

**MICROPROCESSORS  
AND  
PERIPHERALS**

**DATA BOOK**



REPRODUCTION  
OF  
THE  
MOUNTAIN

NO. 10, 1910

<b>GENERAL INFORMATION</b>	<b>1</b>
<b>QUALITY AND RELIABILITY</b>	<b>2</b>
<b>CMOS MICROPROCESSORS</b>	<b>3</b>
<b>NMOS MICROPROCESSORS</b>	<b>4</b>
<b>INTELLIGENT PERIPHERAL CONTROLLERS</b>	<b>5</b>
<b>CMOS SYSTEM SUPPORT PRODUCTS</b>	<b>6</b>
<b>NMOS + BIPOLAR SYSTEM SUPPORT PRODUCTS</b>	<b>7</b>
<b>DEVELOPMENT TOOLS</b>	<b>8</b>
<b>SENTRY TEST TAPE OVERVIEW</b>	<b>9</b>
<b>PACKAGING INFORMATION</b>	<b>10</b>



## TABLE OF CONTENTS

### Section 1 – General Information

Introduction .....	1.3
CMOS Microprocessor Selection Guide .....	1.4
NMOS and HMOS Microprocessor Selection Guide .....	1.5
Intelligent Peripheral Controller Selection Guide .....	1.6
Magnetic Media Controllers .....	1.6
Communication Controllers .....	1.7
Graphics Controllers .....	1.7
CMOS System Support Product Selection Guide .....	1.8
NMOS System Support Product Selection Guide .....	1.9
μPD70108/116/208/216/320/330 Software Development Tool Selection Guide .....	1.10
μPD70108/116/208/216/320/330/616 Hardware Development Tool Selection Guide .....	1.11
μPD70108/116/208/216/320/616 Evaluation Boards Selection Guide .....	1.11
μPD70108/116/208/216/616 Operating System Selection Guide .....	1.12
μPD70616/116/632 (V60/V70) Software Development Tool Selection Guide .....	1.12
Intelligent Peripherals Evaluation/Demonstration Tool Selection Guide .....	1.12
Dataflow Processor (IMPP) Hardware/Software Development Tool Selection Guide .....	1.12

### Section 2 – Quality and Reliability

Introduction .....	2.3
Technology Description .....	2.3
Reliability Testing .....	2.3
Failure Rate Calculation and Prediction .....	2.6
Reliability Test Results .....	2.7
NEC's Goals on Failure Rates .....	2.7
Infant Mortality Failure Screening .....	2.8
Life Test .....	2.8
Built-in Quality and Reliability .....	2.10
Approaches to Total Quality Control .....	2.10
Summary and Conclusion .....	2.12

### Section 3 – CMOS Microprocessors/Microcomputer

μPD70008/A	8-Bit Microprocessors .....	3.3
μPD70108	8/16-Bit High-Performance Microprocessor (V20™) .....	3.31
μPD70116	16-Bit High-Performance Microprocessor (V30™) .....	3.63
μPD70208	8/16-Bit High-Integration Microprocessor (V40™) .....	3.95
μPD70216	16-Bit High-Integration Microprocessor (V50™) .....	3.167
μPD70320/322	16-Bit Single-Chip Microcomputer (V25™) .....	3.239
μPD70616	32-Bit Virtual Memory Microprocessor (V60™) .....	3.313

### Section 4 – NMOS Microprocessors

μPD780	High Performance CP/M®-Compatible 8-Bit Microprocessor .....	4.3
μPD8085A/AH	8-Bit, Single-Chip Microprocessors .....	4.27
μPD8086	16-Bit Microprocessor .....	4.47
μPD8088	High-Performance 8-Bit Microprocessor .....	4.59

CP/M is a registered trademark of Digital Research Inc.

## Section 5 – Intelligent Peripheral Controllers

### Magnetic Media Controllers

μPD765A/7265	Single/Double Density Floppy Disk Controllers .....	5.3
μPD72065/66		
μPB9201	Floppy-Disk Interface .....	5.61
μPD71065/66	Floppy-Disk Interface .....	5.69
μPD72067	Advanced Floppy Disk Controller .....	5.77
μPD72068	Single Chip Floppy Disk Controller with Digital Data Separator .....	5.87
μPD7260	Hard and Floppy Disk Controller .....	5.91
μPD7261A/B	Hard-Disk Controllers .....	5.111
μPD9306/A	CMOS Hard-Disk Interface .....	5.147
μPD7262	Enhanced Small Device Interface Controller .....	5.155

### Communications Controllers

μPD7201A	Multiprotocol Serial Communications Controller .....	5.167
μPD72001	CMOS Multiprotocol Serial Communications Controller .....	5.181
μPD72105	OMNINET Local Area Network Controller .....	5.199
μPD7210	Intelligent GPIB Controller .....	5.215

### Display/Graphics Controllers

μPD7220A	High-Performance Graphics Display Controller .....	5.227
μPD72022	Intelligent Display Processor .....	5.257
μPD72120	Advanced Graphic Display Controller .....	5.275
μPD7225	LCD 7/14 Segment Controller/Driver .....	5.297
μPD7227	LCD 8 x 40 Dot Matrix Controller/Driver .....	5.311
μPD7228/A	LCD 8 x 50/16 x 42 Dot Matrix Controller/Driver .....	5.315
μPD72030	LCD 640 x 256 Dot Panel Controller .....	5.339
μPD6307	LCD Row Driver for μPD72030 .....	5.374
μPD6308	LCD Column Driver for μPD72030 .....	5.383

### Image Processing

μPD7281	Data-flow based Image Pipelined Processor .....	5.389
μPD9305	Support chip for μPD7281 .....	5.431

## Section 6 – CMOS System Support Products

μPD71011	Clock pulse Generator/Driver .....	6.3
μPD71051	Serial Control Unit .....	6.13
μPD71054	Programmable Timer/Counter .....	6.33
μPD71055	Parallel Interface Unit .....	6.49
μPD71059	Interrupt Control Unit .....	6.71
μPD71071	DMA Control Unit .....	6.97
μPD71082/83	8-Bit Latches .....	6.135
μPD71084	Clock Pulse Generator/Driver .....	6.139
μPD71086/87	8-Bit Bus Buffer/Drivers .....	6.147
μPD71088	System Bus Controller .....	6.151
μPD71611	Clock Pulse Generator/Driver for V60 .....	6.159
μPD71613	System Bus Controller for V60 .....	6.169
μPD82C43	CMOS Input/Output Expander for μPD8048/C48 Family .....	6.177

## Section 7 – NMOS System Support Products

μPD8155/56	2048-Bit Static MOS RAM with I/O Ports and Timer	7.3
μPB8216/26	4-Bit Parallel Bidirectional Bus Drivers	7.11
μPD8237A	High-Performance Programmable DMA Controller	7.15
μPD8243/H	Input/Output Expander for μPD8048 Family	7.33
μPD8251A/AF	Programmable Communications Interface (USART)	7.39
μPD8253	Programmable Interval Timer	7.57
μPD8255A	Programmable Peripheral Interface	7.69
μPD8257	Programmable DMA Controller	7.79
μPD8259A	Programmable Interrupt Controller	7.91
μPD8279	Programmable Keyboard/Display Interface	7.109

### BIPOLAR System Support Products

μPB8282/83	8-Bit Latches	7.117
μPB8284A	Clock Generator and Driver for 8086/8088 Microprocessors	7.121
μPB8286/87	8-Bit Bus Transceivers	7.127
μPB8288	CPU System Bus Controller	7.131
μPB8289	Bus Arbiter	7.139

## Section 8 – Development Tools V-Series Microprocessors

RA70116	Relocatable Assembler for V20/V30/V40/V50	8.3
RA70320	Relocatable Assembler for V25/V35	8.7
CC70116	C-compiler for V20/V30/V40/V50/V25 and V35	8.11
RC70116	Relocatable Object Format (ROF) Converter	8.13
NECTOMS	Converter to change relocatable object code from NEC format into Microsoft (MS) format	8.15
MSTONEC	Converter to change relocatable object code from Microsoft (MS) format into NEC format	8.17
PKG70616	Software Generation Tool package for V60/V70	8.19
IE-700K	Stand alone In-Circuit Emulator for V-Series	8.23
Mini-IE	Mini emulator for V40/V50 and V25	8.43
EBIBM-70108/70116	Design-/Evaluation board for V20 and V30	8.47
EBIBM-70208/70216	Design-/Evaluation board for V40 and V50	8.49
EBIBM-7061UNIX	V60 coprocessor slave board with UNIX for PC-XT/AT	8.51
EBIBM-70616SBC	V60 single board computer for Multibus 1	8.53

### Intelligent Peripherals and Dataflow Processors

EBIBM-72001	IBM-XT/AT board for μPD72001	8.55
EBIBM-7220A	IBM-XT/AT board for μPD7220A	8.57
EBIBM-72120	IBM-XT/AT board for μPD72120	8.59
EBIBM-72067	IBM-XT board for μPD72067	8.61
EBIBM-7261A	IBM-XT board for μPD7261A and μPD2306A	8.65
EBIBM-ATDISK	IBM-AT board for μPD7261A, μPD9306A, μPD72065 and μPD71066	8.67
EBIBM-7262	IBM-AT board for μPD7262 and μPD72067	8.71
Functional Assembler	Software package for μPD7281, functional Assembler,	
μPD7281	Simulator and Object Converter	8.73
SX-7281	Multibus boards for μPD7281	8.75
EBIBM-7281	IBM-XT/AT board for μPD7281 and μPD9305	8.77
CDMSD-15DD-7229	Character generation software for μPD7229 LCD controller	8.79

## Section 9 – Sentry Test Tape Overview

Sentry Test Tape .....	9.3
------------------------	-----

## Section 10 – Packaging Information

Package/Device Cross Reference .....	10.3
16-Pin Plastic DIP (300 mil) .....	10.5
18-Pin Plastic DIP (300 mil) .....	10.5
18-Pin Cerdip (300 mil) .....	10.6
20-Pin Plastic DIP (300 mil) .....	10.6
20-Pin Cerdip (300 mil) .....	10.7
20-Pin Plastic SO (Small Outline) .....	10.7
24-Pin Plastic DIP (600 mil) .....	10.8
24-Pin Plastic Skinny DIP (400 mil) .....	10.8
28-Pin Plastic DIP (600 mil) .....	10.9
28-Pin Plastic SO (Small Outline) .....	10.9
28-Pin PLCC (Plastic Leaded Chip Carrier) .....	10.10
30-Pin Plastic Shrink DIP (400 mil) .....	10.10
40-Pin Plastic DIP (600 mil) .....	10.11
40-Pin Ceramic DIP (600 mil) .....	10.11
40-Pin Cerdip (600 mil) .....	10.12
44-Pin Plastic Flat Package .....	10.12
44-Pin PLCC (Plastic Leaded Chip Carrier) .....	10.13
48-Pin Plastic DIP (600 mil) .....	10.13
48-Pin Ceramic DIP (600 mil) .....	10.14
52-Pin Plastic Flat Package .....	10.14
52-Pin PLCC (Plastic Leaded Chip Carrier) .....	10.15
64-Pin Plastic Flat Package .....	10.16
68-Pin PLCC (Plastic Leaded Chip Carrier) .....	10.17
69-Pin Ceramic PGA (Pin Grid Array) .....	10.17
80-Pin Plastic Flat Package .....	10.18
84-Pin PLCC (Plastic Leaded Chip Carrier) .....	10.19
94-Pin Plastic Flat Package .....	10.20
132-Pin Ceramic PGA (Pin Grid Array) .....	10.21

## GENERAL INFORMATION

1



## Section 1 – General Information

Introduction .....	1.3
CMOS Microprocessor Selection Guide .....	1.4
NMOS and HMOS Microprocessor Selection Guide .....	1.5
Intelligent Peripheral Controller Selection Guide .....	1.6
Magnetic Media Controllers .....	1.6
Communication Controllers .....	1.7
Graphics Controllers .....	1.7
CMOS System Support Product Selection Guide .....	1.8
NMOS System Support Product Selection Guide .....	1.9
$\mu$ PD70108/116/208/216/320/330 Software Development Tool Selection Guide .....	1.10
$\mu$ PD70108/116/208/216/320/330/616 Hardware Development Tool Selection Guide .....	1.11
$\mu$ PD70108/116/208/216/320/616 Evaluation Boards Selection Guide .....	1.11
$\mu$ PD70108/116/208/216/616 Operating System Selection Guide .....	1.12
$\mu$ PD70616/116/632 (V60/V70) Software Development Tool Selection Guide .....	1.12
Intelligent Peripherals Evaluation/Demonstration Tool Selection Guide .....	1.12
Dataflow Processor (IMPP) Hardware/Software Development Tool Selection Guide .....	1.12

### Introduction

NEC offers a wide variety of microprocessors and support products for you to choose from. This volume COVERS microprocessor, standard peripherals and intelligent peripheral controller products. These products are offered in a variety of processes (NMOS and CMOS) and in a variety of package types. This extraordinary selection of products provides the systems designer with a wide assortment of design alternatives with products that truly fit your data processing, communications, instrumentation, industrial, and telecommunications needs.

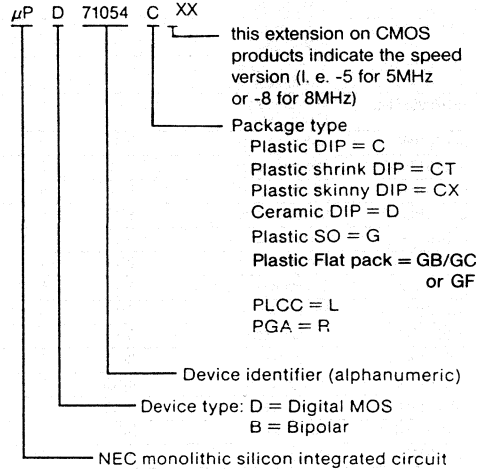
The NEC microcomputer data book is divided into the following sections.

- 1. General Information.** This section includes ordering information and product selection guides.
- 2. Quality and Reliability.** The NEC concepts of designed-in quality and total quality control as a company-wide activity are discussed here.
- 3. CMOS Microprocessors.** This section covers NEC's 8-bit, 16-bit and 32-bit lines of CMOS microprocessors. Included is NEC's proprietary V-Series of advanced CMOS microprocessors and co-processors.
- 4. NMOS Microprocessors.** This section covers NEC's 8-bit, 16-bit lines of NMOS microprocessors.
- 5. Intelligent Peripheral Controllers.** This section covers NEC's line of Magnetic Media, Communication and Graphic Controllers. Included are floppy disk, hard disk, disk support chips, serial controllers, LAN controller, and GPIB controller.
- 6. CMOS System Support Products.** This section covers NEC's line of CMOS System Support Products. These products are CMOS versions of most of the NMOS System Support Products. They have been specially designed for low cost, low power system needs.
- 7. NMOS System Support Products.** This section covers NEC's line of 8085/8088/8086 NMOS and Bipolar system support chips.
- 8. Development Tools and Software.** This section covers all Software- and Hardware Development Tools and all Development Aids (Evaluation Boards) as well as the Operating Systems available for NEC's microprocessors.
- 9. Overview on Sentry test tapes.** This section gives an overview of all test tapes which are available for Sentry VII system.
- 10. Packaging.** This section provides dimensioned package drawings and a cross-reference from package type to device numbers.

### Ordering Information

Part numbers for ordering microcomputer products are listed on the first page of each data sheet. NEC's part numbers consist of four elements as shown in the example that follows.

### Part Numbering System



## GENERAL INFORMATION

### CMOS Microprocessor Selection Guide

Device	Description	Data Bits	Clock (MHz)	Supply Voltage (V)	Power Dissipation		Package	Pins
					Active (mA)	Standby (mA)		
μPD70008C	Microprocessor (CMOS Z80)	8	4	+5	30	.5	Plastic DIP	40
μPD70008AC-4	Microprocessor (CMOS Z80)	8	4	+5	30	.4	Plastic DIP	40
μPD70008AC-6	Microprocessor (CMOS Z80)	8	6	+5	30	.4	Plastic DIP	40
μPD70008AG-4	Microprocessor (CMOS Z80)	8	4	+5	30	.4	Plastic Flat pack	44
μPD70008AG-6	Microprocessor (CMOS Z80)	8	6	+5	30	.4	Plastic Flat pack	44
μPD70008AL-6	Microprocessor (CMOS Z80)	8	6	+5	30	.4	PLCC	44
μPD70108C-5	Microprocessor	8/16	5	+5	30	5	Plastic DIP	40
μPD70108C-8	Microprocessor	8/16	8	+5	45	6	Plastic DIP	40
μPD70108C-10	Microprocessor	8/16	10	+5	60	7	Plastic DIP	40
μPD70108D-5	Microprocessor	8/16	5	+5	30	5	Cerdip DIP	40
μPD70108D-8	Microprocessor	8/16	8	+5	45	6	Cerdip DIP	40
μPD70108D-10	Microprocessor	8/16	10	+5	60	7	Cerdip DIP	40
μPD70108G-5	Microprocessor	8/16	5	+5	30	5	Plastic Flat pack	52
μPD70108G-8	Microprocessor	8/16	8	+5	45	6	Plastic Flat pack	52
μPD70108L-5	Microprocessor	8/16	5	+5	30	5	PLCC	44
μPD70108L-8	Microprocessor	8/16	8	+5	45	6	PLCC	44
μPD70116C-5	Microprocessor	16/16	5	+5	30	5	Plastic DIP	40
μPD70116C-8	Microprocessor	16/16	8	+5	45	6	Plastic DIP	40
μPD70116C-10	Microprocessor	16/16	10	+5	60	7	Plastic DIP	40
μPD70116D-5	Microprocessor	16/16	5	+5	30	5	Cerdip DIP	40
μPD70116D-8	Microprocessor	16/16	8	+5	45	6	Cerdip DIP	40
μPD70116D-10	Microprocessor	16/16	10	+5	60	7	Cerdip DIP	40
μPD70116G-5	Microprocessor	16/16	5	+5	30	5	Plastic Flat pack	52
μPD70116G-8	Microprocessor	16/16	8	+5	45	6	Plastic Flat pack	52
μPD70116L-5	Microprocessor	16/16	5	+5	30	5	PLCC	44
μPD70116L-8	Microprocessor	16/16	8	+5	45	6	PLCC	44
μPD70208R-8	Microprocessor	8/16	8	+5	60	12	PGA	68
μPD70208L-8	Microprocessor	8/16	8	+5	60	12	PLCC	68
μPD70208G-8	Microprocessor	8/16	8	+5	60	12	Plastic Flat pack	80
μPD70208R-10	Microprocessor	8/16	10	+5	80	16	PGA	68
μPD70208L-10	Microprocessor	8/16	10	+5	80	16	PLCC	68
μPD70208G-10	Microprocessor	8/16	10	+5	80	16	Plastic Flat pack	80
μPD70216R-8	Microprocessor	16/16	8	+5	60	12	PGA	68
μPD70216L-8	Microprocessor	16/16	8	+5	60	12	PLCC	68
μPD70216G-8	Microprocessor	16/16	8	+5	60	12	Plastic Flat pack	80
μPD70216R-10	Microprocessor	16/16	10	+5	80	16	PGA	68
μPD70216L-10	Microprocessor	16/16	10	+5	80	16	PLCC	68
μPD70216G-10	Microprocessor	16/16	10	+5	80	16	Plastic Flat pack	80
μPD70616R-16	Microprocessor	16/32	16	+5	450	-	PGA	68

### NMOS and HMOS Microprocessor Selection Guide

Device	Description	Data Bits	Clock (MHz)	Supply Voltage (V)	Power Dissipation		Package	Pins
					Active (mA)	Standby (mA)		
$\mu$ PD780C	NMOS Microprocessor	8	2.5	+5	150	—	Plastic DIP	40
$\mu$ PD780C-1	NMOS Microprocessor	8	4	+5	200	—	Plastic DIP	40
$\mu$ PD780C-2	NMOS Microprocessor	8	6	+5	200	—	Plastic DIP	40
$\mu$ PD8085AC-2	NMOS Microprocessor	8	5	+5	170	—	Plastic DIP	40
$\mu$ PD8085AHC	HMOS Microprocessor	8	3	+5	135	—	Plastic DIP	40
$\mu$ PD8085AHC-2	HMOS Microprocessor	8	5	+5	135	—	Plastic DIP	40
$\mu$ PD8086D	HMOS Microprocessor	16	5	+5	340	—	Cerdip	40
$\mu$ PD8086D-2	HMOS Microprocessor	16	8	+5	350	—	Cerdip	40
$\mu$ PD8088D	HMOS Microprocessor	8	5	+5	340	—	Cerdip	40
$\mu$ PD8088D-2	HMOS Microprocessor	8	8	+5	350	—	Cerdip	40

## GENERAL INFORMATION

### Intelligent Peripheral Controller Selection Guide

Device	Description	Data Bits	Clock (MHz)	Supply Voltage (V)	Power Dissipation		Package	Pins
					Active (mA)	Standby (mA)		
<b>Magnetic Media Controllers</b>								
μPD765AC	NMOS Single/ Double Density FDC	8	8	+5	150	—	Plastic DIP	40
μPD765AC-2	NMOS Single/ Double Density FDC	8	8	+5	150	—	Plastic DIP	40
μPD7265C	NMOS Single/ Double Density FDC	8	8	+5	140	—	Plastic DIP	40
μPD7265C-2	NMOS Single/ Double Density FDC	8	8	+5	140	—	Plastic DIP	40
μPD72065C	CMOS Single/ Double Density FDC	8	8	+5	10	.5	Plastic DIP	40
μPD72065G	CMOS Single/ Double Density FDC	8	8	+5	10	.5	Plastic Flat pack	52
μPD72065L	CMOS Single/ Double Density FDC	8	8	+5	10	.5	PLCC	44
μPD72066C	CMOS Single/ Double Density FDC	8	8	+5	10	.5	Plastic DIP	40
μPD72066G	CMOS Single/ Double Density FDC	8	8	+5	10	.5	Plastic Flat pack	52
μPD72066L	CMOS Single/ Double Density FDC	8	8	+5	10	.5	PLCC	44
μPB9201C	Bipolar Floppy Disk Interface	—	16	+5	270	—	Plastic DIP	40
μPD71065G	CMOS Analog Phase-Locked Loop	—	19.2	+5	25	—	Plastic SO	28
μPD71066CT	CMOS Analog Phase-Locked Loop	—	19.2	+5	25	—	Plastic Shrink DIP	30
μPD72067	Advanced Floppy Disk Controller	8	19.2/ 32	+5	170	—	Plastic DIP PLCC	48 52
μPD7260D	NMOS Hard/Floppy Disk Controller	8	12	+5	320	—	Ceramic DIP	40
μPD7261AD	NMOS Hard Disk Controller	8	12	+5	320	—	Ceramic DIP	40
μPD7261BD-18 BD-23	NMOS Hard Disk Controller	8	18 23	+5	320	—	Ceramic DIP	40
μPD9306AC	CMOS Hard Disk Interface	—	10	+5	30	—	Plastic DIP	28
μPD7262D-12 D-18	NMOS ESDI Hard Disk Controller	8	12 18	+5	320	—	Ceramic DIP	40

### Dataflow Processing

Device	Description	Data Bits	Clock (MHz)	Supply Voltage (V)	Power Dissipation		Package	Pins
					Active (mA)	Standby (mA)		
μPD7281D	NMOS Image Pipelined Processor	16	5	+5	500	—	Ceramic DIP	40
μPD9305R	CMOS IMPP Support Chip	8/16	10	+5	100	—	PGA	132

### Intelligent Peripheral Controller Selection Guide (cont)

Device	Description	Data Bits	Clock (MHz)	Supply Voltage (V)	Power Dissipation		Package	Pins
					Active (mA)	Standby (mA)		
<b>Communications Controllers</b>								
μPD7201AC	NMOS MPS Communications Controller	8	4	+5	230	—	Plastic DIP	40
μPD7201AD	NMOS MPS Communications Controller	8	4	+5	230	—	Ceramic DIP	40
μPD72001C	CMOS MPS Communications Controller	8	8	+5	40	2	Plastic DIP	40
μPD72001C-11			11					
μPD72001L	CMOS MPS Communications Controller	8	8	+5	40	2	PLCC	44
μPD72105C	CMOS Omnet Local Network Controller	8	8	+5	40	2	Plastic DIP	48
μPD72105L	CMOS Omnet Local Network Controller	8	8	+5	40	2	PLCC	52
μPD7210C	NMOS Intelligent GPIB Controller for Listener, Talker, Controller	8	8	+5	180	—	Plastic DIP	40
<b>Graphics Controllers</b>								
μPD7220AD	NMOS Graphics Display Controller	8	6	+5	270	—	Ceramic DIP	40
μPD7220AD-1	NMOS Graphics Display Controller	8	7	+5	270	—	Ceramic DIP	40
μPD7220AD-2	NMOS Graphics Display Controller	8	8	+5	270	—	Ceramic DIP	40
μPD72022	CMOS Intelligent Display Processor	8	8	+5	150	—	PLCC	68
μPD72120	CMOS Advanced Graphic Display Processor	8/16	8	+5	120	—	PLCC	84
μPD7225G-00	LCD 7/14 Segment Controller/Driver	8 (serial)	0.2	2.7-5.5	0.1	—	Plastic Flat, Bent Leads	52
μPD7225G-01	LCD 7/14 Segment Controller/Driver	8 (serial)	0.2	2.7-5.5	0.1	—	Plastic Flat, Straight Leads	52
μPD7227G-11	LCD 8 x 40 Dot Matrix Controller/Driver	8 (serial)	1.1	5	0.1	—	Plastic Flat, Straight Leads	64
μPD7227G-12	LCD 8 X 40 Dot Matrix Controller/Driver	8 (serial)	1.0	5	0.1	—	Plastic Flat, Bent Leads	64
μPD7228G-12	LCD 8 x 50/16 x 42 Dot Matrix Controller/Driver	8 (serial) 4 (parallel)	1.1	5	0.2	—	Plastic Flat, Bent Leads	80
μPD7228AG-12	LCD 8 x 50/16 x 42 Dot Matrix Controller/Driver with 12.5 V for VLCD	8 (serial) 4 (parallel)	1.1	5	0.2	0.02	Plastic Flat, Bent Leads	80
μPD7229G-12	LCD 8 x 50/16 x 42 Dot Matrix Controller Driver with Customized Character Set	8 (serial) 4 (parallel)	1.1	5	0.2	0.02	Plastic Flat, Bent Leads	80

# GENERAL INFORMATION

## CMOS System Support Product Selection Guide

Device	Description	Data Bits	Clock (MHz)	Supply Voltage (V)	Power Dissipation		Package	Pins
					Active (mA)	Standby (mA)		
μPD71011C	Clock Pulse Generator/Driver	—	20	+5	30	.2	Plastic DIP	18
μPD71011C-10	Clock Pulse Generator/Driver	—	20	+5	30	.2	Plastic DIP	18
μPD71011G	Clock Pulse Generator/Driver	—	20	+5	30	.2	Plastic SO	20
μPD71051C	Serial Control Unit	8	8	+5	10	.05	Plastic DIP	28
μPD71051G	Serial Control Unit	8	8	+5	10	.05	Plastic Flat pack	44
μPD71051L	Serial Control Unit	8	8	+5	10	.05	PLCC	28
μPD71054C	Programmable Timer/Counter	8	8	+5	20	.05	Plastic DIP	24
μPD71054G	Programmable Timer/Counter	8	8	+5	20	.05	Plastic Flat pack	44
μPD71054L	Programmable Timer/Counter	8	8	+5	20	.05	PLCC	28
μPD71055C	Parallel Interface Unit	8	8	+5	10	.05	Plastic DIP	40
μPD71055G	Parallel Interface Unit	8	8	+5	10	.05	Plastic Flat pack	44
μPD71055L	Parallel Interface Unit	8	8	+5	10	.05	PLCC	44
μPD71059C	Interrupt Control Unit	8	8	+5	9	.05	Plastic DIP	28
μPD71059G	Interrupt Control Unit	8	8	+5	9	.05	Plastic Flat pack	44
μPD71059L	Interrupt Control Unit	8	8	+5	9	.05	PLCC	28
μPD71071C	DMA Controller	8/16	10	+5	15	.01	Plastic DIP	48
μPD71071D	DMA Controller	8/16	8	+5	15	.01	Ceramic DIP	48
μPD71071C	DMA Controller	8/16	10	+5	15	.01	Plastic Flat pack	52
μPD71071L	DMA Controller	8/16	10	+5	15	.01	PLCC	52
μPD71082C	Transparent Latch	8	8	+5	20	.08	Plastic DIP	20
μPD71082G	Transparent Latch	8	8	+5	20	.08	Plastic SO	20
μPD71083C	Transparent Latch, inverted	8	8	+5	20	.08	Plastic DIP	20
μPD71083G	Transparent Latch, inverted	8	8	+5	20	.08	Plastic SO	20
μPD71084C	Clock Pulse Generator/Driver	—	25	+5	30	.2	Plastic DIP	18
μPD71084G	Clock Pulse Generator/Driver	—	25	+5	30	.2	Plastic SOP	20
μPD71086C	Bus Buffer/Driver	8	8	+5	40	.08	Plastic DIP	20
μPD71086G	Bus Buffer/Driver	8	8	+5	40	.08	Plastic SO	20
μPD71087C	Bus Buffer/Driver, inverted	8	8	+5	40	.08	Plastic DIP	20
μPD71087G	Bus Buffer/Driver, inverted	8	8	+5	40	.08	Plastic SO	20
μPD71088C	System Bus Controller	—	8	+5	20	.08	Plastic SO	20
μPD71088C-10	System Bus Controller	—	10	+5	20	.08	Plastic DIP	18
μPD71088G	System Bus Controller	—	8	+5	20	.08	Plastic SO	20
μPD71611C	Clock Pulse Generator/Driver for V60	—	32	+5	50	.2	Plastic DIP	20
μPD71613C	System Bus Generator for V60	—	16	+5	30	.08	Plastic DIP	20
μPD82C43C	Input/Output Expander	—	5	+5	40	—	Plastic DIP	24
μPD82C43G	Input/Output Expander	—	5	+5	40	—	Plastic SO	24

### NMOS System Support Product Selection Guide

Device	Description	Data Bits	Clock (MHz)	Supply Voltage (V)	Power Dissipation		Package	Pins
					Active (mA)	Standby (mA)		
μPD8155C	256 x 8 RAM with I/O Ports and Timer	8	3	4.5 to 5.5	180	—	Plastic DIP	40
μPD8155C-2	256 x 8 RAM with I/O Ports and Timer	8	5	4.5 to 5.5	180	—	Plastic DIP	40
μPD8155HC	256 x 8 RAM with I/O Ports and Timer	8	3	4.5 to 5.5	180	—	Plastic DIP	40
μPD8155HC-2	256 x 8 RAM with I/O Ports and Timer	8	5	4.5 to 5.5	180	—	Plastic DIP	40
μPD8156C	256 x 8 RAM with I/O Ports and Timer	8	3	4.5 to 5.5	180	—	Plastic DIP	40
μPD8156C-2	256 x 8 RAM with I/O Ports and Timer	8	5	4.5 to 5.5	180	—	Plastic DIP	40
μPD8156HC	256 x 8 RAM with I/O Ports and Timer	8	3	4.5 to 5.5	180	—	Plastic DIP	40
μPD8156HC-2	256 x 8 RAM with I/O Ports and Timer	8	5	4.5 to 5.5	180	—	Plastic DIP	40
μPD8216C	Parallel Bidirectional Bus Driver	4	—	4.5 to 5.5	130	—	Plastic DIP	16
μPB8216C	Parallel Bidirectional Bus Driver	4	—	4.5 to 5.5	130	—	Plastic DIP	16
μPB8226C	Parallel Bidirectional Bus Driver	4	—	4.5 to 5.5	120	—	Plastic DIP	16
μPD8237AC-5	Programmable DMA Controller	8	5	4.5 to 5.5	150	—	Plastic DIP	40
μPD8243C	Input/Output Expander	—	5	4.5 to 5.5	20	—	Plastic DIP	24
μPD8243HC	HMOS Input/Output Expander	—	5	4.5 to 5.5	20	—	Plastic DIP	24
μPD8251AC	Programmable Communication Interface	8	3/5	4.5 to 5.5	100	—	Plastic DIP	28
μPD8251AFC	Programmable Communication Interface	8	3/5	4.5 to 5.5	100	—	Plastic DIP	28
μPD8253C-2	Programmable Internal Timer	8	5	4.5 to 5.5	140	—	Plastic DIP	24



## GENERAL INFORMATION

### NMOS System Support Product Selection Guide (cont)

Device	Description	Data Bits	Clock (MHz)	Supply Voltage (V)	Power Dissipation		Package	Pins
					Active (mA)	Standby (mA)		
$\mu$ PD8259AC-2	Programmable Interrupt Controller	8	5	4.5 to 5.5	85	—	Plastic DIP	28
$\mu$ PD8279C-2	Programmable Keyboard/Display Interface	—	5	4.5 to 5.5	120	—	Plastic DIP	40
$\mu$ PB8282C	Octal Latch	8	8	4.5 to 5.5	160	—	Plastic DIP	20
$\mu$ PB8283C	Octal Latch	8	8	4.5 to 5.5	160	—	Plastic DIP	20
$\mu$ PB8284AD	Clock Generator/Driver	—	24	4.5 to 5.5	140	—	Cerdip	18
$\mu$ PB8286C	Octal Bus Transceiver	8	8	4.5 to 5.5	160	—	Plastic DIP	20
$\mu$ PB8287C	Octal Bus Transceiver	8	8	4.5 to 5.5	130	—	Plastic DIP	20
$\mu$ PB8288D	CPU System Bus Controller	—	10	4.5 to 5.5	230	—	Cerdip	20
$\mu$ PB8289D	Bus Arbiter	—	8	4.5 to 5.5	165	—	Cerdip	20

### $\mu$ PD70108/116/208/216/320/330 (V20/V30/V40/V50/V25/V35) Software Development Tool Selection Guide

#### Relocatable Assembler for $\mu$ PD70108/116/208/216

Device	Host	Operating System
RAC86-15DD-70116	IBM-XT	CP/M-86 *
RAPCD-15DD-70116	IBM-XT	PC-DOS*
RASPD-15DD-70116	PCD (Siemens)	MS-DOS*
RAIS3-08SD-70116	Intel MDS	ISIS3*
RAUNX-0T16-70116	VAX	ULTRIX*
RAVMS-0T16-70116	VAX	VMS*

#### Relocatable Assembler for $\mu$ PD70320/330

Device	Host	Operating System
RAC86-15DD-70320	IBM-XT	CP/M-86 *
RAPCD-15DD-70320	IBM-XT	PC-DOS*
RASPD-15DD-70320	PCD (Siemens)	MS-DOS*
RAIS3-08SD-70320	Intel MDS	ISIS3*
RAUNX-0T16-70320	VAX	ULTRIX*
RAVMS-0T16-70320	VAX	VMS*

#### Object Format Converter

Device	Host	Operating System
RCMSD-15DD-70116	IBM-XT/AT	MS-DOS*
RCIS3-08SD-70116	Intel MDS	ISIS III*
RCVMS-0T16-70116	VAX	VMS*
RCUNX-0T16-70116	VAX	ULTRIX*

#### C-Compiler

Device	Host	Operating System
CCC86-15DD-70116	IBM-XT	CP/M-86 *
CCPCD-15DD-70116	IBM-XT	PC-DOS *
CCSPD-15DD-70116	PCD (Siemens)	MS-DOS *
CCIS3-08SD-70116	Intel MDS	ISIS3 *
CCUNX-0T16-70116	VAX	ULTRIX *
CCVMS-0T16-70116	VAX	VMS *

\*CP/M-86 is a registered trade mark of DIGITAL Research Inc.

\*PC-DOS is a registered trade mark of IBM Corp.

\*ISIS is a registered trade mark of INTEL Corp.

\*VAX/ULTRIX/VMS is a registered trade mark of DEC. Inc.

\*MS-DOS is a registered trade mark of MICROSOFT Inc.

