. Z8531 Read and Write Timing

SWITCHING CHARACTERISTICS (Cont'd.)

Z8031 Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 17, 18, 19)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
23	TdDS(DRz)	$\overline{DS}$ ↑ to Read Data Float Delay (Note 3)		70		45	ns
24	TdA(DR)	Address Required Valid to Read Data Valid Delay		570		310	ns
25	TdDS(W)	$\overline{DS}$ ↓ to Wait Valid Delay (Note 4)		240		200	ns
26	TdDSf(REQ)	$\overline{DS}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200	ns
27	TdDSr(REQ)	$\overline{DS}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		5TcPC +300		5TcPC +250	ns
28	TdAS(INT)	$\overline{AS}$ ↑ to $\overline{INT}$ Valid Delay (Note 4)		500		500	ns
29	TdAS(DSA)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ (Acknowledge) Delay (Note 5)	250		250		ns
30	TwDSA	$\overline{DS}$ (Acknowledge) Low Width	390		250		ns
31	TdDSA(DR)	$\overline{DS}$ ↓ (Acknowledge) to Read Data Valid Delay		250		180	ns
32	TsIEI(DSA)	IEI to $\overline{DS}$ ↓ (Acknowledge) Setup Time	120		100		ns
33	ThIEI(DSA)	IEI to $\overline{DS}$ ↑ (Acknowledge) Hold Time	0		0		ns
34	TdIEI(IEO)	IEI to IEO Delay		120		100	ns
35	TdAS(IEO)	$\overline{AS}$ ↑ to IEO Delay (Note 6)		250		250	ns
36	TdDSA(INT)	$\overline{DS}$ ↓ (Acknowledge) to $\overline{INT}$ Inactive Delay (Note 4)		500		500	ns
37	TdDS(ASQ)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay for No Reset	30		15		ns
38	TdASQ(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay for No Reset	30		30		ns
39	TwRES	$\overline{AS}$ and $\overline{DS}$ Coincident Low for Reset (Note 7)	250		200		ns
40	TwPCl	PCLK Low Width	105	2000	70	1000	ns
41	TwPCh	PCLK High Width	105	2000	70	1000	ns
42	TcPC	PCLK Cycle Time	250	4000	165	2000	ns
43	TrPC	PCLK Rise Time		20		10	ns
44	TfPC	PCLK Fall Time		20		10	ns

Notes: 3. Float delay is defined as the time required for a ±0.5V change in the output with a maximum D.C. load and minimum A.C. load.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any Z-SCC in the daisy chain, TdAS(DSA) must be greater than the sum of TdAS(IEO) for the highest priority device in the daisy chain, TsIEI(DSA) for the Z-SCC, and TdIEIf(IEO) for each device separating them in the daisy chain.
6. Parameter applies only to a 8031 pulling $\overline{INT}$ Low at the beginning of the Interrupt Acknowledge transaction.
7. Internal circuitry allows for the reset provided by the Z8 to be recognized as a reset by the Z-SCC.

All timing references assume 2.0 V for a logic "1" and 0.8 V for a logic "0".

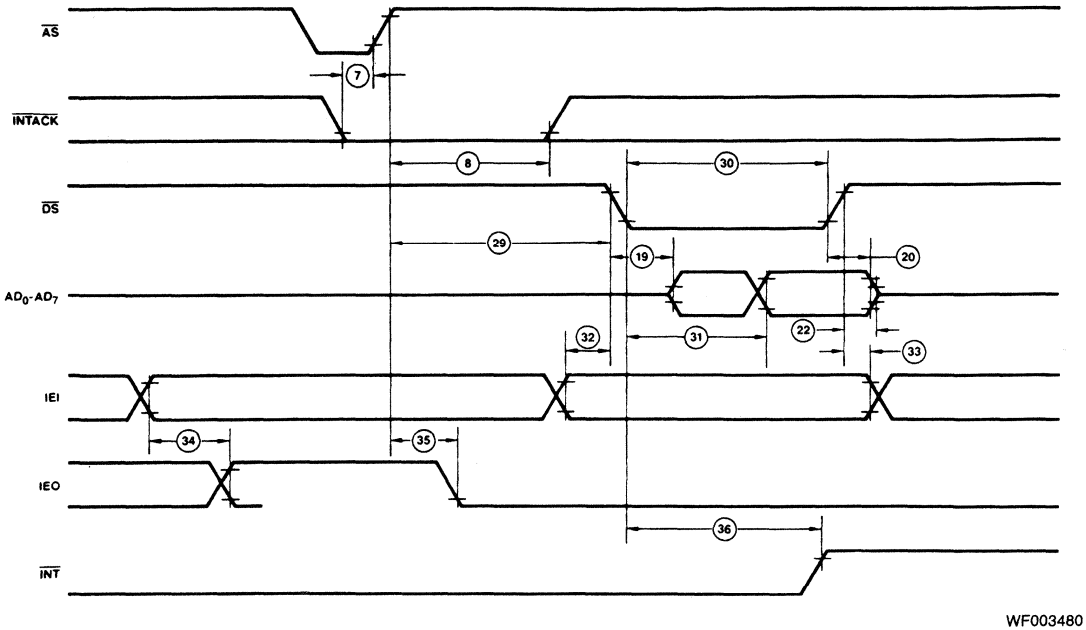


Figure 17. Z8031 Interrupt Acknowledge Timing

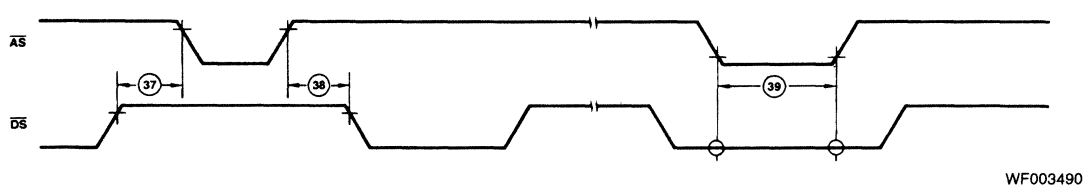


Figure 18. Z8031 Reset Timing

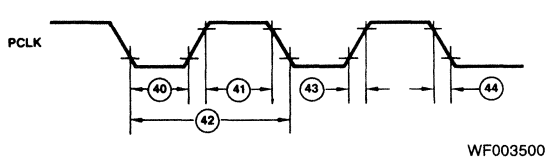


Figure 19. Z8031 Cycle Timing

SWITCHING CHARACTERISTICS (Cont'd.)

Z8531 Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 21, 22, 23)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		400		350	ns
28	TwWRI	$\overline{WR}$ Low Width	240		200		ns
29	TsDW(WR)	Write Data to $\overline{WR}$ ↓ Setup Time	10		10		ns
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		ns
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 4)		240		200	ns
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 4)		240		200	ns
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200	ns
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200	ns
35	TdWRr(REQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		4TcPC		4TcPC	ns
36	TdRDr(REQ)	$\overline{RD}$↑ to $\overline{DTR/REQ}$ Not Valid Delay		4TcPC		4TcPC	ns
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 4)		500		500	ns
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 5)	250		200		ns
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	250		200		ns
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		250		180	ns
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	120		100		ns
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		ns
43	TdIEI(IEO)	IEI to IEO Delay Time		120		100	ns
44	TdPC(IEO)	PCLK ↑ to IEO Delay		250		250	ns
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 4)		500		500	ns
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	30		15		ns
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	30		30		ns
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	250		200		ns
49	Trc	Valid Access Recovery Time (Note 3)	6TcPC + 200		6TcPC + 130		ns

Notes: 3. Parameter applies only between transactions involving the SCC.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEIf(IEO) for each device separating them in the daisy chain.