### μPD70108/116/208/216/320/330/616 (V20/V30/V40/V50/V25/V35/V60) Hardware Development Tool Selection Guide

#### In-circuit Emulator IE-70k

Device	Description
IE-70108-S	In circuit emulator for μPD70108 (with V20 pod)
IE-70116-S	In-circuit emulator for μPD70116 (with V30 pod)
IE-70208-S008	In-circuit emulator for μPD70208 (with V40 pod)
IE-70216-S008	In-circuit emulator for μPD70216 (with V50 pod)
IE-70108-001	Optional pod unit for μPD70108 emulation
IE-70116-001	Optional pod unit for μPD70116 emulation
IE-70208-1008	Optional pod unit for μPD70208 emulation
IE-70216-1008	Optional pod unit for μPD70216 emulation
IE-70116-1508	Converts IE-70208/70216-S008 to IE-70108/70116-S
IE-70216-1508	Converts IE-70108/70116-S to IE-70208/70216-S008

#### PGA – PLCC Adapters

ADAPT68PGA68PLCC	Adapts V40/V50 pod (PGA plug) to PLCC socket
ADAPT84PGA84PLCC	Adapts 84 pin PGA to 84 pin PLCC socket

#### Mini Emulators for μPD70208/216/320/330

Device	Description
EBIBM-V40 MINI-IE	Low cost Mini Emulator for PC to debug V40
EBIBM-V50 MINI-IE	Low cost Mini Emulator for PC to debug V50
EBIBM-V25 MINI-IE	Low cost Mini emulator for PC to debug V25
EBIBM-V35 MINI-IE	Low cost Mini Emulator for PC to debug V35

### μPD70108/116/208/216/320/616 (V20/V30/V40/V50/V25/V60) Evaluation Boards Selection Guide

Device	Description
EBIBM-70108	Development board (SBC) for V20 with ROM based monitor
EBIBM-70116	Development board (SBC) for V30 with ROM based monitor
EBIBM-70208	Development board (SBC) for V40 with ROM based monitor
EBIBM-70216	Development board (SBC) for V50 with ROM based monitor
EBIBM-70320	Development/Evaluation board (SBC) for V25 with ROM based Monitor
EBIBM-70616UNIX	V60 UNIX board for IBM-PC AT/XT (as slave card) including operating system UNIX V
EBIBM-70616SBC	Multibus I board with V60 CPU

#### In-circuit Emulator IE-700k

Device	Description
IE-70108-A010	In-circuit emulator for μPD70108 (with V20 pod)
IE-70116-A010	In-circuit emulator for μPD70116 (with V30 pod)
IE-70208-A008	In-circuit emulator for μPD70208 (with V40 pod)
IE-70216-A008	In-circuit emulator for μPD70216 (with V50 pod)
IE-70320-A008	In-circuit emulator for μPD70320 (with V25 pod)
IE-70330-A008	In-circuit emulator for μPD70320 (with V35 pod)
IE-70616-A016	In-circuit emulator for μPD70616 (with V60 pod)
IE-70108-2010	Optional pod unit for μPD70108 emulation
IE-70116-2010	Optional pod unit for μPD70116 emulation
IE-70208-2008	Optional pod unit for μPD70208 emulation
IE-70216-2008	Optional pod unit for μPD70216 emulation
IE-70320-2008	Optional pod unit for μPD70320 emulation
IE-70330-2008	Optional pod unit for μPD70330 emulation
IE-70616-2008	Optional pod unit for μPD70616 emulation
IE-70000-2954	Logic probe to support external trace and break input
IE-70000-2957	Emulation expansion memory (512kb/no wait states)
IE-70000-2960	System expansion memory (512 kB/no wait states)
IE-70000-2958	68-pin PLCC attachment for IE-V40/V50
IE-70000-2959	68-pin PGA attachment for IE-V40/V50
EP-70320L	80-pin PLCC attachment for IE-V25/V35
EP-70616R	68-pin PGA attachment for IE-V60

## GENERAL INFORMATION

### μPD70108/116/208/216/616(V20/V30/V40/V50/V60) Operating System Selection Guide

Device	Description	Host for porting
RXMSD-ISDD-70116	Real-time operating system for V20, V30, V40, V50 (RX116)	IBM-XT/AT with PC-DOS
RXUNX-OT16-70116	Real-time operating system for V20, V30, V40, V50 (RX116)	VAX under ULTRIX
RXVMS-OT16-70116	Real-time operating system for V20, V30, V40, V50 (RX116)	VAX under VMS
RXUNX-OT16-70616	Real-time operating system for V60, V70 (RX616)	VAX under ULTRIX
RXVMS-OT16-70616	Real-time operating system for V60, V70 (RX616)	VAX under VMS

### μPD70616/632 (V60/V70) Software Development Tool Selection Guide

Device	Host	Operating System	Description
PKGUX-OT16-70616	VAX	ULTRIX	Software Package for V60/V70 with Rel. Assembler, C-Compiler
PKGVS-OT16-70616	VAX	VMS	Linker, Configurator plus Libraries

### Evaluation/Demonstration Tools Hardware/Software Selection Guide

Device	Product Description
<b>Communications:</b>	
EBIBM-72001	72001 demo/evaluation board for IBM XT, can also be used in an IBM AT, includes software for HDLC, SDLC, ASYNC, BISYNC
<b>Graphics:</b>	
EBIBM-7220A	7220A and video RAM demo/evaluation board for IBM XT and IBM AT, comes with demo program
EBIBM-72020	72020 and video RAM demo/evaluation board for IBM XT and IBM AT
EBIBM-72120	72120 - advanced graphics display controller board for IBM XT with demo SW
CGMSD-ISDD-7229	μPD7229 character generation and verification software
<b>Magnetics</b>	
EBIBM-7261 A	7261A hard disk controller demo/evaluation board for IBM XT. Evaluation software is also included.
EBIBM-ATDISK	Hard disk (7261A) and floppy disk (72065) demo/evaluation board for IBM AT.
EBIBM-7262	Hard disk (7262) and floppy disk (72067) demo/evaluation board

### Dataflow Processor (IMPP) Development Tools Hardware/Software Selection Guide

Device	Description
EBIBM-7281GS	7281 (Dataflow Processor) based image processing board including frame grabber and camera input for IBM PC AT/XT, includes menu driven demo software
FAC86-08SS-7281	Functional assembler running under CP/M (on 8" floppy) including simulator
FAC86-ISDD-7281	Functional assembler running under CP/M-86 (on 5 1/4" floppy)
FAMSD-ISDD-7281	Functional assembler running under MS-DOS (on 5 1/4" floppy)
FA UNX-OT16-7281	Functional assembler running under UNIX (Tape 1600BPI)
FA VMS-OT16-7281	Functional assembler running under VMS (Tape 1600 BPI)
SX-7281-M	Multibus II compatible execution unit for debugging software written for 7281 (IMPP)
SX-7281-S	Standalone execution unit for debugging software written for 7281 (IMPP)

## QUALITY AND RELIABILITY

2

## **Section 2 – Quality and Reliability**

Introduction .....	2.3
Technology Description .....	2.3
Reliability Testing .....	2.3
Failure Rate Calculation and Prediction .....	2.6
Reliability Test Results .....	2.7
NEC's Goals on Failure Rates .....	2.7
Infant Mortality Failure Screening .....	2.8
Life Tests.....	2.8
Built-in Quality and Reliability .....	2.10
Approaches to Total Quality Control .....	2.10
Summary and Conclusion .....	2.12

### Introduction

As large-scale integration reaches a higher level of density, reliability of devices imposes a profound impact on system reliability. And as device reliability becomes a major factor, test methods to assure acceptable reliability become more complicated. Simply performing a reliability test according to a conventional method cannot satisfy the demanding requirements for higher reliability. At these new, higher levels of LSI density, it is increasingly difficult to activate all the elements in the internal circuits. A different philosophy and methodology is needed for reliability assurance. Moreover, as integration density increases, the degradation of internal elements in an LSI device is seldom detected by measuring characteristics across external terminals.

In order to improve and guarantee a certain level of reliability for large-scale integrated circuits, it is essential to build quality and reliability into the product. Then, the conventional reliability tests are followed to ensure that the product demonstrates an acceptable level of reliability.

NEC has introduced the concept of total quality control (TQC) across its entire semiconductor product line. By adopting TQC, NEC can build quality into the product and thus assure higher reliability. The concept and methodology of total quality control are company-wide activities involving workers, engineers, quality control staffs, and all levels of management.

NEC has also introduced a prescreening method into the production line that helps eliminate potentially defective units. The combination of building quality in and screening projected early failures out has resulted in superior quality and excellent reliability.

### Technology Description

Most large-scale integrated circuits utilize high-density, MOS technology. State-of-the-art high performance has been achieved by introducing fine-line generation techniques. By reducing physical parameters, circuit density and performance increase while active circuit power dissipation decreases. The data presented here shows that this advanced technology yields products as reliable as those from previous technologies.

### Reliability Testing

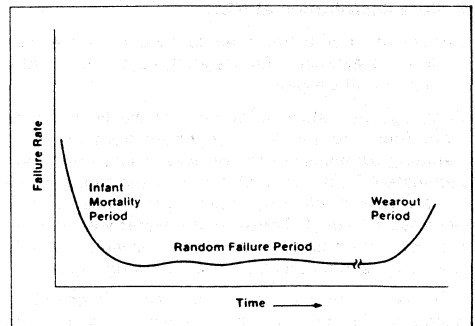
Reliability is defined as the characteristics of an item expressed by the probability that it will perform a required function under stated conditions for a stated period of time. This involves the concept of probability, definition of required function(s), and the critical time used in defining the reliability.

Definition of a required function, by implication, treats the definition of a failure. Failure is defined as the termination of the ability of a device to perform its required function. Furthermore, a device is said to have failed if it shows inability to perform within guaranteed parameters as given in an electrical specification.

Discussion of reliability and failure can be approached in two ways: with respect to systems or to individual devices. The accumulation of normal device failure rates constitutes the expected failure rate of the system hardware. Important considerations here are the constant failure period, the early failure (infant mortality) period, and overall reliability level. With regard to individual devices, areas of prime interest include specific failure mechanisms, failures in accelerated tests, and screening tests.

Some of these failure considerations pertain to both systems and devices. The probability of no failures in a system is the product of the probability of no failure in each of its components. The failure rate of system hardware is then the sum of the failure rates of the components used to construct the system.

Figure 1. Reliability Life (Bathtub) Curve



**Life Distribution**

The fundamental principles of reliability engineering predict that the failure rate of a group of devices will follow the well-known bathtub curve in figure 1. The curve is divided into three regions: infant mortality, random failures, and wearout failures.

Infant mortality, as the name implies, represents the early-life failures of devices. These failures are usually associated with one or more manufacturing defects.

After some period of time, the failure rate reaches a low value. This is the random failure portion of the curve, representing the useful portion of the life of a device. During this random failure period, there is a decline in the failure rate due to the depletion of potential random failures from the general population.

The wearout failures occur at the end of the device's useful life. They are characterized by a rapidly rising failure rate over time as devices wear out both physically and electrically.

Thus, for devices that have very-long life expectancies compared to those of systems, the areas of concern will be the infant mortality and the random failure portions of the population.

The system failure rates are related to the collective device failure rates. In a given system, after elimination of the early failures, the system will be left to the failure rate of its components. In order to make proper projections of the failure rate in the operating environment, time-to-failure must be accelerated in tests in a predictable way.

**Failure Distribution at NEC**

Integrated circuits returned to NEC from the field underwent extensive failure analysis at NEC's Integrated Circuit Division.

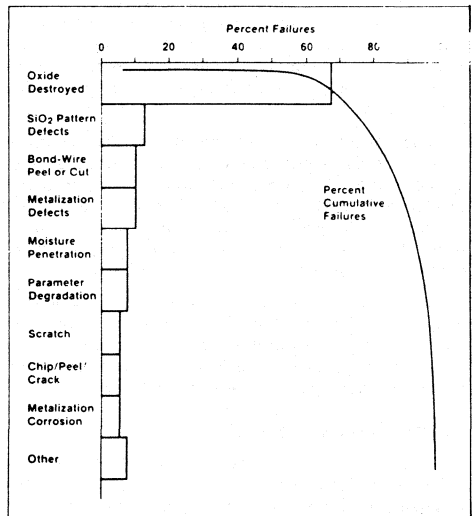
First, approximately 50 percent of the field returns were found to be damaged either from improper handling or misuse of the devices. These units were eliminated from the analysis. The remaining failed units were classified by their failure mechanisms as depicted in figure 2. These failures were then related to the major integrated circuit failure mechanisms and to their origins in a particular manufacturing step.

As shown in figure 2, the first four failure mechanisms accounted for more than 90 percent of total failures. As a result, NEC improved processes and material to reduce these failures. Additionally, NEC introduced screening procedures to detect and eliminate defective devices.

Temperature, humidity, and bias tests are used for testing the moisture resistance of plastic encapsulated integrated circuits. NEC developed a special process to improve the plastic encapsulation material. As a result, moisture-related—thus packaging-related—failures have been drastically reduced.

As a preventive measure, NEC has introduced a special screening procedure embedded in the production line. A burn-in at an elevated temperature is performed for 100 percent of the lots. This burn-in effectively removes the potentially defective units. In addition, improvement of the plastic encapsulation material has lowered the failures in a high-temperature and high-humidity environment.

**Figure 2. Failure Distribution of MOS Integrated Circuits**



### Accelerated Reliability Testing

As an example, assume that an electronic system contains 1000 integrated circuits and can tolerate 1 percent system failures per month. The failure rate per component is:

$$\frac{0.01 \text{ Failures}}{720\text{K Device Hours}} = 13.888 \times 10^{-9} \text{ Failures/Hour} \text{ or } 13.8888 \text{ FITs}$$

where FIT = Failure units per  $10^9$  device hours

To demonstrate this failure rate, note that 13.8888 FITs corresponds to one failure in about 7000 devices during an operating test of 10,000 hours. It is quickly apparent that a test condition is required to accelerate the time-to-failure in a predictable and understandable way. The implicit requirement for the accelerated stress test is that the relationship between the accelerated stress testing condition and the condition of actual use be known.

A most common time-to-failure relationship involves the effect of temperature, which accelerates many physiochemical reactions leading to device failure. Other environmental conditions are voltage, current, humidity, vibration, or some combination of these. Table 1 lists the reliability assurance tests performed at NEC for integrated circuits.

**Table 1. Monthly NEC Reliability Tests**

Test	MIL-STD-883 Method	Test Conditions
<b>Life Test</b>		
High-temperature, operating	1005A, D	$T_A = 100$ to $125^\circ\text{C}$ for 1000 hours
High-temperature, storage	1008C	$T_A = 150^\circ\text{C}$ for 1000 hours
High-temperature, high-humidity test	—	$T_A = 85^\circ\text{C}$ at 85% RH for 1000 hours
Pressure cooker test	—	$T_A = 125^\circ\text{C}$ at 2.3 atm for 168 hours
<b>Environmental Test</b>		
Soldering heat test	2031 (MIL-STD-750)	$T = 260^\circ\text{C}$ for 10 s without flux
Temperature cycle	1010C	$T = -65$ to $+150^\circ\text{C}$ for 10 cycles
Thermal shock	1011A	$T = 0$ to $100^\circ\text{C}$ for 15 cycles
Lead fatigue	2004B2	at 250 gm: 3 leads, 3 bends
Solderability	2003	$T = 230^\circ\text{C}$ for 5 s with flux

**Temperature Effect.** The effect of temperature that concerns us is that which responds to the Arrhenius relationship. This relates the reaction rate to temperature.

$$R = R_0 \exp(-E_a/kT)$$

where  $R_0$  = Constant  
 $E_a$  = Activation energy in eV  
 $k$  = Boltzmann's constant =  $8.617 \times 10^{-5}$  eV/K  
 $T$  = Absolute temperature in kelvin (K)

The significance of this relationship is that the failure mechanisms of semiconductor devices are directly applicable to it. A linear relationship between failure mechanism and time is assumed.

**Activation Energy.** Associated with each failure mechanism is an activation energy value. Table 2 lists some of the more common failure mechanisms and the associated activation energy of each.

**Table 2. Activation Energy and Detection of Failure Mechanisms**

Failure Mechanism	Activation Energy	Detection
Oxide defect	0.3 eV	High-temperature operating life test
Silicon defect	0.3 eV	
Ionic contamination	1.0-1.35 eV	
Electromigration	0.4-0.8 eV	
Charge injection	1.3 eV	
Gold-aluminum interface	0.8 eV	
Metal corrosion	0.7 eV	High-humidity operating life test

**High-Temperature Operating Life Test.** This test is used to accelerate failure mechanisms by operating the devices at an elevated temperature of  $125^\circ\text{C}$ . The data obtained is translated to a lower temperature by using the Arrhenius relationship.

**High-Temperature and High-Humidity Test.** Semiconductor integrated circuits are highly sensitive to the general accelerating effect of humidity in causing electrolytic corrosion between biased lines. The high-temperature and high-humidity test is performed to detect failure mechanisms that are accelerated by these conditions. This test is effective in accelerating leakage-related failures and drifts in device parameters due to process instability.



## QUALITY AND RELIABILITY

**High-Temperature Storage Test.** Another common test is the high-temperature storage test in which devices are subjected to elevated temperatures with no applied bias. This test is used to detect mechanical problems and process instability.

**Environmental Test.** Other environmental tests are performed to detect problems related to the package, material, susceptibility to extremes in environment, and problems related to usage of the devices.

### Failure Rate Calculation and Prediction

Analysis of integrated circuit failure rates can serve many useful purposes. For example, the early-life failure rate helps establish a warranty period, while the mature-life failure rate aids in estimating repair costs, spare parts stock requirements, or product downtime. Accurate prediction of failure rates can also be used for process control.

The following sections describe the failure rate calculation and prediction methods used by NEC's Integrated Circuit Division.

### The Arrhenius Model

Most integrated circuit failure mechanisms depend to some degree on temperature. This relationship can be represented by the Arrhenius model, which includes the effects of temperature and activation energy of the failure mechanisms.

As applied to accelerated life testing of integrated circuits, the Arrhenius model assumes that degradation of a performance parameter is linear with time. Temperature dependence is taken to be the exponential function that defines the probability of occurrence. The relationship of failure rate to temperature is expressed as:

$$F_1 = F_2 \exp\left[\frac{E_a}{k} \left(\frac{1}{T_1} - \frac{1}{T_2}\right)\right]$$

Where:  $F_2$  = Failure rate at  $T_2$   
 $F_1$  = Failure rate at  $T_1$   
 $E_a$  = Activation energy in eV  
 $k$  = Boltzmann's constant  
 $T$  = Operating junction temperature in kelvin (K)

The equation explains the thermal dependence of integrated circuit failure rates and is used for derating the resulting failure rate to a more realistic temperature.

### Acceleration Factor

The acceleration factor is the factor by which the failure rate can be accelerated by increased temperature. This factor is derived from the Arrhenius failure rate expression, resulting in the following form.

$$A = F_1/F_2 = \exp\left[\frac{E_a}{k} \left(\frac{1}{T_1} - \frac{1}{T_2}\right)\right]$$

where  $A$  = Acceleration factor  
 $F_2$  = Failure rate at  $T_2$   
 $F_1$  = Failure rate at  $T_1$

In calculating the field reliability of an integrated circuit, it is necessary to calculate the junction temperature. In general, the junction temperature will depend on the ambient temperature, cooling, package type, operating cycle time, and power dissipation of the circuit itself. In these terms, the junction temperature ( $T_J$ ) is expressed as:

$$T_J = T_A + P_d A_f \theta_{JA}$$

where  $T_J$  = Junction temperature  
 $T_A$  = Ambient temperature  
 $P_d$  = Power dissipation  
 $A_f$  = Air flow factor  
 $\theta_{JA}$  = Package thermal resistance

Table 3 lists derating factors of various failure mechanisms. This table is generated assuming that an accelerated test is performed at a junction temperature of 125°C. The result is then derated to 55°C junction temperature. The acceleration factor may then be obtained by taking the inverse of the derating factor.

**Table 3. Derating Factors of Failure Mechanisms**

Failure Mechanisms	Activation Energy, eV	Derating Factor
Oxide defect	0.3	0.1546
Silicon defect	0.3	0.1546
Ionic contamination	1.0	0.001984
Electromigration	0.4	0.08307
Charge injection	1.3	0.0003067
Metal corrosion	0.7	0.01315
Gold-aluminum interface	0.8	0.006886

The acceleration of failure mechanisms in a high-humidity and high-temperature environment must be expressed as a function not only of temperature but also of humidity.

According to the reliability test statistics, the acceleration factor in such an environment can best be approximated with Peck's model as follows.

$$A = \exp[(E_a/k) \times (1/T_1 - 1/T_2)] \times (H_2/H_1)^{4.5}$$

where  $E_a$  = Activation energy  
 $k$  = Boltzmann's constant  
 $T$  = Junction temperature  
 $H$  = Relative humidity

For example, the acceleration factor for high-humidity and high-temperature or pressure cooker tests ranges from 100 to 1000 times that of the normal operating environment.

### Failure Rate Calculation

As an example, suppose that product samples are submitted to a 1000-hour life test at 125°C junction temperature and two failures are encountered: one oxide and one metalization defect. The sample size is 885 units.

Thus, the oxide failure rate is 0.11 percent per 1000 hours and the metalization failure rate is 0.11 percent per 1000 hours. Therefore, the total failure rate at 125°C sums to 0.22 percent per 1000 hours at 1K hours.

### Failure Rate Prediction

To derate these failure rates to a normal operating environment, use the derating factors listed in table 3.

Oxide failures =  $0.11 \times 0.1546 = 0.01701\%$  per 1K hrs  
 Metal failures =  $0.11 \times 0.01315 = 0.00145\%$   
 per 1K hrs  
 Total failures =  $0.01846\%$  per 1K hrs

Note that the example above is a snapshot of the high-temperature life test performed on a particular lot. It is not accumulated data that can be used to represent overall reliability. This conservative illustration, however, shows that the failure rate in a normal operating environment is approximately one-twelfth the failure rate in a higher-temperature environment.

The failure rate prediction takes different activation energies into account whenever the causes of failures are known through performing failure analysis. In some cases, however, an activation energy is assumed in order to accomplish a quick first-order approximation. To yield a conservative estimate of failure rates, NEC assumes an average activation energy of 0.7 eV whenever the exact failure mechanism is not known.

### Reliability Test Results

Before introducing new technologies or products, NEC's internal reliability goals must be attained. Several categories of testing are used in the internal qualification program to assure that product reliability meets NEC's reliability goals. Once the product is qualified, its reliability level is regularly monitored in a monthly reliability test.

### NEC's Goals on Failure Rates

NEC's approach to achieving high reliability is to build quality into the product, as opposed to merely screening out defective units. The use of distributed control methods embedded in the production line, in conjunction with conventional screening methods, results in the highest reliability at the lowest cost.

NEC's maximum failure rate goals for infant mortality and long-term device operation are listed in table 4.

**Table 4. Infant Mortality and Long-Term Failure Rates**

Type	Failure Rate Percent/1000 Hours
Infant mortality	0.10 max
Long-term	
1.2M device hours average	0.02 max
3.0M device hours average	0.01 max

### Infant Mortality Failure Rate

The infant mortality goal for each product group is set at 0.10 percent maximum. When a failure rate exceeds this level, there is prompt remedial action.

### Long-Term Failure Rate

The long-term failure rate goal is based on the following conditions:

- A minimum of 1.2 million device hours at 125°C is accumulated to resolve 0.02 percent per 1000 hours at 55°C with a 60-percent confidence level.
- A minimum of 3 million device hours at 125°C is accumulated to resolve 0.01 percent per 1000 hours at 55°C with a 60-percent confidence level.

**Infant Mortality Failure Screening**

It is logical to assume the integrated circuit that fails at one temperature would also fail at another temperature, except it would fail sooner at a higher temperature. As can be expected, the failure rate is a function of activation energy. Establishing infant mortality screening, therefore, requires knowledge of the likely failure mechanisms and their associated activation energy.

The most likely mechanisms associated with infant mortality failures are generally manufacturing defects and process anomalies. These generally consist of contamination, cracked chips, wire bond shorts, or bad wire bonds. Since these describe a number of possible mechanisms, any one of which might predominate at a given time, the activation energy for infant mortality might be expected to vary considerably.

The effectiveness of a screening condition, preferably at some stress level in order to shorten the time, varies greatly with the failure mechanism being screened for. Another factor is the economics of the screening process introduced into the production line. Optimal conditions and duration of a screening process will be a compromise of these two factors.

For example, failures due to ionic contamination have an activation energy of approximately 1.0 eV. Therefore, a 15-hour stress at 125°C junction temperature would be the equivalent of approximately 90 days of operation at a junction temperature of 55°C. On the other hand, failures due to oxide defects have an activation energy of approximately 0.3 eV, and a 15-hour stress at 125°C junction temperature would be the equivalent of approximately one week's operation at 55°C junction temperature. As indicated by this, the condition and duration of infant mortality screening would be a strong function of the allowable component failures, hence the system failure, in the field.

Empirical data, gathered over more than a year at NEC, indicates that early failure does occur after less than 4 hours of stress at 125°C ambient temperature. This fact is supported by the life test of the same lot, where the failure rate shows random distribution, as opposed to a decreasing failure rate that then runs into the random failure region.

NEC has adopted the initial infant mortality burn-in at 125°C as a standard production screening procedure. As a result, the field reliability of NEC devices is an order of magnitude higher than the goals set for NEC's integrated circuit products.

**Life Tests**

The most significant difference between NEC's products and those of other integrated circuit manufacturers is that NEC's have been prescreened for their infant mortality defects. The products delivered to customers are operating at the beginning of the random failure region of the life curve. The life test data also reflects this fact, as will be shown.

The failure mechanism distribution from field failures, as previously shown in figure 2, also contains a very low percentage due to infant mortality. The majority of failures are long-term life failures, and these can be eliminated by stringent process control. Usually, these failure mechanisms have low activation energy associated with them.

Another significant improvement devised by NEC is plastic encapsulation and passivation. As a result, NEC products show excellent reliability in both high-humidity and high-temperature environments. Following is life test data accumulated over more than a year for large-scale integrated circuits.

**High-Temperature Operating Life Test**

This test is used to accelerate failure mechanisms by operating the devices at an elevated temperature. For large-scale integrated circuits, the failure rate is 0.242 percent per 1000 hours at 125°C. This is equivalent to 0.0071 percent per 1000 hours in an operating environment of 55°C (table 5).

**Table 5. High-Temperature Operating Life Test**

Number of Samples	Number of Failures at				
	48 hrs	96 hrs	168 hrs	500 hrs	1K hrs
3317	0	0	1	4	3
Total number of failures at 1K hrs	= 8				
Failure rate at 1K hrs at 125°C	= 0.242% per 1K hrs				
Projected failure rate at 1K hrs at 55°C	= 0.007% per 1K hrs				

**High-Temperature and High-Humidity Life Test**

This test is used to accelerate failure mechanisms by operating the devices at high temperature and high humidity. Leakage-related failures and device parameter drift are accelerated by this test. For these large-scale integrated circuits, the failure rate is 0.091 percent per 1000 hours. This is equivalent to 0.0027 percent per 1000 hours in an operating environment of 55°C. The test conditions are T<sub>A</sub> = 85°C and relative humidity (RH) = 80% (table 6).

**Table 6. High-Temperature and High-Humidity Life Test**

Number of Samples	Number of Failures at				
	48 hrs	96 hrs	168 hrs	500 hrs	1K hrs
2190	0	0	0	0	2
Total number of failures at 1K hrs = 2					
Failure rate at 1K hrs at 85°C/80% RH = 0.091% per 1K hrs					
Projected failure rate at 1K hrs at 55°C/60% RH = 0.003% per 1K hrs					

### High-Temperature Storage Life Test

This test is effective in accelerating the failure mechanisms related to mechanical reliability problems and process instability. For these LSI devices, the failure rate is 0.207 percent per 1000 hours at 125°C. This is equivalent to 0.0061 percent per 1000 hours in an operating environment of 55°C (table 7).

**Table 7. High-Temperature Storage Life Test**

Number of Samples	Number of Failures at				
	48 hrs	96 hrs	168 hrs	500 hrs	1K hrs
2410	0	0	0	1	4
Total number of failures at 1K hrs = 5					
Failure rate at 1K hrs at 125°C = 0.207% per 1K hrs					
Projected failure rate at 1K hrs at 55°C = 0.006% per 1K hrs					

### Pressure Cooker Test

This test is effective in accelerating failure mechanisms related to metalization corrosion due to moisture. The failure rate is 0.52 percent per 1000 hours at  $T_A = 125^\circ\text{C}$  and 2.3 atm at 100 percent humidity. This is equivalent to 0.0013 percent per 1000 hours at 55°C and an environment of 60 percent humidity (table 8).

**Table 8. Pressure Cooker Test**

Number of Samples	Number of Failures at				
	48 hrs	96 hrs	168 hrs	500 hrs	1K hrs
1718	0	4	5	No test performed	
Total number of failures at 168 hrs = 9					
Failure rate at 125°C = 0.54% per 1K hrs					
Projected failure rate at 55°C = 0.001% per 1K hrs					

### Life Test Data Summary

Table 9 summarizes the life test results and projected failure rates in the normal operating environment. The failure rate shows random distribution as opposed to a decreasing failure rate. This is a result of infant mortality screening.

**Table 9. Life Test Data**

Test Time	Number of Samples	Number of Failures at				Total Number of Failures
		96 hrs	168 hrs	500 hrs	1K hrs	
High-temperature life test	3317	0	1	4	3	8
High-humidity life test	2190	0	0	0	2	2
High-temperature storage life test	2410	0	0	1	4	5
Pressure cooker test	1718	4	5	No test performed		9
Total	9635	4	6	5	9	24

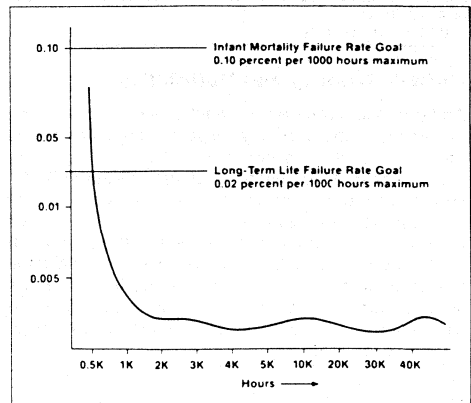
The projected failure rate in the normal operating environment is calculated assuming that the average activation energy is 0.7 eV.

Figure 3 shows the life distribution of NEC integrated circuits as a form of the bathtub curve.

This life test data shows improvements of approximately an order of magnitude better than NEC's goal. The hours of operation are equivalent to the normal operating environment. Wear-out failures, which had been the main target for reliability improvement, have also been significantly reduced. This result comes mainly from process improvements and stringent manufacturing process control.

NEC's main goal has been to improve reliability with respect to infant mortality and long-term life failures. This can be achieved by introducing an effective screening method for infant mortality and building quality into the product.

**Figure 3. Plot of Life Test Results**



## QUALITY AND RELIABILITY

### Thermal Stress Tests

Temperature cycling and thermal shock test the thermal compatibility of material and metal used to make integrated circuits. Table 10 lists the reliability test results of thermal stress tests.

**Table 10. Thermal Stress Tests**

Test Item	Number of Samples	Number of Failures
Soldering heat test $T_A = 260^\circ\text{C}$ for 10 seconds	1891	0
Temperature cycle $T_A = -65$ to $+150^\circ\text{C}$ , 10 cycles	1891	0
Thermal shock test $T_A = 0$ to $+100^\circ\text{C}$ , 15 cycles	1891	0

### Mechanical Stress Tests

In addition to the device life test, NEC performs mechanical stress tests to detect reliability problems related to the package, material, and device susceptibility to an extreme environment. Table 11 lists mechanical stress test results.

**Table 11. Mechanical Stress Tests**

Test Item	Number of Samples	Number of Failures
Mechanical shock test at 15 kg, 3 axis	315	0
Vibration test at 100 Hz to 2 kHz, 20 g	315	0
Constant acceleration at 20 kg, 3 axis	315	0
Lead fatigue test at 240 grams	538	0
Solderability test at $230^\circ\text{C}$ for 5 seconds	638	0

### Built-In Quality and Reliability

As large-scale integration reaches even higher levels of density, simple quality inspections cannot assure adequate levels of product quality and reliability. In order to ensure the reliability of state-of-the-art VLSI, NEC has adopted another approach. Highest reliability and superior quality of a device can only be achieved by building these characteristics into the product at each process step. NEC, therefore, has introduced the notion of total quality control (TQC) into its entire semiconductor production line. Quality control is distributed into each process step and then summed to form a consolidated system.

### Approaches to Total Quality Control

First, the quality control function is embedded into each process. This method enables early detection of possible causes of failure and immediate feedback.

Second, the reliability and quality assurance policy is an integral part of the entire organization. This enables a companywide quality control activity. At NEC, everyone in the company is involved with the concept and methodology of total quality control.

Third, there is an ongoing research and development effort to set even higher standards of device quality and reliability.

Fourth, extensive failure analysis is performed periodically and corrective actions are taken as preventive measures. Process control is based on statistical data gathered from this analysis.

The goal is to maintain the superior product quality and reliability that has become synonymous with the NEC name. The new standard is continuously upgraded and the iterative process continues.

### Implementation of Distributed Quality Control

Building quality into a product requires early detection of possible causes of failure at each process step. Then, immediate feedback to remove the causes is a must. A fixed station quality inspection is often lacking in immediate feedback. It is, therefore, necessary to distribute quality control functions to each process step, including the conceptual stage. NEC has implemented a distributed quality control function at each step of the process. Following is a breakdown of the significant steps:

- Product development phase
- Wafer processing
- Chip mounting and packaging
- Electrical testing and thermal aging
- Incoming material inspection

**Product Development Phase.** The product development phase includes conception of a product, review of the device proposal, organization and physical element design, engineering evaluation, and finally, transfer of the product to manufacturing. Quality and reliability are considered at every step. More significantly, at the design review stage and prior to product transfer, the quality and reliability requirements have to be examined and determined to be satisfactory. This often adds 2 to 3 months to the product development cycle. Building in high reliability, however, cannot be sacrificed.

**Wafer Processing Stage Inspection.** The in-process quality inspections that occur at the wafer fabrication stage are listed in table 12.

**Table 12. Wafer Processing Inspection**

Process	Inspection Item
Wafer	Resistivity, dimension, and appearance. (lot sampling inspection)
Mask	
Photolithography	Alignment and etching (100% inspection)
Cleaning	
Diffusion and oxidation	Oxide thickness, sheet resistivity (lot sampling inspection)
Metalization and passivation	Thickness, $V_{th}$ , C-V characteristics (lot sampling)
Wafer sort and scribe	Dc parameters (100% inspection)
Die sort	100% visual inspection

**Chip Mounting and Packaging.** The in-process quality inspections done at the chip mounting and packaging stage are listed in table 13.

**Table 13. Chip Mounting and Packaging Inspection**

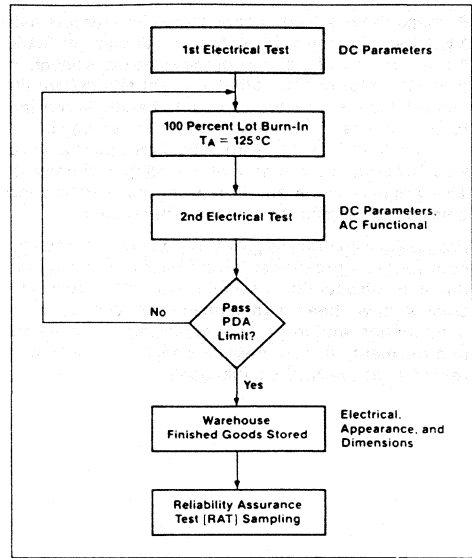
Process	Inspection Item
Die	Incoming material inspection
Die attach	Appearance (lot sampling inspection)
Wire bonding	Bond strength, appearance (lot sampling)
Packaging	100% appearance inspection
Fine leak*	Lot sampling
Gross leak*	100% inspection

\*For ceramic package devices only.

**Electrical Testing and Screening.** Electrical testing and infant mortality screening are performed at this stage. A flowchart of the process is depicted in figure 4.

At the first electrical test, dc parameters are tested according to the electrical specifications on 100% of each lot. This is a prescreening prior to the infant mortality test. At the second electrical test, ac functional tests as well as dc parameter tests are performed on 100% of the subjected lot. If the percentage of defective units exceeds the limit, the lot is subjected to an additional burn-in. During this time, the defective units are undergoing a failure analysis, the results of which are then fed back into the process for corrective action.

**Figure 4. Electrical Testing and Screening**



**Incoming Material Inspection.** Prior to warehouse storage, lots are subjected to an incoming inspection according to the following sampling plan.

- Electrical test: Dc parameters LTPD 3%
- Functional test LTPD 3%
- Appearance LTPD 3%

### Reliability Assurance Test

Samples are continually taken from the warehouse and subjected to monthly reliability tests as discussed previously. They are taken from similar process groups so that it can be assumed that any device is representative of the reliability of the group.

**In-Process Screening**

Perhaps the most significant preventive measure that NEC has implemented is the introduction of 100% burn-in as an integral part of the standard production process. Most of the potential infant failures are effectively screened from every lot, thereby improving reliability. Assuming average activation energy of 0.7 eV, burn-in at  $T_A = 125^\circ\text{C}$  for 4 hours is equivalent to a week's operation in a normal operating environment. This appears to be ample time for accelerating the time-to-failure mechanisms for early failures.

Process automation, as previously mentioned, has also contributed a great deal toward improving reliability. Since its introduction, assembly related failure mechanisms have been substantially reduced. And, in combination with in-process screening and materials improvement, it has helped establish quality and reliability above NEC's initial goals.

**Summary and Conclusion**

As has been discussed, building quality and reliability into products is the most efficient way to ensure product reliability. NEC's approach of distributing quality control functions to process steps, then forming a consolidated quality control system, has produced superior quality and excellent reliability.

Prescreening, introduced as an integral part of large-scale integrated circuit protection, has been a major factor in improving reliability. The most recent year's production clearly demonstrates continuation of NEC's high reliability and the effectiveness of this method.

Reliability assurance tests (RATs), performed monthly, have ensured high outgoing quality levels. The combination of building quality into products, effective prescreening of potential failures, and the reliability assurance test has established a singularly high standard of quality and reliability for NEC's large-scale integrated circuits.

With a companywide quality control program, NEC is committed to building superior quality and highest reliability into all its products. Through continuous research and development activities, extensive failure analysis, and process improvements, a higher standard of quality and reliability will continuously be set and maintained.

# CMOS MICROPROCESSORS/ MICROCOMPUTER

3



### Section 3 – CMOS Microprocessors/Microcomputer

μPD70008/A	8-Bit Microprocessors .....	3.3
μPD70108	8/16-Bit High-Performance Microprocessor (V20™) .....	3.31
μPD70116	16-Bit High-Performance Microprocessor (V30™) .....	3.63
μPD70208	8/16-Bit High-Integration Microprocessor (V40™) .....	3.95
μPD70216	16-Bit High-Integration Microprocessor (V50™) .....	3.167
μPD70320/322	16-Bit Single-Chip Microcomputer (V25™) .....	3.239
μPD70616	32-Bit Virtual Memory Microprocessor (V60™) .....	3.313

## Description

The  $\mu$ PD70008 and  $\mu$ PD70008A are power saving, high performance, general purpose 8-bit microprocessor. It is a CMOS-process part with a standby mode that greatly reduces power consumption.

## Features

- High performance  $\mu$ PD780 instruction set
- Instruction cycle:  
1  $\mu$ s at 4 MHz ( $\mu$ PD70008,  $\mu$ PD70008A-4)  
0.66  $\mu$ s at 6 MHz ( $\mu$ PD70008A-6)
- Direct addressing of up to 64 K bytes of memory
- Memory refresh function
- Interrupt functions:
  - Maskable external interrupt ( $\overline{\text{INT}}$ )
  - Nonmaskable external interrupt (NMI)
- Low-power standby mode (HALT)
- CMOS standby mode (HALT)
- Single power supply

## Ordering Information

Part Number	Package Type	Max Frequency of Operation
$\mu$ PD70008C	40-pin plastic DIP	4 MHz
$\mu$ PD70008AC-4	40-pin plastic DIP	4 MHz
$\mu$ PD70008AC-6	40-pin plastic DIP	6 MHz
$\mu$ PD70008AG-4	44-pin plastic flat pack	4 MHz
$\mu$ PD70008AG-6	44-pin plastic flat pack	6 MHz
$\mu$ PD70008AL-6	44-pin PLCC	6 MHz

## Absolute Maximum Ratings

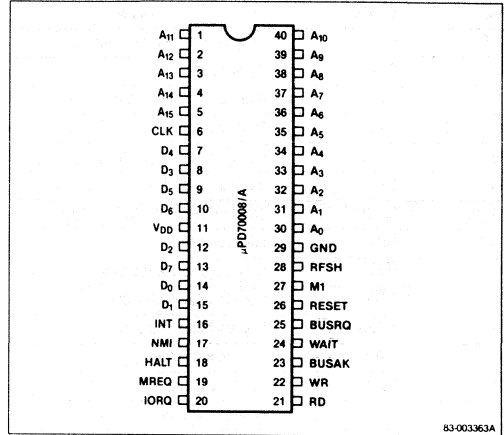
$T_A = 25^\circ\text{C}$

Power supply voltage, $V_{DD}$	-0.3 V to +7 V
Input voltage, $V_{IN}$	-0.3 V to $V_{DD} + 0.3$ V
Output voltage, $V_O$	-0.3 V to $V_{DD} + 0.3$ V
Operating temperature, $T_{OPT}$	
$\mu$ PD70008	-10°C to +70°C
$\mu$ PD70008A	-45°C to +85°C
Storage temperature, $T_{STG}$	-65°C to +150°C

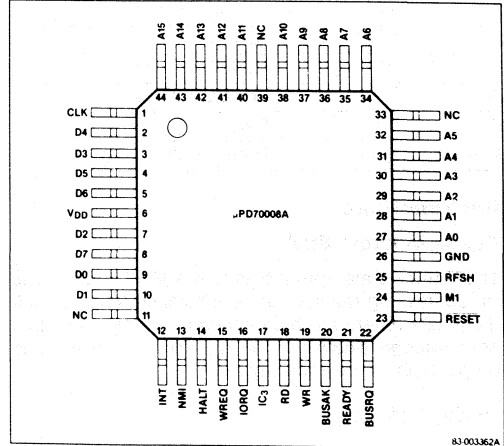
**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## Pin Configurations

### 40-Pin Plastic DIP



### 44-Pin Plastic Flat Pack



## Pin Identification

### 40-Pin Plastic DIP

No.	Symbol	Function
1-5	A <sub>11</sub> -A <sub>15</sub>	Address bus, high bits, outputs
6	CLK	Clock input
7-10	D <sub>3</sub> -D <sub>6</sub>	Data bus, bits 3-6, inputs / outputs
11	V <sub>DD</sub>	Power supply
12	D <sub>2</sub>	Data bus, bit 2, input / output
13	D <sub>7</sub>	Data bus, bit 7, input / output
14, 15	D <sub>0</sub> , D <sub>1</sub>	Data bus, bits 0, 1, inputs / outputs
16	$\overline{\text{INT}}$	Interrupt input
17	$\overline{\text{NMI}}$	Nonmaskable interrupt input
18	$\overline{\text{HALT}}$	Halt / standby mode output
19	$\overline{\text{MREQ}}$	Memory request output
20	$\overline{\text{IORQ}}$	I / O request output
21	$\overline{\text{RD}}$	Read strobe output
22	$\overline{\text{WR}}$	Write strobe output
23	$\overline{\text{BUSAK}}$	Bus acknowledge output
24	$\overline{\text{WAIT}}$	Wait input
25	$\overline{\text{BUSRQ}}$	Bus request input
26	$\overline{\text{RESET}}$	Reset input
27	$\overline{\text{M1}}$	Machine cycle 1 output
28	$\overline{\text{RFSH}}$	Refresh request output
29	GND	Ground
30-40	A <sub>0</sub> -A <sub>10</sub>	Address bus, low bits, outputs

## Pin Functions

### A<sub>15</sub>-A<sub>0</sub> (Address Bus)

These three-state output pins form a 16-bit address bus for addressing memory or peripheral devices. The address bus enters the high impedance state when bus acknowledge is active. In the standby mode, these pins output high- or low-level signals.

### D<sub>7</sub>-D<sub>0</sub> (Data Bus)

These three-state pins form an 8-bit bidirectional data bus. On this bus data is transferred between the  $\mu$ PD70008/A and memory or peripheral devices. This bus enters the high impedance state when bus acknowledge is active. In the standby mode, these pins are high-level.

### $\overline{\text{INT}}$ (Interrupt)

This pin is an active-low interrupt input which can be masked by software. NMI has a lower priority than NMI and  $\overline{\text{BUSRQ}}$ .  $\overline{\text{INT}}$  releases the standby mode.

### 44-Pin Plastic Miniflat

No.	Symbol	Function
40-44	A <sub>11</sub> -A <sub>15</sub>	Address bus, high bits, outputs
1	CLK	Clock input
2-5	D <sub>3</sub> -D <sub>6</sub>	Data bus, bits 3-6, inputs / outputs
6	V <sub>DD</sub>	Power supply
7	D <sub>2</sub>	Data bus, bit 2, input / output
8	D <sub>7</sub>	Data bus, bit 7, input / output
9-10	D <sub>0</sub> , D <sub>1</sub>	Data bus, bits 0, 1, inputs / outputs
12	$\overline{\text{INT}}$	Interrupt input
13	$\overline{\text{NMI}}$	Nonmaskable interrupt input
14	$\overline{\text{HALT}}$	Halt / standby mode output
15	$\overline{\text{MREQ}}$	Memory request output
16	$\overline{\text{IORQ}}$	I / O request output
18	$\overline{\text{RD}}$	Read strobe output
19	$\overline{\text{WR}}$	Write strobe output
20	$\overline{\text{BUSAK}}$	Bus acknowledge output
21	$\overline{\text{WAIT}}$	Wait input
22	$\overline{\text{BUSRQ}}$	Bus request input
23	$\overline{\text{RESET}}$	Reset input
24	$\overline{\text{M1}}$	Machine cycle 1 output
25	$\overline{\text{RFSH}}$	Refresh request output
26	GND	Ground
28-32, 34-38	A <sub>0</sub> -A <sub>10</sub>	Address bus, low bits, outputs
17	IC	Internally connected
11, 33, 39	NC	Not connected

### $\overline{\text{NMI}}$ (Nonmaskable Interrupt)

This pin inputs an interrupt which is not maskable by software. NMI is active-low in the  $\mu$ PD70008, and is falling edge triggered in the  $\mu$ PD70008A.  $\overline{\text{NMI}}$  has a higher priority than  $\overline{\text{INT}}$ , but a lower priority than  $\overline{\text{BUSRQ}}$  and  $\overline{\text{RESET}}$ . NMI releases the standby mode.

### $\overline{\text{MREQ}}$ (Memory Request)

This three-state pin is an active-low output. The  $\mu$ PD70008/A asserts MREQ to indicate that the information on the address bus is a memory address. This pin enters the high impedance state when bus acknowledge is active. MREQ is inactive (high) in the standby mode.

## Pin Functions (cont)

### $\overline{\text{IORQ}}$ (I/O Request)

This three-state pin is an active-low output. The μPD70008/A asserts  $\overline{\text{IORQ}}$  to indicate that the information on the address bus is a peripheral device address.  $\overline{\text{IORQ}}$  is also asserted during a maskable interrupt service to request the interrupting device to output its interrupt vector to the data bus. This pin enters the high impedance state when bus acknowledge is active.  $\overline{\text{IORQ}}$  is inactive (high) in the standby mode.

### $\overline{\text{RD}}$ (Read Strobe)

This three-state active-low output provides a read strobe for the memory and peripheral devices. The pin enters the high impedance state when the bus acknowledge is active.  $\overline{\text{RD}}$  is inactive (high) in the standby mode.

### $\overline{\text{WR}}$ (Write Strobe)

This three-state active-low output provides a write strobe for the memory and peripheral devices. This pin enters the high impedance state when bus acknowledge is active.  $\overline{\text{WR}}$  is inactive (high) in the standby mode.

### $\overline{\text{BUSRQ}}$ (Bus Request)

This is an active-low input. Peripheral devices assert  $\overline{\text{BUSRQ}}$  to request the μPD70008/A to release control of the address bus (A<sub>15</sub>-A<sub>0</sub>), data bus (D<sub>7</sub>-D<sub>0</sub>) and control bus (MREQ,  $\overline{\text{IORQ}}$ ,  $\overline{\text{RD}}$  and  $\overline{\text{WR}}$ ) and assert bus acknowledge.  $\overline{\text{BUSRQ}}$  has a higher priority than either  $\overline{\text{INT}}$  or  $\overline{\text{NMI}}$ , but is lower in priority than  $\overline{\text{RESET}}$ .  $\overline{\text{BUSRQ}}$  will temporarily suspend the standby mode. The μPD70008/A leaves standby mode when  $\overline{\text{BUSRQ}}$  is asserted, but returns to the standby mode when  $\overline{\text{BUSRQ}}$  is released.

### $\overline{\text{BUSAK}}$ (Bus Acknowledge)

This active-low output indicates that the data bus (D<sub>7</sub>-D<sub>0</sub>), address bus (A<sub>15</sub>-A<sub>0</sub>), and control bus (MREQ,  $\overline{\text{IORQ}}$ ,  $\overline{\text{RD}}$  and  $\overline{\text{WR}}$ ) have entered the high impedance state. This releases the buses from CPU control and makes them available to the peripheral devices for data exchange. This state cannot be released by  $\overline{\text{NMI}}$  or  $\overline{\text{INT}}$ , but responds only to  $\overline{\text{RESET}}$  or the release of  $\overline{\text{BUSRQ}}$ .

### $\overline{\text{HALT}}$ (Halt/Standby Mode)

This active-low output is asserted after the halt command has been executed and indicates that the μPD70008/A has entered the standby mode.

### $\overline{\text{WAIT}}$ (Wait)

This pin is an active-low input. Memory and peripheral devices assert this signal to increase read or write access time. When  $\overline{\text{WAIT}}$  is asserted, the μPD70008/A inserts wait states (TW) into the machine cycle until  $\overline{\text{WAIT}}$  is released.

### $\overline{\text{RESET}}$ (Reset)

This active-low input is used to reset the μPD70008/A. The standby mode is released on Reset. Reset has the highest priority.

$$\overline{\text{INTI}} < \overline{\text{NMI}} < \overline{\text{BUSRQ}} < \overline{\text{RESET}}$$

### $\overline{\text{RFSH}}$ (Refresh Request)

This pin is an active-low output. The μPD70008/A asserts  $\overline{\text{RFSH}}$  to trigger the external memory refresh operation. When  $\overline{\text{RFSH}}$  is low, the lower seven bits of the address bus (A<sub>6</sub>-A<sub>0</sub>) are a refresh address. This pin is inactive (high) in the standby mode.

### $\overline{\text{M1}}$ (Machine Cycle 1)

This pin is an active-low output. When  $\overline{\text{M1}}$  is asserted, it indicates that the μPD70008/A is in the opcode fetch cycle, M1.

### CLK (Clock)

This pin is the system clock input.

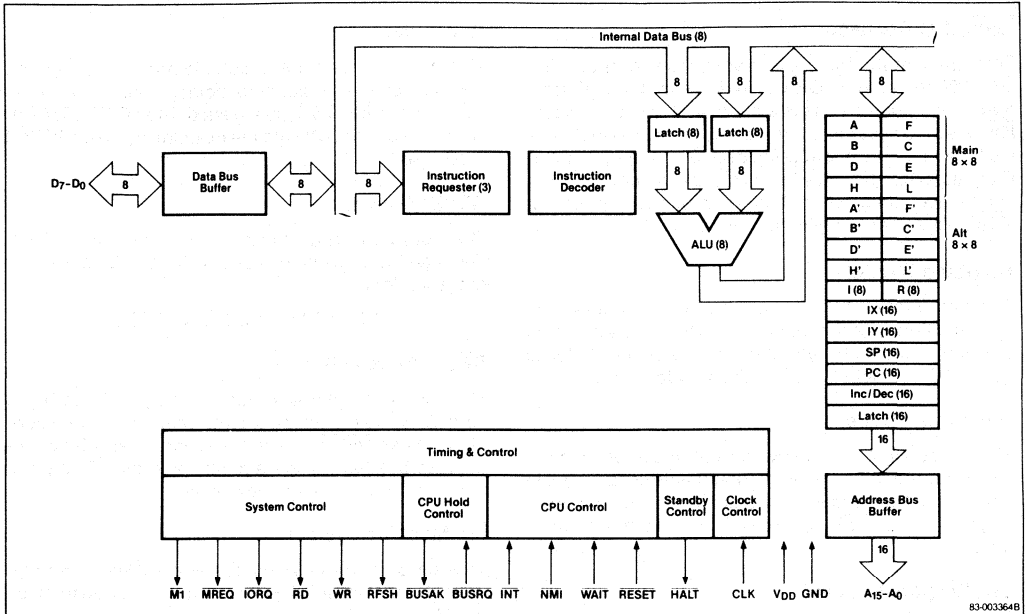
### VDD (Power Supply)

This pin is the +5 V power supply input.

### GND (Ground)

This pin is the ground pin.

**Block Diagram**



**DC Characteristics**

μPD70008:  $T_A = -10^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ , μPD70008A:  $T_A = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{DD} = +5\text{V} \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage high	$V_{IH1}$	2.2		$V_{DD}$	V	Except CLK, RESET
	$V_{IH2}$	$V_{DD} - 0.6$		$V_{DD} + 0.3$	V	CLK, RESET
Input voltage low	$V_{IL1}$	-0.3		0.8	V	Except CLK, RESET
	$V_{IL2}$	-0.3	0.45		V	CLK, RESET
Output voltage high	$V_{OH}$	2.4			V	$I_{OH} = -400\ \mu\text{A}$
Output voltage low	$V_{OL}$		0.4		V	$I_{OL} = 2.5\ \text{mA}$
Input leakage current high	$I_{LIH}$			10	$\mu\text{A}$	$V_I = V_{DD}$
Input leakage current low	$I_{LIL}$			-10	$\mu\text{A}$	$V_I = 0\ \text{V}$
Output leakage current high	$I_{LOH}$			10	$\mu\text{A}$	$V_O = V_{DD}$
Output leakage current low	$I_{LOL}$			-10	$\mu\text{A}$	$V_O = 0\ \text{V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Supply current (Note 1)						
μPD70008	$I_{DD1}$		10	30	mA	$t_{CYK} = 0.25\ \mu\text{s}$
	$I_{DD2}$		500		$\mu\text{A}$	$t_{CYK} = 0.25\ \mu\text{s}$
μPD70008A-4	$I_{DD1}$		9	20	mA	$t_{CYK} = 0.25\ \mu\text{s}$
	$I_{DD2}$		80	240	$\mu\text{A}$	$t_{CYK} = 0.25\ \mu\text{s}$
μPD70008A-6	$I_{DD1}$		14	30	mA	$t_{CYK} = 0.165\ \mu\text{s}$
	$I_{DD2}$		120	360	$\mu\text{A}$	$t_{CYK} = 0.165\ \mu\text{s}$

**Note:**

- (1)  $I_{DD1}$  is normal operating current.
- $I_{DD2}$  is standby mode current.

## Capacitance

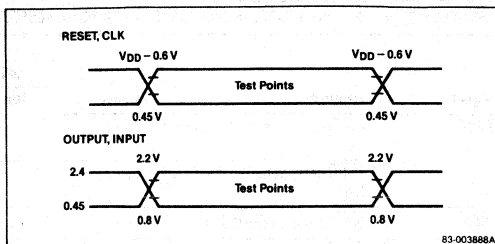
$T_A = 25^\circ\text{C}$ ,  $f_C = 1\text{MHz}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
CLK input capacitance	$C_K$		35		pF	(Note 1)
Input capacitance	$C_I$		5		pF	(Note 1)
Output capacitance	$C_O$		10		pF	(Note 1)
I/O capacitance	$C_{I/O}$		10		pF	(Note 1)

### Note:

(1) All unmeasured pins returned to 0V.

## AC Test Points



## AC Characteristics

μPD70008:  $T_A = -10^\circ\text{C}$  to  $+70^\circ\text{C}$ , μPD70008A:  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$

Signal	Parameter	Symbol	Limits				Unit	Test Conditions
			μPD70008IA-4		μPD70008A-6			
			Min	Max	Min	Max		
CLK	Clock period	$t_{CYK}$	0.25	(Note 1)	0.165	(Note 8)	$\mu\text{s}$	
	Clock pulse width high	$t_{KHH}$	0.11	200	0.065	200	$\mu\text{s}$	
	Clock pulse width low	$t_{KLL}$	110	2000	65	2000	ns	
	Clock pulse rise and fall time	$t_{KR}$ , $t_{KF}$		30		20	ns	
A <sub>15</sub> -A <sub>0</sub>	Address output delay	$t_{DKA}$		110		90	ns	$C_L = 100\text{pF}$
	Address delay to float	$t_{FKA}$		90		80	ns	$C_L = 100\text{pF}$
	Address stable prior to $\overline{\text{MREQ}}$ , memory cycle	$t_{SAM}$	(Note 2)		(Note 9)		ns	$C_L = 100\text{pF}$
	Address stable prior to $\overline{\text{IORQ}}$ in I/O cycle	$t_{SAI}$	$t_{CYK} - 70$		(Note 10)		ns	$C_L = 100\text{pF}$
	Address stable from $\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{IORQ}}$ , $\overline{\text{MREQ}}$	$t_{HRA}$	(Note 3)		(Note 11)		ns	$C_L = 100\text{pF}$
	Address stable from $\overline{\text{RD}}$ , $\overline{\text{WR}}$ during float	$t_{FCA}$	(Note 4)		(Note 12)		ns	$C_L = 100\text{pF}$
D <sub>7</sub> -D <sub>0</sub>	Data output delay	$t_{DKD}$		180		130	ns	$C_L = 100\text{pF}$
	Delay to float during write cycle	$t_{FKD}$		90		80	ns	$C_L = 100\text{pF}$
	Data setup time to CLK during M1 cycle	$t_{SDKR}$	35		30		ns	$C_L = 100\text{pF}$
	Data setup time to CLK during M2 to M5 cycles	$t_{SDKF}$	50		40		ns	$C_L = 100\text{pF}$
	Data stable prior to $\overline{\text{WR}}$ (memory cycle)	$t_{SMDW}$	$t_{CYK} - 170$		(Note 13)		ns	$C_L = 100\text{pF}$
	Data stable prior to $\overline{\text{WR}}$ (I/O cycle)	$t_{SIDW}$	$t_{KLL} + t_{KR} - 170$		(Note 14)		ns	$C_L = 100\text{pF}$
	Data stable from $\overline{\text{WR}}$	$t_{FCD}$	(Note 5)		(Note 15)		ns	$C_L = 100\text{pF}$
$\overline{\text{WR}}$	$\overline{\text{WR}}$ delay from CLK $\uparrow$ to $\overline{\text{WR}}$ low	$t_{DKRWL}$		65		60	ns	
	$\overline{\text{WR}}$ delay from CLK $\downarrow$ to $\overline{\text{WR}}$ low	$t_{DKFWL}$		80		70	ns	
	$\overline{\text{WR}}$ delay from CLK $\downarrow$ to $\overline{\text{WR}}$ high	$t_{DKFWH}$		80		70	ns	
	$\overline{\text{WR}}$ low pulse width	$t_{WWL}$	$t_{CYK} - 30$		(Note 18)		ns	
$\overline{\text{M1}}$	$\overline{\text{M1}}$ delay from CLK $\uparrow$ to $\overline{\text{M1}}$ low	$t_{DKM1L}$		100		80	ns	$C_L = 100\text{pF}$
	$\overline{\text{M1}}$ delay from CLK $\uparrow$ to $\overline{\text{M1}}$ high	$t_{DKM1H}$		100		80	ns	$C_L = 100\text{pF}$

**AC Characteristics (cont)**

μPD70008: T<sub>A</sub> = -10°C to +70°C, μPD70008A: T<sub>A</sub> = -40°C to +85°C, V<sub>CC</sub> = 5 V ± 10%

Signal	Parameter	Symbol	Limits				Unit	Test Conditions
			μPD70008/A-4		μPD70008A-6			
			Min	Max	Min	Max		
RFSH	RFSH delay from CLK ↑ to RFSH low	t <sub>DKRFL</sub>		130		110	ns	C <sub>L</sub> = 100 pF
	RFSH delay from CLK ↑ to RFSH high	t <sub>DKRFH</sub>		120		100	ns	C <sub>L</sub> = 100 pF
WAIT	WAIT setup time to CLK ↓	t <sub>SWTK</sub>	70		60		ns	
HALT	HALT delay from CLK ↓	t <sub>DKHT</sub>		300		260	ns	C <sub>L</sub> = 100 pF
INT	INT setup time to CLK ↑	t <sub>SITK</sub>	80		70		ns	
NMI	NMI low pulse width	t <sub>NNL</sub>	80		70		ns	
BUSRQ	BUSRQ setup time to CLK ↓	t <sub>SBQK</sub>	50		50		ns	
BUSAK	BUSAK delay from CLK ↑ to BUSAK low	t <sub>DKRBA</sub>		100		90	ns	C <sub>L</sub> = 100 pF
	BUSAK delay from CLK ↓ to BUSAK high	t <sub>DKFBA</sub>		100		90	ns	C <sub>L</sub> = 100 pF
RESET	RESET setup to CLK	t <sub>SRSK</sub>	60		60		ns	
Other	Delay to float (MREQ, IORQ, RD, WR)	t <sub>FKC</sub>		80		70	ns	
	M1 stable prior to IORQ (interrupt acknowledge)	t <sub>SM11</sub>	(Note 7)		(Note 19)		ns	
	Hold time for setup time	t <sub>H</sub>	0		0		ns	
MREQ	MREQ delay from CLK ↓ to MREQ low	t <sub>DKFML</sub>		85		70	ns	C <sub>L</sub> = 100 pF
	MREQ delay from CLK ↑ to MREQ high	t <sub>DKRMH</sub>		85		70	ns	C <sub>L</sub> = 100 pF
	MREQ delay from CLK ↓ MREQ high	t <sub>DKFMH</sub>		85		70	ns	C <sub>L</sub> = 100 pF
	Pulse width MREQ low	t <sub>MML</sub>	t <sub>CYK</sub> - 30		(Note 16)		ns	C <sub>L</sub> = 100 pF
	Pulse width MREQ high	t <sub>MMH</sub>	(Note 6)		(Note 17)		ns	C <sub>L</sub> = 100 pF
IORQ	IORQ delay from CLK ↑ to IORQ low	t <sub>DKRIL</sub>		75		65	ns	C <sub>L</sub> = 100 pF
	IORQ delay from CLK ↓ to IORQ low	t <sub>DKFIL</sub>		85		70	ns	C <sub>L</sub> = 100 pF
	IORQ delay from CLK ↑ to IORQ high	t <sub>DKRIH</sub>		85		70	ns	C <sub>L</sub> = 100 pF
	IORQ delay from CLK ↓ to IORQ high	t <sub>DKFIH</sub>		85		70	ns	C <sub>L</sub> = 100 pF
RD	RD delay from CLK ↑ to RD low	t <sub>DKRRL</sub>		85		70	ns	C <sub>L</sub> = 100 pF
	RD delay from CLK ↓ to RD low	t <sub>DKRFL</sub>		95		80	ns	C <sub>L</sub> = 100 pF
	RD delay from CLK ↑ to RD high	t <sub>DKRRH</sub>		85		70	ns	C <sub>L</sub> = 100 pF
	RD delay from CLK ↓ to RD high	t <sub>DKFRH</sub>		85		70	ns	C <sub>L</sub> = 100 pF

- Note:**
- (1) t<sub>CYK</sub> = t<sub>KKH</sub> + t<sub>KKL</sub> + t<sub>KR</sub> + t<sub>KF</sub>
  - (2) t<sub>SAM</sub> = t<sub>KKH</sub> + t<sub>KF</sub> - 65
  - (3) t<sub>HRA</sub> = t<sub>KKL</sub> + t<sub>HR</sub> - 50
  - (4) t<sub>FCA</sub> = t<sub>KKL</sub> + t<sub>KR</sub> - 45
  - (5) t<sub>FCD</sub> = t<sub>KKL</sub> + t<sub>KR</sub> - 70
  - (6) t<sub>MMH</sub> = t<sub>KKH</sub> + t<sub>KF</sub> - 20
  - (7) t<sub>SM11</sub> = 2t<sub>CYK</sub> + t<sub>KKH</sub> + t<sub>KF</sub> - 65
  - (8) t<sub>CYK</sub> = t<sub>KKH</sub> + t<sub>KKL</sub> + t<sub>KR</sub> + t<sub>KF</sub>
  - (9) t<sub>SAM</sub> = t<sub>KKH</sub> + t<sub>KF</sub> - 50
  - (10) t<sub>SAI</sub> = t<sub>CYK</sub> - 55
  - (11) t<sub>HRA</sub> = t<sub>KKL</sub> + t<sub>KR</sub> - 50
  - (12) t<sub>FCA</sub> = t<sub>KKL</sub> + t<sub>KR</sub> - 40
  - (13) t<sub>SMDW</sub> = t<sub>CYK</sub> - 140
  - (14) t<sub>SIDW</sub> = t<sub>KKL</sub> + t<sub>KR</sub> - 140
  - (15) t<sub>FCD</sub> = t<sub>KKL</sub> + t<sub>KR</sub> - 55
  - (16) t<sub>MML</sub> = t<sub>CYK</sub> - 30
  - (17) t<sub>MMH</sub> = t<sub>KKH</sub> + t<sub>KF</sub> - 20
  - (18) t<sub>WWL</sub> = t<sub>CYK</sub> - 30
  - (19) t<sub>SM11</sub> = 2t<sub>CYK</sub> + t<sub>KKH</sub> + t<sub>KF</sub> - 50





### Register Configuration

#### Program Counter (PC)

The 16-bit program counter contains the address of the next instruction to be fetched and executed. It is set to 0000H at reset.

#### Stack Pointer (SP)

The 16-bit stack pointer stores the first address of the portion of main memory used as a LIFO stack. SP is decremented when a CALL or PUSH is executed, or when an interrupt occurs. It is incremented when a RET, POP, or interrupt return is executed.

#### Index Registers (IX, IY)

These two 16-bit registers are used to perform indexed addressing.

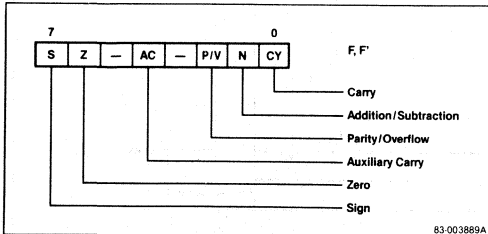
#### Accumulators (A, A')

The μPD70008/A has two 8-bit accumulators: the main accumulator (A) which is used to perform arithmetic and logic operations, and an alternate accumulator (A'). The contents of the main and alternate accumulators can be exchanged using the (EX) instruction. The alternate accumulator can be used for background operation, or to save the data in the main accumulator when an interrupt is processed.

#### Flag Registers (F, F')

The μPD70008/A has two 8-bit flag registers: main (F) and alternate (F') of the format shown in figure 1. The main flag register (F) has the status flags resulting from normal operation. The contents of the main and alternate registers can be exchanged using the exchange (EX) instruction. The alternate (F') register can be used for background operation, or to save the state of the main flag register when an interrupt is processed.

Figure 1. Flag Register Format



### General Purpose Registers

The μPD70008/A has twelve 8-bit general purpose registers: six main registers (B, C, D, E, H, and L) and six alternate registers (B', C', D', E', H', and L'). Each register can be used individually as an 8-bit register, or can be used in pairs as 16-bit registers (BC, B'C', DE, D'E', HL, and H'L').

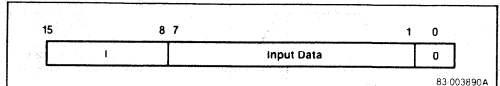
The main registers are used when instructions are executed normally. The contents of the main and alternate registers are exchanged using the EX instruction. The alternate registers may be used for background operation or to save the contents of the main registers when an interrupt is processed.

#### Interrupt Page Address Register (I)

This 8-bit register is used to generate addresses in maskable interrupt mode 2. See figure 2. These addresses are used with externally input data to reference an interrupt start address table.

This register is cleared to 00H at reset.

Figure 2. Interrupt Reference Address



#### Memory Refresh Register (R)

This 7-bit register retains the refresh address for the external dynamic memory. The contents of this register are automatically incremented in each opcode fetch (M1) cycle. The contents of this register are output on the lower 7 bits of the address bus (A<sub>6</sub>-A<sub>0</sub>).

This register is cleared to 00H at reset.

### Timing

This section describes read and write timing for memory and I/O devices in connection with CPU operation timing. A single clock cycle (from one leading edge to the next) is defined as one timing state. The nth state is represented as T<sub>n</sub>. A single instruction consists of two to six machine cycles. A single machine cycle requires three to six timing states. The nth machine cycle is represented as M<sub>n</sub>.

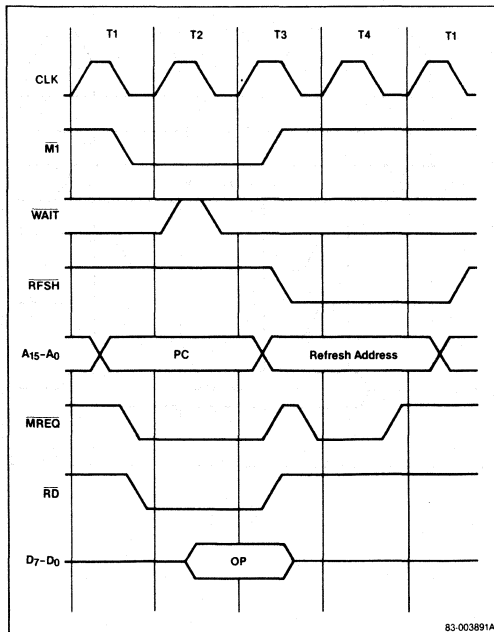
Table 1 lists the number of states normally required by each cycle.

**Table 1. Timing States per Cycle**

Cycle	Number of States per Machine Cycle
Opcode fetch	4
Memory read	3
Memory write	3
I/O read	4
I/O write	4

The four states for I/O read and write include a single wait state (TW). The μPD70008/A inserts one wait state in every I/O read or write. Slower external devices may assert the WAIT signal to request longer read and write access times. This time will be added to the original number of clock states. The WAIT signal is monitored on the trailing edge of clock state T2. If WAIT is asserted, a wait state (TW) is generated. The μPD70008/A continues to monitor WAIT on the clock's trailing edge, and supplies additional wait states as long as that signal is asserted. When WAIT is released the μPD70008/A proceeds to the T3 state.

**Figure 3. Opcode Fetch Cycle**



### Opcode Fetch Cycle

The first machine cycle of each instruction, M1, is the opcode fetch cycle. See figure 3. The opcode is fetched from memory during the first half of this cycle, and the dynamic memory is refreshed during the latter half.

The memory outputs the opcode to the data bus when MREQ, RD, or M1 is asserted. It is then read into the CPU at the leading edge of clock state T3.

The CPU outputs a refresh address onto A6-A0 during T3. It is applied to the dynamic memory when RFSH or MREQ are asserted.

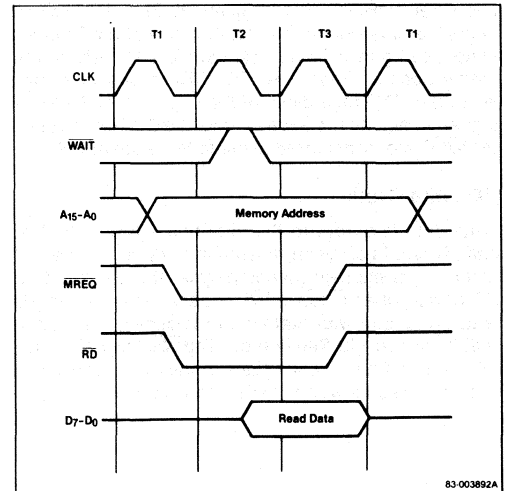
### Memory Read Cycle

The memory contents are read out to the data bus when MREQ or RD is asserted. The μPD70008/A reads data from the data bus on the trailing edge of T3. See figure 4.

### Memory Write Cycle

Write data is output to the data bus between the last half of state T1 of the current machine cycle and the first half of state T1 of the next cycle. It is written to memory when WR or MREQ is asserted. See figure 5.

**Figure 4. Memory Read Cycle**



3

Figure 5. Memory Write Cycle

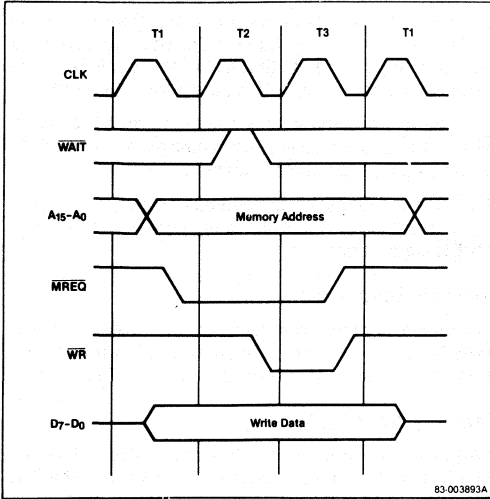
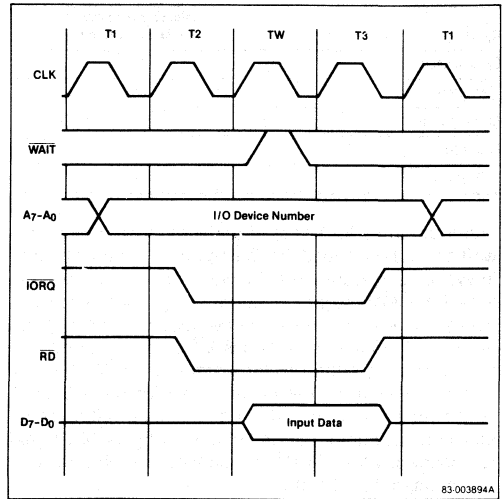


Figure 6. I/O Read Cycle



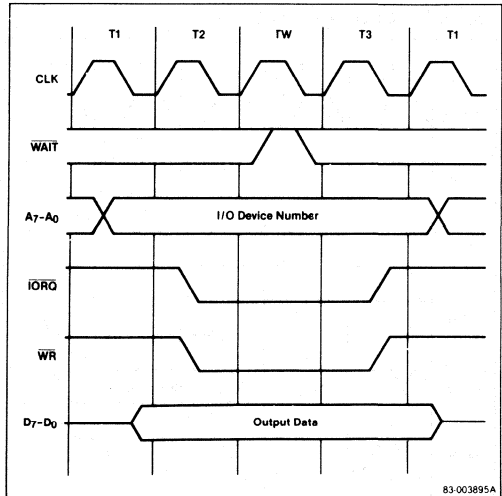
### I/O Read Cycle

The contents of an I/O device are read out to the data bus when  $\overline{IORQ}$  or  $\overline{RD}$  is asserted. The  $\mu$ PD70008/A reads the data bus on the trailing edge of the T3 clock state. See figure 6. To compensate for I/O devices with longer access times, the  $\mu$ PD70008/A generates one wait state (TW) regardless of the condition of the WAIT signal. To extend the access time, the CPU must detect the WAIT signal asserted at the falling edge of TW.

### I/O Write Cycle

Write data is output to the data bus between the last half of state T1 of the current machine cycle and the first half of T1 of the next machine cycle. It is written to an I/O device when  $\overline{IORQ}$  or  $\overline{WR}$  is asserted. As in the I/O read cycle, one wait state is automatically inserted in the I/O write cycle. See figure 7. The WAIT signal is used to insert additional wait states in the I/O write cycle in exactly the same way as in the read cycle.

Figure 7. I/O Write Cycle



## Bus Request State

The bus request causes the μPD70008/A address bus (A<sub>15</sub>-A<sub>0</sub>), data bus (D<sub>7</sub>-D<sub>0</sub>), and control bus (MREQ, IORQ, RD, and WR) pins to enter the high impedance state. This makes the buses available to external devices for DMA access.

The bus request state is controlled by the bus request (BUSRQ) signal. See figure 8. The μPD70008/A detects BUSRQ at the rising edge of the last state of each machine cycle. If it is active (low) the μPD70008/A does not move on the next machine cycle, but enters the bus request state. The μPD70008/A asserts BUSAK to indicate that the BUSRQ signal has been received, and the three buses have entered the high impedance state.

BUSRQ is checked at the rising edge of all clock states. When it becomes inactive the μPD70008/A leaves the bus request state, and proceeds to the next cycle.

BUSRQ temporarily suspends the standby mode. When BUSRQ is asserted, the μPD70008/A leaves the standby mode and enters the bus request state. When BUSRQ is released the μPD70008/A returns to the standby mode.

Interrupts are disabled during the bus request state.

## Interrupts

The μPD70008/A has two types of interrupt: maskable (INT) and nonmaskable (NMI). The nonmaskable interrupt request cannot be masked by software. It will be acknowledged unless the μPD70008/A is in the bus

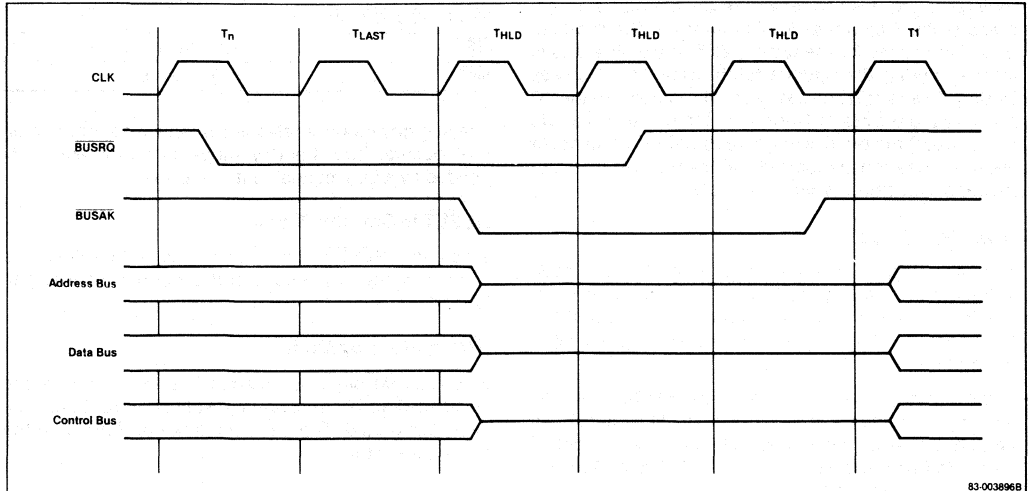
request state. The maskable interrupt can be masked by software. It is controlled by setting or resetting the interrupt enable flip-flop (IFF) using the EI or DI instructions. INT has a lower priority than the nonmaskable interrupt. The maskable interrupt will therefore not be acknowledged if there is a nonmaskable interrupt, or if the μPD70008/A is in the bus request state.

$$\text{INTI} < \text{NMI} < \text{BUSRQ} < \text{RESET}$$

## Nonmaskable Interrupt Operation

The falling edge of NMI always sets the nonmaskable interrupt flip-flop. The μPD70008/A checks the flip-flop at the rising edge of the last clock state of an instruction. If it is set, the μPD70008/A transfers control to the nonmaskable interrupt service routine. The interrupt process starts at the opcode fetch cycle (M1, 5 states) but the opcode fetched at this point is ignored. The contents of the PC are stored on the stack in the next two machine cycles (M2, 3 states and M3, 3 states). At the same time, the address 0066H is loaded into the PC, and the state of the interrupt enable flip-flop is saved to an exclusive flip-flop. The entire interrupt routine requires 3 machine cycles (T1 states). The contents of the PC and IFF are restored by the execution of the RETN instruction at the end of the interrupt procedure.

Figure 8. Bus Request State



83-003896B

### Maskable Interrupt Operation

Maskable interrupts are processed in three modes. In each mode, the  $\overline{\text{INT}}$  signal is detected at the rising edge of the last clock state of each instruction. The M1 instruction specifies which mode is to be used.

**Mode 0.** In this mode, the data placed on the bus by the interrupting device is treated as an instruction. It is fetched in the opcode fetch cycle (M1, 7 states) and executed. The instruction used in this mode is usually a CALL (3 bytes) or RST (1 byte).