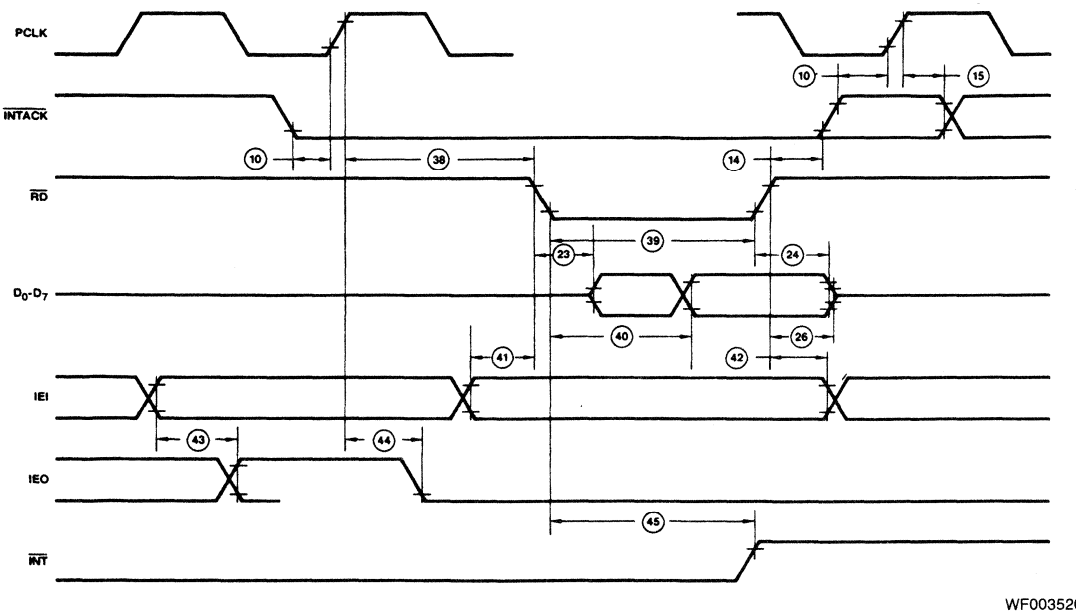


Figure 21. Z8531 Interrupt Acknowledge Timing

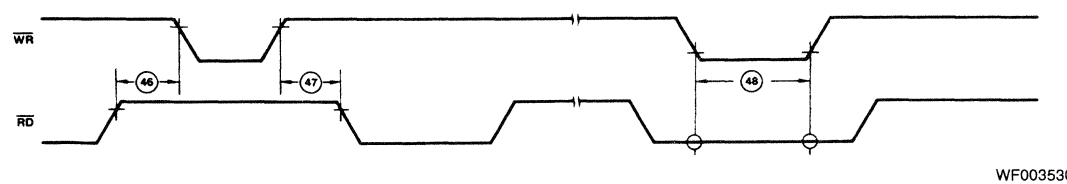


Figure 22. Z8531 Reset Timing

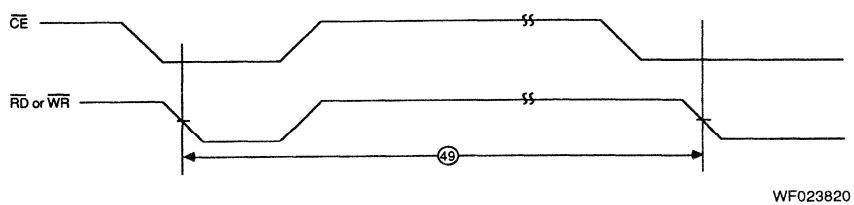


Figure 23. Z8531 Cycle Timing

GENERAL INFORMATION

AMD LITERATURE

To obtain literature in the U.S., write or call the AMD Literature Distribution Center, 901 Thompson Place, P.O. Box 3453 — M/S 82, Sunnyvale, CA 94088; (408) 732-2400, TOLL FREE (800) 538-8450. To obtain literature from international locations, contact the nearest AMD sales office or distributor (see listings in the back of this publication).

PACKAGE MATERIAL CONFIGURATION

	MULTILAYER CERAMIC			CERAMIC		PLASTIC				HEADERS	
	BRAZED PACKAGES		LEADLESS CHIP CARRIER	PIN GRID ARRAY	DIP		LEADED CHIP CARRIER	DIP			
PACKAGE BODY MATERIAL	90% Alumina (min.)		90% Alumina (min.)	90% Alumina (min.)	90% Alumina (min.)		Novolac Epoxy	Novolac Epoxy		Alloy 42	
DIE ATTACH PAD METALLIZATION	Gold		Gold	Gold	Gold	Silver Palladium	Silver	Gold	Silver	Gold	
DIE ATTACH MATERIAL	Gold/Silicon		Gold/Silicon	Gold/Silicon	Gold/Silicon		Silver Epoxy	Gold Silicon	Silver Epoxy	Gold/Silicon	
DIE ATTACH TEMPERATURE	440°C max.		440°C max.	440°C max.	440°C max.		180°C (max. temp.)	440°C max.	180°C (max. temp.)	390°C max.	
BOND FINGER METALLIZATION	Gold		Gold	Gold	Aluminum		Silver	Gold	Silver	Gold	
BONDING WIRE	Aluminum/ 1% Silicon		Aluminum/ 1% Silicon	Aluminum/ 1% Silicon	Aluminum/ 1% Silicon		Gold	Gold		Aluminum/ 1% Silicon	
BONDING METHOD	Ultrasonic		Ultrasonic	Ultrasonic	Ultrasonic		Thermal Sonic	Thermal Sonic		Ultrasonic	
SEAL RING METALLIZATION	Gold		Gold	Gold	N/A		N/A	N/A		Gold	Nickel
SEAL MATERIAL	Gold/Tin Eutectic		Gold/Tin Eutectic	Gold/Tin Eutectic	Vitreous Glass		N/A	N/A		N/A	N/A
LID MATERIAL	Alloy 42 (Gold Plated)	Alloy 42 (Tin Plated)	Alloy 42 (Gold Plated)	Alloy 42 (Gold Plated)	90% Alumina (min.)		N/A	N/A		Nickel/Nickel Clad Stainless Steel	
SEAL TEMPERATURE	370°C max.		370°C max.	370°C max.	460°C max.		185°C max. (Mold Temp.)	185°C max. (Mold Temp.)		Resistance Weld	
SEAL AMBIENT	Nitrogen		Nitrogen	Nitrogen	Air		N/A	N/A		Nitrogen	
LEAD MATERIAL	Alloy 42		N/A	Alloy 42	Alloy 42		Copper	Copper		Alloy 42	
LEAD FINISH	Gold	Tin	Gold (Solder Pad)	Gold	Tin		Solder	Solder		Gold	Nickel
HEATSINK	Aluminum (52/64 Lead Only)		N/A	Aluminum	N/A		N/A	N/A		N/A	
HEATSINK ATTACH MATERIAL	Epoxy (52/64 Lead Only)		N/A	Epoxy	N/A		N/A	N/A		N/A	
HEATSINK ATTACH TEMPERATURE	155°C (52/64 Lead Only)		N/A	155°C	N/A		N/A	N/A		N/A	

Note: N/A = Not Applicable

6

TECHNICAL REPORT

No. TR202

THERMAL CHARACTERIZATION OF PACKAGE DEVICES

by James D. Hayward

Publication #	Rev.	Amendment
08127	B	/0

Issue Date: June 1987

ABSTRACT

Determination of the Thermal Resistance of Packaged Devices is of concern to the designer of new devices and to AMD customers. The Advanced Package and Material Development group has undertaken the task of characterizing current AMD products and quantifying package-related influences on Thermal Resistance. This report describes some of these effects and the technique used to measure Thermal Resistance.

1.0 DEFINITION OF THERMAL RESISTANCE

The reliability of an integrated circuit is largely dependent on the maximum temperature which the device will attain during operation. Because the stability of a semiconductor junction declines with increasing temperature, knowledge of the thermal properties of the packaged device becomes an important factor during device design. In order to increase the operating lifetime of a given device, the junction temperatures must be minimized. This demands knowledge of the thermal resistance of the completed assembly and specification of the conditions in which the device will function properly. As devices become both smaller and more complex and the requirement for high speed operation becomes more important, heat dissipation will become an ever more critical parameter.

Thermal resistance is defined as the temperature rise per unit power dissipation above some referenced condition. The unit of measure is typically °C/watt. The relationship between junction temperature and thermal resistance is given by:

$$T_J = T_X + P_D\ \theta_{JX} \qquad (1)$$

where: T_J = junction temperature
T_X = reference temperature
P_D = power dissipation
θ_{JX} = thermal resistance
X = some defined test condition

In general, one of three conditions is defined for measurement of thermal resistance:

θ_{JC}	– thermal resistance measured with reference to the temperature at some specified point on the package surface.
θ_{JA} (still air)	– thermal resistance measured with respect to the temperature of a specified volume of still air.
θ_{JA} (moving air)	– thermal resistance measured with respect to the temperature of air moving at a specified velocity.