If a 1-byte RST instruction is executed, the contents of the PC are saved to the stack. A fixed address specified by the opcode is loaded into the PC during the next M2 (3 states) and M3 (3 states). The execution of this interrupt requires 3 machine cycles or 13 states.

If a 3-byte CALL instruction is executed, the second and third bytes are fetched during the M2 and M3 cycles (3 states each). During M4 and M5 (3 states each), the contents of the PC are saved to the stack and the second and third bytes of the CALL instruction are loaded into the PC. This interrupt requires 5 machine cycles and a total of 19 clock states.

**Mode 1.** In this mode, the data fetched during M1 (7 states) is ignored, and the μPD70008/A proceeds to the next cycle. During the M2 and M3 machine cycles (3 states each), the contents of the PC are saved to the stack and replaced by the interrupt address 0038H. This interrupt requires 3 machine cycles or 13 states.

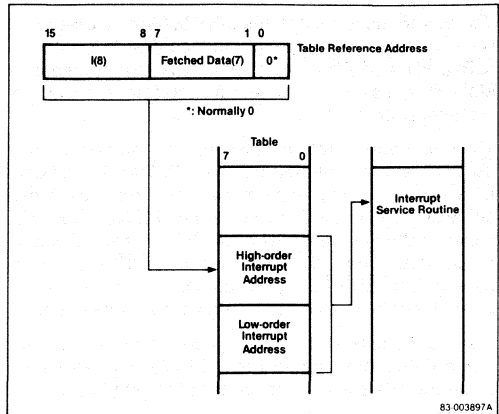
**Mode 2.** In this mode, the data fetched from the interrupting device and the contents of the interrupt page register (I) are used to reference an interrupt start address table. Program execution jumps to the 16-bit address referenced by the table. See figure 9. The data is fetched during the opcode fetch cycle (M1, 7 states). During M2 and M3 (3 states each) the contents of the PC are saved to the stack. During the M4 and M5 cycles, (3 states each) the table is referenced and the contents of the table location are loaded into the PC. This interrupt requires 5 machine cycles or 19 states.

### Standby Mode

The μPD70008/A is provided with a standby mode (HALT). In the standby mode, power consumption is approximately 2% of normal operating power consumption. The standby mode is set by executing the HALT instruction.

In the standby mode, the state of the μPD70008/A is retained. The contents of all registers and the state of all flags are retained as well. Clock signals are supplied only to indispensable circuits in the μPD70008/A to minimize power consumption.

Figure 9. Interrupt Address Table



External operations such as memory access and memory refresh are not performed in the standby mode.

Table 2 shows the state of each output pin in the standby mode.

Table 2. Standby Mode

Pin	Status
Data bus D <sub>7</sub> -D <sub>0</sub>	High level, pulled up through internal resistance
Address bus A <sub>15</sub> -A <sub>0</sub>	High or low level signals
Control bus RD, WR, MREQ, IORQ, M1	High level (inactive)
RFSH	High level (inactive)
HALT	Low level (active)

The standby mode is released when a reset or an interrupt occurs. The standby mode is temporarily suspended by a bus request, but not released.

### RESET in Standby Mode

When the RESET signal becomes active (low) the standby mode is released and a normal reset is performed.

### NMI in Standby Mode

When the NMI signal is asserted (low) the standby mode is released and normal nonmaskable interrupt processing is performed. The interrupt is not performed in the bus request state.

## INT in Standby Mode

When the  $\overline{\text{INT}}$  signal is asserted (low) the standby mode is released. If the interrupt is enabled, normal interrupt processing is performed. If the interrupt is disabled, execution resumes at the instruction following the HALT instruction.

## BUSRQ in Standby Mode

The  $\overline{\text{BUSRQ}}$  signal is detected at the rising edge of each clock in the standby mode. If the  $\overline{\text{BUSRQ}}$  signal is active (low) the μPD70008/A leaves the standby mode, and enters the bus request state. When  $\overline{\text{BUSRQ}}$  is released, the standby mode is resumed. The standby mode is not released by the  $\overline{\text{BUSRQ}}$  signal.

## RESET

The  $\overline{\text{RESET}}$  signal must be asserted (low) for over 3 clock cycles to be recognized. The following steps are the reset initialization process:

- The program counter (PC) is cleared to 0000H.
- The interrupt enable flip-flop (IFF) is reset to 0, disabling maskable interrupts. The interrupt mode is set to 0.
- The interrupt page address register (I) is cleared to 00H.
- The memory refresh register (R) is cleared to 00H.
- The address bus (A<sub>15</sub>-A<sub>0</sub>) and data bus (D<sub>7</sub>-D<sub>0</sub>) are set to high impedance.
- All control outputs are set in their inactive state.
- The standby mode is released.

The following registers are undefined at reset:

- Stack pointer (SP)
- Accumulators (A, A')
- Flag registers (F, F')
- General purpose registers (B, B', C, C', D, D', E, E', H, H', L, L')
- Index registers (IX, IY)

When  $\overline{\text{RESET}}$  is released the program will begin execution from location 0000H.

## Instruction Set

Each operand should be written in the operand column of an instruction according to the description in table 3. Capital letters are keywords and should be written as they appear.

**Table 3. Operand Description**

Identifier	Description
addr	16-bit immediate data or label
faddr	00H, 08H, 10H, 18H, 20H, 28H, 30H, 38H immediate data or label
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label (bit specification of 8-bit register / memory)
d	8-bit displacement (signed 2's complement)
r	A, B, C, D, E, H, L
r'	A', B', C', D', E', H', L'
rp	BC, DE, HL, AF
rp1	BC, DE, HL, SP
rp2	BC, DE, IX, SP
rp3	BC, DE, IY, SP
e	Displacement for relative jump (signed 2's complement)

## Selection of Register and Condition

rp	qq	rpl	ss, ddd	rp2	pp	rp3	rr
BC	00	BC	00	BC	00	BC	00
DE	01	DE	01	DE	01	DE	01
HL	10	HL	10	IX	10	IY	10
AF	11	SP	11	SP	11	SP	11

r, r'	r, r'	bit	b	faddr	t
B B'	000	0	000	00H	000
C C'	001	1	001	08H	001
D D'	010	2	010	10H	010
E E'	011	3	011	18H	011
H H'	100	4	100	20H	100
L L'	101	5	101	28H	101
A A'	111	6	110	30H	110
		7	111	38H	111

## Flag Operation

(Blank): Flag not affected

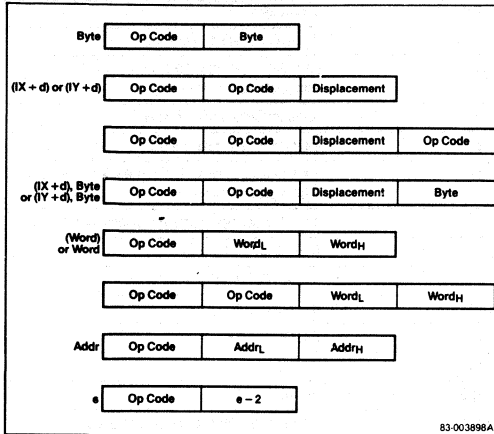
0: Flag reset

1: Flag set

X: Flag affected according to result of operation

U: Flag unknown

Structure of Instruction Byte for Addressing



## Instruction Set

Mnemonic	Operands	Operation	Operation Code																Flags							
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	No. of Clocks	No. of Bytes	S	Z	H	P	V	N
LD	r, r'	r ← r'	0	1	r	r	r'											4	1							
	r, byte	r ← byte	0	0	r	1	1	0										7	2							
	r, (HL)	r ← (HL)	0	1	r	1	1	0										7	1							
	r, (IX + d)	r ← (IX + disp)	1	1	0	1	1	0	1	0	1	r	1	1	0	19	3									
		disp																								
	r, ((Y + d)	r ← ((Y + disp)	1	1	1	1	1	0	1	0	1	r	1	1	0	19	3									
		disp																								
	(HL), r	(HL) ← r	0	1	1	1	0	r										7	1							
	(IX + d), r	(IX + disp) ← r	1	1	0	1	1	0	1	0	1	1	1	0	r	19	3									
		disp																								
(Y + d), r	(Y + disp) ← r	1	1	1	1	1	0	1	0	1	1	1	0	r	19	3										
	disp																									
(HL), byte	(HL) ← byte	0	0	1	1	0	1	1	0								10	2								
(IX + d), byte	(IX + disp) ← byte	1	1	0	1	1	1	0	1	0	1	1	0	1	1	0	19	4								
	disp																									
(Y + d), byte	(Y + disp) ← byte	1	1	1	1	1	0	1	0	1	1	0	1	1	0	19	4									
	disp																									
A, (BC)	A ← (BC)	0	0	0	0	1	0	1	0								7	1								
A, (DE)	A ← (DE)	0	0	0	1	0	1	0									7	1								
A, (word)	A ← (word)	0	0	1	1	0	0	1	0								13	3								
(BC), A	(BC) ← A	0	0	0	0	0	1	0									7	1								
(DE), A	(DE) ← A	0	0	0	1	0	0	1	0								7	1								
(word), A	(word) ← A	0	0	1	1	0	0	1	0								13	3								
A, I	A ← I	1	1	1	0	1	1	0	1	0	1	0	1	1	1	1	9	2	x	x	0	IFF	0			
A, R	A ← R	1	1	1	0	1	1	0	1	0	1	0	1	1	1	1	9	2	x	x	0	IFF	0			
I, A	I ← A	1	1	1	0	1	1	0	1	0	1	0	0	1	1	1	9	2								
R, A	R ← A	1	1	1	0	1	1	0	1	0	1	0	0	1	1	1	9	2								
rp1, word	rp1 ← word	0	0	d	0	0	0	1									10	3								
IX, word	IX ← word	1	1	0	1	1	0	1	0	1	0	0	1	0	0	1	14	4								
IY, word	IY ← word	1	1	1	1	1	0	1	0	1	0	0	1	0	0	1	14	4								
HL, (word)	H ← (word + 1), L ← (word)	0	0	1	0	1	0	1	0								16	3								



Instruction Set (cont)

Mnemonic	Operands	Operation	Operation Code																No. of Cycles	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			S	Z	H	PIV	N	C
LD	rp <sub>L</sub> (word)	rp <sub>L</sub> ← (word + 1), rp <sub>L</sub> ← (word)	1	1	1	0	1	1	0	1	0	1	0	1	d	d	1	0	1	1	20	4				
	IX <sub>L</sub> (word)	IX <sub>L</sub> ← (word + 1), IX <sub>L</sub> ← (word)	1	1	0	1	1	1	0	1	0	0	1	0	1	0	1	0	1	0	20	4				
	IY <sub>L</sub> (word)	IY <sub>L</sub> ← (word + 1), IY <sub>L</sub> ← (word)	1	1	1	1	1	1	0	1	0	0	1	0	1	0	1	0	1	0	20	4				
	(word), HL	(word + 1) ← H, (word) ← L	0	0	1	0	0	0	1	0											16	3				
	(word), rp <sub>L</sub>	(word + 1) ← rp <sub>L</sub> , (word) ← rp <sub>L</sub>	1	1	1	0	1	1	0	1	0	1	d	d	0	0	1	1	20	4						
	(word), IX	(word + 1) ← IX <sub>L</sub> , (word) ← IX <sub>L</sub>	1	1	0	1	1	0	1	0	1	d	d	0	0	1	1	20	4							
	(word), IY	(word + 1) ← IY <sub>L</sub> , (word) ← IY <sub>L</sub>	1	1	1	1	1	0	1	0	0	1	0	0	0	1	0	20	4							
	SP, HL	SP ← HL	1	1	1	1	1	0	0	1											6	1				
	SP, IX	SP ← IX	1	1	0	1	1	0	1	1	1	1	1	0	0	1	10	2								
	SP, IY	SP ← IY	1	1	1	1	1	0	1	1	1	1	1	0	0	1	10	2								
PUSH	rp	(SP - 1) ← rp <sub>L</sub> , (SP - 2) ← rp <sub>H</sub> , SP ← SP - 2	1	1	q	q	0	1	0	1											11	1				
	IX	(SP - 1) ← IX <sub>L</sub> , (SP - 2) ← IX <sub>H</sub> , SP ← SP - 2	1	1	0	1	1	1	0	1	1	1	0	0	1	0	1	15	2							
	IY	(SP - 1) ← IY <sub>L</sub> , (SP - 2) ← IY <sub>H</sub> , SP ← SP - 2	1	1	1	1	1	1	0	1	1	1	1	0	0	1	0	1	15	2						
	rp	rp <sub>L</sub> ← (SP), rp <sub>H</sub> ← (SP + 1), SP ← SP + 2	1	1	q	q	0	0	0	1											10	1				
IX	IX <sub>L</sub> ← (SP), IX <sub>H</sub> ← (SP + 1), SP ← SP + 2	1	1	0	1	1	1	0	1	1	1	1	0	0	0	1	14	2								
	IY <sub>L</sub> ← (SP), IY <sub>H</sub> ← (SP + 1), SP ← SP + 2	1	1	1	1	1	1	0	1	1	1	1	0	0	0	1	14	2								
	IY <sub>L</sub> ← (SP), IY <sub>H</sub> ← (SP + 1), SP ← SP + 2	1	1	1	1	1	1	0	1	1	1	1	0	0	0	1	14	2								

## Instruction Set (cont)

Mnemonic	Operands	Operation	Operation Code																Flags										
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	Clocks	Bytes	S	Z	H	P/V	N	C			
<b>Data Conversion Instructions</b>																													
EX	DE, HL	DE ↔ HL	1	1	1	0	1	0	1	1	1	1	1	0	1	1	4	1											
	AF, AF'	A ↔ A', F ↔ F'	0	0	0	0	1	0	0	0	0	0	0	0	0	0	4	1											
EXX		BC ↔ BC', DE ↔ DE', HL ↔ HL'	1	1	0	1	1	0	0	1							4	1											
EX	(SP), HL	(SP) ↔ L, (SP + 1) ↔ H, SP → SP + 2	1	1	1	0	0	0	1	1							19	1											
	(SP), IX	(SP) ↔ IXL, (SP + 1) ↔ IXH, SP → SP + 2	1	1	0	1	1	1	0	1	1	1	1	0	0	0	1	1	23										
	(SP), IY	(SP) ↔ IYL, (SP + 1) ↔ IYH, SP → SP + 2	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	23										
<b>Block Transfer Instructions</b>																													
LDI		(DE) ↔ (HL), DE → DE + 1, HL → HL + 1, BC → BC - 1	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	16	2	0	x	0							
LDIR		(DE) ↔ (HL), DE → DE + 1, HL → HL + 1, BC → BC - 1, End if BC = 0	1	1	1	0	1	1	0	1	1	0	1	0	1	0	0	21/16(1)	2	0	0	0							
LDD		(DE) ↔ (HL), DE → DE - 1, HL → HL - 1, BC → BC - 1	1	1	1	0	1	1	0	1	1	0	1	0	1	0	0	16	2	0	x	0							
LDDR		(DE) ↔ (HL), DE → DE - 1, HL → HL - 1, BC → BC - 1, End if BC = 0	1	1	1	0	1	1	0	1	1	0	1	0	1	0	0	21/16(1)	2	0	0	0							
<b>Block Search Instructions</b>																													
CPI		A - (HL), HL → HL + 1, BC → BC - 1	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	16	2	x	x	x	x	x	x	x	x	x	1
CPRI		A - (HL), HL → HL + 1, BC → BC - 1, End if A = (HL) or BC = 0	1	1	1	0	1	1	0	1	1	0	1	0	1	0	1	21/16(2)	2	x	x	x	x	x	x	x	x	x	1

**Instruction Set (cont)**

Mnemonic	Operands	Operation	Operation Code																Flags										
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	No. of Clocks	No. of Bytes	S	Z	H	P	V	N	C		
Block Search Instructions (cont)																													
CPD		A ← (HL), HL ← HL - 1, BC ← BC - 1	1	1	1	0	1	1	0	1	1	0	1	0	1	0	0	1	1	6	2								
CPDR		A ← (HL), HL ← HL - 1, BC ← BC - 1, End if A = (HL) or BC = 0	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	21/16(2)	2										
Eight-Bit Arithmetic Operation Instructions																													
ADD	A, r	A ← A + r	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4	1											
	A, byte	A ← A + byte	1	1	0	0	0	1	1	0	0	0	0	0	0	7	2												
	A, (HL)	A ← A + (HL)	1	0	0	0	0	1	1	0	0	0	0	0	7	1													
	A, (IX + d)	A ← A + (IX + disp)	1	1	0	1	1	0	1	1	0	0	0	1	1	0	19	3											
	A, (IY + d)	A ← A + (IY + disp)	1	1	1	1	1	0	1	1	0	0	0	1	1	0	19	3											
disp																													
disp																													
ADC	A, r	A ← A + r + CY	1	0	0	0	1	0	0	0	0	0	0	0	4	1													
	A, byte	A ← A + byte + CY	1	1	0	0	1	1	1	0	0	0	0	0	7	2													
	A, (HL)	A ← A + (HL) + CY	1	0	0	0	1	1	1	0	0	0	0	7	1														
	A, (IX + d)	A ← A + (IX + disp) + CY	1	1	0	1	1	0	1	1	0	0	1	1	0	19	3												
	A, (IY + d)	A ← A + (IY + disp) + CY	1	1	1	1	1	0	1	1	0	0	1	1	0	19	3												
disp																													
disp																													
SUB	A, r	A ← A - r	1	0	0	1	0	0	0	0	0	0	0	4	1														
	A, byte	A ← A - byte	1	1	0	1	0	1	1	0	0	0	0	7	2														
	A, (HL)	A ← A - (HL)	1	0	0	1	0	1	1	0	0	0	0	7	1														
	A, (IX + d)	A ← A - (IX + disp)	1	1	0	1	1	0	1	1	0	0	1	1	0	19	3												
	A, (IY + d)	A ← A - (IY + disp)	1	1	1	1	1	0	1	1	0	0	1	1	0	19	3												
disp																													
disp																													

## Instruction Set (cont)

Mnemonic	Operands	Operation	Operation Code																Flags																						
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	No. of Clocks	No. of Bytes	S	Z	H	PIV	N	C															
<b>Eight-Bit Arithmetic Operation Instructions (cont)</b>																																									
SBC	A, r	A ← A - r - CY	1	0	0	1	1	r																	4	1	x	x	x	V	1	x									
	A, byte	A ← A - byte - CY	1	1	0	1	1	1	0																	7	2	x	x	x	V	1	x								
	A, (HL)	A ← A - (HL) - CY	1	0	0	1	1	1	0																	7	1	x	x	x	V	1	x								
	A, (IX + d)	A ← A - (IX + disp) - CY	1	1	0	1	1	0	1	1	0	0	1	1	1	0																	19	3	x	x	x	V	1	x	
		disp																																							
	A, (IV + d)	A ← A - (IV + disp) - CY	1	1	1	1	1	0	1	1	0	0	1	1	1	1	0																	19	3	x	x	x	V	1	x
		disp																																							
<b>Eight-Bit Logical Operation Instructions</b>																																									
AND	A, r	A ← A AND r	1	0	1	0	0	r																	4	1	x	x	1	P	0	0									
	A, byte	A ← A AND byte	1	1	1	0	0	1	1	0																	7	2	x	x	1	P	0	0							
	A, (HL)	A ← A AND (HL)	1	0	1	0	0	1	1	0																	7	1	x	x	1	P	0	0							
	A, (IX + d)	A ← A AND (IX + disp) - CY	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0																	19	3	x	x	1	P	0	0
		disp																																							
	A, (IV + d)	A ← A AND (IV + disp)	1	1	1	1	1	0	1	1	0	1	0	0	1	1	0																	19	3	x	x	1	P	0	0
		disp																																							
OR	A, r	A ← A OR r	1	0	1	1	0	r																	4	1	x	x	0	P	0	0									
	A, byte	A ← A OR byte	1	1	1	1	0	1	1	0																	7	2	x	x	0	P	0	0							
	A, (HL)	A ← A OR (HL)	1	0	1	1	0	1	1	0																	7	1	x	x	0	P	0	0							
	A, (IX + d)	A ← A OR (IX + disp)	1	1	0	1	1	0	1	1	0	1	0	1	1	0																	19	3	x	x	0	P	0	0	
		disp																																							
	A, (IV + d)	A ← A OR (IV + disp)	1	1	1	1	1	0	1	1	0	1	0	1	1	0																	19	3	x	x	0	P	0	0	
		disp																																							
XOR	A, r	A ← A XOR r	1	0	1	0	1	r																	4	1	x	x	0	P	0	0									
	A, byte	A ← A XOR byte	1	1	1	0	1	1	0																	7	2	x	x	0	P	0	0								
	A, (HL)	A ← A XOR (HL)	1	0	1	0	1	1	0																	7	1	x	x	0	P	0	0								
	A, (IX + d)	A ← A XOR (IX + disp)	1	1	0	1	1	0	1	1	0	1	0	1	1	0																	19	3	x	x	0	P	0	0	
		disp																																							
	A, (IV + d)	A ← A XOR (IV + disp)	1	1	1	1	1	0	1	1	0	1	0	1	1	0																	19	3	x	x	0	P	0	0	
		disp																																							



## Instruction Set (cont)

Mnemonic	Operands	Operation	Operation Code													No. of				Flags												
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	Clocks	Bytes	S	Z	H	PIV	N	C						
<b>Accumulator Operation Instructions</b>																																
DAA		Decimal adjust accumulator	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	4	1	x	x	x	x	P	x							
CPL		$A \leftarrow \bar{A}$	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	4	1					1								
NEG		$A \leftarrow \bar{A} + 1$	1	1	1	0	1	0	1	0	1	0	1	0	0	1	0	8	2	x	x	x	x	V	1	x						
CCF		$CY \leftarrow \bar{CY}$	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	4	1					U	0	x						
SCF		$CY \leftarrow 1$	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	4	1					0	0	1						
<b>Rotate Instructions</b>																																
RLCA			0	0	0	0	0	1	1	1	1	1	1	1	1	1	4	1					0	0	0	x						
RLA			0	0	0	1	0	1	1	1	1	1	1	1	1	1	4	1					0	0	0	x						
RRCA			0	0	0	0	1	1	1	1	1	1	1	1	1	1	4	1					0	0	0	x						
RRA			0	0	0	1	1	1	1	1	1	1	1	1	1	1	4	1					0	0	0	x						
RLC	r		1	1	0	0	1	0	1	1	0	0	0	0	0	0	r	8	2	x	x	0	P	0	x							
	(HL)		1	1	0	0	1	0	1	1	0	0	0	0	0	1	1	15	2	x	x	0	P	0	x							
	(IX + d)		1	1	0	1	1	0	1	1	1	0	0	1	0	1	1	23	4	x	x	0	P	0	x							
(IV + d)	r, (HL), (IX + disp), (IV + disp)		disp													0	0	0	0	1	1	0										
			1	1	1	1	1	0	1	1	1	0	0	1	0	1	1	23	4	x	x	0	P	0	x							
			disp													0	0	0	0	1	1	0										
RL	r		1	1	0	0	1	0	1	1	0	0	0	1	0	r	8	2	x	x	0	P	0	x								
	(HL)		1	1	0	0	1	0	1	1	0	0	0	1	0	1	15	2	x	x	0	P	0	x								
	(IX + d)		1	1	0	1	1	0	1	1	1	0	1	0	1	1	23	4	x	x	0	P	0	x								
(IV + d)	r, (HL), (IX + disp), (IV + disp)		disp													0	0	0	1	0	1	1	0									
			1	1	1	1	1	0	1	1	1	0	0	1	0	1	1	23	4	x	x	0	P	0	x							
			disp													0	0	0	1	0	1	1	0									
RRC	r		1	1	0	0	1	0	1	1	0	0	0	0	1	r	8	2	x	x	0	P	0	x								
	(HL)		1	1	0	0	1	0	1	1	0	0	0	0	1	1	15	2	x	x	0	P	0	x								
	(IX + d)		1	1	0	1	1	0	1	1	1	0	0	1	0	1	23	4	x	x	0	P	0	x								
(IV + d)	r, (HL), (IX + disp), (IV + disp)		disp													0	0	0	1	1	1	0										
			1	1	1	1	1	0	1	1	1	0	0	1	0	1	1	23	4	x	x	0	P	0	x							
			disp													0	0	0	1	1	1	0										

**Instruction Set (cont)**

Mnemonic	Operands	Operation	Operation Code										No. of				Flags											
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	Clocks	Bytes	S	Z	H	P	V	N	C	
<b>Rotate Instructions (cont)</b>																												
RR	r		1	1	0	0	1	0	1	1	0	0	0	1	1	r	8	2	x	x	0	P	0	x				
	(HL)		1	1	0	0	1	0	1	1	0	0	0	1	1	1	15	2	x	x	0	P	0	x				
	(IX + d)		1	1	0	1	1	0	1	1	0	0	1	0	1	1	23	4	x	x	0	P	0	x				
		disp														0												
	(Y + d)	r, (HL), (IX + disp), (Y + disp)	1	1	1	1	0	1	1	1	0	0	1	0	1	1	23	4	x	x	0	P	0	x				
		disp														0												
RLD			1	1	1	0	1	1	0	1	0	1	1	1	1	1	18	2	x	x	0	P	0					
RRD			1	1	1	0	1	1	0	1	1	0	1	1	1	1	18	2	x	x	0	P	0					
<b>Shift Instructions</b>																												
SLA	r		1	1	0	0	1	0	1	1	0	0	1	0	0	r	8	2	x	x	0	P	0	x				
	(HL)		1	1	0	0	1	0	1	1	0	0	1	0	0	1	15	2	x	x	0	P	0	x				
	(IX + d)		1	1	0	1	1	0	1	1	0	0	1	0	1	1	23	4	x	x	0	P	0	x				
		disp														0												
	(Y + d)	r, (HL), (IX + disp), (Y + disp)	1	1	1	1	0	1	1	1	0	0	1	0	1	1	23	4	x	x	0	P	0	x				
		disp														0												
SRA	r		1	1	0	0	1	0	1	1	0	0	1	0	1	r	8	2	x	x	0	P	0	x				
	(HL)		1	1	0	0	1	0	1	1	0	0	1	0	1	1	15	2	x	x	0	P	0	x				
	(IX + d)		1	1	0	1	1	0	1	1	0	0	1	0	1	1	23	4	x	x	0	P	0	x				
		disp														0												
	(Y + d)	r, (HL), (IX + disp), (Y + disp)	1	1	1	1	0	1	1	1	0	0	1	0	1	1	23	4	x	x	0	P	0	x				
		disp														0												
SRL	r		1	1	0	0	1	0	1	1	0	0	1	1	1	r	8	2	x	x	0	P	0	x				
	(HL)		1	1	0	0	1	0	1	1	0	0	1	1	1	1	15	2	x	x	0	P	0	x				
	(IX + d)		1	1	0	1	1	0	1	1	0	0	1	0	1	1	23	4	x	x	0	P	0	x				
		disp														0												
	(Y + d)	r, (HL), (IX + disp), (Y + disp)	1	1	1	1	0	1	1	1	0	0	1	0	1	1	23	4	x	x	0	P	0	x				
		disp														0												

## Instruction Set (cont)

Mnemonic	Operands	Operation	Operation Code																Flags																				
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	No. of Checks	No. of Bytes	S	Z	H	P	V	N	C												
<b>Bit Operation Instructions</b>																																							
BIT	bit, r	$Z \leftarrow \overline{f_b}$	1	1	0	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	8	2	U	x	1	U	0			
	bit, (HL)	$Z \leftarrow \overline{(HL)_b}$	1	1	0	0	1	0	1	1	0	1	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	12	2	U	x	1	U	0	
	bit, (IX + d)	$Z \leftarrow \overline{(IX + disp)_b}$	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	0	1	0	20	4	U	x	1	U	0
		disp		0	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	0	1	0	20	4	U	x	1	U	0
SET	bit, r	$f_b \leftarrow 1$	1	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	8	2						
	bit, (HL)	$(HL)_b \leftarrow 1$	1	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	15	2						
	bit (IX + d)	$(IX + disp)_b \leftarrow 1$	1	1	0	1	1	0	1	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	23	4					
		disp		1	1	0	1	1	0	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	1	1	23	4					
RES	bit, r	$f_b \leftarrow 0$	1	1	0	0	1	0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	8	2						
	bit, (HL)	$(HL)_b \leftarrow 0$	1	1	0	0	1	0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	15	2						
	bit (IX + d)	$(IX + disp)_b \leftarrow 0$	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1	1	0	1	0	1	1	23	4						
		disp		1	1	0	1	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1	1	1	0	1	0	1	1	1	1	23	4					
Jump Instructions	addr	$PC \leftarrow addr$	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	3						
	NZ, addr	$If Z = 0, PC \leftarrow addr$	1	1	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	3						
	Z, addr	$If Z = 1, PC \leftarrow addr$	1	1	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	3						
	NC, addr	$If C = 0, PC \leftarrow addr$	1	1	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	3						
JP	C, addr	$If C = 1, PC \leftarrow addr$	1	1	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	3						
	PO, addr	$If P = 0, PC \leftarrow addr$	1	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	3						
	PE, addr	$If P = 1, PC \leftarrow addr$	1	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	3						
	P, addr	$If S = 0, PC \leftarrow addr$	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	3						
M, addr	$If S = 1, PC \leftarrow addr$	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	3						



**Instruction Set (cont)**

Mnemonic	Operands	Operation	Operation Code																Flags								
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	No. of Clocks	No. of Bytes	S	Z	H	P	V	N	C
JR	e	PC ← PC + e	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	12	2								
	NZ, e	If Z = 0, PC ← PC + e	0	0	1	0	0	0	0	0	0	0	0	0	0	0	12/7(3)	2									
	Z, e	If Z = 1, PC ← PC + e	0	0	1	0	1	0	0	0	0	0	0	0	0	0	12/7(3)	2									
	NC, e	If C = 0, PC ← PC + e	0	0	1	1	0	0	0	0	0	0	0	0	0	0	12/7(3)	2									
	C, e	If C = 1, PC ← PC + e	0	0	1	1	1	0	0	0	0	0	0	0	0	0	12/7(3)	2									
JP	(HL)	PC ← HL	1	1	1	0	1	0	0	0	1						4	1									
(IX)	PC ← IX	1	1	0	1	1	0	1	1	1	1	0	1	0	0	1	8	2									
(IV)	PC ← IV	1	1	1	1	1	0	1	1	1	1	0	1	0	0	1	8	2									
DJNZ	e	B ← B - 1; if B ≠ 0, PC ← PC + e	0	0	0	1	0	0	0	0	0						8/13(4)	2									
<b>Call instructions</b>																											
CALL	addr	(SP - 1) ← PC <sub>H</sub> , (SP - 2) ← PC <sub>L</sub> , SP ← SP - 2, PC ← addr	1	1	0	0	1	1	0	1							17	3									
NZ, addr		If conditions met, (SP - 1) ← PC <sub>H</sub> , (SP - 2) ← PC <sub>L</sub> , SP ← SP - 2, PC ← addr	1	1	0	0	0	1	0	0							17/10(5)	3									
Z, addr		If conditions met, (SP - 1) ← PC <sub>H</sub> , (SP - 2) ← PC <sub>L</sub> , SP ← SP - 2, PC ← addr	1	1	0	0	1	1	0	0							17/10(5)	3									
NC, addr		If conditions met, (SP - 1) ← PC <sub>H</sub> , (SP - 2) ← PC <sub>L</sub> , SP ← SP - 2, PC ← addr	1	1	0	1	0	1	0	0							17/10(5)	3									
C, addr		If conditions met, (SP - 1) ← PC <sub>H</sub> , (SP - 2) ← PC <sub>L</sub> , SP ← SP - 2, PC ← addr	1	1	0	1	1	1	0	0							17/10(5)	3									
P0, addr		If conditions met, (SP - 1) ← PC <sub>H</sub> , (SP - 2) ← PC <sub>L</sub> , SP ← SP - 2, PC ← addr	1	1	1	0	0	1	0	0							17/10(5)	3									

## Instruction Set (cont)

Mnemonic	Operands	Operation	Operation Code																No. of Cycles	No. of Bytes	Flags														
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			S	Z	H	P	I	V	N	C							
CALL		PE, addr	If conditions met, (SP - 1) ← 1 1 1 0 1 1 0 0																17/10(5)	3															
		PC <sub>H</sub> , (SP - 2) ← PC <sub>L</sub> , SP ← SP - 2, PC ← addr																																	
P, addr			If conditions met, (SP - 1) ← 1 1 1 1 0 1 0 0																17/10(5)	3															
		PC <sub>H</sub> , (SP - 2) ← PC <sub>L</sub> , SP ← SP - 2, PC ← addr																																	
M, addr			If conditions met, (SP - 1) ← 1 1 1 1 1 1 0 0																17/10(5)	3															
		PC <sub>H</sub> , (SP - 2) ← PC <sub>L</sub> , SP ← SP - 2, PC ← addr																																	
RST		faddr	If conditions met, (SP - 1) ← 1 1 1 1 1 1 1 1																11	1															
		PC <sub>L</sub> ← faddr																																	
Return Instructions																																			
RET			PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP + 1), SP ← SP + 2																10	1															
		NZ	If conditions met, PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP + 1), SP ← SP + 2																11/5(6)	1															
		Z	If conditions met, PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP + 1), SP ← SP + 2																11/5(6)	1															
		NC	If conditions met, PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP + 1), SP ← SP + 2																11/5(6)	1															
		C	If conditions met, PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP + 1), SP ← SP + 2																11/5(6)	1															
		PO	If conditions met, PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP + 1), SP ← SP + 2																11/5(6)	1															
		PE	If conditions met, PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP + 1), SP ← SP + 2																11/5(6)	1															
		P	If conditions met, PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP + 1), SP ← SP + 2																11/5(6)	1															
		M	If conditions met, PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP + 1), SP ← SP + 2																11/5(6)	1															
RETI			Return from interrupt																14	2															
RETN			Return from interrupt, nonmaskable																14	2															

**Instruction Set (cont)**

Mnemonic	Operands	Operation	Operation Code																Flags									
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	No. of Cycles	No. of Bytes	S	Z	H	PIV	N	C		
Return Instructions (cont)																												
IN	A, byte	A ← (byte), A <sub>7</sub> -A <sub>0</sub> ← byte, A <sub>15</sub> -A <sub>8</sub> ← byte	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	11	2									
	r (C)	r ← (C), A <sub>7</sub> -A <sub>0</sub> ← C, A <sub>15</sub> -A <sub>8</sub> ← B	1	1	1	0	1	1	0	1	0	1	1	0	0	1	r	0	0	0	12	2	x	x	x	x	P	0
INI		(HL) ← (C), B ← B - 1, HL ← HL + 1, A <sub>7</sub> -A <sub>0</sub> ← C, A <sub>15</sub> -A <sub>8</sub> ← B	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	16	2	U	x	U	U	U	U	1		
IND		(HL) ← (C), B ← B - 1, HL ← HL - 1, A <sub>7</sub> -A <sub>0</sub> ← C, A <sub>15</sub> -A <sub>8</sub> ← B	1	1	1	0	1	1	0	1	1	0	1	0	1	0	1	16	2	U	x	U	U	U	1			
INIR		(HL) ← (C), B ← B - 1, HL ← HL + 1, A <sub>7</sub> -A <sub>0</sub> ← C, A <sub>15</sub> -A <sub>8</sub> ← B, End if B = 0	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	21/16(7)	2	U	1	U	U	U	1		
INDR		(HL) ← (C), B ← B - 1, HL ← HL - 1, A <sub>7</sub> -A <sub>0</sub> ← C, A <sub>15</sub> -A <sub>8</sub> ← B, End if B = 0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	21/16(7)	2	U	1	U	U	U	1		
OUT	byte, A	(byte) → A, A <sub>7</sub> -A <sub>0</sub> ← byte, A <sub>15</sub> -A <sub>8</sub> ← B	1	1	0	1	0	0	1	1	1	0	0	1	1	1	1	11	2									
	(C), r	(C) → r, A <sub>7</sub> -A <sub>0</sub> ← C, A <sub>15</sub> -A <sub>8</sub> ← B	1	1	1	0	1	1	0	1	0	1	1	r	0	0	1	12	2									
OUTI		(C) → (HL), B ← B - 1, HL ← HL + 1, A <sub>7</sub> -A <sub>0</sub> ← C, A <sub>15</sub> -A <sub>8</sub> ← B	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	16	2	U	x	U	U	U	1			
OUTD		(C) → (HL), B ← B - 1, HL ← HL - 1, A <sub>7</sub> -A <sub>0</sub> ← C, A <sub>15</sub> -A <sub>8</sub> ← B	1	1	1	0	1	1	0	1	1	0	1	0	1	0	1	16	2	U	x	U	U	U	1			
OUTIR		(C) → (HL), B ← B - 1, HL ← HL + 1, A <sub>7</sub> -A <sub>0</sub> ← C, A <sub>15</sub> -A <sub>8</sub> ← B, End if B = 0	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	21/16(7)	2	U	1	U	U	U	1		
OUTDR		(C) → (HL), B ← B - 1, HL ← HL - 1, A <sub>7</sub> -A <sub>0</sub> ← C, A <sub>15</sub> -A <sub>8</sub> ← B, End if B = 0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	16	2	U	1	U	U	U	1			

### Instruction Set (cont)

Mnemonic	Operands	Operation	Operation Code																Flags						
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	No. of Clocks	No. of Bytes	S	Z	H	P/V	N
NOP		No operation	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	1						
HALT		Halt	0	1	1	1	0	1	1	0								4	1						
DI		Disable interrupts (IFF ←←0)	1	1	1	1	0	0	1	1								4	1						
EI		Enable interrupts (IFF ←←1)	1	1	1	1	1	0	1	1								4	1						
IM	0	Set interrupt mode 0	1	1	1	0	1	1	0	1	0	1	0	0	1	1	0	8	2						
	1	Set interrupt mode 1	1	1	1	0	1	1	0	1	0	1	0	1	0	1	0	8	2						
	2	Set interrupt mode 2	1	1	1	0	1	1	0	1	0	1	0	1	1	1	0	8	2						

**Note:**

- (1) 21 if BC ≠ 0, 16 if BC = 0
- (2) 21 if BC ≠ 0 and A ≠ (HL), 16 if BC = 0 or A = (HL)
- (3) 12 if condition is met, 7 if not
- (4) 8 if B = 0, 13 if B ≠ 0
- (5) 17 if condition is met, 10 if not
- (6) 11 if condition is met, 5 if not
- (7) 21 if B = 0, 16 if B ≠ 0



### Description

The μPD70108 (V20) is a CMOS 16-bit microprocessor with internal 16-bit architecture and an 8-bit external data bus. The μPD70108 instruction set is a superset of the μPD8086/8088; however, mnemonics and execution times are different. The μPD70108 additionally has a powerful instruction set including bit processing, packed BCD operations, and high-speed multiplication/division operations. The μPD70108 can also execute the entire 8080 instruction set and comes with a standby mode that significantly reduces power consumption. It is software-compatible with the μPD70116 16-bit microprocessor.