The relationship between θ_{JC} and θ_{JA} is

$$\theta_{JA} = \theta_{JC} + \theta_{CA}$$

where θ_{CA} is a measure of the heat dissipation due to natural convection (still air) or forced convection (moving air) and the effect of heat radiation and mounting techniques. θ_{JC} is dependent solely on material properties and package geometry; θ_{JA} includes the influence of the surface area of the package and environmental conditions. Each of these definitions of thermal resistance is an attempt to simulate some manner in which the package device may be used.

The thermal resistance of a packaged device, however measured, is a summation of the thermal resistances of the individual components of the assembly. These in turn are functions of the thermal conductivity of the component materials and the geometry of the heat flow paths. Like other material properties, thermal conductivity is usually temperature dependent. For alumina and silicon, two common package materials, this dependence can amount to a 30% variation in thermal conductivity over the operating temperature range of the device. The thermal resistance of a component is given by

$$\theta = \frac{L}{K(T)A} \qquad (2)$$

where: L = length of the heat flow path
A = cross sectional area of the heat flow path
K(T) = thermal conductivity as a function of temperature

and the overall thermal resistance of the assembly (discounting convective effects) will be:

$$\theta = \Sigma\theta_n = \Sigma\ \frac{L_n}{K_n A_n}$$

but since the heat flow path through a component is influenced by the materials surrounding it, determination of L and A is not always straightforward.

A second factor that affects the thermal resistance of a packaged device is the power dissipation level and, more particularly, the relationship between power level and die geometry, i.e., power distribution and power density. By rearrangement of equation 1 to

$$P_d = \frac{1}{\theta_{JX}}(T_J - T_X) = \frac{1}{\Sigma\theta_N}(T_J - T_X) \qquad (3)$$

the relationship between P_d and T_j can be more clearly seen. Thus, to dissipate a greater quantity of heat for a given geometry, T_J must increase and, since the individual θ_n will also increase with temperature, the increase in T_J will not be a linear function of increasing power levels.

A third factor of concern is the quality of the material interfaces. In terms of package construction, this relates specifically to the die attach bond, and for those packages having a heatsink, the heatsink attach bond. The quality of the die attach bond will most severely influence the package thermal resistance as this is the area which first impedes the transfer of heat out of the silicon die. Indeed, it seems likely that the initial thermal response of a powered device can be directly related to the quality of the die attach bond.

2.0 EXPERIMENTAL METHOD

The technique for measurement of thermal resistance involves the identification of a temperature-sensitive parameter on the device and monitoring this parameter while the device is powered. For bipolar integrated circuits the forward voltage of the substrate isolation diode provides a convenient parameter to measure and has the advantage of a linear dependence on temperature. MOS devices which do not have an accessible substrate diode present greater measurement difficulties and may require simulation through use of a specially designed thermal test die. Choice of the parameter to be measured must be made with some care to ensure that the results of the measurement are truly representative of the thermal state of the device being investigated. Thus measurement of the substrate isolation diode which is generally diffused across the area of the die yields a weighted average of the condition of the individual junctions across the die surface. Measurement of a more local source would yield a less generalized result.

For MOS devices, simulation is accomlished using the thermal test die. The basis for this test die is a 25 mil square cell containing an isolated diode and a 1 KΩ resistor. The resistors are interconnected from cell to cell on the wafer before it is cut

into mulitple arrays of the basic unit cell. In use the device is powered via the resistors with voltage or current adjusted for the proper level and the voltage drop of the individual diodes is monitored as in the case of actual devices.

Prior to the thermal resistance test, the diode voltage/temperature calibration must be determined. This is done by measuring the forward voltage at 1 mA current level at two different temperatures. The diode calibration factor is then:

$$K_f = \frac{T_2 - T_1}{V_2 - V_1} = \frac{\Delta T}{\Delta V} \qquad (4)$$

in units of °C/mV. For most diodes used for this test the voltage/temperature relationship is linear and these two measurement points are sufficient to determine the calibration.

The actual thermal resistance measurement has two alternating phases: measurement and power on. The device under test is pulse powered with an ON duty cycle of 99% and a repetition rate of < 100 Hz. During the brief OFF states the device is reverse-biased with a 1 mA current and the voltage drop is measured. The series of voltage readings are averaged over short periods and compared to the voltage reading obtained before the device was first powered ON. The thermal resistance is then computed as:

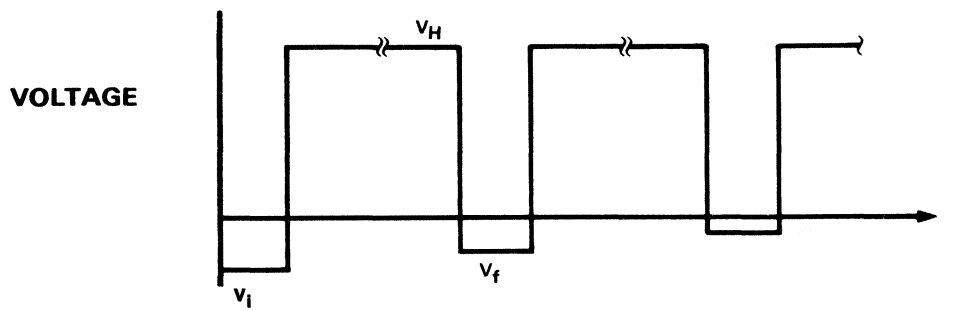
$$\theta_{jx} = \frac{K_F(V_F - V_I)}{V_H I_H} = \frac{K_f \Delta V}{P_D} \qquad (5)$$

where: K_F = calibration factor
V_I = initial forward voltage value
V_F = current forward voltage value
V_H = heating voltage
I_H = heating current

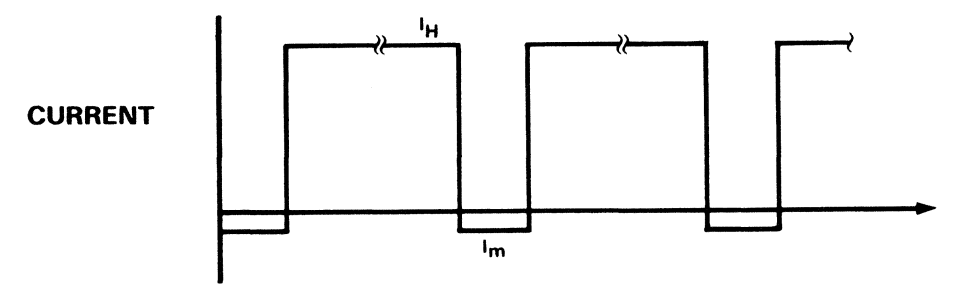
The pulsing measurement is continued until the device has reached thermal equilibrium and the final value measured is the equilibrium thermal resistance of the device under test.

When the end result desired is θ_{JA} (still air), the device and the test fixture (typically a standard burn-in socket) are enclosed in a box containing approximately 1 cubic foot of air. For θ_{JC} measurements the device is attached to a large metal heatsink. This ensures that the reference point on the device surface is maintained at a constant temperature. The requirements for measurement of θ_{JA} (moving air) are rather more comple x and involve the use of a small wind tunnel with capability for monitoring air pressure, temperature and velocity in the area immediately surrounding the device tested. Standardization of this last test requires much careful attention.

WAVEFORMS FOR PULSED THERMAL RESISTANCE TEST



WF009091



WF009080

3.0 EXPERIMENTAL RESULTS

The thermal resistance data included in the attached table was extrapolated from data collected using the procedure outlined in the preceding section. This data has resulted from an ongoing program undertaken by members of the Material Technology Development group.

Updated data will replace the data in this table as each device is measured or revised data becomes available.

TABLE 1. THERMAL RESISTANCE OF AMD PRODUCTS

Device	Package	Process	Gates[1] (Equiv.)	Area (KSq Mils)	Power[2] (mW)	θ_{JA}[3] (C/W)	θ_{JC}[3] (C/W)
5380	SD 040						
	PD 040						
Z8001	SD 048	MOS	5.9K	63.9	1625	35	10
	PD 048	MOS	5.9K	63.9	1625	50	
	CL 052	MOS	5.9K	63.9	1625	44	7
Z8002	SD 040	MOS	5.9K	63.9	1625	34	7
	CD 040	MOS	5.9K	63.9	1625	34	7
	PD 040	MOS	5.9K	63.9	1625	44	
	CL 052	MOS	5.9K	63.9	1625	44	7
80186	CA2068	MOS	23.0K	88.9	2750	31	8
	CGX068	MOS	23.0K	88.9	2750	34	8
	PL 068	MOS	23.0K	88.9	2750	35	
80188	CA2068	MOS	23.0K	88.9	2750	31	8
	CGX068	MOS	23.0K	88.9	2750	34	8
	PL 068	MOS	23.0K	88.9	2750	35	
80286	CA2068	MOS	40.0K	98.0	3000	31	8
	CGX068	MOS	40.0K	98.0	3000	34	8
80L286	PL 068	MOS	40.0K	98.0	2250	35	
Z8030/31	CD 040	MOS	5.9K	48.8	1250	34	7
	PD 040	MOS	5.9K	48.8	1250	44	
8031AH	CD 040	MOS	7.3K	34.2	625	34	7
	PD 040	MOS	7.3K	34.2	625	44	
	PL 044	MOS	7.3K	34.2	625	47	
80C31	CD 040	CMOS	7.5K	35.0	150		
	PD 040	CMOS	7.5K	35.0	150		
	PL 044	CMOS	7.5K	35.0	150		
8051AH	CD 040	MOS	7.3K	34.2	625	34	7
	PD 040	MOS	7.3K	34.2	625	44	
	PL 044	MOS	7.3K	34.2	625	47	
80C51	CD 040	CMOS	7.5K	35.0	150		
	PD 040	CMOS	7.5K	35.0	150		
80515	PL 068	MOS	8.5K		880		
80C521	CD 040	CMOS	7.5K	46.0	175		
8053AH	CD 040	MOS	7.5K	38.8	625	34	7
	PD 040	MOS	7.5K	38.8	625	44	
	PL 044	MOS	7.5K	38.8	625	47	
80535	PL 068	MOS	8.5K		880		
Z8065	CD 040	MOS			1250	34	
Z8068	CD 040	MOS	6.4K	55.1	1250	34	7
	PD 040	MOS	6.4K	55.1	1250	44	
	PL 044	MOS	6.4K	55.1	1250	47	
8080A	CD 040	MOS	1.5K	22.4	1245	37	9
	PD 040	MOS	1.5K	22.4	1245	46	
8085AH	CD 040	MOS	2.4K	36.6	1000	34	7
	PD 040	MOS	2.4K	36.6	1000	44	

TABLE 1. THERMAL RESISTANCE OF AMD PRODUCTS (Cont'd.)

Device	Package	Process	Gates[1] (Equiv.)	Area (KSq Mils)	Power[2] (mW)	θ_{JA}[3] (C/W)	θ_{JC}[3] (C/W)
8086	CD 040	MOS	9.6K	52.9	1700	34	7
	PD 040	MOS	9.6K	52.9	1700	44	
	PL 044	MOS	9.6K	52.9	1700	47	
8088	CD 040	MOS	9.6K	36.1	1700	34	7
	PD 040	MOS	9.6K	36.1	1700	44	
	PL 044	MOS	9.6K	36.1	1700	47	
8155	CD 040	MOS	5.0K	44.2	900	34	7
	PD 040	MOS	5.0K	44.2	900	44	
8156	CD 040	MOS	5.0K	44.2	900	34	7
	PD 040	MOS	5.0K	44.2	900	44	
8155H	CD 040	MOS	5.0K	44.2	625	34	7
	PD 040	MOS	5.0K	44.2	625	44	
8156H	CD 040	MOS	5.0K	44.2	625	34	7
	PD 040	MOS	5.0K	44.2	625	44	
82284	CD 018	BPL	32	8.9	725	60	14
	PD 018	BPL	32	8.9	725	58	
82C288	CD 020	CMOS		16.0	120		
	PD 020	CMOS		16.0	120		
8231A	CD 024	MOS	7.2K	58.6	1700	45	11
8232	CD 024	MOS	7.7K	61.7	1615	45	11
8237A	CD 040	MOS	2.2K	30.6	750	34	7
	PD 040	MOS	2.2K	30.6	750	44	
8251	CD 028	MOS	1.2K	25.3	400	37	9
	PD 028	MOS	1.2K	25.3	400	47	
8251A	CD 028	MOS	1.2K	28.2	500	37	9
	PD 028	MOS	1.2K	28.2	500	47	
	PL 028	MOS	1.2K	28.2	500	52	
8253	CD 024	MOS	1.3K	22.7	700	45	11
	PD 024	MOS	1.3K	22.7	700	55	
	PL 028	MOS	1.3K	22.7	700	52	
82C54	CD 024	CMOS		24.0	70		
	PD 024	CMOS		24.0	70		
8255A	CD 040	MOS	0.6K	16.5	600	34	7
	PD 040	MOS	0.6K	16.5	600	44	
	PL 044	MOS	0.6K	16.5	600	42	
8259A	CD 028	MOS	0.8K	16.9	425	37	9
	PD 028	MOS	0.8K	16.9	425	47	
	PL 028	MOS	0.8K	16.9	425	52	
8284A	CD 018	BPL	35	11.7	810	60	14
	PD 018	BPL	35	11.7	810	58	
8288	CD 020	BPL			1150	70	14
	PD 020	BPL			1150	80	
Z8530/31	CD 040	MOS	5.0K	48.8	1250	34	7
	PD 040	MOS	5.0K	48.8	1250	44	
	PL 044	MOS	5.0K	48.8	1250	47	
8751H	CDV040	MOS	7.5K	57.0	1250	34	7
	CLV044	MOS	7.5K	57.0	1250	47	7
8753H	CDV040	MOS	7.5K	57.0	1250	34	7
	CLV044	MOS	7.5K	57.0	1250	47	
9080A	CD 040	MOS	9.6K	36.1	1700	34	7
	PD 040	MOS	9.6K	36.1	1700	44	
9511A	CD024	MOS	7.2K	58.6	1700	45	11

TABLE 1. THERMAL RESISTANCE OF AMD PRODUCTS (Cont'd.)

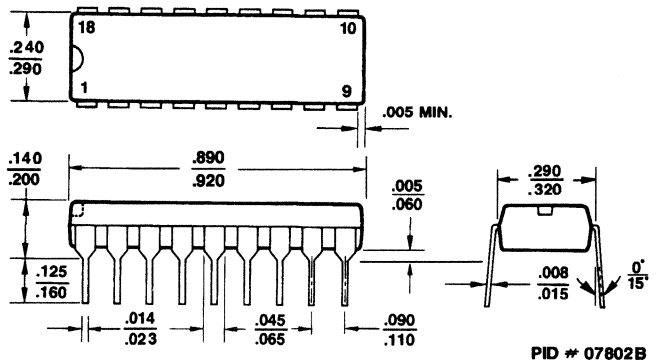
Device	Package	Process	Gates[1] (Equiv.)	Area (KSq Mils)	Power[2] (mW)	θ_{JA} [3] (C/W)	θ_{JC} [3] (C/W)
9513A	CD 040	MOS	4.5K	39.3	1375	34	7
	PD 040	MOS	4.5K	39.3	1375	44	
	CL 044	MOS	4.5K	39.3	1375	47	7
	PL 044	MOS	4.5K	39.3	1375	47	
9516A	SD 048	MOS	3.7K	52.4	1750	35	10
	PD 048	MOS	3.7K	52.4	1750	50	
9517A	CD 040	MOS	2.2K	30.6	750	34	7
	PD 040	MOS	2.2K	30.6	750	44	
	PL 044	MOS	2.2K	30.6	750	42	
9518	CD 040	MOS	6.4K	55.1	1250	34	7
	PD 040	MOS	6.4K	55.1	1250	44	
	PL 044	MOS	6.4K	55.1	1250	47	
9519A	CD 028	MOS	1.5K	39.3	725	37	9
	PD 028	MOS	1.5K	39.3	725	47	
9520/21	CD 040	MOS	4.0K	50.1	1375	34	7
	PD 040	MOS	4.0K	50.1	1375	44	
9551	CD 028	MOS	1.2K	25.3	400	37	9
	PD 028	MOS	1.2K	25.3	400	47	
9568	CD 040	MOS	14.0K	56.0	1250	34	7
	PD 040	MOS	14.0K	56.0	1250	44	
9580A	CA2068	MOS	150.0K	126.2	3500	31	8
9582	CD 028	BPL			1500		
	PD 028	BPL			1500		
95C85	PL 044	CMOS	53.3K	98.6	1000		

Notes: 1. The gate counts are only an approximation (total devices / 3).
2. Power is the highest I_{CC} over the temperature times the nominal power supply voltage.
3. θ_{JA} and θ_{JC} values were estimated by extrapolating measured data. Values are to be considered as "typical" for the device.
4. There is no accepted industry definition for θ_{JC} for molded plastic packages at this time.
5. CMOS data is not available.