### Features

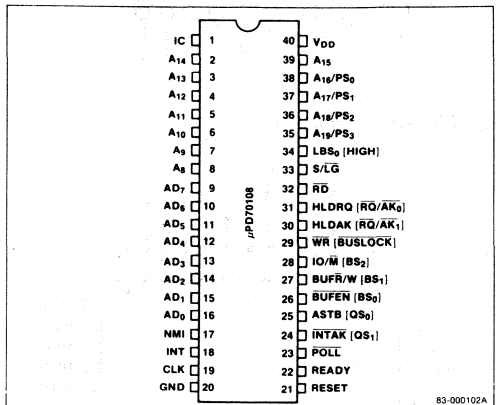
- Minimum instruction execution time: 250 ns (at 8 MHz)
- Maximum addressable memory: 1 Mbyte
- Abundant memory addressing modes
- 14 x 16-bit register set
- 101 instructions
- Instruction set is a superset of μPD8086/8088 instruction set
- Bit, byte, word, and block operations
- Bit field operation instructions
- Packed BCD instructions
- Multiplication/division instruction execution time: 4 μs to 6 μs (at 8 MHz)
- High-speed block transfer instructions: 1 Mbyte/s (at 8 MHz)
- High-speed calculation of effective addresses: 2 clock cycles in any addressing mode
- Maskable (INT) and nonmaskable (NMI) interrupt inputs
- IEEE-796 bus compatible interface
- 8080 emulation mode
- CMOS technology
- Low-power consumption
- Low-power standby mode
- Single power supply
- 5 MHz, 8 MHz or 10 MHz clock

### Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD70108C-5	40-pin plastic DIP	5 MHz
μPD70108C-8	40-pin plastic DIP	8 MHz
μPD70108C-10	40-pin plastic DIP	10 MHz
μPD70108D-5	40-pin ceramic DIP	5 MHz
μPD70108D-8	40-pin ceramic DIP	8 MHz
μPD70108D-10	40-pin ceramic DIP	10 MHz
μPD70108GC-5	52-pin plastic flat pack	5 MHz
μPD70108GC-8	52-pin plastic flat pack	8 MHz
μPD70108L-5	44-pin PLCC	5 MHz
μPD70108L-8	44-pin PLCC	8 MHz

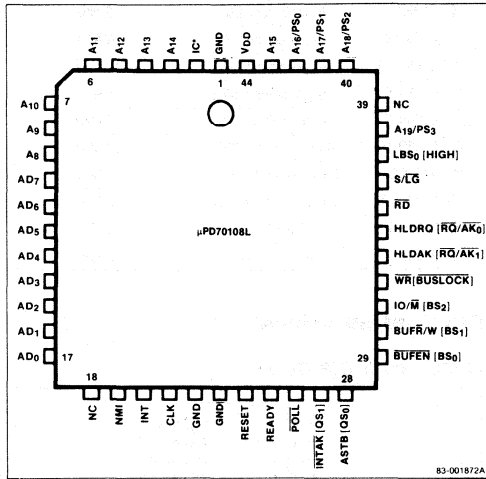
### Pin Configurations

#### 40-Pin Plastic DIP/Cerdip

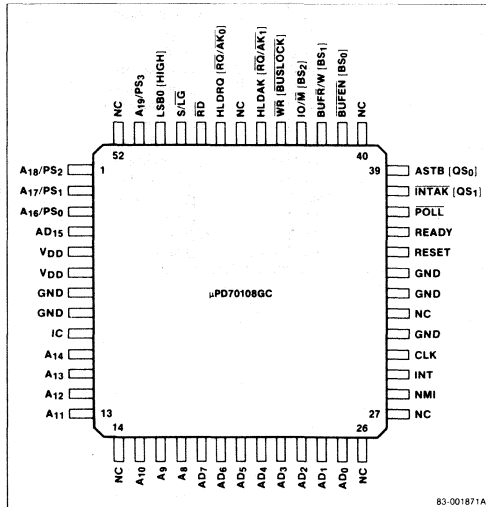


Pin Configurations (cont)

44-Pin Plastic Leaded Chip Carrier (PLCC)



52-PIN Plastic Flat Pack



Pin Identification

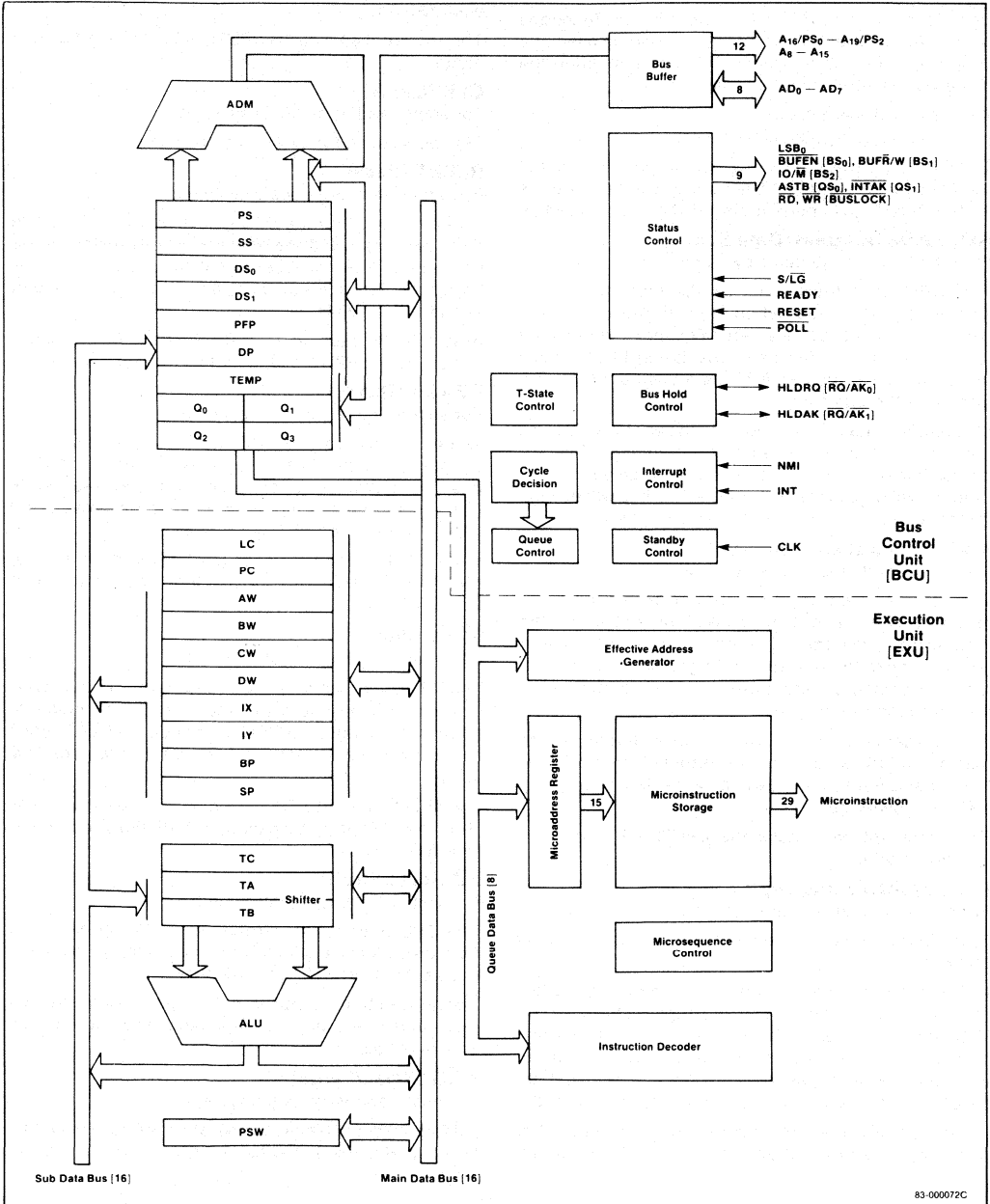
Symbol	Direction	Function
IC*		Internally connected
A <sub>14</sub> - A <sub>8</sub>	Out	Address bus, middle bits
AD <sub>7</sub> - AD <sub>0</sub>	In/Out	Address/data bus
NMI	In	Nonmaskable interrupt input
INT	In	Maskable interrupt input
CLK	In	Clock input
GND		Ground potential
RESET	In	Reset input
READY	In	Ready input
POLL	In	Poll input
INTAK (QS <sub>1</sub> )	Out	Interrupt acknowledge output (queue status bit 1 output)
ASTB (QS <sub>0</sub> )	Out	Address strobe output (queue status bit 0 output)
BUFEN (BS <sub>0</sub> )	Out	Buffer enable output (bus status bit 0 output)
BUF R/W (BS <sub>1</sub> )	Out	Buffer read/write output (bus status bit 1 output)
IO/M (BS <sub>2</sub> )	Out	Access is I/O or memory (bus status bit 2 output)
WR (BUSLOCK)	Out	Write strobe output (bus lock output)
HLD A K (RQ/AK <sub>1</sub> )	Out (In/Out)	Hold acknowledge output, (bus hold request input/acknowledge output 1)
HLD R Q (RQ/AK <sub>0</sub> )	In (In/Out)	Hold request input (bus hold request input/acknowledge output 0)
RD	Out	Read strobe output
S/LG	In	Small-scale/large-scale system input
LBS <sub>0</sub> (HIGH)	Out	Latched bus status output 0 (always high in large-scale systems)
A <sub>19</sub> /PS <sub>3</sub> - A <sub>16</sub> /PS <sub>0</sub>	Out	Address bus, high bits or processor status output
A <sub>15</sub>	Out	Address bus, bit 15
V <sub>DD</sub>		Power supply

Notes: \* IC should be connected to ground.

Where pins have different functions in small- and large-scale systems, the large-scale system pin symbol and function are in parentheses.

Unused input pins should be tied to ground or V<sub>DD</sub> to minimize power dissipation and prevent the flow of potentially harmful currents.

## Block Diagram





**Pin Functions**

Some pins of the μPD70108 have different functions according to whether the microprocessor is used in a small- or large-scale system. Other pins function the same way in either type of system.

**A<sub>15</sub> - A<sub>8</sub> [Address Bus]**

For small- and large-scale systems.

The CPU uses these pins to output the middle 8 bits of the 20-bit address data. They are three-state outputs and become high impedance during hold acknowledge.

**AD<sub>7</sub> - AD<sub>0</sub> [Address/Data Bus]**

For small- and large-scale systems.

The CPU uses these pins as the time-multiplexed address and data bus. When high, an AD bit is a one; when low, an AD bit is a zero. This bus contains the lower 8 bits of the 20-bit address during T<sub>1</sub> of the bus cycle and is used as an 8-bit data bus during T<sub>2</sub>, T<sub>3</sub>, and T<sub>4</sub> of the bus cycle.

Sixteen-bit data I/O is performed in two steps. The low byte is sent first, followed by the high byte. The address/data bus is a three-state bus and can be at a high or low level during standby mode. The bus will be high impedance during hold and interrupt acknowledge.

**NMI [Nonmaskable Interrupt]**

For small- and large-scale systems.

This pin is used to input nonmaskable interrupt requests. NMI cannot be masked by software. This input is positive edge triggered and must be held high for five clocks to guarantee recognition. Actual interrupt processing begins, however, after completion of the instruction in progress.

The contents of interrupt vector 2 determine the starting address for the interrupt-servicing routine. Note that a hold request will be accepted even during NMI acknowledge.

This interrupt will cause the μPD70108 to exit the standby mode.

**INT [Maskable Interrupt]**

For small- and large-scale systems.

This pin is an interrupt request that can be masked by software.

INT is active high level and is sensed during the last clock of the instruction. The interrupt will be accepted if the interrupt enable flag IE is set. The CPU outputs the INTAK signal to inform external devices that the interrupt request has been granted. INT must be asserted until the interrupt acknowledge is returned.

If NMI and INT interrupts occur at the same time, NMI has higher priority than INT and INT cannot be

accepted. A hold request will be accepted during INT acknowledge.

This interrupt causes the μPD70108 to exit the standby mode.

**CLK [Clock]**

For small- and large-scale systems.

This pin is used for external clock input.

**RESET [Reset]**

For small- and large-scale systems.

This pin is used for the CPU reset signal. It is an active high level. Input of this signal has priority over all other operations. After the reset signal input returns to a low level, the CPU begins execution of the program starting at address FFFF0H.

In addition to causing normal CPU start, RESET input will cause the μPD70108 to exit the standby mode.

**READY [Ready]**

For small- and large-scale systems.

When the memory or I/O device being accessed cannot complete data read or write within the CPU basic access time, it can generate a CPU wait state (T<sub>w</sub>) by setting this signal to inactive (low level) and requesting a read/write cycle delay.

If the READY signal is active (high level) during either the T<sub>3</sub> or T<sub>w</sub> state, the CPU will not generate a wait state.

**POLL [Poll]**

For small- and large-scale systems.

The CPU checks this input upon execution of the POLL instruction. If the input is low, then execution continues. If the input is high, the CPU will check the POLL input every five clock cycles until the input becomes low again.

The POLL and READY functions are used to synchronize CPU program execution with the operation of external devices.

**RD [Read Strobe]**

For small- and large-scale systems.

The CPU outputs this strobe signal during data read from an I/O device or memory. The IO/M signal is used to select between I/O and memory.

The three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

**S/LG [Small/Large]**

For small- and large-scale systems.

This signal determines the operation mode of the CPU. This signal is fixed at either a high or low level. When

this signal is a high level, the CPU will operate in small-scale system mode, and when low, in the large-scale system mode. A small-scale system will have at most one bus master such as a DMA controller device on the bus. A large-scale system can have more than one bus master accessing the bus as well as the CPU.

### INTAK [Interrupt Acknowledge]

For small-scale systems.

The CPU generates the  $\overline{\text{INTAK}}$  signal low when it accepts an INT signal.

The interrupting device synchronizes with this signal and outputs the interrupt vector to the CPU via the data bus (AD<sub>7</sub> - AD<sub>0</sub>).

### ASTB [Address Strobe]

For small-scale systems.

The CPU outputs this strobe signal to latch address information at an external latch.

ASTB is held at a low level during standby mode and hold acknowledge.

### BUFEN [Buffer Enable]

For small-scale systems.

This is used as the output enable signal for an external bidirectional buffer. The CPU generates this signal during data transfer operations with external memory or I/O devices or during input of an interrupt vector.

This three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

### BUFR/W [Buffer Read/Write]

For small-scale systems.

The output of this signal determines the direction of data transfer with an external bidirectional buffer. A high output causes transmission from the CPU to the external device; a low signal causes data transfer from the external device to the CPU.

BUFR/W is a three-state output and becomes high impedance during hold acknowledge.

### IO/M [IO/Memory]

For small-scale systems.

The CPU generates this signal to specify either I/O access or memory access. A high-level output specifies I/O and a low-level signal specifies memory.

IO/M's output is three state and becomes high impedance during hold acknowledge.

### WR [Write Strobe]

For small-scale systems.

The CPU generates this strobe signal during data write to an I/O device or memory. Selection of either I/O or memory is performed by the IO/M signal.

This three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

### HLDAK [Hold Acknowledge]

For small-scale systems.

The HLDAK signal is used to indicate that the CPU accepts the hold request signal (HLDRQ). When this signal is a high level, the address bus, address/data bus, and the control lines become high impedance.

### HLDRQ [Hold Request]

For small-scale systems.

This input signal is used by external devices to request the CPU to release the address bus, address/data bus, and the control bus.

### LBS<sub>0</sub> [Latched Bus Status 0]

For small-scale systems.

The CPU uses this signal along with the IO/M and BUFR/W signals to inform an external device what the current bus cycle is.

IO/M	BUFR/W	LBS <sub>0</sub>	Bus Cycle
0	0	0	Program fetch
0	0	1	Memory read
0	1	0	Memory write
0	1	1	Passive state
1	0	0	Interrupt acknowledge
1	0	1	I/O read
1	1	0	I/O write
1	1	1	Halt

## $\mu$ PD70108 (V20)

### **A<sub>15</sub>/PS<sub>3</sub> - A<sub>16</sub>/PS<sub>0</sub> [Address Bus/Processor Status]** For small- and large-scale systems.

These pins are time multiplexed to operate as an address bus and as processor status signals.

When used as the address bus, these pins are the high 4 bits of the 20-bit memory address. During I/O access, all 4 bits output data 0.

The processor status signals are provided for both memory and I/O use. PS<sub>3</sub> is always 0 in the native mode and 1 in 8080 emulation mode. The interrupt enable flag (IE) is pin on pin PS<sub>2</sub>. Pins PS<sub>1</sub> and PS<sub>0</sub> indicate which memory segment is being accessed.

A <sub>17</sub> /PS <sub>1</sub>	A <sub>16</sub> /PS <sub>0</sub>	Segment
0	0	Data segment 1
0	1	Stack segment
1	0	Program segment
1	1	Data segment 0

The output of these pins is three state and becomes high impedance during hold acknowledge.

### **QS<sub>1</sub>, QS<sub>0</sub> [Queue Status]** For large-scale systems.

The CPU uses these signals to allow external devices, such as the floating-point arithmetic processor chip, ( $\mu$ PD72091) to monitor the status of the internal CPU instruction queue.

QS <sub>1</sub>	QS <sub>0</sub>	Instruction Queue Status
0	0	NOP.(queue does not change)
0	1	First byte of instruction
1	0	Flush queue
1	1	Subsequent bytes of instruction

The instruction queue status indicated by these signals is the status when the execution unit (EXU) accesses the instruction queue. The data output from these pins is therefore valid only for one clock cycle immediately following queue access. These status signals are provided so that the floating-point processor chip can monitor the CPU's program execution status and synchronize its operation with the CPU when control is passed to it by the FPO (Floating Point Operation) instructions.

### **BS<sub>2</sub> - BS<sub>0</sub> [Bus Status]** For large-scale systems.

The CPU uses these status signals to allow an external bus controller to monitor what the current bus cycle is.

The external bus controller decodes these signals and generates the control signals required to perform access of the memory or I/O device.

BS <sub>2</sub>	BS <sub>1</sub>	BS <sub>0</sub>	Bus Cycle
0	0	0	Interrupt acknowledge
0	0	1	I/O read
0	1	0	I/O write
0	1	1	Halt
1	0	0	Program fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive state

The output of these signals is three state and becomes high impedance during hold acknowledge.

### **BUSLOCK [Bus Lock]** For large-scale systems.

The CPU uses this signal to secure the bus while executing the instruction immediately following the BUSLOCK prefix instruction, or during an interrupt acknowledge cycle. It is a status signal to the other bus masters in a multiprocessor system, inhibiting them from using the system bus during this time.

The output of this signal is three state and becomes high impedance during hold acknowledge. BUSLOCK is high during standby mode except if the HALT instruction has a BUSLOCK prefix.

### **$\overline{RQ}/\overline{AK}_1, \overline{RQ}/\overline{AK}_0$ [Hold Request/Acknowledge]** For large-scale systems.

These pins function as bus hold request inputs ( $\overline{RQ}$ ) and as bus hold acknowledge outputs ( $\overline{AK}$ ).  $\overline{RQ}/\overline{AK}_0$  has a higher priority than  $\overline{RQ}/\overline{AK}_1$ .

These pins have three-state outputs with on-chip pull-up resistors which keep the pin at a high level when the output is high impedance.

### **V<sub>DD</sub> [Power Supply]** For small- and large-scale systems.

This pin is used for the +5 V power supply.

### **GND [Ground]** For small- and large-scale systems.

This pin is used for ground.

### **IC [Internally Connected]** This pin is used for tests performed at the factory by NEC. The $\mu$ PD70108 is used with this pin at ground potential.

### Absolute Maximum Ratings

T<sub>A</sub> = +25°C

Power supply voltage, V <sub>DD</sub>	-0.5 V to +7.0 V
Power dissipation, P <sub>D</sub> MAX	0.5 W
Input voltage, V <sub>I</sub>	-0.5 V to V <sub>DD</sub> + 0.3 V
CLK input voltage, V <sub>K</sub>	-0.5 V to V <sub>DD</sub> + 1.0 V
Output voltage, V <sub>O</sub>	-0.5 V to V <sub>DD</sub> + 0.3 V
Operating temperature, T <sub>OP</sub> T	-40°C to +85°C
Storage temperature, T <sub>STG</sub>	-65°C to +150°C

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### DC Characteristics

μPD70108-5, T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V ± 10%

μPD70108-8, μPD70108-10, T<sub>A</sub> = -10°C to +70°C, V<sub>DD</sub> = +5 V ± 5%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage high	V <sub>IH</sub>	2.2		V <sub>DD</sub> + 0.3	V	
Input voltage low	V <sub>IL</sub>	-0.5		0.8	V	
CLK input voltage high	V <sub>KH</sub>	3.9		V <sub>DD</sub> + 1.0	V	
CLK input voltage low	V <sub>KL</sub>	-0.5		0.6	V	
Output voltage high	V <sub>OH</sub>	0.7 × V <sub>DD</sub>			V	I <sub>OH</sub> = -400 μA
Output voltage low	V <sub>OL</sub>			0.4	V	I <sub>OL</sub> = 2.5 mA
Input leakage current high	I <sub>LIH</sub>			10	μA	V <sub>I</sub> = V <sub>DD</sub>
Input leakage current low	I <sub>LIL</sub>			-10	μA	V <sub>I</sub> = 0 V
Output leakage current high	I <sub>LOH</sub>			10	μA	V <sub>O</sub> = V <sub>DD</sub>
Output leakage current low	I <sub>LOL</sub>			-10	μA	V <sub>O</sub> = 0 V
Supply current	I <sub>DD</sub>	70108-5	30	60	mA	Normal operation
		5 MHz	5	10	mA	Standby mode
		70108-8	45	80	mA	Normal operation
		8 MHz	6	12	mA	Standby mode
		70108-10	60	100	mA	Normal operation
		10 MHz	7	14	mA	Standby mode

### Capacitance

T<sub>A</sub> = +25°C, V<sub>DD</sub> = 0 V

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input capacitance	C <sub>I</sub>		15	pF	f <sub>c</sub> = 1 MHz Unmeasured pins returned to 0 V
I/O capacitance	C <sub>I/O</sub>		15	pF	

**AC Characteristics**

μPD70108-5, T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V ± 10%  
 μPD70108-8, μPD70108-10, T<sub>A</sub> = -10°C to +70°C, V<sub>DD</sub> = +5 V ± 5%

Parameter	Symbol	μPD70108-5		μPD70108-8		μPD70108-10		Unit	Conditions
		Min	Max	Min	Max	Min	Max		
<b>Small/Large Scale</b>									
Clock cycle	t <sub>CYK</sub>	200	500	125	500	100	500	ns	
Clock pulse width high	t <sub>KKH</sub>	69		44		41		ns	V <sub>KH</sub> = 3.0 V
Clock pulse width low	t <sub>KKL</sub>	90		60		49		ns	V <sub>KL</sub> = 1.5 V
Clock rise time	t <sub>KR</sub>		10		8		5	ns	1.5 V to 3.0 V
Clock fall time	t <sub>KF</sub>		10		7		5	ns	3.0 V to 1.5 V
READY inactive setup to CLK↓	t <sub>SRYLK</sub>	-8		-8		-10		ns	
READY inactive hold after CLK↑	t <sub>HKRYH</sub>	30		20		20		ns	
READY active setup to CLK↑	t <sub>SRYHK</sub>	t <sub>KKL</sub> - 8		t <sub>KKL</sub> - 8		t <sub>KKL</sub> - 10		ns	
READY active hold after CLK↑	t <sub>HKRYL</sub>	30		20		20		ns	
Data setup time to CLK ↓	t <sub>SDK</sub>	30		20		10		ns	
Data hold time after CLK ↓	t <sub>HKD</sub>	10		10		10		ns	
NMI, INT, POLL setup time to CLK ↑	t <sub>SIK</sub>	30		15		15		ns	
Input rise time (except CLK)	t <sub>IR</sub>		20		20		20	ns	0.8 V to 2.2 V
Input fall time (except CLK)	t <sub>IF</sub>		12		12		12	ns	2.2 V to 0.8 V
Output rise time	t <sub>OR</sub>		20		20		20	ns	0.8 V to 2.2 V
Output fall time	t <sub>OF</sub>		12		12		12	ns	2.2 V to 0.8 V
Reset pulse width high	t <sub>RESET1</sub>		50		50		50	μs	After power on
	t <sub>RESET2</sub>		4t <sub>CYK</sub>		4t <sub>CYK</sub>		4t <sub>CYK</sub>		During operation
<b>Small Scale</b>									
Address delay time from CLK	t <sub>DKA</sub>	10	90	10	60	10	48	ns	
Address hold time from CLK	t <sub>HKA</sub>	10		10		10		ns	
PS delay time from CLK ↓	t <sub>DKP</sub>	10	90	10	60	10	50	ns	
PS float delay time from CLK ↑	t <sub>FKP</sub>	10	80	10	60	10	50	ns	
Address setup time to ASTB ↓	t <sub>SAST</sub>	t <sub>KKL</sub> - 60		t <sub>KKL</sub> - 30		t <sub>KKL</sub> - 30		ns	
Address float delay time from CLK ↓	t <sub>FKA</sub>	t <sub>HKA</sub>	80	t <sub>HKA</sub>	60	t <sub>HKA</sub>	50	ns	C <sub>L</sub> = 100 pF
ASTB ↑ delay time from CLK ↓	t <sub>DKSTH</sub>		80		50		40	ns	
ASTB ↓ delay time from CLK ↑	t <sub>DKSTL</sub>		85		55		45	ns	
ASTB width high	t <sub>STST</sub>	t <sub>KKL</sub> - 20		t <sub>KKL</sub> - 10		t <sub>KKL</sub> - 10		ns	
Address hold time from ASTB ↓	t <sub>HSTA</sub>	t <sub>KKH</sub> - 10		t <sub>KKH</sub> - 10		t <sub>KKH</sub> - 10		ns	

### AC Characteristics (cont)

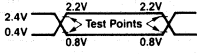
μPD70108-5, T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V ± 10%

μPD70108-8, μPD70108-10, T<sub>A</sub> = -10°C to +70°C, V<sub>DD</sub> = +5 V ± 5%

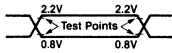
Parameter	Symbol	μPD70108-5		μPD70108-8		μPD70108-10		Unit	Conditions
		Min	Max	Min	Max	Min	Max		
<b>Small Scale (cont)</b>									
Control delay time from CLK	t <sub>DKCT</sub>	10	110	10	65	10	55	ns	
Address float to $\overline{RD}$ ↓	t <sub>AFRL</sub>	0		0		0		ns	
$\overline{RD}$ ↓ delay time from CLK ↓	t <sub>DKRL</sub>	10	165	10	80	10	70	ns	
$\overline{RD}$ ↑ delay time from CLK ↓	t <sub>DKRH</sub>	10	150	10	80	10	60	ns	
Address delay time from $\overline{RD}$ ↑	t <sub>DRHA</sub>	t <sub>CYK</sub> - 45		t <sub>CYK</sub> - 40		t <sub>CYK</sub> - 35		ns	
$\overline{RD}$ width low	t <sub>RR</sub>	2t <sub>CYK</sub> - 75		2t <sub>CYK</sub> - 50		2t <sub>CYK</sub> - 40		ns	C <sub>L</sub> = 100 pF
Data output delay time from CLK ↓	t <sub>DKD</sub>	10	90	10	60	10	50	ns	
Data float delay time from CLK ↓	t <sub>FKD</sub>	10	80	10	60	10	50	ns	
$\overline{WR}$ width low	t <sub>WW</sub>	2t <sub>CYK</sub> - 60		2t <sub>CYK</sub> - 40		2t <sub>CYK</sub> - 35		ns	
HLD $\overline{RQ}$ setup time to CLK ↑	t <sub>SHQK</sub>	35		20		20		ns	
HLD $\overline{AK}$ delay time from CLK ↓	t <sub>DKHA</sub>	10	160	10	100	10	60	ns	
<b>Large Scale</b>									
Address delay time from CLK	t <sub>DKA</sub>	10	90	10	60	10	48	ns	
Address hold time from CLK	t <sub>HKA</sub>	10		10		10		ns	
PS delay time from CLK ↓	t <sub>DKP</sub>	10	90	10	60	10	50	ns	
PS float delay time from CLK ↑	t <sub>FKP</sub>	10	80	10	60	10	50	ns	
Address float delay time from CLK ↓	t <sub>FKA</sub>	t <sub>HKA</sub>	80	t <sub>HKA</sub>	60	t <sub>HKA</sub>	50	ns	
Address delay time from $\overline{RD}$ ↑	t <sub>DRHA</sub>	t <sub>CYK</sub> - 45		t <sub>CYK</sub> - 40		t <sub>CYK</sub> - 35		ns	
ASTB delay time from BS ↓	t <sub>DBST</sub>		15		15		15	ns	
BS ↓ delay time from CLK ↑	t <sub>DKBL</sub>	10	110	10	60	10	50	ns	
BS ↑ delay time from CLK ↓	t <sub>DKBH</sub>	10	130	10	65	10	50	ns	
$\overline{RD}$ ↓ delay time from address float	t <sub>DAFRL</sub>	0		0		0		ns	C <sub>L</sub> = 100 pF
$\overline{RD}$ ↓ delay time from CLK ↓	t <sub>DKRL</sub>	10	165	10	80	10	70	ns	
$\overline{RD}$ ↑ delay time from CLK ↓	t <sub>DKRH</sub>	10	150	10	80	10	60	ns	
$\overline{RD}$ width low	t <sub>RR</sub>	2t <sub>CYK</sub> - 75		2t <sub>CYK</sub> - 50		2t <sub>CYK</sub> - 40		ns	
Data output delay time from CLK ↓	t <sub>DKD</sub>	10	90	10	60	10	50	ns	
Data float delay time from CLK ↑	t <sub>FKD</sub>	10	80	10	60	10	50	ns	
$\overline{AK}$ delay time from CLK ↓	t <sub>DKAK</sub>		70		50		40	ns	
$\overline{RQ}$ setup time to CLK ↑	t <sub>SRQK</sub>	20		10		9		ns	
$\overline{RQ}$ hold time after CLK ↑	t <sub>HLRQ2</sub>	40		30		20		ns	
CLK ↓ to $\overline{RQ}$ hold time	t <sub>HKRQ1</sub>	0		0		0		ns	

Timing Waveforms

AC Test Input Waveform [Except CLK]

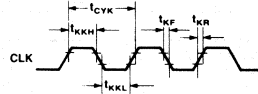


AC Output Test Points



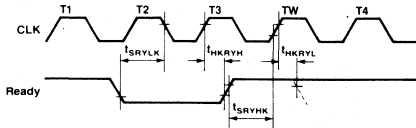
24-34-16-2

Clock Timing

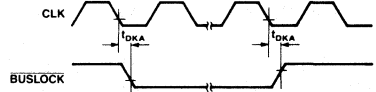


49 00029A

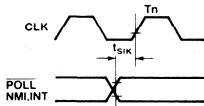
Wait [Ready] Timing



BUSLOCK Output Timing



POLL, NMI, INT Input Timing

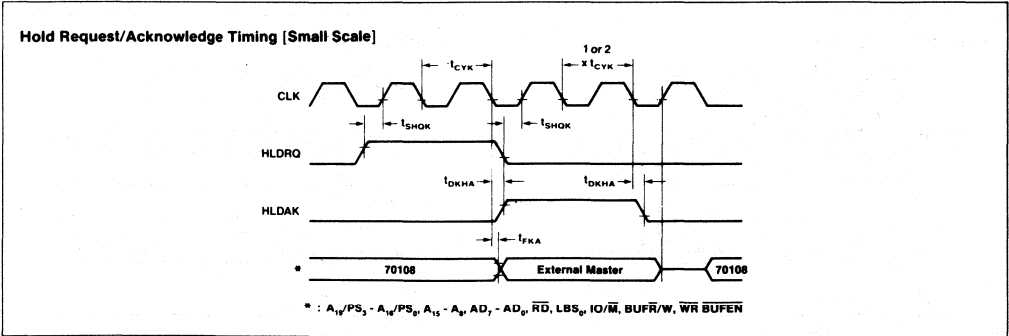
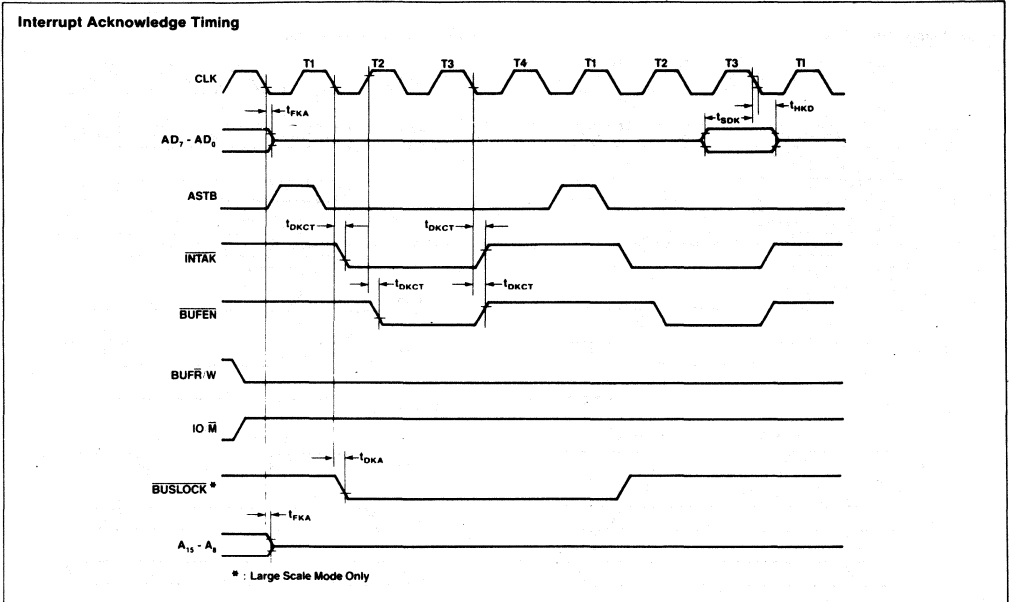


34 (MC)40A

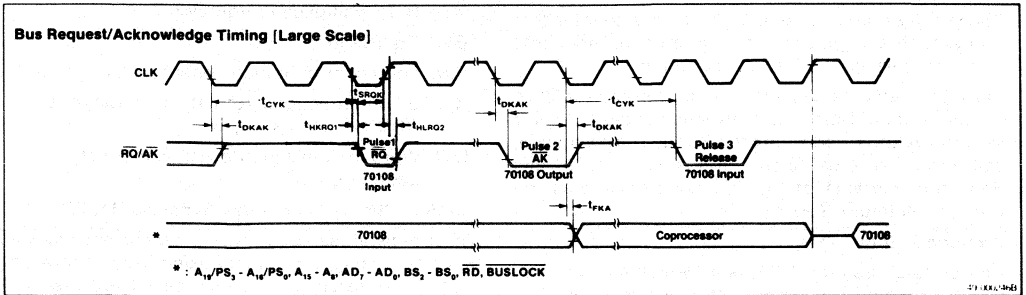




Timing Waveforms (cont)



## Timing Waveforms (cont)



### Register Configuration

#### Program Counter [PC]

The program counter is a 16-bit binary counter that contains the segment offset address of the next instruction which the EXU is to execute.

The PC increments each time the microprogram fetches an instruction from the instruction queue. A new location value is loaded into the PC each time a branch, call, return, or break instruction is executed. At this time, the contents of the PC are the same as the Prefetch Pointer (PPF).

#### Prefetch Pointer [PPF]

The prefetch pointer (PPF) is a 16-bit binary counter which contains a segment offset which is used to calculate a program memory address that the bus control unit (BCU) uses to prefetch the next byte for the instruction queue. The contents of PPF are an offset from the PS (Program Segment) register.

The PPF is incremented each time the BCU prefetches an instruction from the program memory. A new location will be loaded into the PPF whenever a branch, call, return, or break instruction is executed. At that time the contents of the PPF will be the same as those of the PC (Program Counter).

#### Segment Registers [PS, SS, DS<sub>0</sub>, and DS<sub>1</sub>]

The memory addresses accessed by the μPD70108 are divided into 64K-byte logical segments. The starting (base) address of each segment is specified by a 16-bit segment register, and the offset from this starting address is specified by the contents of another register or by the effective address.

These are the four types of segment registers used.

Segment Register	Default Offset
PS (Program Segment)	PPF
SS (Stack Segment)	SP, effective address
DS <sub>0</sub> (Data Segment 0)	IX, effective address
DS <sub>1</sub> (Data Segment 1)	IY

#### General-Purpose Registers [AW, BW, CW, and DW]

There are four 16-bit general-purpose registers. Each one can be used as one 16-bit register or as two 8-bit registers by dividing them into their high and low bytes (AH, AL, BH, BL, CH, CL, DH, DL).

Each register is also used as a default register for processing specific instructions. The default assignments are:

AW: Word multiplication/division, word I/O, data conversion

AL: Byte multiplication/division, byte I/O, BCD rotation, data conversion, translation

AH: Byte multiplication/division

BW: Translation

CW: Loop control branch, repeat prefix

CL: Shift instructions, rotation instructions, BCD operations

DW: Word multiplication/division, indirect addressing I/O

#### Pointers [SP, BP] and Index Registers [IX, IY]

These registers serve as base pointers or index registers when accessing the memory using based addressing, indexed addressing, or based indexed addressing.

These registers can also be used for data transfer and arithmetic and logical operations in the same manner as the general-purpose registers. They cannot be used as 8-bit registers.

Also, each of these registers acts as a default register for specific operations. The default assignments are:

SP: Stack operations

IX: Block transfer (source), BCD string operations

IY: Block transfer (destination), BCD string operations

#### Program Status Word [PSW]

The program status word consists of the following six status and four control flags.

##### Status Flags

- V (Overflow)
- S (Sign)
- Z (Zero)
- AC (Auxiliary Carry)
- P (Parity)
- CY (Carry)

##### Control Flags

- MD (Mode)
- DIR (Direction)
- IE (Interrupt Enable)
- BRK (Break)

When the PSW is pushed on the stack, the word images of the various flags are as shown here.

PSW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	1	1	1	V	D	I	B	S	Z	O	A	O	P	1	C	
D						I	E	R			C				Y	
						R	K									

The status flags are set and reset depending upon the result of each type of instruction executed.

Instructions are provided to set, reset, and complement the CY flag directly.

Other instructions set and reset the control flags and control the operation of the CPU.

### High-Speed Execution of Instructions

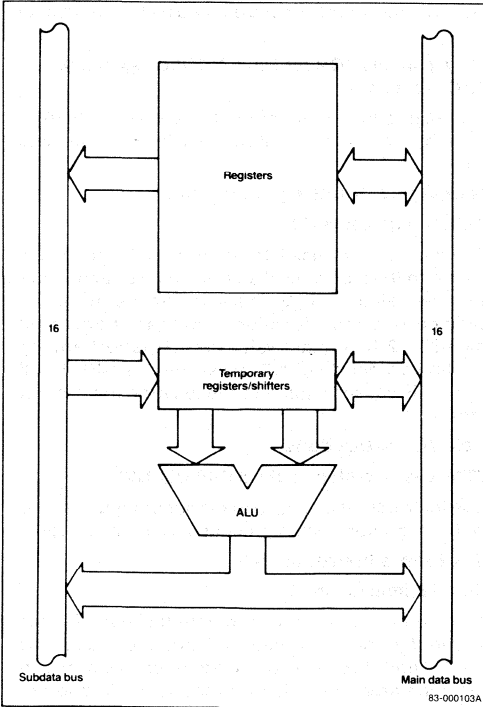
This section highlights the major architectural features that enhance the performance of the μPD70108.

- Dual data bus in EXU
- Effective address generator
- 16/32-bit temporary registers/shifters (TA, TB)
- 16-bit loop counter
- PC and PFP

#### Dual Data Bus Method

To reduce the number of processing steps for instruction execution, the dual data bus method has been adopted for the μPD70108 (figure 1). The two data buses (the main data bus and the subdata bus) are both 16 bits wide. For addition/subtraction and logical and comparison operations, processing time has been speeded up some 30% over single-bus systems.

Figure 1. Dual Data Buses



#### Example

ADD AW, BW ; AW ← AW + BW

##### Single Bus

Step 1 TA ← AW

Step 2 TB ← BW

Step 3 AW ← TA + TB

##### Dual Bus

TA ← AW, TB ← BW

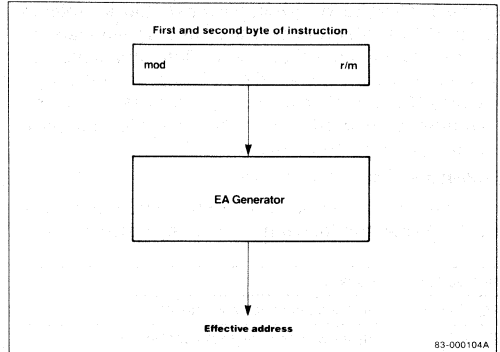
AW ← TA + TB

#### Effective Address Generator

This circuit (figure 2) performs high-speed processing to calculate effective addresses for accessing memory.

Calculating an effective address by the microprogramming method normally requires 5 to 12 clock cycles. This circuit requires only two clock cycles for addresses to be generated for any addressing mode. Thus, processing is several times faster.

Figure 2. Effective Address Generator



#### 16/32-Bit Temporary Registers/Shifters (TA, TB)

These 16-bit temporary registers/shifters (TA, TB) are provided for multiplication/division and shift/rotation instructions.

These circuits have decreased the execution time of multiplication/division instructions. In fact, these instructions can be executed about four times faster than with the microprogramming method.

TA + TB: 32-bit temporary register/shifter for multiplication and division instructions.

TB: 16-bit temporary register/shifter for shift/rotation instructions.

## μPD70108 (V20)

### Loop Counter [LC]

This counter is used to count the number of loops for a primitive block transfer instruction controlled by a repeat prefix instruction and the number of shifts that will be performed for a multiple bit shift/rotation instruction.

The processing performed for a multiple bit rotation of a register is shown below. The average speed is approximately doubled over the microprogram method.

#### Example

RORC AW, CL ; CL = 5

**Microprogram method**    **LC method**

8 + (4 x 5) = 28 clocks    7 + 5 = 12 clocks

### Program Counter and Prefetch Pointer [PC and PFP]

The μPD70108 microprocessor has a program counter, (PC) which addresses the program memory location of the instruction to be executed next, and a prefetch pointer (PFP), which addresses the program memory location to be accessed next. Both functions are provided in hardware. A time saving of several clocks is realized for branch, call, return, and break instruction execution, compared with microprocessors that have only one instruction pointer.

### Enhanced Instructions

In addition to the μPD8088/86 instructions, the μPD70108 has the following enhanced instructions.

Instruction	Function
PUSH imm	Pushes immediate data onto stack
PUSH R	Pushes 8 general registers onto stack
POP R	Pops 8 general registers from stack
MUL imm	Executes 16-bit multiply of register or memory contents by immediate data
SHL imm8 SHR imm8 SHRA imm8 ROL imm8 ROR imm8 ROLC imm8 RORC imm8	Shifts/rotates register or memory by immediate value
CHKIND	Checks array index against designated boundaries
INM	Moves a string from an I/O port to memory
OUTM	Moves a string from memory to an I/O port
PREPARE	Allocates an area for a stack frame and copies previous frame pointers
DISPOSE	Frees the current stack frame on a procedure exit

### Enhanced Stack Operation Instructions

#### PUSH imm

This instruction allows immediate data to be pushed onto the stack.

#### PUSH R/POP R

These instructions allow the contents of the eight general registers to be pushed onto or popped from the stack with a single instruction.

### Enhanced Multiplication Instructions

#### MUL reg16, imm16/MUL mem16, imm16

These instructions allow the contents of a register or memory location to be 16-bit multiplied by immediate data.

### Enhanced Shift and Rotate Instructions

#### SHL reg, imm8/SHR reg, imm8/SHRA reg, imm8

These instructions allow the contents of a register to be shifted by the number of bits defined by the immediate data.

#### ROL reg, imm8/ROR reg, imm8/ROLC reg, imm8/RORC reg, imm8

These instructions allow the contents of a register to be rotated by the number of bits defined by the immediate data.

### Check Array Boundary Instruction

#### CHKIND reg16, mem32

This instruction is used to verify that index values pointing to the elements of an array data structure are within the defined range. The lower limit of the array should be in memory location mem32, the upper limit in mem32 + 2. If the index value in reg16 is not between these limits when CHKIND is executed, a BRK 5 will occur. This causes a jump to the location in interrupt vector 5.

### Block I/O Instructions

#### OUTM DW, src-block/INM dst-block, DW

These instructions are used to output or input a string to or from memory, when preceded by a repeat prefix.

### Stack Frame Instructions

#### PREPARE imm16, imm8

This instruction is used to generate the stack frames required by block-structured languages, such as PASCAL and Ada. The stack frame consists of two areas. One area has a pointer that points to another frame which has variables that the current frame can access. The other is a local variable area for the current procedure.

### DISPOSE

This instruction releases the last stack frame generated by the PREPARE instruction. It returns the stack and base pointers to the values they had before the PREPARE instruction was used to call a procedure.

### Unique Instructions

In addition to the μPD8088/86 instructions and the enhanced instructions, the μPD70108 has the following unique instructions.

Instruction	Function
INS	Insert bit field
EXT	Extract bit field
ADD4S	Adds packed decimal strings
SUB4S	Subtracts one packed decimal string from another
CMP4S	Compares two packed decimal strings
ROL4	Rotates one BCD digit left through AL lower 4 bits
ROR4	Rotates one BCD digit right through AL lower 4 bits
TEST1	Tests a specified bit and sets/resets Z flag
NOT1	Inverts a specified bit
CLR1	Clears a specified bit
SET1	Sets a specified bit
REPC	Repeats next instruction until CY flag is cleared
REPNC	Repeats next instruction until CY flag is set
FP02	Additional floating point processor call

### Variable Length Bit Field Operation Instructions

This category has two instructions: INS (Insert Bit Field) and EXT (Extract Bit Field). These instructions are highly effective for computer graphics and high-level languages. They can, for example, be used for data structures such as packed arrays and record type data used in PASCAL.

#### INS reg8, reg8/INS reg8, imm4

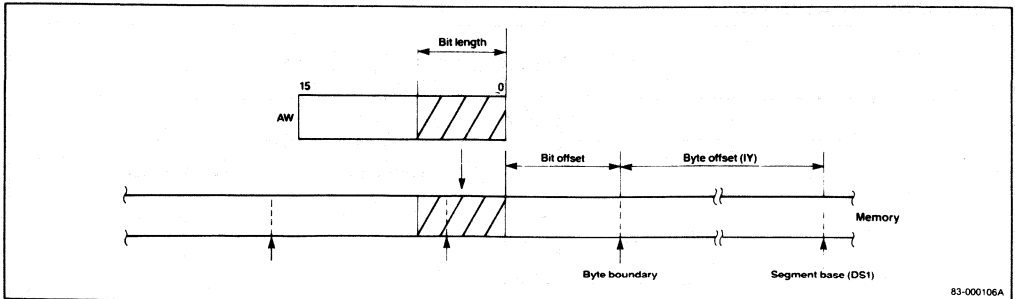
This instruction (figure 3) transfers low bits from the 16-bit AW register (the number of bits is specified by the second operand) to the memory location specified by the segment base (DS<sub>1</sub> register) plus the byte offset (IY register). The starting bit position within this byte is specified as an offset by the lower 4-bits of the first operand.

After each complete data transfer, the IY register and the register specified by the first operand are automatically updated to point to the next bit field.

Either immediate data or a register may specify the number of bits transferred (second operand). Because the maximum transferable bit length is 16-bits, only the lower 4-bits of the specified register (00H to 0FH) will be valid.

Bit field data may overlap the byte boundary of memory.

Figure 3. Bit Field Insertion



**EXT reg8, reg8/EXT reg8, imm4**

This instruction (figure 4) loads to the AW register the bit field data whose bit length is specified by the second operand of the instruction from the memory location that is specified by the DS0 segment register (segment base), the IX index register (byte offset), and the lower 4-bits of the first operand (bit offset).

After the transfer is complete, the IX register and the lower 4-bits of the first operand are automatically updated to point to the next bit field.

Either immediate data or a register may be specified for the second operand. Because the maximum transferable bit length is 16 bits, however, only the lower 4-bits of the specified register (0H to 0FH) will be valid.

Bit field data may overlap the byte boundary of memory.

**Packed BCD Operation Instructions**

The instructions described here process packed BCD data either as strings (ADD4S, SUB4S, CMP4S) or byte-format operands (ROR4, ROL4). Packed BCD strings may be from 1 to 254 digits in length.

When the number of digits is even, the zero and carry flags will be set according to the result of the operation. When the number of digits is odd, the zero and carry flags may not be set correctly in this case, (CL = odd), the zero flag will not be set unless the upper 4 bits of the highest byte are all zero. The carry flag will not be set unless there is a carry out of the upper 4 bits of the highest byte. When CL is odd, the contents of the upper 4 bits of the highest byte of the result are undefined.

**ADD4S**

This instruction adds the packed BCD string addressed by the IX index register to the packed BCD string addressed by the IY register, and stores the result in the string addressed by the IY register. The length of the string (number of BCD digits) is specified by the CL register, and the result of the operation will affect the overflow flag (V), the carry flag (CY), and zero flag (Z).

$$\text{BCD string (IY, CL)} \leftarrow \text{BCD string (IY, CL)} + \text{BCD string (IX, CL)}$$

**SUB4S**

This instruction subtracts the packed BCD string addressed by the IX index register from the packed BCD string addressed by the IY register, and stores the result in the string addressed by the IY register. The length of the string (number of BCD digits) is specified by the CL register, and the result of the operation will affect the overflow flag (V), the carry flag (CY), and zero flag (Z).

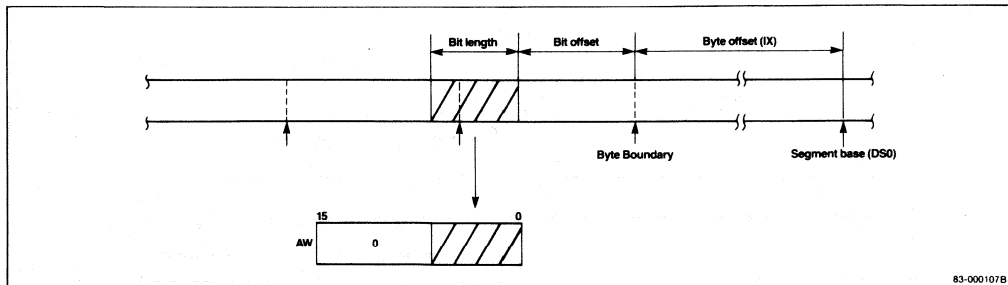
$$\text{BCD string (IY, CL)} \leftarrow \text{BCD string (IY, CL)} - \text{BCD String (IX, CL)}$$

**CMP4S**

This instruction performs the same operation as SUB4S except that the result is not stored and only the overflow (V), carry flags (CY) and zero flag (Z) are affected.

$$\text{BCD string (IY, CL)} - \text{BCD string (IX, CL)}$$

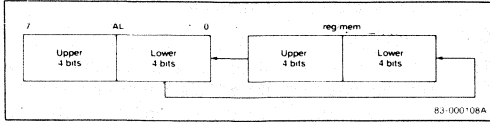
**Figure 4. Bit Field Extraction**



### ROL4

This instruction (figure 5) treats the byte data of the register or memory directly specified by the instruction byte as BCD data and uses the lower 4-bits of the AL register (AL<sub>L</sub>) to rotate that data one BCD digit to the left.

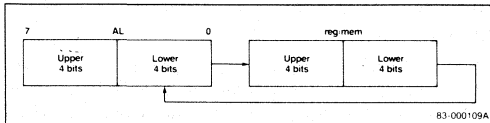
Figure 5. BCD Rotate Left (ROL4)



### ROR4

This instruction (figure 6) treats the byte data of the register or memory directly specified by the instruction byte as BCD data and uses the lower 4-bits of the AL register (AL<sub>L</sub>) to rotate that data one BCD digit to the right.

Figure 6. BCD Rotate Right (ROR4)



### Bit Manipulation Instructions

#### TEST1

This instruction tests a specific bit in a register or memory location. If the bit is 1, the Z flag is reset to 0. If the bit is 0, the Z flag is set to 1.

#### NOT1

This instruction inverts a specific bit in a register or memory location.

#### CLR1

This instruction clears a specific bit in a register or memory location.

#### SET1

This instruction sets a specific bit in a register or memory location.

### Repeat Prefix Instructions

#### REPC

This instruction causes the μPD70108 to repeat the following primitive block transfer instruction until the CY flag becomes cleared or the CW register becomes zero.

#### REPC

This instruction causes the μPD70108 to repeat the following primitive block transfer instruction until the CY flag becomes set or the CW register is decremented to zero.

### Floating Point Instruction

#### FPO2

This instruction is in addition to the μPD8088/86 floating point instruction, FPO1. These instructions are covered in a later section.

### Mode Operation Instructions

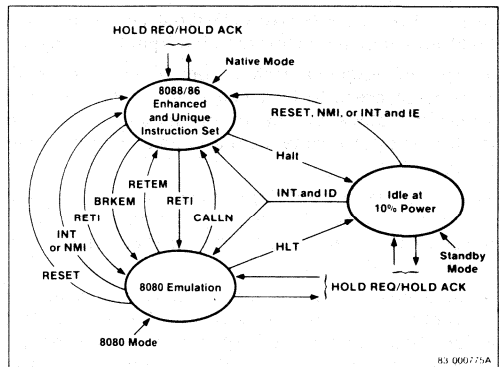
The μPD70108 has two operating modes (figure 7). One is the native mode which executes μPD8088/86, enhanced and unique instructions. The other is the 8080 emulation mode in which the instruction set of the μPD8080AF is emulated. A mode flag (MD) is provided to select between these two modes. Native mode is selected when MD is 1 and emulation mode when MD is 0. MD is set and reset, directly and indirectly, by executing the mode manipulation instructions.

Two instructions are provided to switch operation from the native mode to the emulation mode and back: BRKEM (Break for Emulation), and RETEM (Return from Emulation).

Two instructions are used to switch from the emulation mode to the native mode and back: CALLN (Call Native Routine), and RETI (Return from Interrupt).

The system will return from the 8080 emulation mode to the native mode when the RESET signal is present, or when an external interrupt (NMI or INT) is present.

Figure 7. V20 Modes





**BRKEM imm8**

This is the basic instruction used to start the 8080 emulation mode. This instruction operates exactly the same as the BRK instruction, except that BRKEM resets the mode flag (MD) to 0. PSW, PS, and PC are saved to the stack. MD is then reset and the interrupt vector specified by the operand imm8 of this command is loaded into PS and PC.

The instruction codes of the interrupt processing routine jumped to are then fetched. Then the CPU executes these codes as μPD8080AF instructions.

In 8080 emulation mode, registers and flags of the μPD8080AF are performed by the following registers and flags of the μPD70108.

	μPD8080AF	μPD70108
<b>Registers:</b>	A	AL
	B	CH
	C	CL
	D	DH
	E	DL
	H	BH
	L	BL
	SP	BP
	PC	PC
	<b>Flags:</b>	C
Z		Z
S		S
P		P
AC		AC

In the native mode, SP is used for the stack pointer. In the 8080 emulation mode this function is performed by BP.

This use of independent stack pointers allows independent stack areas to be secured for each mode and keeps the stack of one of the modes from being destroyed by an erroneous stack operation in the other mode.

The SP, IX, IY and AH registers and the four segment registers (PS, SS, DS<sub>0</sub>, and DS<sub>1</sub>) used in the native mode are not affected by operations in 8080 emulation mode.

In the 8080 emulation mode, the segment register for instructions is determined by the PS register (set automatically by the interrupt vector) and the segment register for data is the DS<sub>0</sub> register (set by the programmer immediately before the 8080 emulation mode is entered).

It is prohibited to nest BRKEM instructions.

**RETEM [no operand]**

When RETEM is executed in 8080 emulation mode (interpreted by the CPU as a μPD8080AF instruction), the CPU restores PS, PC, and PSW (as it would when returning from an interrupt processing routine), and returns to the native mode. At the same time, the contents of the mode flag (MD) which was saved to the stack by the BRKEM instruction, is restored to MD = 1. The CPU is set to the native mode.

**CALLN imm8**

This instruction makes it possible to call the native mode subroutines from the 8080 emulation mode. To return from subroutine to the emulation mode, the RETI instruction is used.

The processing performed when this instruction is executed in the 8080 emulation mode (it is interpreted by the CPU as μPD8080AF instruction), is similar to that performed when a BRK instruction is executed in the native mode. The imm8 operand specifies an interrupt vector type. The contents of PS, PC, and PSW are pushed on the stack and an MD flag value of 0 is saved. The mode flag is set to 1 and the interrupt vector specified by the operand is loaded into PS and PC.

**RETI [no operand]**

This is a general-purpose instruction used to return from interrupt routines entered by the BRK instruction or by an external interrupt in the native mode. When this instruction is executed at the end of a subroutine entered by the execution of the CALLN instruction, the operation that restores PS, PC, and PSW is exactly the same as the native mode execution. When PSW is restored, however, the 8080 emulation mode value of the mode flag (MD) is restored, the CPU is set in emulation mode, and all subsequent instructions are interpreted and executed as μPD8080AF instructions.

RETI is also used to return from an interrupt procedure initiated by an NMI or INT interrupt in the emulation mode.

**Floating Point Operation Chip Instructions****FPO1 fp-op, mem/FPO2 fp-op, mem**

These instructions are used for the external floating point processor. The floating point operation is passed to the floating point processor when the CPU fetches one of these instructions. From this point the CPU performs only the necessary auxiliary processing (effective address calculation, generation of physical addresses, and start-up of the memory read cycle).

The floating point processor always monitors the instructions fetched by the CPU. When it interprets one as an instruction to itself, it performs the appropriate processing. At this time, the floating point processor chip uses either the address alone or both the address and read data of the memory read cycle executed by the CPU. This difference in the data used depends on which of these instructions is executed.

**Note:** During the memory read cycle initiated by the CPU for FPO1 or FPO2 execution, the CPU does not accept any read data on the data bus from memory. Although the CPU generates the memory address, the data is used by the floating point processor.

### Interrupt Operation

The interrupts used in the μPD70108 can be divided into two types: interrupts generated by external interrupt requests and interrupts generated by software processing. These are the classifications.

#### External Interrupts

- (a) NMI input (nonmaskable)
- (b) INT input (maskable)

#### Software Processing

As the result of instruction execution

- When a divide error occurs during execution of the DIV or DIVU instruction
- When a memory-boundary-over error is detected by the CHKIND instruction

Conditional break instruction

- When V = 1 during execution of the BRKV instruction

Unconditional break instructions

- 1-byte break instruction: BRK3
- 2-byte break instruction: BRK imm8

Flag processing

- When stack operations are used to set the BRK flag

8080 Emulation mode instructions

- BRKEM imm8
- CALLN imm8

### Interrupt Vectors

Starting addresses for interrupt processing routines are either determined automatically by a single location of the interrupt vector table or selected each time interrupt processing is entered.

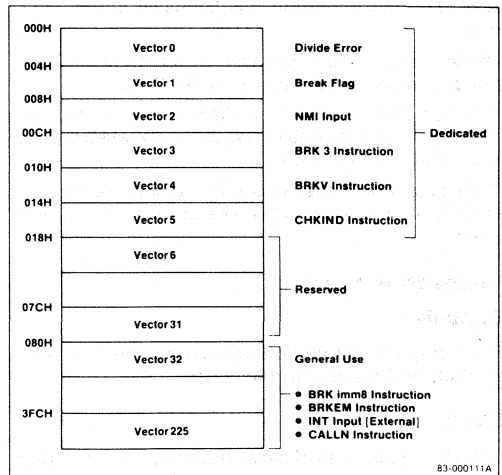
The interrupt vector table is shown in figure 8. The table uses 1K bytes of memory addresses 000H to 3FFH and can store starting address data for a maximum of 256 vectors (4 bytes per vector).

The corresponding interrupt sources for vectors 0 to 5 are predetermined and vectors 6 to 31 are reserved. These vectors consequently cannot be used for general applications.

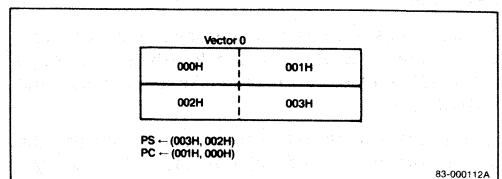
The BRKEM instruction and CALLN instruction (in the emulation mode) and the INT input are available for general applications for vectors 32 to 255.

A single interrupt vector is made up of 4 bytes (figure 9). The 2 bytes in the low addresses of memory are loaded into PC as the offset, and the high 2 bytes are loaded into PS as the base address. The bytes are combined in reverse order. The lower-order bytes in the vector become the most significant bytes in the PC and PS, and the higher-order bytes become the least significant bytes.

**Figure 8. Interrupt Vector Table**



**Figure 9. Interrupt Vector 0**



Based on this format, the contents of each vector should be initialized at the beginning of the program.

The basic steps to jump to an interrupt processing routine are now shown.

- (SP - 1, SP - 2) ← PSW
- (SP - 3, SP - 4) ← PS
- (SP - 5, SP - 6) ← PC
- SP ← SP - 6
- IE ← 0, BRK ← 0, MD ← 1
- PS ← vector high bytes
- PC ← vector low bytes

**Standby Function**

The μPD70108 has a standby mode to reduce power consumption during program wait states. This mode is set by the HALT instruction in both the native and the emulation mode.

In the standby mode, the internal clock is supplied only to those circuits related to functions required to release this mode and bus hold control functions. As a result, power consumption can be reduced to 1/10 the level of normal operation in either native or emulation mode.

The standby mode is released by inputting a RESET signal or an external interrupt (NMI, INT).

The bus hold function is effective during standby mode. The CPU returns to standby mode when the bus hold request is removed.

During standby mode, all control outputs are disabled and the address/data bus will be at either high or low levels.

**Instruction Set**

**Symbols**

Preceding the instruction set, several tables explain symbols, abbreviations, and codes.

**Clocks**

In the Clocks column of the instruction set, the numbers cover these operations: instruction decoding, effective address calculation, operand fetch, and instruction execution.

Clock timings assume the instruction has been pre-fetched and is present in the four-byte instruction queue. Otherwise, add four clocks for each byte not present.

For instructions that reference memory operands, the number on the left side of the slash (/) is for byte operands and the number on the right side is for word operands.

For conditional control transfer or branch instructions, the number on the left side of the slash is applicable if the transfer or branch takes place. The number on the right side is applicable if it does not take place.

If a range of numbers is given, the execution time depends on the operands involved.

**Symbols**

Symbol	Meaning
acc	Accumulator (AW or AL)
disp	Displacement (8 or 16 bits)
dmem	Direct memory address
dst	Destination operand or address
ext-disp8	16-bit displacement (sign-extension byte + 8-bit displacement)
far_label	Label within a different program segment
far_proc	Procedure within a different program segment
fp_op	Floating point instruction operation
imm	8- or 16-bit immediate operand
imm3/4	3/4-bit immediate bit offset
imm8	8-bit immediate operand
imm16	16-bit immediate operand
mem	Memory field (000 to 111); 8- or 16-bit memory location
mem8	8-bit memory location
mem16	16-bit memory location
mem32	32-bit memory location
memptr16	Word containing the destination address within the current segment
memptr32	Double word containing a destination address in another segment
mod	Mode field (00 to 10)
near_label	Label within the current segment
near_proc	Procedure within the current segment
offset	Immediate offset data (16 bits)
pop_value	Number of bytes to discard from the stack
reg	Register field (000 to 111); 8- or 16-bit general-purpose register
reg8	8-bit general-purpose register
reg16	16-bit general-purpose register
regptr	16-bit register containing a destination address within the current segment
regptr16	Register containing a destination address within the current segment
seg	Immediate segment data (16 bits)
short_label	Label between -128 and +127 bytes from the end of the current instruction

### Symbols (cont)

Symbol	Meaning
sr	Segment register
src	Source operand or address
temp	Temporary register (8/16/32 bits)
tmpcy	Temporary carry flag (1 bit)
AC	Auxiliary carry flag
AH	Accumulator (high byte)
AL	Accumulator (low byte)
AND	Logical product
AW	Accumulator (16 bits)
BH	BW register (high byte)
BL	BW register (low byte)
BP	BP register
BRK	Break flag
BW	BW register (16 bits)
CH	CW register (high byte)
CL	CW register (low byte)
CW	CW register (16 bits)
CY	Carry flag
DH	DW register (high byte)
DIR	Direction flag
DL	DW register (low byte)
DS0	Data segment 0 register (16 bits)
DS1	Data segment 1 register (16 bits)
DW	DW register (16 bits)
IE	Interrupt enable flag
IX	Index register (source) (16 bits)

### Symbols

Symbol	Meaning
IY	Index register (destination) (16 bits)
MD	Mode flag
OR	Logical sum
P	Parity flag
PC	Program counter (16 bits)
PS	Program segment register (16 bits)
PSW	Program status word (16 bits)
R	Register set
S	Sign extend operand field S = 0 No sign extension S = 1 Sign extend immediate byte operand
S	Sign flag
SP	Stack pointer (16 bits)
SS	Stack segment register (16 bits)
V	Overflow flag
W	Word/byte field (0 to 1)
X, XXX, YYY, ZZZ	Data to identify the instruction code of the external floating point arithmetic chip
XOR	Exclusive logical sum
XXH	Two-digit hexadecimal value
XXXXH	Four-digit hexadecimal value
Z	Zero flag
( )	Values in parentheses are memory contents
←	Transfer direction
+	Addition
-	Subtraction
x	Multiplication
÷	Division
%	Modulo

**Flag Operations**

Symbol	Meaning
(blank)	No change
0	Cleared to 0
1	Set to 1
x	Set or cleared according to result
u	Undefined
R	Restored to previous state

**Memory Addressing Modes**

mem	mod = 00	mod = 01	mod = 10
000	BW + IX	BW + IX + disp8	BW + IX + disp16
001	BW + IY	BW + IY + disp8	BW + IY + disp16
010	BP + IX	BP + IX + disp8	BP + IX + disp16
011	BP + IY	BP + IY + disp8	BP + IY + disp16
100	IX	IX + disp8	IX + disp16
101	IY	IY + disp8	IY + disp16
110	Direct	BP + disp8	BP + disp16
111	BW	BW + disp8	BW + disp16

**Register Selection (mod = 11)**

reg	W = 0	W = 1
000	AL	AW
001	CL	CW
010	DL	DW
011	BL	BW
100	AH	SP
101	CH	BP
110	DH	IX
111	BH	IY

**Segment Register Selection**

sr	Segment Register
00	DS1
01	PS
10	SS
11	DS0

### Instruction Set

Mnemonic	Operand	Opcode														Clocks	Bytes	Flags				
		7	6	5	4	3	2	1	0	7	6	5	4	3	2			1	0	AC	CY	V
<b>Data Transfer Instructions</b>																						
MOV	reg, reg	1	0	0	0	1	0	1	W	1	1	reg	reg	2	2							
	mem, reg	1	0	0	0	1	0	0	W	mod	reg	mem	9/13	2-4								
	reg, mem	1	0	0	0	1	0	1	W	mod	reg	mem	11/15	2-4								
	mem, imm	1	1	0	0	0	1	1	W	mod	reg	mem	11/15	3-6								
	reg, imm	1	0	1	1	W	reg	4	2-3													
	acc, dmem	1	0	1	0	0	0	0	W	10/14	3											
	dmem, acc	1	0	1	0	0	0	1	W	9/13	3											
	sr, reg16	1	0	0	0	1	1	1	0	1	1	0	sr	reg	2	2						
	sr, mem16	1	0	0	0	1	1	1	0	mod	0	sr	mem	11/15	2-4							
	reg16, sr	1	0	0	0	1	1	0	0	1	1	0	sr	reg	2	2						
	mem16, sr	1	0	0	0	1	1	0	0	mod	0	sr	mem	10/14	2-4							
	DS0, reg16, mem32	1	1	0	0	0	1	0	1	mod	reg	mem	18/26	2-4								
	DS1, reg16, mem32	1	1	0	0	0	1	0	0	mod	reg	mem	18/26	2-4								
	AH, PSW	1	0	0	1	1	1	1	1	2	1											
	PSW, AH	1	0	0	1	1	1	1	0	3	1						x	x	x	x	x	
LDEA	reg16, mem16	1	0	0	0	1	1	0	1	mod	reg	mem	4	2-4								
TRANS	src_table	1	1	0	1	0	1	1	1	9	1											
XCH	reg, reg	1	0	0	0	0	1	1	W	1	1	reg	reg	3	2							
	mem, reg	1	0	0	0	0	1	1	W	mod	reg	mem	16/26	2-4								
	AW, reg16	1	0	0	1	0	reg	3	1													

### Repeat Prefixes

REPC	0	1	1	0	0	1	0	1	2	1										
REPNC	0	1	1	0	0	1	0	0	2	1										
REP	1	1	1	1	0	0	1	1	2	1										
REPE																				
REPZ																				
REPNE	1	1	1	1	0	0	1	0	2	1										
REPZ																				

### Block Transfer Instructions

MOV BK	dst, src	1	0	1	0	0	1	0	W	11 + 8n	1											
CMP BK	dst, src	1	0	1	0	0	1	1	W	7 + 14n	1							x	x	x	x	x
CMP M	dst	1	0	1	0	1	1	1	W	7 + 10n	1							x	x	x	x	x
LDM	src	1	0	1	0	1	1	0	W	7 + 9n	1											
STM	dst	1	0	1	0	1	0	1	W	7 + 4n	1											

n = number of transfers

### I/O Instructions

IN	acc, imm8	1	1	1	0	0	1	0	W	9/13	2											
	acc, DW	1	1	1	0	1	1	0	W	8/12	1											
OUT	imm8, acc	1	1	1	0	0	1	1	W	8/12	2											
	DW, acc	1	1	1	0	1	1	1	W	8/12	1											
INM	dst, DW	0	1	1	0	1	1	0	W	9 + 8n	1											
OUTM	DW, src	0	1	1	0	1	1	1	W	9 + 8n	1											

n = number of transfers

**Instruction Set (cont)**

Mnemonic	Operand	Opcode										Clocks	Bytes	Flags												
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P	S	Z	
<b>BCD Instructions</b>																										
ADJBA		0	0	1	1	0	1	1	1	1									3	1	x	x	u	u	u	u
ADJ4A		0	0	1	0	0	1	1	1	1									3	1	x	x	u	x	x	x
ADJBS		0	0	1	1	1	1	1	1	1									7	1	x	x	u	u	u	u
ADJ4S		0	0	1	0	1	1	1	1	1									7	1	x	x	u	x	x	x
ADD4S	dst, src	0	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	0	7 + 19n	2	u	x	u	u	u	x
SUB4S	dst, src	0	0	0	0	1	1	1	1	1	0	0	1	0	0	0	1	0	7 + 19n	2	u	x	u	u	u	x
CMP4S	dst, src	0	0	0	0	1	1	1	1	1	0	0	1	0	0	1	1	0	7 + 19n	2	u	x	u	u	u	x
ROL4	reg8	0	0	0	0	1	1	1	1	1	0	0	1	0	1	0	0	0	25	3						
		1	1	0	0	0				reg																
	mem8	0	0	0	0	1	1	1	1	1	0	0	1	0	1	0	0	0	28	3-5						
		mod	0	0	0																					
ROR4	reg8	0	0	0	0	1	1	1	1	1	0	0	1	0	1	0	1	0	29	3						
		1	1	0	0	0																				
	mem8	0	0	0	0	1	1	1	1	1	0	0	1	0	1	0	1	0	33	3-5						
		mod	0	0	0																					

n = number of BCD digits divided by 2

**Data Type Conversion Instructions**

CVTBD		1	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	15	2	u	u	u	x	x	x
CVTDB		1	1	0	1	0	1	0	1	0	0	0	0	0	1	0	1	0	7	2	u	u	u	x	x	x
CVTBW		1	0	0	1	1	0	0	0	0									2	1						
CVTWL		1	0	0	1	1	0	0	1	0									4-5	1						

**Arithmetic Instructions**

ADD	reg, reg	0	0	0	0	0	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x		
	mem, reg	0	0	0	0	0	0	0	W	mod	reg	mem	16/24	2-4	x	x	x	x	x	x			
	reg, mem	0	0	0	0	0	0	1	W	mod	reg	mem	11/15	2-4	x	x	x	x	x	x			
	reg, imm	1	0	0	0	0	0	S	W	1	1	0	0	0	reg	4	3-4	x	x	x	x	x	x
	mem, imm	1	0	0	0	0	0	S	W	mod	0	0	0	mem	18/26	3-6	x	x	x	x	x	x	
	acc, imm	0	0	0	0	0	1	0	W						4	2-3	x	x	x	x	x	x	
ADDC	reg, reg	0	0	0	1	0	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x		
	mem, reg	0	0	0	1	0	0	0	W	mod	reg	mem	16/24	2-4	x	x	x	x	x	x			
	reg, mem	0	0	0	1	0	0	1	W	mod	reg	mem	11/15	2-4	x	x	x	x	x	x			
	reg, imm	1	0	0	0	0	0	S	W	1	1	0	1	0	reg	4	3-4	x	x	x	x	x	x
	mem, imm	1	0	0	0	0	0	S	W	mod	0	1	0	mem	18/26	3-6	x	x	x	x	x	x	
	acc, imm	0	0	0	1	0	1	0	W						4	2-3	x	x	x	x	x	x	
SUB	reg, reg	0	0	1	0	1	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x		
	mem, reg	0	0	1	0	1	0	0	W	mod	reg	mem	16/24	2-4	x	x	x	x	x	x			
	reg, mem	0	0	1	0	1	0	1	W	mod	reg	mem	11/15	2-4	x	x	x	x	x	x			
	reg, imm	1	0	0	0	0	0	S	W	1	1	1	0	1	reg	4	3-4	x	x	x	x	x	x

### Instruction Set (cont)

Mnemonic	Operand	Opcode											Clocks	Bytes	Flags								
		7	6	5	4	3	2	1	0	7	6	5			4	3	2	1	0	AC	CY	V	P
<b>Arithmetic Instructions (cont)</b>																							
	mem, imm	1	0	0	0	0	0	S	W	mod	1	0	1	mem	18/26	3-6	x	x	x	x	x	x	
	acc, imm	0	0	1	0	1	1	0	W						4	2-3	x	x	x	x	x	x	
SUBC	reg, reg	0	0	0	1	1	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x		
	mem, reg	0	0	0	1	1	0	0	W	mod	reg	mem	16/24	2-4	x	x	x	x	x	x			
	reg, mem	0	0	0	1	1	0	1	W	mod	reg	mem	11/15	2-4	x	x	x	x	x	x			
	reg, imm	1	0	0	0	0	0	S	W	1	1	0	1	1	reg	4	3-4	x	x	x	x	x	x
	mem, imm	1	0	0	0	0	0	S	W	mod	0	1	1	mem	18/26	3-6	x	x	x	x	x	x	
	acc, imm	0	0	0	1	1	1	0	W					4	2-3	x	x	x	x	x	x		
INC	reg8	1	1	1	1	1	1	1	0	1	1	0	0	0	reg	2	2	x	x	x	x	x	x
	mem	1	1	1	1	1	1	1	W	mod	0	0	0	mem	16/24	2-4	x	x	x	x	x	x	
	reg16	0	1	0	0	0	reg						2	1	x	x	x	x	x	x			
DEC	reg8	1	1	1	1	1	1	1	0	1	1	0	0	1	reg	2	2	x	x	x	x	x	x
	mem	1	1	1	1	1	1	1	W	mod	0	0	1	mem	16/24	2-4	x	x	x	x	x	x	
	reg16	0	1	0	0	1	reg						2	1	x	x	x	x	x	x			
MULU	reg	1	1	1	1	0	1	1	W	1	1	1	0	0	reg	21-30	2	u	x	x	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	0	0	mem	27-36	2-4	u	x	x	u	u	u	
MUL	reg	1	1	1	1	0	1	1	W	1	1	1	0	1	reg	33-47	2	u	x	x	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	0	1	mem	39-57	2-4	u	x	x	u	u	u	
	reg16, reg16, imm8	0	1	1	0	1	0	1	1	1	1	reg	reg	28-34	3	u	x	x	u	u	u		
	reg16, mem16, imm8	0	1	1	0	1	0	1	1	mod	reg	mem	34-44	3-5	u	x	x	u	u	u			
	reg16, reg16, imm16	0	1	1	0	1	0	0	1	1	1	reg	reg	36-42	4	u	x	x	u	u	u		
	reg16, mem16, imm16	0	1	1	0	1	0	0	1	mod	reg	mem	46-52	4-6	u	x	x	u	u	u			
DIVU	reg	1	1	1	1	0	1	1	W	1	1	1	1	0	reg	19-25	2	u	u	u	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	1	0	mem	25-35	2-4	u	u	u	u	u	u	
DIV	reg	1	1	1	1	0	1	1	W	1	1	1	1	1	reg	29-43	2	u	u	u	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	1	1	mem	35-53	2-4	u	u	u	u	u	u	
<b>Comparison Instructions</b>																							
CMP	reg, reg	0	0	1	1	1	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x		
	mem, reg	0	0	1	1	1	0	0	W	mod	reg	mem	11/15	2-4	x	x	x	x	x	x			
	reg, mem	0	0	1	1	1	0	1	W	mod	reg	mem	11/15	2-4	x	x	x	x	x	x			
	reg, imm	1	0	0	0	0	0	S	W	1	1	1	1	1	reg	4	3-4	x	x	x	x	x	x
	mem, imm	1	0	0	0	0	0	S	W	mod	1	1	1	mem	13/17	3-6	x	x	x	x	x	x	
	acc, imm	0	0	1	1	1	1	0	W					4	2-3	x	x	x	x	x	x		
<b>Logical Instructions</b>																							
NOT	reg	1	1	1	1	0	1	1	W	1	1	0	1	0	reg	2	2						
	mem	1	1	1	1	0	1	1	W	mod	0	1	0	mem	16/24	2-4							
NEG	reg	1	1	1	1	0	1	1	W	1	1	0	1	1	reg	2	2	x	x	x	x	x	x
	mem	1	1	1	1	0	1	1	W	mod	0	1	1	mem	16/24	2-4	x	x	x	x	x	x	
TEST	reg, reg	1	0	0	0	0	1	0	W	1	1	reg	reg	2	2	u	0	0	x	x	x		
	mem, reg	1	0	0	0	0	1	0	W	mod	reg	mem	10/14	2-4	u	0	0	x	x	x			
	reg, imm	1	1	1	1	0	1	1	W	1	1	0	0	0	reg	4	3-4	u	0	0	x	x	x