PACKAGE OUTLINES*

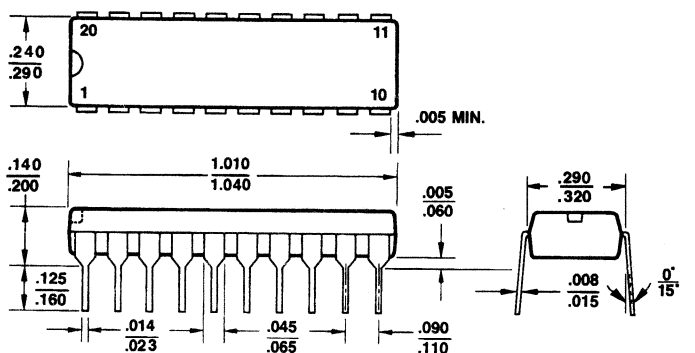
Plastic Dual-In-Line Packages (PD)

PD 018



PID # 07802B

PD 020

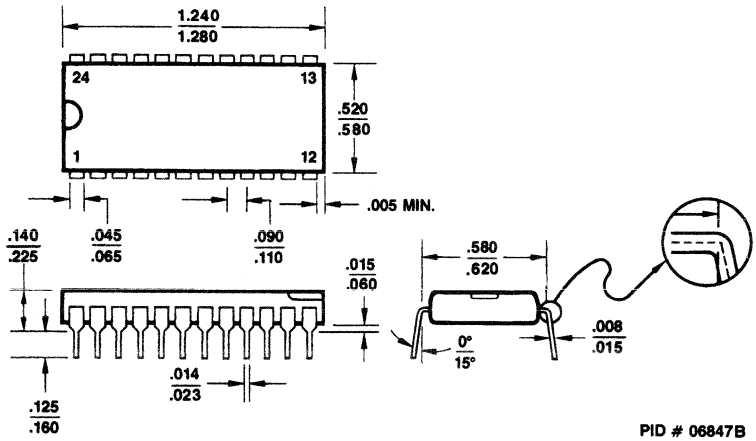


PID # 07552A

*For reference only.

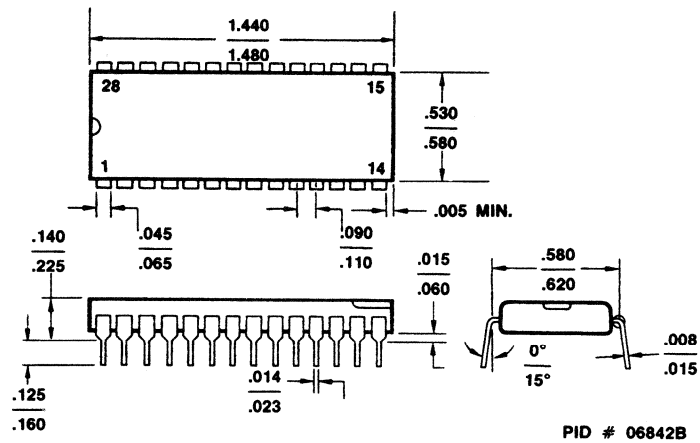
Plastic Dual-In-Line Packages (PD) (Cont'd.)

PD 024



PID # 06847B

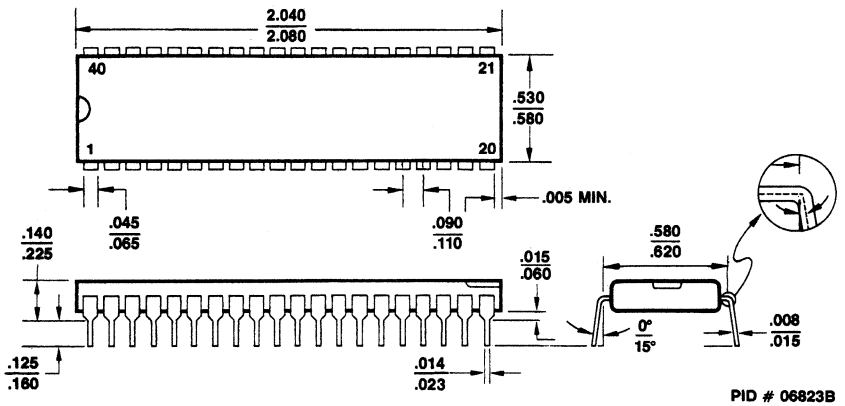
PD 028



PID # 06842B

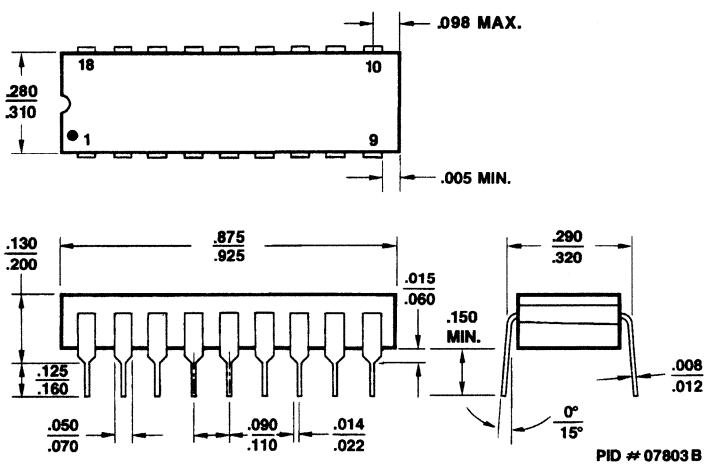
Plastic Dual-In-Line Packages (PD) (Cont'd.)

PD 040



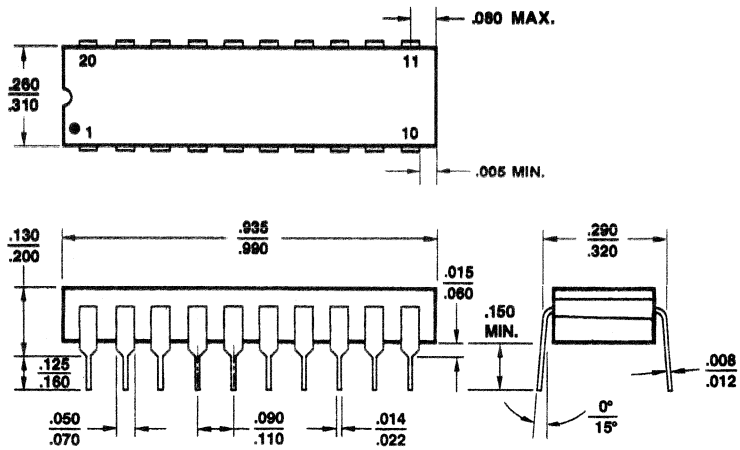
Ceramic Hermetic Dual-In-Line Packages (CD)

CD 018



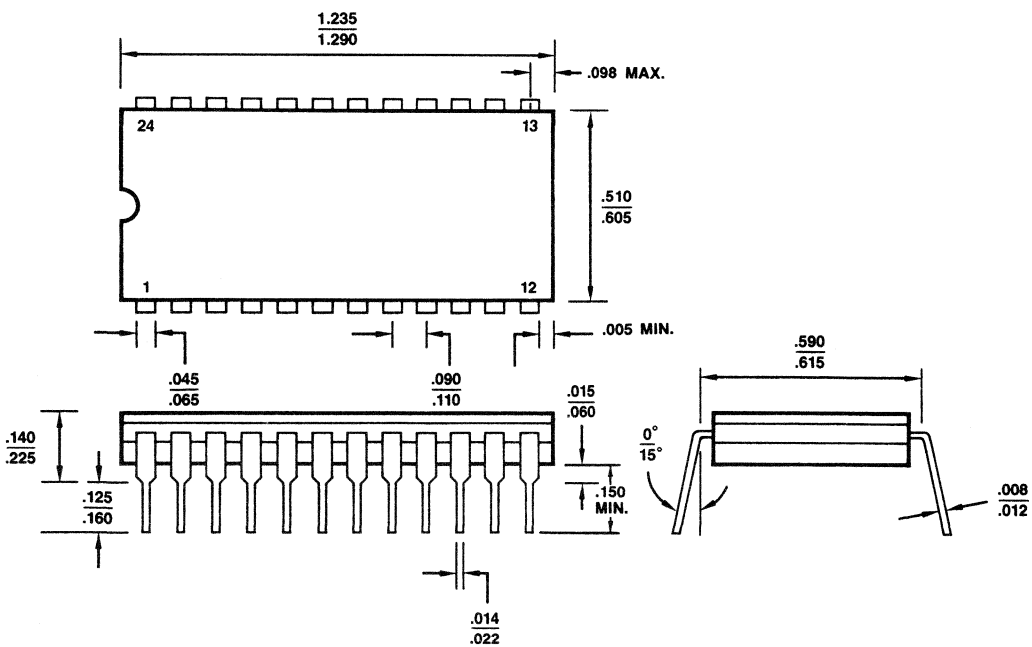
Ceramic Hermetic Dual-In-Line Packages (CD) (Cont'd.)