Instruction Set (cont)

Mnemonic	Operand	Opcode											Clocks	Bytes	Flags										
		7	6	5	4	3	2	1	0	7	6	5			4	3	2	1	0	AC	CY	V	P	S	Z
<b>Logical Instructions (cont)</b>																									
	mem, imm	1	1	1	1	0	1	1	W	mod	0	0	0	mem	11/15	3-6	u	0	0	x	x	x			
	acc, imm	1	0	1	0	1	0	0	W						4	2-3	u	0	0	x	x	x			
AND	reg, reg	0	0	1	0	0	0	1	W	1	1	reg	reg	2	2	u	0	0	x	x	x				
	mem, reg	0	0	1	0	0	0	0	W	mod	reg	mem	16/20	2-4	u	0	0	x	x	x					
	reg, mem	0	0	1	0	0	0	1	W	mod	reg	mem	11/15	2-4	u	0	0	x	x	x					
	reg, imm	1	0	0	0	0	0	0	W	1	1	1	0	0	reg	4	3-4	u	0	0	x	x	x		
	mem, imm	1	0	0	0	0	0	0	W	mod	1	0	0	mem	18/26	3-6	u	0	0	x	x	x			
	acc, imm	0	0	1	0	0	1	0	W						4	2-3	u	0	0	x	x	x			
OR	reg, reg	0	0	0	0	1	0	1	W	1	1	reg	reg	2	2	u	0	0	x	x	x				
	mem, reg	0	0	0	0	1	0	0	W	mod	reg	mem	16/24	2-4	u	0	0	x	x	x					
	reg, mem	0	0	0	0	1	0	1	W	mod	reg	mem	11/15	2-4	u	0	0	x	x	x					
	reg, imm	1	0	0	0	0	0	0	W	1	1	0	0	1	reg	4	3-4	u	0	0	x	x	x		
	mem, imm	1	0	0	0	0	0	0	W	mod	0	0	1	mem	18/26	3-6	u	0	0	x	x	x			
	acc, imm	0	0	1	0	0	1	0	W						4	2-3	u	0	0	x	x	x			
XOR	reg, reg	0	0	1	1	0	0	1	W	1	1	reg	reg	2	2	u	0	0	x	x	x				
	mem, reg	0	0	1	1	0	0	0	W	mod	reg	mem	16/24	2-4	u	0	0	x	x	x					
	reg, mem	0	0	1	1	0	0	1	W	mod	reg	mem	11/15	2-4	u	0	0	x	x	x					
	reg, imm	1	0	0	0	0	0	0	W	1	1	1	1	0	reg	4	3-4	u	0	0	x	x	x		
	mem, imm	1	0	0	0	0	0	0	W	mod	1	1	0	mem	18/26	3-6	u	0	0	x	x	x			
	acc, imm	0	0	1	0	0	1	0	W						4	2-3	u	0	0	x	x	x			
<b>Bit Manipulation Instructions</b>																									
INS	reg8, reg8	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	1	35-133	3						
	reg8, imm8	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	1	35-133	4						
		1	1	0	0	0	reg																		
EXT	reg8, reg8	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1	34-59	3						
	reg8, imm8	0	0	0	0	1	1	1	1	0	0	1	1	1	0	1	1	34-59	4						
		1	1	0	0	0	reg																		
TEST1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	W	3	3	u	0	0	u	u	x
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	W	12/16	3-5	u	0	0	u	u	x
		mod	0	0	0	mem																			
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0	W	4	4	u	0	0	u	u	x
	mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0	W	13/21	4-6	u	0	0	u	u	x
		mod	0	0	0	mem																			
SET1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	0	W	4	3						
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	0	W	13/21	3-5						
		mod	0	0	0	mem																			

### Instruction Set (cont)

Mnemonic	Operands	Opcode														Clocks	Bytes	Flags					
		7	6	5	4	3	2	1	0	7	6	5	4	3	2			1	0	AC	CY	V	P
<b>Bit Manipulation Instructions (cont)</b>																							
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0	W	5	4				
		1	1	0	0	0	0	reg															
	mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0	W	14/22	4-6				
		mod	0	0	0	mem																	
	CY	1	1	1	1	1	0	0	1									2	1		1		
	DIR	1	1	1	1	1	0	1										2	1				
CLR1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	W	5	3				
		1	1	0	0	0	reg																
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	W	14/22	3-5				
		mod	0	0	0	mem																	
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	W	6	4				
		1	1	0	0	0	reg																
	mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	W	15/27	4-6				
		mod	0	0	0	mem																	
	CY	1	1	1	1	1	0	0	0									2	1		0		
	DIR	1	1	1	1	1	0	0										2	1				
NOT1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	1	W	4	3				
		1	1	0	0	0	reg																
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	1	W	18/26	3-5				
		mod	0	0	0	mem																	
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	W	5	4				
		1	1	0	0	0	reg																
	mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	W	19/27	4-6				
		mod	0	0	0	mem																	
	CY	1	1	1	1	0	1	0	1									2	1		x		
<b>Shift/Rotate Instructions</b>																							
SHL	reg, 1	1	1	0	1	0	0	0	W	1	1	1	0	0	reg	2	2	u	x	x	x	x	x
	mem, 1	1	1	0	1	0	0	0	W	mod	1	0	0	mem	16/24	2-4	u	x	x	x	x	x	
	reg, CL	1	1	0	1	0	0	1	W	1	1	1	0	0	reg	7 + n	2	u	x	u	x	x	x
	mem, CL	1	1	0	1	0	0	1	W	mod	1	0	0	mem	19/27 + n	2-4	u	x	u	x	x	x	
	reg, imm8	1	1	0	0	0	0	0	W	1	1	1	0	0	reg	7 + n	3	u	x	u	x	x	x
	mem, imm8	1	1	0	0	0	0	0	W	mod	1	0	0	mem	19/27 + n	3-5	u	x	u	x	x	x	
SHR	reg, 1	1	1	0	1	0	0	0	W	1	1	1	0	1	reg	2	2	u	x	x	x	x	x
	mem, 1	1	1	0	1	0	0	0	W	mod	1	0	1	mem	16/24	2-4	u	x	x	x	x	x	
	reg, CL	1	1	0	1	0	0	1	W	1	1	1	0	1	reg	7 + n	2	u	x	u	x	x	x
	mem, CL	1	1	0	1	0	0	1	W	mod	1	0	1	mem	19/27 + n	2-4	u	x	u	x	x	x	
	reg, imm8	1	1	0	0	0	0	0	W	1	1	1	0	1	reg	7 + n	3	u	x	u	x	x	x
	mem, imm8	1	1	0	0	0	0	0	W	mod	1	0	1	mem	19/27 + n	3-5	u	x	u	x	x	x	

n = number of shifts

**Instruction Set (cont)**

Mnemonic	Operands	Opcode										Clocks	Bytes	Flags									
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P
<b>Shift/Rotate Instructions (cont)</b>																							
SHRA	reg, 1	1	1	0	1	0	0	0	W	1	1	1	1	1	reg	2	2	u	x	0	x	x	x
	mem, 1	1	1	0	1	0	0	0	W	mod	1	1	1	mem	16/24	2-4	u	x	0	x	x	x	
	reg, CL	1	1	0	1	0	0	1	W	1	1	1	1	1	reg	7 + n	2	u	x	u	x	x	x
	mem, CL	1	1	0	1	0	0	1	W	mod	1	1	1	mem	19/27 + n	2-4	u	x	u	x	x	x	
	reg, imm8	1	1	0	0	0	0	0	W	1	1	1	1	1	reg	7 + n	3	u	x	u	x	x	x
	mem, imm8	1	1	0	0	0	0	0	W	mod	1	1	1	mem	19/27 + n	3-5	u	x	u	x	x	x	
ROL	reg, 1	1	1	0	1	0	0	0	W	1	1	0	0	0	reg	2	2			x	x		
	mem, 1	1	1	0	1	0	0	0	W	mod	0	0	0	mem	16/24	2-4			x	x			
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	0	0	reg	7 + n	2			x	u		
	mem, CL	1	1	0	1	0	0	1	W	mod	0	0	0	mem	19/27 + n	2-4			x	u			
	reg, imm	1	1	0	0	0	0	0	W	1	1	0	0	0	reg	7 + n	3			x	u		
	mem, imm	1	1	0	0	0	0	0	W	mod	0	0	0	mem	19/27 + n	3-5			x	u			
ROR	reg, 1	1	1	0	1	0	0	0	W	1	1	0	0	1	reg	2	2			x	u		
	mem, 1	1	1	0	1	0	0	0	W	mod	0	0	1	mem	16/24	2-4			x	x			
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	0	1	reg	7 + n	2			x	u		
	mem, CL	1	1	0	1	0	0	1	W	mod	0	0	1	mem	19/27 + n	2-4			x	u			
	reg, imm8	1	1	0	0	0	0	0	W	1	1	0	0	1	reg	7 + n	3			x	u		
	mem, imm8	1	1	0	0	0	0	0	W	mod	0	0	1	mem	19/27 + n	3-5			x	u			
ROLC	reg, 1	1	1	0	1	0	0	0	W	1	1	0	1	0	reg	2	2			x	x		
	mem, 1	1	1	0	1	0	0	0	W	mod	0	1	0	mem	16/28	2-4			x	x			
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	1	0	reg	7 + n	2			x	u		
	mem, CL	1	1	0	1	0	0	1	W	mod	0	1	0	mem	19/27 + n	2-4			x	u			
	reg, imm8	1	1	0	0	0	0	0	W	1	1	0	1	0	reg	7 + n	3			x	u		
	mem, imm8	1	1	0	0	0	0	0	W	mod	0	1	0	mem	19/27 + n	3-5			x	u			
RORC	reg, 1	1	1	0	1	0	0	0	W	1	1	0	1	1	reg	2	2			x	x		
	mem, 1	1	1	0	1	0	0	0	W	mod	0	1	1	mem	16/24	2-4			x	x			
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	1	1	reg	7 + n	2			x	u		
	mem, CL	1	1	0	1	0	0	1	W	mod	0	1	1	mem	19/27 + n	2-4			x	u			
	reg, imm8	1	1	0	0	0	0	0	W	1	1	0	1	1	reg	7 + n	3			x	u		
	mem, imm8	1	1	0	0	0	0	0	W	mod	0	1	1	mem	19/27 + n	3-5			x	u			

n = number of shifts

**Stack Manipulation Instructions**

PUSH	mem16	1	1	1	1	1	1	1	mod	1	1	0	mem	26	2-4						
	reg16	0	1	0	1	0	reg							12	1						
	sr	0	0	0	sr		1	1	0					12	1						
	PSW	1	0	0	1	1	1	0	0					12	1						
	R	0	1	1	0	0	0	0	0					67	1						
	imm	0	1	1	0	1	0	S	0					11/12	2-3						

### Instruction Set (cont)

Mnemonic	Operands	Opcode																Clocks	Bytes	Flags					
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
<b>Stack Manipulation Instructions (cont)</b>																									
POP	mem16	1	0	0	0	1	1	1	1	mod	0	0	0	0	mem	25	2-4								
	reg16	0	1	0	1	1			reg							12	1								
	sr	0	0	0		sr	1	1	1							12	1								
	PSW	1	0	0	1	1	1	0	1							12	1	R	R	R	R	R	R		
	R	0	1	1	0	0	0	0	1							75	1								
PREPARE	imm16, imm8	1	1	0	0	1	0	0	0							*	4								
										*imm8 = 0 : 12 imm8 > 1 : 19 + 8 (imm8 - 1)															
DISPOSE		1	1	0	0	1	0	0	1							10	1								
<b>Control Transfer Instructions</b>																									
CALL	near_proc	1	1	1	0	1	0	0	0							20	3								
	regptr	1	1	1	1	1	1	1	1	1	1	0	1	0	reg	18	1								
	memptr16	1	1	1	1	1	1	1	1	mod	0	1	0	0	mem	31	2-4								
	far_proc	1	0	0	1	1	0	1	0							29	5								
	memptr32	1	1	1	1	1	1	1	1	mod	0	1	1	1	mem	47	2-4								
RET		1	1	0	0	0	0	1	1							19	1								
	pop_value	1	1	0	0	0	0	1	0							24	3								
		1	1	0	0	1	0	1	1							29	1								
	pop_value	1	1	0	0	1	0	1	0							32	3								
BR	near_label	1	1	1	0	1	0	0	1							13	3								
	short_label	1	1	1	0	1	0	0	1							12	2								
	reg	1	1	1	1	1	1	1	1	1	1	1	0	0	reg	11	2								
	memptr16	1	1	1	1	1	1	1	1	mod	1	0	0	0	mem	24	2-4								
	far_label	1	1	1	0	1	0	1	0							15	5								
	memptr32	1	1	1	1	1	1	1	1	mod	1	0	1	1	mem	35	2-4								
BV	near_label	0	1	1	1	0	0	0	0							14/4	2								
BNV	near_label	0	1	1	1	0	0	0	1							14/4	2								
BC, BL	near_label	0	1	1	1	0	0	1	0							14/4	2								
BNC, BNL	near_label	0	1	1	1	0	0	1	1							14/4	2								
BE, BZ	near_label	0	1	1	1	0	1	0	0							14/4	2								
BNE, BNZ	near_label	0	1	1	1	0	1	0	1							14/4	2								
BNH	near_label	0	1	1	1	0	1	1	0							14/4	2								
BH	near_label	0	1	1	1	0	1	1	1							14/4	2								
BN	near_label	0	1	1	1	1	0	0	0							14/4	2								
BP	near_label	0	1	1	1	1	0	0	1							14/4	2								
BPE	near_label	0	1	1	1	1	0	1	0							14/4	2								
BPO	near_label	0	1	1	1	1	0	1	1							14/4	2								
BLT	near_label	0	1	1	1	1	1	0	0							14/4	2								
BGE	near_label	0	1	1	1	1	1	0	1							14/4	2								

**Instruction Set (cont)**

Mnemonic	Operand	Opcode										Clocks	Bytes	Flags											
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P	S	Z
<b>Control Transfer Instructions (cont)</b>																									
BLE	near_Label	0	1	1	1	1	1	1	0								14/4	2							
BGT	near_Label	0	1	1	1	1	1	1	1								14/4	2							
DBNZNE	near_Label	1	1	1	0	0	0	0	0								14/5	2							
DBNZE	near_Label	1	1	1	0	0	0	0	1								14/5	2							
DBNZ	near_Label	1	1	1	0	0	0	1	0								13/5	2							
BCWZ	near_Label	1	1	1	0	0	0	1	1								13/5	2							
<b>Interrupt Instructions</b>																									
BRK	3	1	1	0	0	1	1	0	0								50	1							
	imm8	1	1	0	0	1	1	0	1								50	2							
BRKV	imm8	1	1	0	0	1	1	1	1								52/3	1							
RETI		1	1	0	0	1	1	1	0								39	1	R	R	R	R	R	R	
CHKIND	reg16, mem32	0	1	1	0	0	0	1	0	mod	reg		mem				73-76/26	2-4							
BRKEM	imm8	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	50	3							
<b>CPU Control Instructions</b>																									
HALT		1	1	1	1	0	1	0	0								2	1							
BUSLOCK		1	1	1	1	0	0	0	0								2	1							
FP01	fp_op	1	1	0	1	1	X	X	X	1	1	Y	Y	Z	Z	Z	2	2							
	fp_op, mem	1	1	0	1	1	X	X	X	mod	Y	Y	Y	mem			15	2-4							
FP02	fp_op	0	1	1	0	0	1	1	X	1	1	Y	Y	Z	Z	Z	2	2							
	fp_op, mem	0	1	1	0	0	1	1	X	mod	Y	Y	Y	mem			15	2-4							
POLL		1	0	0	1	1	0	1	1								2 + 5n	1							
NOP		1	0	0	1	0	0	0	0								3	1							
DI		1	1	1	1	0	1	0								2	1								
EI		1	1	1	1	1	0	1	1								2	1							
<b>8080 Instruction Set Enhancements</b>																									
RETEM		1	1	1	0	1	1	0	1	1	1	1	1	1	0	1	39	2	R	R	R	R	R	R	
CALLN	imm8	1	1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	58	3						

### Description

The μPD70116 (V30) is a CMOS 16-bit microprocessor with an internal 16-bit architecture and a 16-bit external data bus. The μPD70116 instruction set is a superset of the μPD8086/8088; however, mnemonics and execution times are different. The μPD70116 additionally has a powerful instruction set including bit processing, packed BCD operations, and high-speed multiplication/division operations. The μPD70116 can also execute the entire 8080 instruction set and comes with a standby mode that significantly reduces power consumption. It is software-compatible with the μPD70108 microprocessor.

### Features

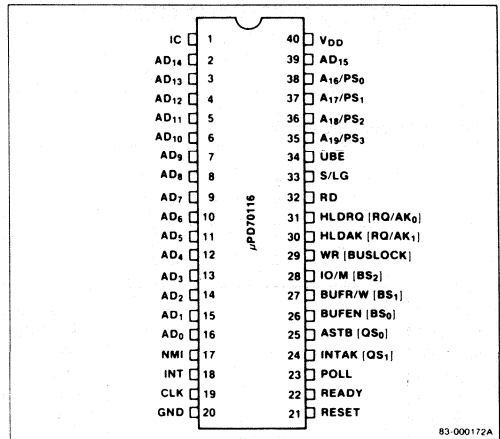
- Minimum instruction execution time: 250 ns (at 8 MHz)
- Maximum addressable memory: 1 Mbyte
- Abundant memory addressing modes
- 14 x 16-bit register set
- 101 instructions
- Instruction set is a superset of μPD8086/8088 instruction set
- Bit, byte, word, and block operations
- Bit field operation instructions
- Packed BCD instructions
- Multiplication/division instruction execution time: 4 μs to 6 μs (at 8 MHz)
- High-speed block transfer instructions: 2 Mbyte/s (at 8 MHz)
- High-speed calculation of effective addresses: 2 clock cycles in any addressing mode
- Maskable (INT) and nonmaskable (NMI) interrupt inputs
- IEEE-796 bus compatible interface
- 8080 emulation mode
- CMOS technology
- Low-power consumption
- Low-power standby mode
- Single power supply
- 5-MHz, 8-MHz or 10-MHz clock

### Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD70116C-5	40-pin plastic DIP	5 MHz
μPD70116C-8	40-pin plastic DIP	8 MHz
μPD70116C-10	40-pin plastic DIP	10 MHz
μPD70116D-5	40-pin ceramic DIP	5 MHz
μPD70116D-8	40-pin ceramic DIP	8 MHz
μPD70116D-10	40-pin ceramic DIP	10 MHz
μPD0116GC-5	52-pin plastic flat pack	5 MHz
μPD70116GC-8	52-pin plastic flat pack	8 MHz
μPD70116L-5	44-pin PLCC	5 MHz
μPD70116L-8	44-pin PLCC	8 MHz

### Pin Configurations

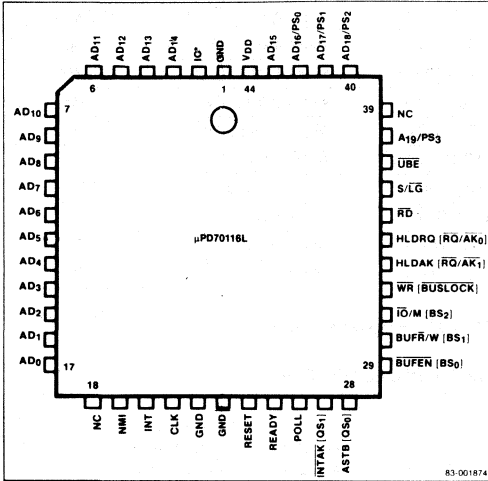
#### 40-Pin Plastic DIP/Cerdip



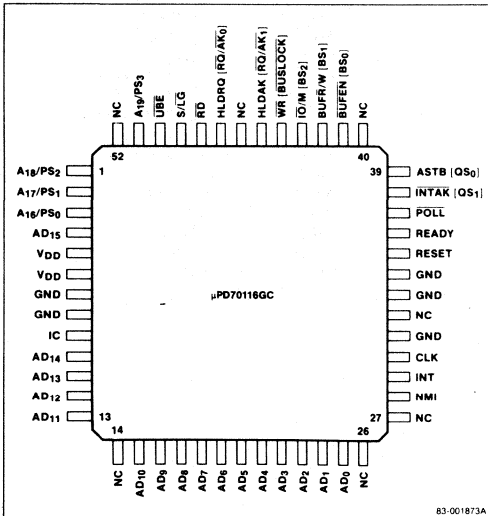
83-000172A

**Pin Configurations (cont)**

**44-Pin Plastic Leadless Chip Carrier (PLCC)**



**52-Pin Plastic Flat Pack**



**Pin Identification**

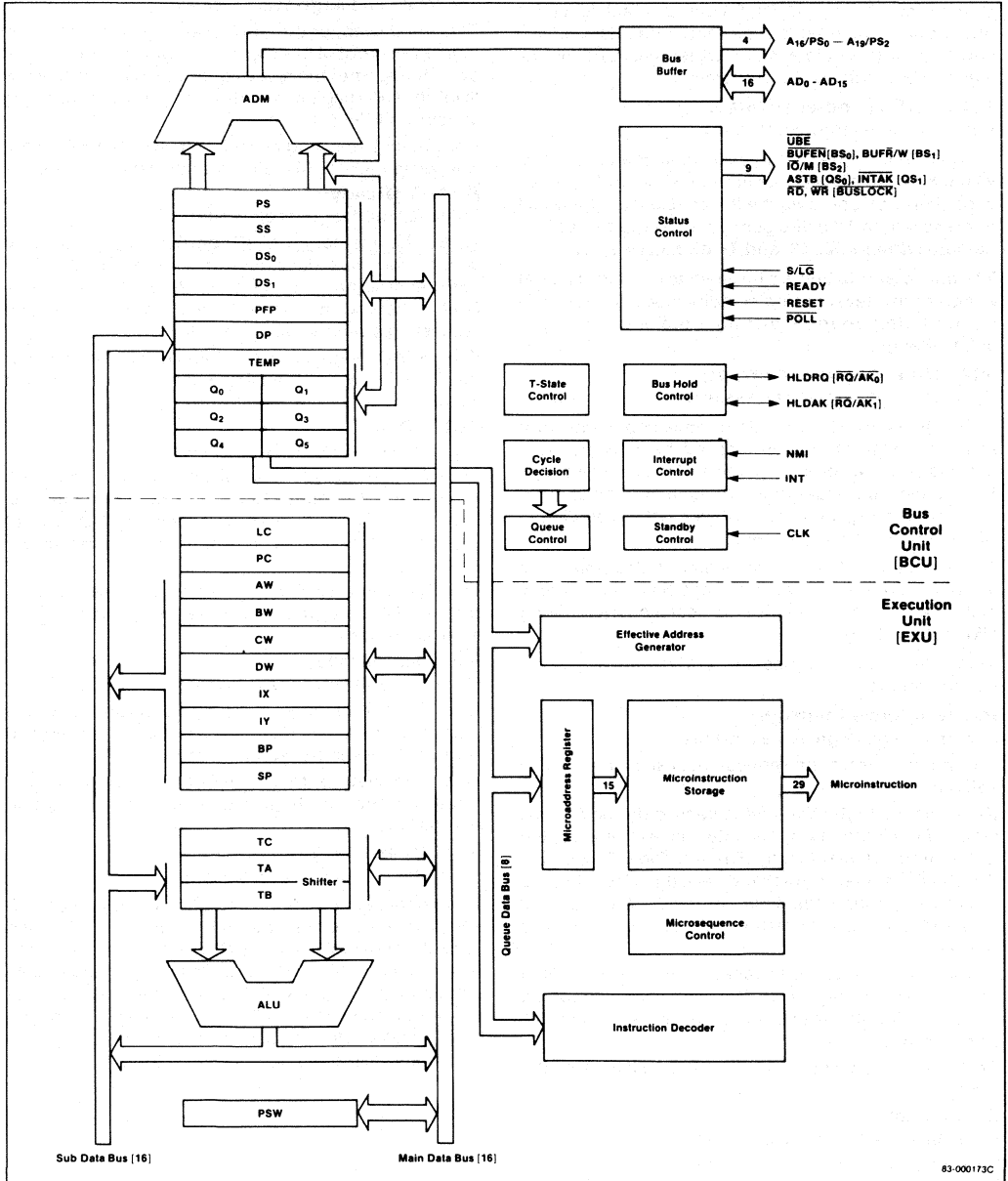
Symbol	Direction	Function
IC*		Internally connected
AD <sub>14</sub> - AD <sub>0</sub>	In/Out	Address/data bus
NMI	In	Nonmaskable interrupt input
INT	In	Maskable interrupt input
CLK	In	Clock input
GND		Ground potential
RESET	In	Reset input
READY	In	Ready input
POLL	In	Poll input
INTAK (QS <sub>1</sub> )	Out	Interrupt acknowledge output (queue status bit 1 output)
ASTB (OS <sub>0</sub> )	Out	Address strobe output (queue status bit 0 output)
BUFEN (BS <sub>0</sub> )	Out	Buffer enable output (bus status bit 0 output)
BUFR/W (BS <sub>1</sub> )	Out	Buffer read/write output (bus status bit 1 output)
I/O/M (BS <sub>2</sub> )	Out	Access is I/O or memory (bus status bit 2 output)
WR (BUSLOCK)	Out	Write strobe output (bus lock output)
HLDK (RQ/AK <sub>1</sub> )	Out (In/Out)	Hold acknowledge output. (bus hold request input/acknowledge output 1)
HLDK (RQ/AK <sub>0</sub> )	In (In/Out)	Hold request input (bus hold request input/acknowledge output 0)
RD	Out	Read strobe output
S/LG	In	Small-scale/large-scale system input
UBE	Out	Upper byte enable
A <sub>19</sub> /PS <sub>3</sub> - A <sub>16</sub> /PS <sub>0</sub>	Out	Address bus, high bits or processor status output
AD <sub>15</sub>	In/Out	Address/data bus, bit 15
V <sub>DD</sub>		Power supply

**Notes:** \* IC should be connected to ground.

Where pins have different functions in small- and large-scale systems, the large-scale system pin symbol and function are in parentheses.

Unused input pins should be tied to ground or V<sub>DD</sub> to minimize power dissipation and prevent the flow of potentially harmful currents.

## Block Diagram





**Pin Functions**

Some pins of the μPD70116 have different functions according to whether the microprocessor is used in a small- or large-scale system. Other pins function the same way in either type of system.

**AD<sub>15</sub> - AD<sub>0</sub> [Address/Data Bus]**

For small- and large-scale systems.

AD<sub>15</sub> - AD<sub>0</sub> is a time-multiplexed address and data bus. When high, an AD bit is a one; when low, an AD bit is a zero. This bus contains the lower 16 bits of the 20-bit address during T<sub>1</sub> of the bus cycle. It is used as a 16-bit data bus during T<sub>2</sub>, T<sub>3</sub>, and T<sub>4</sub> of the bus cycle.

The address/data bus is a three-state bus and can be at a high or low level during standby mode. The bus will be high impedance during hold and interrupt acknowledge.

**NMI [Nonmaskable Interrupt]**

For small- and large-scale systems.

This pin is used to input nonmaskable interrupt requests. NMI cannot be masked by software. This input is positive edge triggered and must be held high for five clocks to guarantee recognition. Actual interrupt processing begins, however, after completion of the instruction in progress.

The contents of interrupt vector 2 determine the starting address for the interrupt-servicing routine. Note that a hold request will be accepted even during NMI acknowledge.

This interrupt will cause the μPD70116 to exit the standby mode.

**INT [Maskable Interrupt]**

For small- and large-scale systems.

This pin is an interrupt request that can be masked by software.

INT is active high level and is sensed during the last clock of the instruction. The interrupt will be accepted if the interrupt enable flag IE is set. The CPU outputs the  $\overline{\text{INTAK}}$  signal to inform external devices that the interrupt request has been granted. INT must be asserted until the interrupt acknowledge signal is returned.

If NMI and INT interrupts occur at the same time, NMI has higher priority than INT and INT cannot be accepted. A hold request will be accepted during INT acknowledge.

This interrupt causes the μPD70116 to exit the standby mode.

**CLK [Clock]**

For small- and large-scale systems.

This pin is used for external clock input.

**RESET [Reset]**

For small- and large-scale systems.

This pin is used for the CPU reset signal. It is an active high level. Input of this signal has priority over all other operations. After the reset signal input returns to a low level, the CPU begins execution of the program starting at address FFFF0H.

In addition to causing normal CPU start, RESET input will cause the μPD70116 to exit the standby mode.

**READY [Ready]**

For small- and large-scale systems.

When the memory or I/O device being accessed cannot complete data read or write within the CPU basic access time, it can generate a CPU wait state (T<sub>w</sub>) by setting this signal to inactive (low level) and requesting a read/write cycle delay.

If the READY signal is active (high level) during either the T<sub>3</sub> or T<sub>w</sub> state, the CPU will not generate a wait state.

**POLL [Poll]**

For small- and large-scale systems.

The CPU checks this input upon execution of the POLL instruction. If the input is low, then execution continues. If the input is high, the CPU will check the  $\overline{\text{POLL}}$  input every five clock cycles until the input becomes low again.

The  $\overline{\text{POLL}}$  and READY functions are used to synchronize CPU program execution with the operation of external devices.

 **$\overline{\text{RD}}$  [Read Strobe]**

For small- and large-scale systems.

The CPU outputs this strobe signal during data read from an I/O device or memory. The  $\overline{\text{IO/M}}$  signal is used to select between I/O and memory.

This three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

**S/ $\overline{\text{LG}}$  [Small/Large]**

For small- and large-scale systems.

This signal determines the operation mode of the CPU. This signal is fixed at either a high or low level. When this signal is a high level, the CPU will operate in small-scale system mode, and when low, in the large-scale system mode. A small-scale system will have at most one bus master such as a DMA controller device on the bus. A large-scale system can have more than one bus master accessing the bus as well as the CPU.

### **INTAK** [Interrupt Acknowledge]

For small-scale systems.

The CPU generates the  $\overline{\text{INTAK}}$  signal low when it accepts an INT signal.

The interrupting device synchronizes with this signal and outputs the interrupt vector to the CPU via the data bus (AD<sub>7</sub> - AD<sub>0</sub>).

### **ASTB** [Address Strobe]

For small-scale systems.

The CPU outputs this strobe signal to latch address information at an external latch.

ASTB is held at a low level during standby mode and hold acknowledge.

### **BUFEN** [Buffer Enable]

For small-scale systems.

This is used as the output enable signal for an external bidirectional buffer. The CPU generates this signal during data transfer operations with external memory or I/O devices or during input of an interrupt vector.

This three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

### **BUF $\overline{\text{R}}$ /W** [Buffer Read/Write]

For small-scale systems.

The output of this signal determines the direction of data transfer with an external bidirectional buffer. A high output causes transmission from the CPU to the external device; a low signal causes data transfer from the external device to the CPU.

BUF $\overline{\text{R}}$ /W is a three-state output and enters the high-impedance state during hold acknowledge.

### **$\overline{\text{IO}}$ /M** [IO/Memory]

For small-scale systems.

The CPU generates this signal to specify either I/O access or memory access. A low-level output specifies I/O and a high-level signal specifies memory.

$\overline{\text{IO}}$ /M's output is three state and becomes high impedance during hold acknowledge.

### **WR** [Write Strobe]

For small-scale systems.

The CPU generates this strobe signal during data write to an I/O device or memory. Selection of either I/O or memory is performed by the  $\overline{\text{IO}}$ /M signal.

This three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

### **HLD $\overline{\text{AK}}$** [Hold Acknowledge]

For small-scale systems.

The HLD $\overline{\text{AK}}$  signal is used to indicate that the CPU accepts the hold request signal (HLDRQ). When this signal is a high level, the address bus, address/data bus, the control lines become high impedance.

### **HLDRQ** [Hold Request]

For small-scale systems.

This input signal is used by external devices to request the CPU to release the address bus, address/data bus, and the control bus.

### **$\overline{\text{UBE}}$** [Upper Byte Enable]

For small- and large-scale systems.

$\overline{\text{UBE}}$  indicates the use of the upper eight bits (AD<sub>16</sub> - AD<sub>8</sub>) of the address/data bus during a bus cycle. This signal is active low during T1 for read, write, and interrupt acknowledge cycles when AD<sub>15</sub> - AD<sub>8</sub> are to be used. Bus cycles in which  $\overline{\text{UBE}}$  is active are shown in the following table.

Type of Bus Operation	$\overline{\text{UBE}}$	AD <sub>0</sub>	Number of Bus Cycles
Word at even address	0	0	1
Word at odd address	0 1	1* 0**	2
Byte at even address	1	0	1
Byte at odd address	0	1	1

Notes: \* First bus cycle

\*\* Second bus cycle

$\overline{\text{UBE}}$  is low continuously during the interrupt acknowledge state.

The three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

### **A<sub>19</sub>/PS<sub>3</sub> - A<sub>16</sub>/PS<sub>0</sub>** [Address Bus/Processor Status]

For small- and large-scale systems.

These pins are time multiplexed to operate as an address bus and as processor status signals.

When used as the address bus, these pins are the high 4 bits of the 20-bit memory address. During I/O access, all 4 bits output data 0.

The processor status signals are provided for both memory and I/O use. PS<sub>3</sub> is always 0 in the native mode and 1 in 8080 emulation mode. The interrupt enable flag (IE) is output on pin PS<sub>2</sub>. Pins PS<sub>1</sub> and PS<sub>0</sub> indicate which memory segment is being accessed.

A <sub>17</sub> /PS <sub>1</sub>	A <sub>16</sub> /PS <sub>0</sub>	Segment
0	0	Data segment 1
0	1	Stack segment
1	0	Program segment
1	1	Data segment 0

The output of these pins is three state and becomes high impedance during hold acknowledge.

### QS<sub>1</sub>, QS<sub>0</sub> [Queue Status]

For large-scale systems.

The CPU uses these signals to allow external devices, such as the floating-point arithmetic processor chip ( $\mu$ PD72091), to monitor the status of the internal CPU instruction queue.

QS <sub>1</sub>	QS <sub>0</sub>	Instruction Queue Status
0	0	NOP (queue does not change)
0	1	First byte of instruction
1	0	Flush queue
1	1	Subsequent bytes of instruction

The instruction queue status indicated by these signals is the status when the execution unit (EXU) accesses the instruction queue. The data output from these pins is therefore valid only for one clock cycle immediately following queue access. These status signals are provided so that the floating-point processor chip can monitor the CPU's program execution status and synchronize its operation with the CPU when control is passed to it by the FPO (Floating Point Operation) instructions.

### BS<sub>2</sub> - BS<sub>0</sub> [Bus Status]

For large-scale systems.

The CPU uses these status signals to allow an external bus controller to monitor what the current bus cycle is.

The external bus controller decodes these signals and generates the control signals required to perform access of the memory or I/O device.

BS <sub>2</sub>	BS <sub>1</sub>	BS <sub>0</sub>	Bus Cycle
0	0	0	Interrupt acknowledge
0	0	1	I/O read
0	1	0	I/O write
0	1	1	Halt
1	0	0	Program fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive state

The output of these signals is three state and becomes high impedance during hold acknowledge.

### BUSLOCK [Bus Lock]

For large-scale systems.

The CPU uses this signal to secure the bus while executing the instruction immediately following the BUSLOCK prefix instruction and during interrupt acknowledge cycles. It is a status signal to the other bus masters in a multiprocessor system inhibiting them from using the system bus during this time.

The output of this signal is three state and becomes high impedance during hold acknowledge. BUSLOCK is high during standby mode except if the HALT instruction has a BUSLOCK prefix.

### RQ/AK<sub>1</sub>, RQ/AK<sub>0</sub> [Hold Request/Acknowledge]

For large-scale systems.

These pins function as bus hold request inputs ( $\overline{RQ}$ ) and as bus hold acknowledge outputs (AK). RQ/AK<sub>0</sub> has a higher priority than  $\overline{RQ}$ /AK<sub>1</sub>.

These pins have three-state outputs with on-chip pull-up resistors which keep the pin at a high level when the output is high impedance.

### V<sub>DD</sub> [Power Supply]

For small- and large-scale systems.

This pin is used for the +5 V power supply.

### GND [Ground]

For small- and large-scale systems.

This pin is used for ground.

### IC [Internally Connected]

This pin is used for tests performed at the factory by NEC. The  $\mu$ PD70116 is used with this pin at ground potential.