CD 020



PID # 07553A

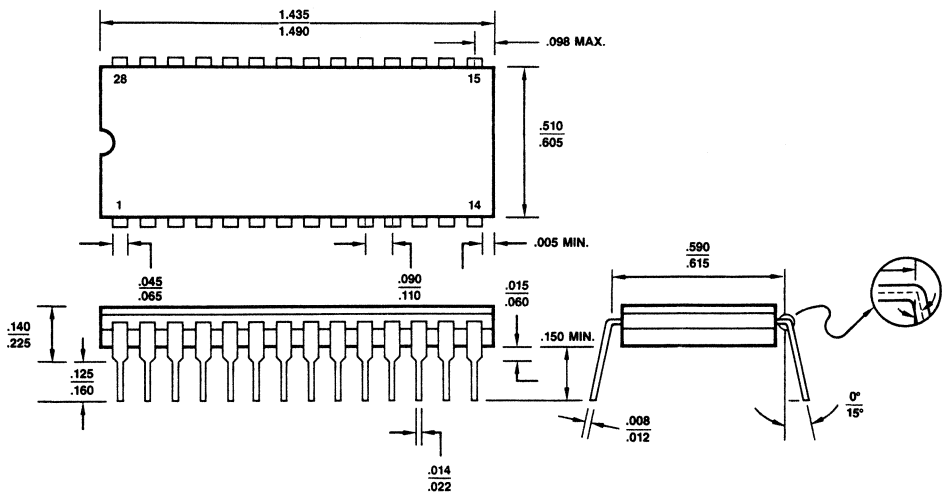
CD 024



PID # 07156A

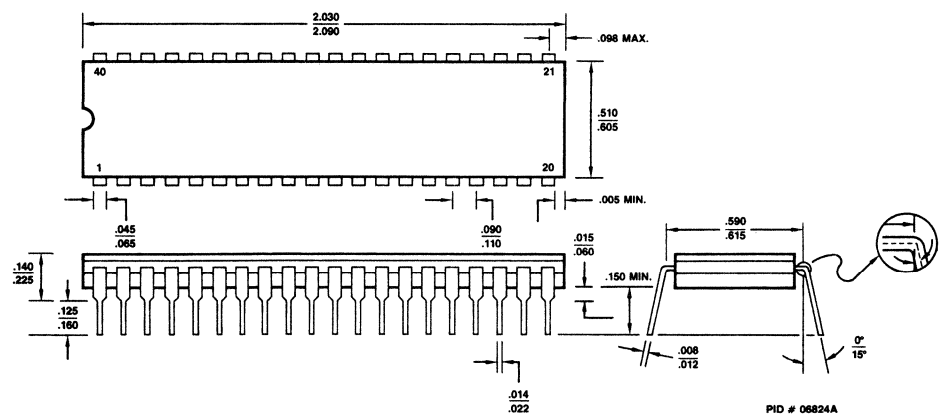
Ceramic Hermetic Dual-In-Line Packages (CD) (Cont'd.)

CD 028



PID # 06837A

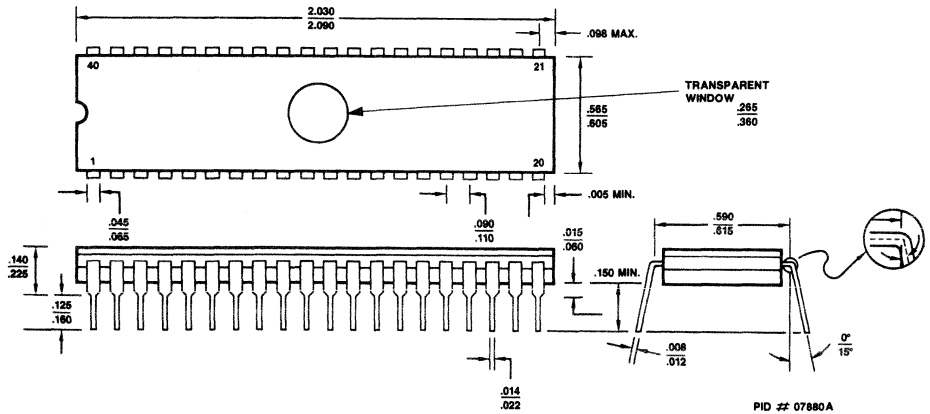
CD 040



PID # 06824A

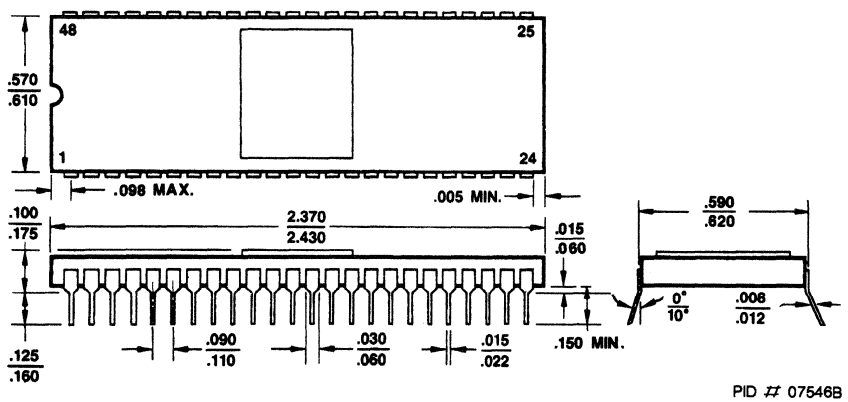
Ceramic Hermetic Dual-In-Line Packages (CD) (Cont'd.)

CDV040



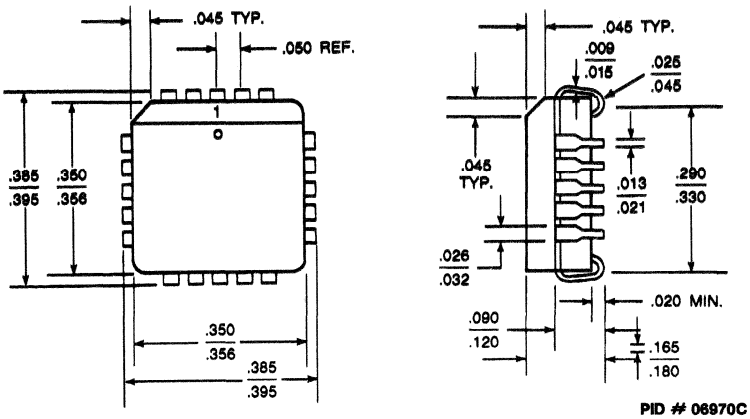
Sidebrazed Hermetic Dual-In-Line Packages (SD)

SD 048



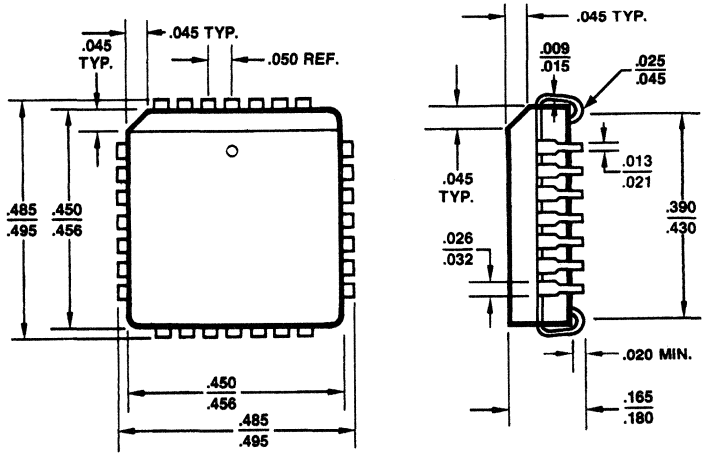
Plastic Leaded Chip Carriers (PL)

PL 020



PID # 06970C

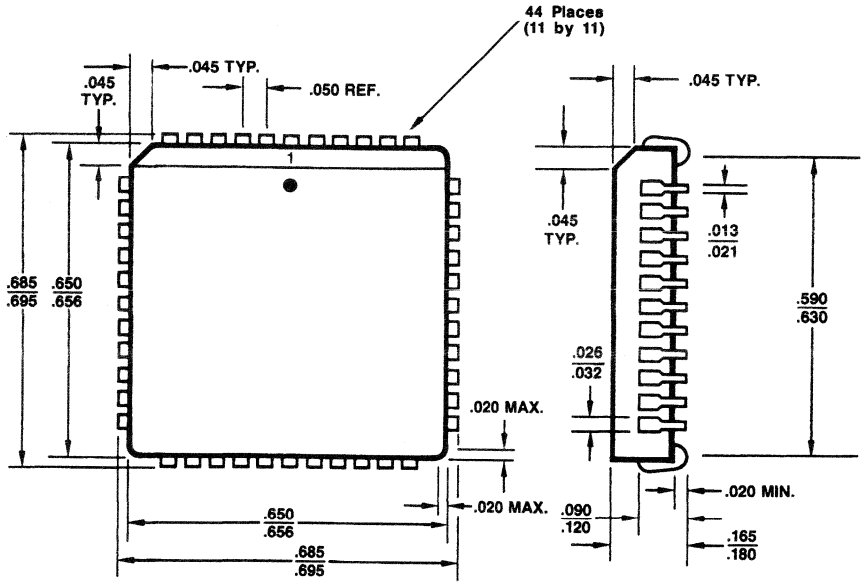
PL 028



PID # 06751D

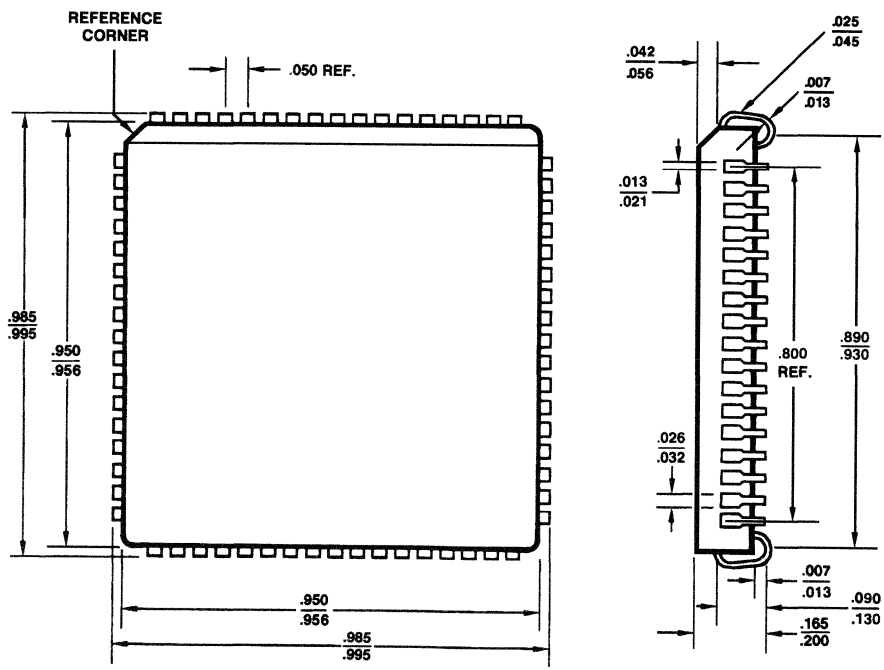
Plastic Leaded Chip Carriers (PL) (Cont'd.)

PL 044



PID # 06752B

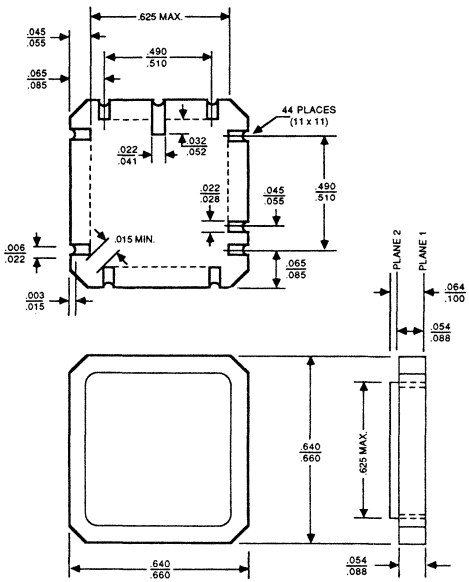
PL 068



PID # 06753G

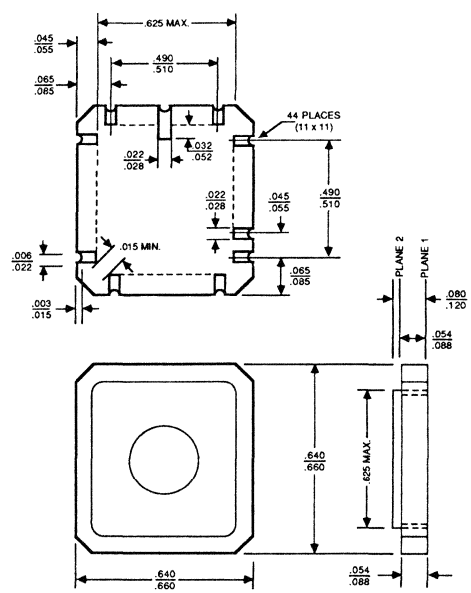
Ceramic Leadless Chip Carriers (CL)

CL 044



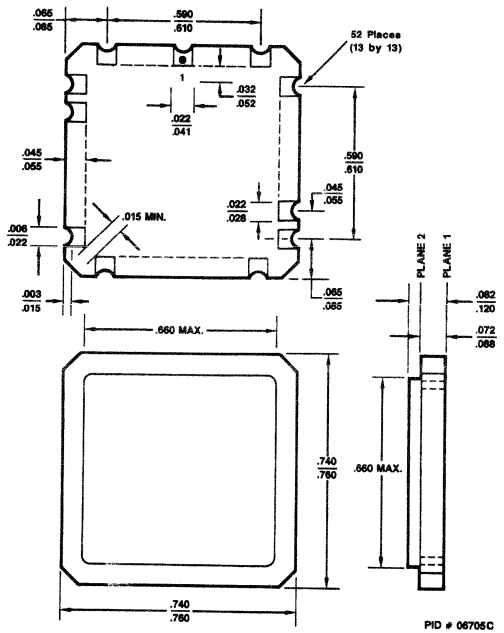
PID #06825D

CLV044

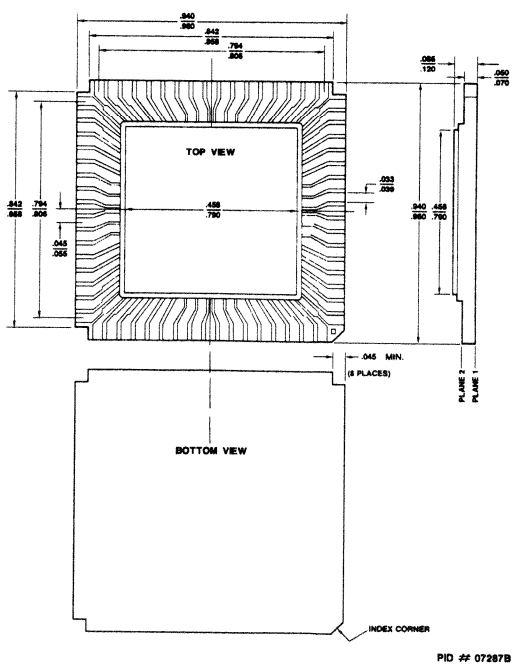


PID #09703A

Ceramic Leadless Chip Carriers (CL) (Cont'd.)
CL 052

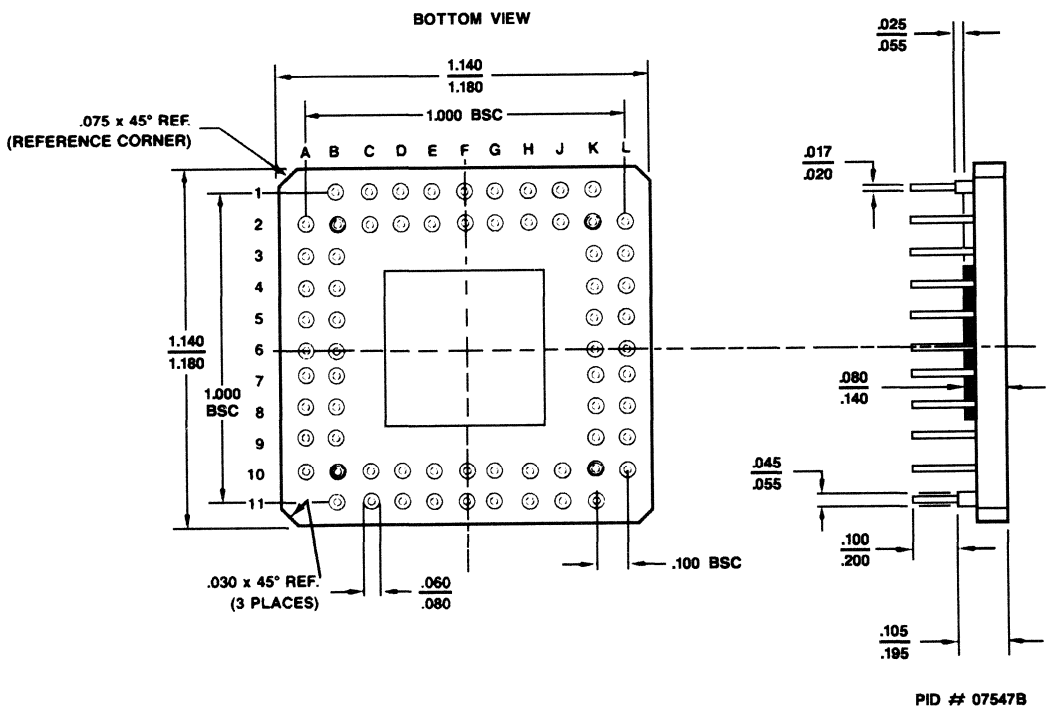


68-Pin Square Leadless Chip Carrier (CA)
CA2068



Ceramic Pin Grid Array Package (CG)

CGX068



PID # 07547B

ADVANCED MICRO DEVICES' U.S. SALES OFFICES

ALABAMA ..... (205) 882-9122
ARIZONA,
Tempe ..... (602) 242-4400
CALIFORNIA,
Culver City ..... (213) 645-1524
Newport Beach ..... (714) 752-6262
San Diego ..... (619) 560-7030
Santa Clara ..... (408) 727-3270
Woodland Hills ..... (818) 992-4155
COLORADO ..... (303) 741-2900
CONNECTICUT ..... (203) 264-7800
FLORIDA,
Clearwater ..... (813) 530-9971
Ft Lauderdale ..... (305) 484-8600
Melbourne ..... (305) 729-0496
Orlando ..... (305) 859-0831
GEORGIA ..... (404) 449-7920
ILLINOIS,
Chicago ..... (312) 773-4422
Naperville ..... (312) 505-9517
INDIANA ..... (317) 244-7207
KANSAS ..... (913) 451-3115
MARYLAND ..... (301) 796-9310
MASSACHUSETTS ..... (617) 273-3970
MINNESOTA ..... (612) 938-0001
MISSOURI ..... (314) 275-4415
NEW JERSEY ..... (201) 299-0002
NEW YORK,
Liverpool ..... (315) 457-5400
Poughkeepsie ..... (914) 471-8180
Woodbury ..... (516) 364-8020
NORTH CAROLINA ..... (919) 847-8471
OREGON ..... (503) 245-0080
OHIO ..... (614) 891-6455
PENNSYLVANIA,
Allentown ..... (215) 398-8006
Willow Grove ..... (215) 657-3101
TEXAS,
Austin ..... (512) 346-7830
Dallas ..... (214) 934-9099
Houston ..... (713) 785-9001
WASHINGTON ..... (206) 455-3600
WISCONSIN ..... (414) 792-0590

ADVANCED MICRO DEVICES' INTERNATIONAL SALES OFFICES

BELGIUM,
Bruxelles ..... TEL .. (02) 771 91 42
FAX .. (02) 762 37 12
TLX 61028
CANADA, Ontario,
Kanata ..... TEL . (613) 592-0060
Willowdale ..... TEL . (416) 224-5193
FAX . (416) 224-5193
FRANCE,
Paris ..... TEL (01) 49-75-10-10
FAX (01) 49-75-10-13
TLX 263282F
GERMANY,
Hannover area ..... TEL ... (05143) 50 55
FAX ... (05143) 55 53
TLX 925287
München ..... TEL ... (089) 41 14-0
FAX ... (089) 406490
TLX 523883
Stuttgart ..... TEL . (0711) 62 33 77
FAX .. (0711) 625187
TLX 721882
HONG KONG,
Kowloon ..... TEL 3-695377
FAX 1234276
TLX 504260AMDAPHX
ITALY, Milano ..... TEL ... (02) 3390541
FAX ... (02) 3498000
TLX 315286
JAPAN,
Tokyo ..... TEL ..... (03) 345-8241
FAX 3425196
TLX J24064AMDTKOJ
Osaka ..... TEL 06-243-3250
FAX 06-243-3253
KOREA, Seoul ..... TEL 82-2-784-7598
FAX 82-2-784-8014
LATIN AMERICA,
Ft. Lauderdale ..... TEL ..... (305) 484-8600
FAX ..... (305) 485-9736
TLX .. 5109554261 AMDFTL
SWEDEN, Stockholm ..... TEL ..... (08) 733 03 50
FAX ..... (08) 733 22 85
TLX 11602
TAIWAN ..... TLX 886-2-7122066
FAX 886-2-7122017
UNITED KINGDOM,
Manchester area ..... TEL ..... (0925) 828008
FAX ..... (0925) 827693
TLX 628524
London area ..... TEL ..... (04862) 22121
FAX ..... (04862) 22179
TLX 859103
TLX 886-2-7122066
FAX 886-2-7122017

NORTH AMERICAN REPRESENTATIVES

CALIFORNIA
I^2 INC ..... OEM (408) 988-3400
DISTI (408) 498-6868
CANADA
Calgary, Alberta
VITEL ELECTRONICS ..... (403) 278-5833
Kanata, Ontario
VITEL ELECTRONICS ..... (613) 592-0090
Mississauga, Ontario
VITAL ELECTRONICS ..... (416) 676-9720
Quebec
VITEL ELECTRONICS ..... (514) 636-5951
IDAHO
INTERMOUNTAIN TECH MKGT ..... (208) 888-6071
INDIANA
SAI MARKETING CORP ..... (317) 253-1668
IOWA
LORENZ SALES ..... (319) 377-4666
KANSAS
LORENZ SALES ..... (913) 384-6556
MICHIGAN
SAI MARKETING CORP ..... (313) 750-1922
MISSOURI
LORENZ SALES ..... (314) 997-4558
NEBRASKA
LORENZ SALES ..... (402) 475-4660
NEW MEXICO
THORSON DESERT STATES ..... (505) 293-8555
NEW YORK
NYCOM, INC ..... (315) 437-8343
OHIO
Columbus
DOLFUSS ROOT & CO ..... (614) 885-4844
Dayton
DOLFUSS ROOT & CO ..... (513) 433-6776
Strongsville
DOLFUSS ROOT & CO ..... (216) 238-0300
PENNSYLVANIA
DOLFUSS ROOT & CO ..... (412) 221-4420
UTAH
R^2 MARKETING ..... (801) 595-0631

ADVANCED MICRO DEVICES 901 Thompson Pl., P.O. Box 3453, Sunnyvale, CA 94088, USA
TEL: (408) 732-2400 • TWX: 910-339-9280 • TELEX: 34-6306 • TOLL FREE: (800) 538-8450