### Absolute Maximum Ratings

$T_A = +25^\circ\text{C}$

Power supply voltage, $V_{DD}$	-0.5 V to +7.0 V
Power dissipation, $P_{D_{MAX}}$	0.5 W
Input voltage, $V_I$	-0.5 V to $V_{DD} + 0.3$ V
CLK input voltage, $V_K$	-0.5 V to $V_{DD} + 1.0$ V
Output voltage, $V_O$	-0.5 V to $V_{DD} + 0.3$ V
Operating temperature, $T_{OPT}$	-40°C to +85°C
Storage temperature, $T_{STG}$	-65°C to +150°C

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### DC Characteristics

μPD70116-5,  $T_A = -40^\circ\text{C}$  to +85°C,  $V_{DD} = +5\text{ V} \pm 10\%$

μPD70116-8, μPD70116-10,  $T_A = -10^\circ\text{C}$  to +70°C,  $V_{DD} = +5\text{ V} \pm 5\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage high	$V_{IH}$	2.2		$V_{DD} + 0.3$	V	
Input voltage low	$V_{IL}$	-0.5		0.8	V	
CLK input voltage high	$V_{KH}$	3.9		$V_{DD} + 1.0$	V	
CLK input voltage low	$V_{KL}$	-0.5		0.6	V	
Output voltage high	$V_{OH}$	$0.7 \times V_{DD}$			V	$I_{OH} = -400\ \mu\text{A}$
Output voltage low	$V_{OL}$			0.4	V	$I_{OL} = 2.5\ \text{mA}$
Input leakage current high	$I_{LIH}$			10	μA	$V_I = V_{DD}$
Input leakage current low	$I_{LIL}$			-10	μA	$V_I = 0\ \text{V}$
Output leakage current high	$I_{LOH}$			10	μA	$V_O = V_{DD}$
Output leakage current low	$I_{LOL}$			-10	μA	$V_O = 0\ \text{V}$
Supply current	$I_{DD}$	70116-5	30	60	mA	Normal operation
		5 MHz	5	10	mA	Standby mode
		70116-8	45	80	mA	Normal operation
		8 MHz	6	12	mA	Standby mode
		70116-10	60	100	mA	Normal operation
		10 MHz	7	14	mA	Standby mode

### Capacitance

$T_A = +25^\circ\text{C}$ ,  $V_{DD} = 0\ \text{V}$

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input capacitance	$C_I$		15	pF	$f_c = 1\ \text{MHz}$ Unmeasured pins returned to 0 V
I/O capacitance	$C_{IO}$		15	pF	

## μPD70116 (V30)

### AC Characteristics

μPD70116-5, T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V ± 10%

μPD70116-8, μPD70116-10 T<sub>A</sub> = -10°C to +70°C, V<sub>DD</sub> = +5 V ± 5%

Parameter	Symbol	μPD70116-5		μPD70116-8		μPD70116-10		Unit	Conditions
		Min	Max	Min	Max	Min	Max		
<b>Small/Large Scale</b>									
Clock cycle	t <sub>CYK</sub>	200	500	125	500	100	500	ns	
Clock pulse width high	t <sub>KKH</sub>	69		44		41		ns	V <sub>KH</sub> = 3.0 V
Clock pulse width low	t <sub>KKL</sub>	90		60		49		ns	V <sub>KL</sub> = 1.5 V
Clock rise time	t <sub>KR</sub>		10		8		5	ns	1.5 V to 3.0 V
Clock fall time	t <sub>KF</sub>		10		7		5	ns	3.0 V to 1.5 V
READY inactive setup to CLK ↓	t <sub>SRYLK</sub>	-8		-8		-10		ns	
READY inactive hold after CLK ↑	t <sub>HKRYH</sub>	30		20		20		ns	
READY active setup to CLK ↑	t <sub>SRYHK</sub>	t <sub>KKL</sub> - 8		t <sub>KKL</sub> - 8		t <sub>KKL</sub> - 10		ns	
READY active hold after CLK ↑	t <sub>HKRYL</sub>	30		20		20		ns	
Data setup time to CLK ↓	t <sub>SDK</sub>	30		20		10		ns	
Data hold time after CLK ↓	t <sub>HKD</sub>	10		10		10		ns	
NMI, INT, POLL setup time to CLK ↑	t <sub>SIK</sub>	30		15		15		ns	
Input rise time (except CLK)	t <sub>IR</sub>		20		20		20	ns	0.8 V to 2.2 V
Input fall time (except CLK)	t <sub>IF</sub>		12		12		12	ns	2.2 V to 0.8 V
Output rise time	t <sub>OR</sub>		20		20		20	ns	0.8 V to 2.2 V
Output fall time	t <sub>OF</sub>		12		12		12	ns	2.2 V to 0.8 V
Reset pulse width high	t <sub>RESET1</sub>	50		50		50		μs	After power on
	t <sub>RESET2</sub>	4t <sub>CYK</sub>		4t <sub>CYK</sub>		4t <sub>CYK</sub>			During operation
<b>Small Scale</b>									
Address delay time from CLK	t <sub>DKA</sub>	10	90	10	60	10	48	ns	
Address hold time from CLK	t <sub>HKA</sub>	10		10		10		ns	
PS delay time from CLK ↓	t <sub>DKP</sub>	10	90	10	60	10	50	ns	
PS float delay time from CLK ↑	t <sub>FKP</sub>	10	80	10	60	10	50	ns	
Address setup time to ASTB ↓	t <sub>SAST</sub>	t <sub>KKL</sub> - 60		t <sub>KKL</sub> - 30		t <sub>KKL</sub> - 30		ns	
Address float delay time from CLK ↓	t <sub>FAK</sub>	t <sub>HKA</sub>	80	t <sub>HKA</sub>	60	t <sub>HKA</sub>	50	ns	C <sub>L</sub> = 100 pF
ASTB ↓ delay time from CLK ↓	t <sub>DKSTH</sub>		80		50		40	ns	
ASTB ↓ delay time from CLK ↑	t <sub>DKSTL</sub>		85		55		45	ns	
ASTB width high	t <sub>STST</sub>	t <sub>KKL</sub> - 20		t <sub>KKL</sub> - 10		t <sub>KKL</sub> - 10		ns	
Address hold time from ASTB ↓	t <sub>HSTA</sub>	t <sub>KKH</sub> - 10		t <sub>KKH</sub> - 10		t <sub>KKH</sub> - 10		ns	

### AC Characteristics (cont)

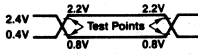
μPD70116-5, T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V ± 10%

μPD70116-8, μPD70116-10, T<sub>A</sub> = -10°C to +70°C, V<sub>DD</sub> = +5 V ± 5%

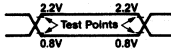
Parameter	Symbol	μPD70116-5		μPD70116-8		μPD70116-10		Unit	Conditions
		Min	Max	Min	Max	Min	Max		
<b>Small Scale (cont)</b>									
Control delay time from CLK	t <sub>DKCT</sub>	10	110	10	65	10	55	ns	
Address float to $\overline{RD}$ ↓	t <sub>AFRL</sub>	0		0		0		ns	
$\overline{RD}$ ↓ delay time from CLK ↓	t <sub>DKRL</sub>	10	165	10	80	10	70	ns	
$\overline{RD}$ ↑ delay time from CLK ↓	t <sub>DKRH</sub>	10	150	10	80	10	60	ns	
Address delay time from $\overline{RD}$ ↑	t <sub>DRHA</sub>	t <sub>CYK</sub> - 45		t <sub>CYK</sub> - 40		t <sub>CYK</sub> - 35		ns	C <sub>L</sub> = 100 pF
$\overline{RD}$ width low	t <sub>RR</sub>	2t <sub>CYK</sub> - 75		2t <sub>CYK</sub> - 50		2t <sub>CYK</sub> - 40		ns	
Data output delay time from CLK ↓	t <sub>DKD</sub>	10	90	10	60	10	50	ns	
Data float delay time from CLK ↓	t <sub>FKD</sub>	10	80	10	60	10	50	ns	
WR width low	t <sub>WW</sub>	2t <sub>CYK</sub> - 60		2t <sub>CYK</sub> - 40		2t <sub>CYK</sub> - 35		ns	
HLDRO setup time to CLK ↑	t <sub>SHOK</sub>	35		20		20		ns	
HLDKAK delay time from CLK ↓	t <sub>DKHA</sub>	10	160	10	100	10	60	ns	
<b>Large Scale</b>									
Address delay time from CLK	t <sub>DKA</sub>	10	90	10	60	10	48	ns	
Address hold time from CLK	t <sub>HKA</sub>	10		10		10		ns	
PS delay time from CLK ↓	t <sub>DKP</sub>	10	90	10	60	10	50	ns	
PS float delay time from CLK ↑	t <sub>FKP</sub>	10	80	10	60	10	50	ns	
Address float delay time from CLK ↓	t <sub>FKA</sub>	t <sub>HKA</sub>	80	t <sub>HKA</sub>	60	t <sub>HKA</sub>	50	ns	
Address delay time from $\overline{RD}$ ↑	t <sub>DRHA</sub>	t <sub>CYK</sub> - 45		t <sub>CYK</sub> - 40		t <sub>CYK</sub> - 35		ns	C <sub>L</sub> = 100 pF
ASTB delay time from BS ↓	t <sub>DBST</sub>		15		15		15	ns	
BS ↓ delay time from CLK ↑	t <sub>DKBL</sub>	10	110	10	60	10	50	ns	
BS ↑ delay time from CLK ↓	t <sub>DKBH</sub>	10	130	10	65	10	50	ns	
$\overline{RD}$ ↓ delay time from address float	t <sub>DAFRL</sub>	0		0		0		ns	
$\overline{RD}$ ↓ delay time from CLK ↓	t <sub>DKRL</sub>	10	165	10	80	10	70	ns	
$\overline{RD}$ ↑ delay time from CLK ↓	t <sub>DKRH</sub>	10	150	10	80	10	60	ns	
$\overline{RD}$ width low	t <sub>RR</sub>	2t <sub>CYK</sub> - 75		2t <sub>CYK</sub> - 50		2t <sub>CYK</sub> - 40		ns	
Data output delay time from CLK ↓	t <sub>DKD</sub>	10	90	10	60	10	50	ns	
Data float delay time from CLK ↑	t <sub>FKD</sub>	10	80	10	60	10	50	ns	
AK delay time from CLK ↓	t <sub>DKAK</sub>		70		50		40	ns	
RQ setup time to CLK ↑	t <sub>SRQK</sub>	20		10		9		ns	
$\overline{RQ}$ hold time after CLK ↑	t <sub>HLRQ2</sub>	40		30		20		ns	
CLK <sub>1</sub> to RQ hold time	t <sub>HKRQ1</sub>	0		0		0		ns	

Timing Waveforms

AC Test Input Waveform [Except CLK]

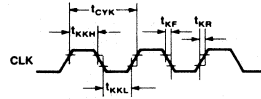


AC Output Test Points



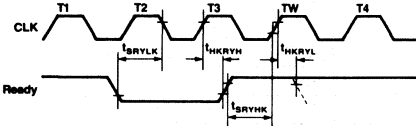
49 000247A

Clock Timing

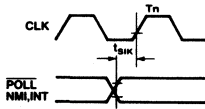


49 000248A

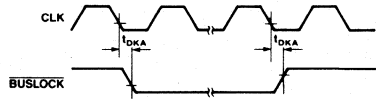
Wait (Ready) Timing



POLL, NMI, INT Input Timing

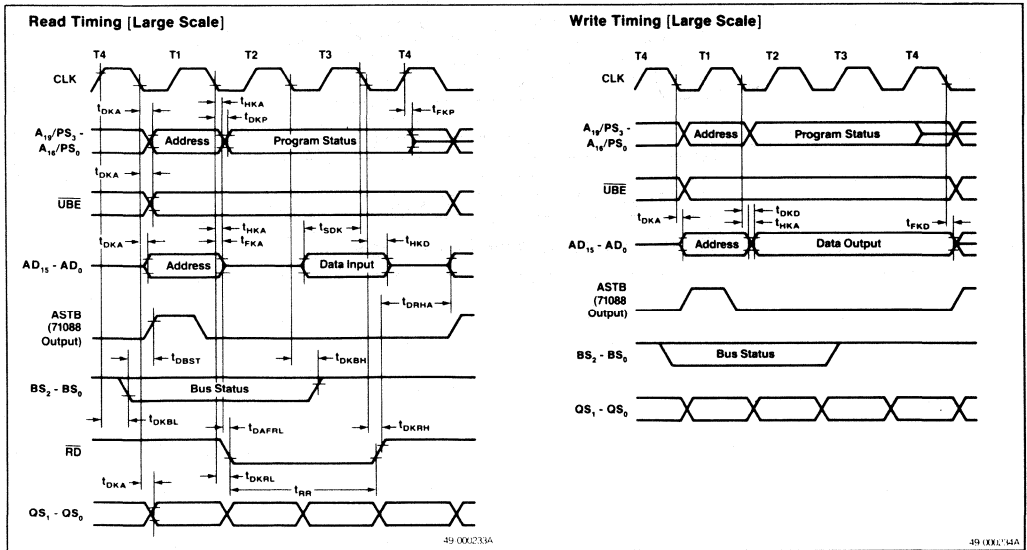
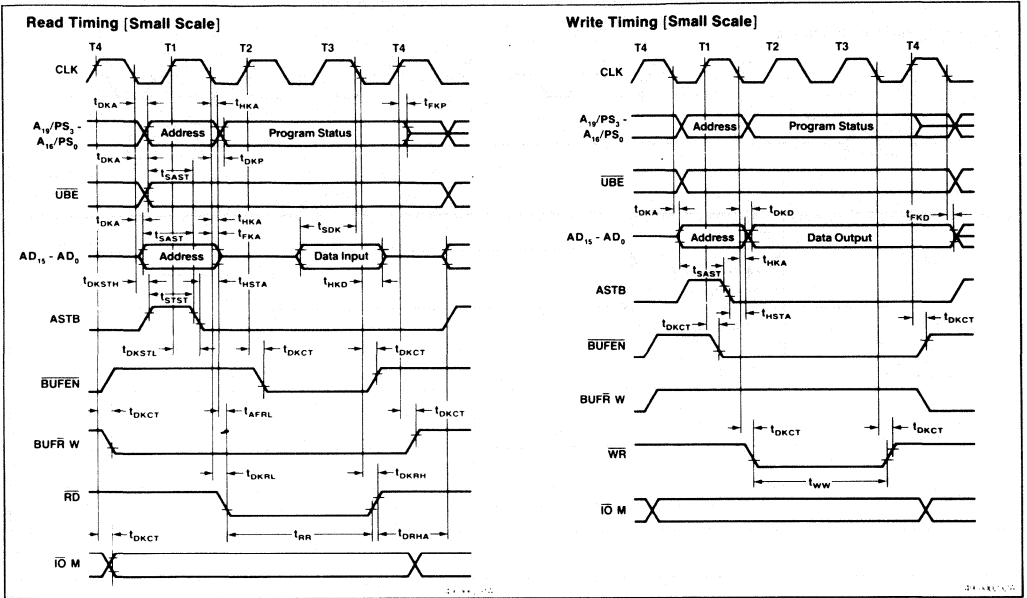


BUSLOCK Output Timing



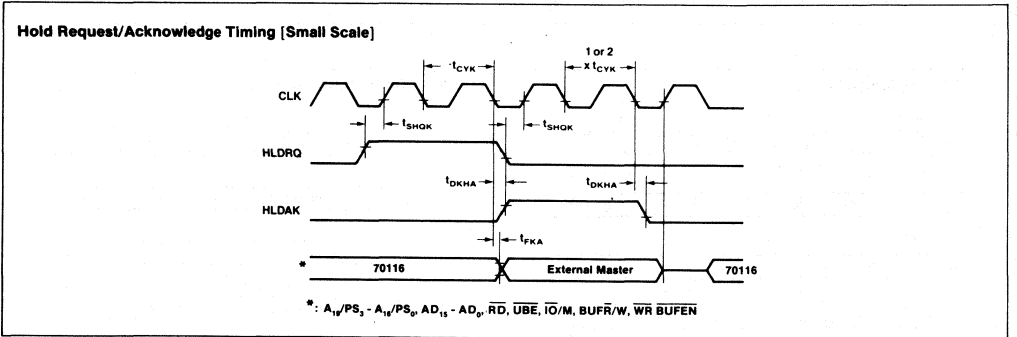
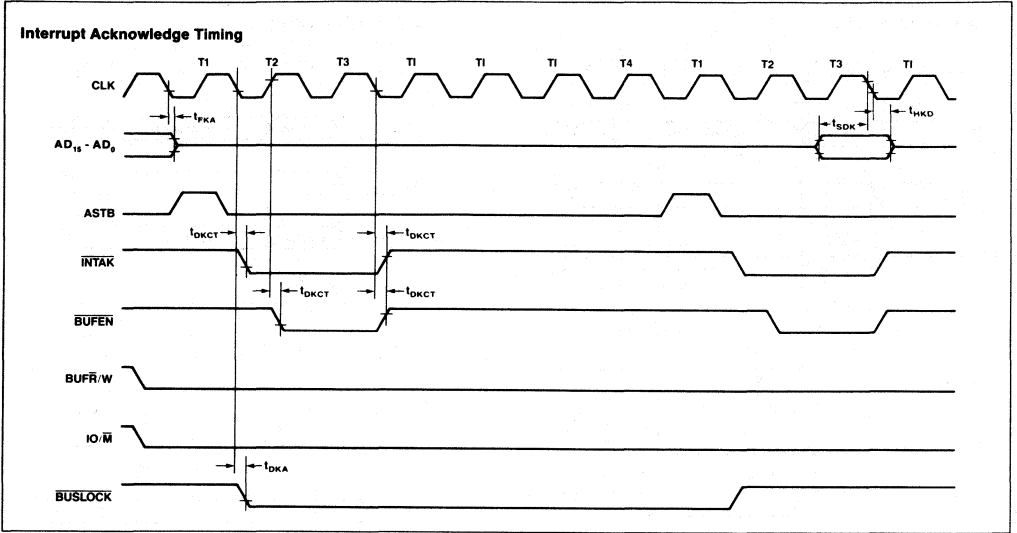
49 000249A

## Timing Waveforms (cont)

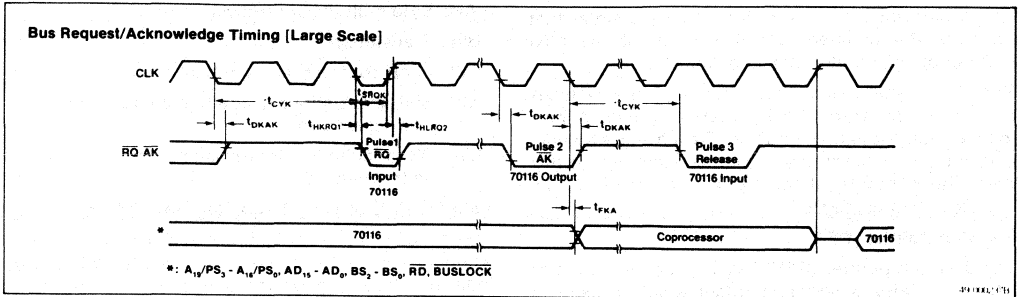




Timing Waveforms (cont)



## Timing Waveforms (cont)



### Register Configuration

#### Program Counter [PC]

The program counter is a 16-bit binary counter that contains the segment offset address of the next instruction which the EXU is to execute.

The PC increments each time the microprogram fetches an instruction from the instruction queue. A new location value is loaded into the PC each time a branch, call, return, or break instruction is executed. At this time, the contents of the PC are the same as the Prefetch Pointer (PFP).

#### Prefetch Pointer [PFP]

The prefetch pointer (PFP) is a 16-bit binary counter which contains a segment offset which is used to calculate a program memory address that the bus control unit (BCU) uses to prefetch the next word for the instruction queue. The contents of PFP are an offset from the PS (Program Segment) register.

The PFP is incremented each time the BCU prefetches an instruction from the program memory. A new location will be loaded into the PFP whenever a branch, call, return, or break instruction is executed. At that time the contents of the PFP will be the same as those of the PC (Program Counter).

#### Segment Registers [PS, SS, DS<sub>0</sub>, and DS<sub>1</sub>]

The memory addresses accessed by the μPD70116 are divided into 64K-byte logical segments. The starting (base) address of each segment is specified by a 16-bit segment register, and the offset from this starting address is specified by the contents of another register or by the effective address.

These are the four types of segment registers used.

Segment Register	Default Offset
PS (Program Segment)	PFP
SS (Stack Segment)	SP, effective address
DS <sub>0</sub> (Data Segment 0)	IX, effective address
DS <sub>1</sub> (Data Segment 1)	IY

#### General-Purpose Registers [AW, BW, CW, and DW]

There are four 16-bit general-purpose registers. Each one can be used as one 16-bit register or as two 8-bit registers by dividing them into their high and low bytes (AH, AL, BH, BL, CH, CL, DH, DL).

Each register is also used as a default register for processing specific instructions. The default assignments are:

AW: Word multiplication/division, word I/O, data conversion

AL: Byte multiplication/division, byte I/O, BCD rotation, data conversion, translation

AH: Byte multiplication/division

BW: Translation

CW: Loop control branch, repeat prefix

CL: Shift instructions, rotation instructions, BCD operations

DW: Word multiplication/division, indirect addressing I/O

#### Pointers [SP, BP] and Index Registers [IX, IY]

These registers serve as base pointers or index registers when accessing the memory using based addressing, indexed addressing, or based indexed addressing.

These registers can also be used for data transfer and arithmetic and logical operations in the same manner as the general-purpose registers. They cannot be used as 8-bit registers.

Also, each of these registers acts as a default register for specific operations. The default assignments are:

SP: Stack operations

IX: Block transfer (source), BCD string operations

IY: Block transfer (destination), BCD string operations

#### Program Status Word [PSW]

The program status word consists of the following six status and four control flags.

##### Status Flags

- V (Overflow)
- S (Sign)
- Z (Zero)
- AC (Auxiliary Carry)
- P (Parity)
- CY (Carry)

##### Control Flags

- MD (Mode)
- DIR (Direction)
- IE (Interrupt Enable)
- BRK (Break)

When the PSW is pushed on the stack, the word images of the various flags are as shown here.

PSW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	1	1	1	V	D	I	B	S	Z	O	A	O	P	I	C	
D																

The status flags are set and reset depending upon the result of each type of instruction executed.

Instructions are provided to set, reset, and complement the CY flag directly.

Other instructions set and reset the control flags and control the operation of the CPU.

## High-Speed Execution of Instructions

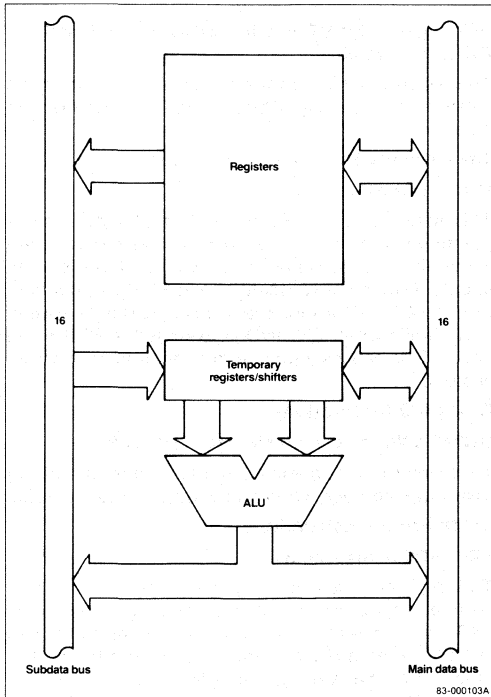
This section highlights the major architectural features that enhance the performance of the μPD70116.

- Dual data bus in EXU
- Effective address generator
- 16/32-bit temporary registers/shifters (TA, TB)
- 16-bit loop counter
- PC and PFP

### Dual Data Bus Method

To reduce the number of processing steps for instruction execution, the dual data bus method has been adopted for the μPD70116 (figure 1). The two data buses (the main data bus and the subdata bus) are both 16 bits wide. For addition/subtraction and logical and comparison operations, processing time has been speeded up some 30% over single-bus systems.

Figure 1. Dual Data Buses



### Example

ADD AW, BW ; AW ← AW + BW

#### Single Bus

Step 1 TA ← AW

Step 2 TB ← BW

Step 3 AW ← TA + TB

#### Dual Bus

TA ← AW, TB ← BW

AW ← TA + TB

### Effective Address Generator

This circuit (figure 2) performs high-speed processing to calculate effective addresses for accessing memory.

Calculating an effective address by the microprogramming method normally requires 5 to 12 clock cycles. This circuit requires only two clock cycles for addresses to be generated for any addressing mode. Thus, processing is several times faster.

### 16/32-Bit Temporary Registers/Shifters [TA, TB]

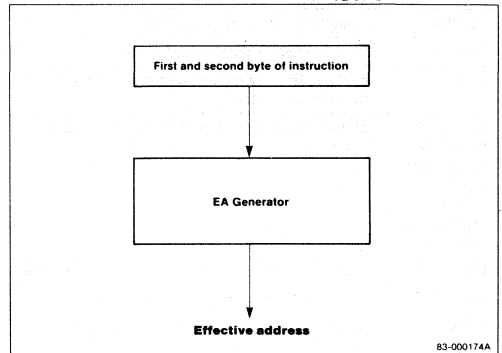
These 16-bit temporary registers/shifters (TA, TB) are provided for multiplication/division and shift/rotation instructions.

These circuits have decreased the execution time of multiplication/division instructions. In fact, these instructions can be executed about four times faster than with the microprogramming method.

TA + TB: 32-bit temporary register/shifter for multiplication and division instructions.

TB: 16-bit temporary register/shifter for shift/rotation instructions.

Figure 2. Effective Address Generator



83-000174A

Loop Counter [LC]

This counter is used to count the number of loops for a primitive block transfer instruction controlled by a repeat prefix instruction and the number of shifts that will be performed for a multiple bit shift/rotation instruction.

The processing performed for a multiple bit rotation of a register is shown below. The average speed is approximately doubled over the microprogram method.

Example

RORC AW, CL ; CL = 5

Microprogram method LC method

8 + (4 x 5) = 28 clocks 7 + 5 = 12 clocks

Program Counter and Prefetch Pointer [PC and PFP]

The μPD70116 microprocessor has a program counter, (PC) which addresses the program memory location of the instruction to be executed next, and a prefetch pointer(PFP), which addresses the program memory location to be accessed next. Both functions are provided in hardware. A time saving of several clocks is realized for branch, call, return, and break instruction execution, compared with microprocessors that have only one instruction pointer.

Enhanced Instructions

In addition to the μPD8088/86 instructions, the μPD70116 has the following enhanced instructions.

Instruction	Function
PUSH imm	Pushes immediate data onto stack
PUSH R	Pushes 8 general registers onto stack
POP R	Pops 8 general registers from stack
MUL imm	Executes 16-bit multiply of register or memory contents by immediate data
SHL imm8 SHR imm8 SHRA imm8 ROL imm8 ROR imm8 ROLC imm8 RORC imm8	Shifts/rotates register or memory by immediate value
CHKIND	Checks array index against designated boundaries
INM	Moves a string from an I/O port to memory
OUTM	Moves a string from memory to an I/O port
PREPARE	Allocates an area for a stack frame and copies previous frame pointers
DISPOSE	Frees the current stack frame on a procedure exit

Enhanced Stack Operation Instructions

PUSH imm

This instruction allows immediate data to be pushed onto the stack.

PUSH R/POP R

These instructions allow the contents of the eight general registers to be pushed onto or popped from the stack with a single instruction.

Enhanced Multiplication Instructions

MUL reg16, imm16/MUL mem16, imm16

These instructions allow the contents of a register or memory location to be multiplied by immediate data.

Enhanced Shift and Rotate Instructions

SHL reg, imm8/SHR reg, imm8/SHRA reg, imm8

These instructions allow the contents of a register to be shifted by the number of bits defined by the immediate data.

ROL reg, imm8/ROR reg, imm8/ROLC reg, imm8/RORC reg, imm8

These instructions allow the contents of a register to be rotated by the number of bits defined by the immediate data.

Check Array Boundary Instruction

CHKIND reg16, mem32

This instruction is used to verify that index values pointing to the elements of an array data structure are within the defined range. The lower limit of the array should be in memory location mem32, the upper limit in mem32 + 2. If the index value in reg16 is not between these limits when CHKIND is executed, a BRK 5 will occur. This causes a jump to the location in interrupt vector 5.

Block I/O Instructions

OUTM DW, src-block/INM dst-block, DW

These instructions are used to output or input a string to or from memory, when preceded by a repeat prefix.

Stack Frame Instructions

PREPARE imm16, imm8

This instruction is used to generate the stack frames required by block-structured languages, such as PASCAL and Ada. The stack frame consists of two areas. One area has a pointer that points to another frame which has variables that the current frame can access. The other is a local variable area for the current procedure.

### DISPOSE

This instruction releases the last stack frame generated by the PREPARE instruction. It returns the stack and base pointers to the values they had before the PREPARE instruction was used to call a procedure.

### Unique Instructions

In addition to the μPD8088/86 instructions and the enhanced instructions, the μPD70116 has the following unique instructions.

Instruction	Function
INS	Insert bit field
EXT	Extract bit field
ADD4S	Adds packed decimal strings
SUB4S	Subtracts one packed decimal string from another
CMP4S	Compares two packed decimal strings
ROL4	Rotates one BCD digit left through AL lower 4 bits
ROR4	Rotates one BCD digit right through AL lower 4 bits
TEST1	Tests a specified bit and sets/resets Z flag
NOT1	Inverts a specified bit
CLR1	Clears a specified bit
SET1	Sets a specified bit
REPC	Repeats next instruction until CY flag is cleared
REPNC	Repeats next instruction until CY flag is set
FPQ2	Additional floating point processor call

### Variable Length Bit Field Operation Instructions

This category has two instructions: INS (Insert Bit Field) and EXT (Extract Bit Field). These instructions are highly effective for computer graphics and high-level languages. They can, for example, be used for data structures such as packed arrays and record type data used in PASCAL.

#### INS reg8, reg8/INS reg8, imm4

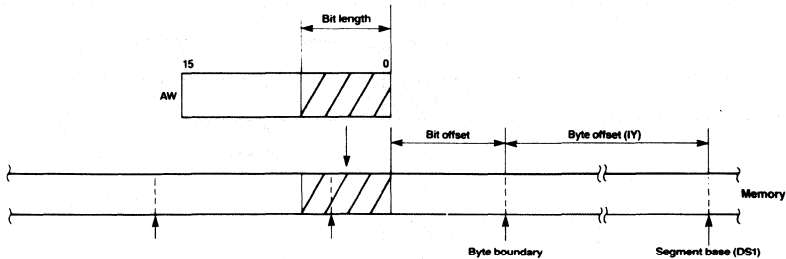
This instruction (figure 3) transfers low bits from the 16-bit AW register (the number of bits is specified by the second operand) to the memory location specified by the segment base (DS<sub>1</sub> register) plus the byte offset (IY register). The starting bit position within this byte is specified as an offset by the lower 4-bits of the first operand.

After each complete data transfer, the IY register and the register specified by the first operand are automatically updated to point to the next bit field.

Either immediate data or a register may specify the number of bits transferred (second operand). Because the maximum transferable bit length is 16-bits, only the lower 4-bits of the specified register (00H to 0FH) will be valid.

Bit field data may overlap the byte boundary of memory.

Figure 3. Bit Field Insertion



83-000106B

### EXT reg8, reg8/EXT reg8, imm4

This instruction (figure 4) loads the AW register the bit field data whose bit length is specified by the second operand of the instruction from the memory location that is specified by the DS0 segment register (segment base), the IX index register (byte offset), and the lower 4-bits of the first operand (bit offset).

After the transfer is complete, the IX register and the lower 4-bits of the first operand are automatically updated to point to the next bit field.

Either immediate data or a register may be specified for the second operand. Because the maximum transferable bit length is 16 bits, however, only the lower 4-bits of the specified register (0H to 0FH) will be valid.

Bit field data may overlap the byte boundary of memory.

### Packed BCD Operation Instructions

The instructions described here process packed BCD data either as strings (ADD4S, SUB4S, CMP4S) or byte-format operands (ROR4, ROL4). Packed BCD strings may be from 1 to 254 digits in length.

When the number of digits is even, the zero and carry flags will be set according to the result of the operation. When the number of digits is odd, the zero and carry flags may not be set correctly in this case, (CL = odd), the zero flag will not be set unless the upper 4 bits of the highest byte are all zero. The carry flag will not be set unless there is a carry out of the upper 4 bits of the highest byte. When CL is odd, the contents of the upper 4 bits of the highest byte of the result are undefined.

### ADD4S

This instruction adds the packed BCD string addressed by the IX index register to the packed BCD string addressed by the IY index register, and stores the result in the string addressed by the IY register. The length of the string (number of BCD digits) is specified by the CL register, and the result of the operation will affect the overflow flag (V), the carry flag (CY), and zero flag (Z).

BCD string (IY, CL) ← BCD string (IY, CL) + BCD string (IX, CL)

### SUB4S

This instruction subtracts the packed BCD string addressed by the IX index register from the packed BCD string addressed by the IY register, and stores the result in the string addressed by the IY register. The length of the string (number of BCD digits) is specified by the CL register, and the result of the operation will affect the overflow flag (V), the carry flag (CY), and zero flag (Z).

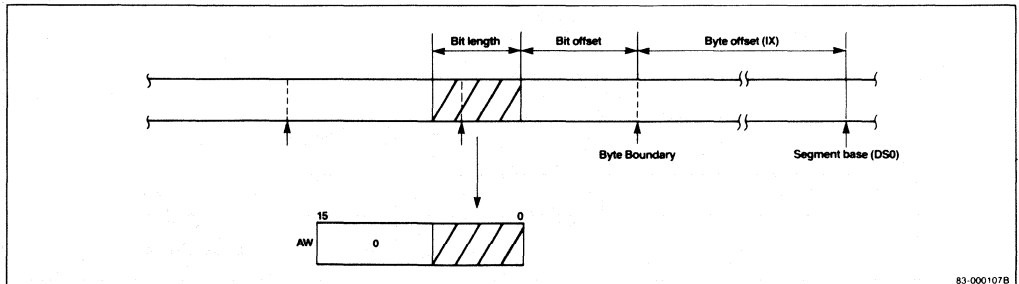
BCD string (IY, CL) ← BCD string (IY, CL) – BCD String (IX, CL)

### CMP4S

This instruction performs the same operation as SUB4S except that the result is not stored and only the overflow (V), carry flags (CY) and zero flag (Z) are affected.

BCD string (IY, CL) – BCD string (IX, CL)

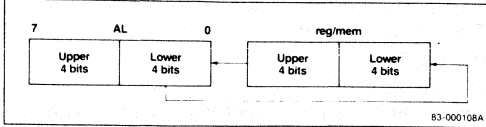
Figure 4. Bit Field Extraction



## ROL4

This instruction (figure 5) treats the byte data of the register or memory operand specified by the instruction as BCD data and uses the lower 4 bits of the AL register (AL<sub>L</sub>) to rotate that data one BCD digit to the left.

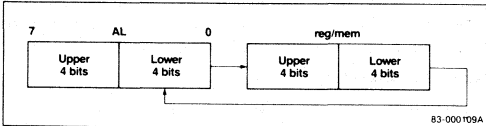
Figure 5. BCD Rotate Left (ROL4)



## ROR4

This instruction (figure 6) treats the byte data of the register or memory specified by the instruction as BCD data and uses the lower 4 bits of the AL register (AL<sub>L</sub>) to rotate that data one BCD digit to the right.

Figure 6. BCD Rotate Right (ROR4)



## Bit Manipulation Instructions

### TEST1

This instruction tests a specific bit in a register or memory location. If the bit is 1, the Z flag is reset to 0. If the bit is 0, the Z flag is set to 1.

### NOT1

This instruction inverts a specific bit in a register or memory location.

### CLR1

This instruction clears a specific bit in a register or memory location.

### SET1

This instruction sets a specific bit in a register or memory location.

## Repeat Prefix Instructions

### REPC

This instruction causes the μPD70116 to repeat the following primitive block transfer instruction until the CY flag becomes cleared or the CW register becomes zero.

## REPNC

This instruction causes the μPD70116 to repeat the following primitive block transfer instruction until the CY flag becomes set or the CW register becomes zero.

## Floating Point Instruction

### FPO2

This instruction is in addition to the μPD8088/86 floating point instruction, FPO1. These instructions are covered in a later section.

## Mode Operation Instruction

The μPD70116 has two operating modes (figure 7). One is the native mode which executes μPD8088/86, enhanced and unique instructions. The other is the 8080 emulation mode in which the instruction set of the μPD8080AF is emulated. A mode flag (MD) is provided to select between these two modes. Native mode is selected when MD is 1 and emulation mode when MD is 0. MD is set and reset, directly and indirectly, by executing the mode manipulation instructions.

Two instructions are provided to switch operation from the native mode to the emulation mode and back.

BRKEM (Break for Emulation)

RETEM (Return from Emulation)

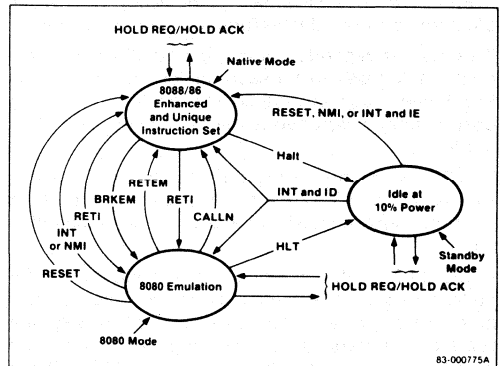
Two instructions are used to switch from the emulation mode to the native mode and back.

CALLN (Call Native Routine)

RETI (Return from Interrupt)

The system will return from the 8080 emulation mode to the native mode when the RESET signal is present, or when an external interrupt (NMI or INT) is present.

Figure 7. V30 Modes





**BRKEM imm8**

This is the basic instruction used to start the 8080 emulation mode. This instruction operates exactly the same as the BRK instruction, except that BRKEM resets the mode flag (MD) to 0. PSW, PS, and PC are saved to the stack. MD is then reset and the interrupt vector specified by the operand imm8 of this command is loaded into PS and PC.

The instruction codes of the interrupt processing routine jumped to are then fetched. Then the CPU executes these codes as μPD8080AF instructions.

In 8080 emulation mode, registers and flags of the μPD8080AF are performed by the following registers and flags of the μPD70116.

	μPD8080AF	μPD70116
<b>Registers:</b>	A	AL
	B	CH
	C	CL
	D	DH
	E	DL
	H	BH
	L	BL
	SP	BP
	PC	PC
<b>Flags:</b>	C	CY
	Z	Z
	S	S
	P	P
	AC	AC

In the native mode, SP is used for the stack pointer. In the 8080 emulation mode this function is performed by BP.

This use of independent stack pointers allows independent stack areas to be secured for each mode and keeps the stack of one of the modes from being destroyed by an erroneous stack operation in the other mode.

The SP, IX, IY and AH registers and the four segment registers (PS, SS, DS<sub>0</sub>, and DS<sub>1</sub>) used in the native mode are not affected by operations in 8080 emulation mode.

In the 8080 emulation mode, the segment register for instructions is determined by the PS register (set automatically by the interrupt vector) and the segment register for data is the DS<sub>0</sub> register (set by the programmer immediately before the 8080 emulation mode is entered).

It is prohibited to nest BRKEM instructions.

**RETEM [no operand]**

When RETEM is executed in 8080 emulation mode (interpreted by the CPU as a μPD8080AF instruction), the CPU restores PS, PC, and PSW (as it would when returning from an interrupt processing routine), and returns to the native mode. At the same time, the contents of the mode flag (MD) which was saved to the stack by the BRKEM instruction, is restored to MD = 1. The CPU is set to the native mode.

**CALLN imm8**

This instruction makes it possible to call the native mode subroutines from the 8080 emulation mode. To return from subroutine to the 8080 emulation mode, the RETI instruction is used.

The processing performed when this instruction is executed in the 8080 emulation mode (it is interpreted by the CPU as μPD8080AF instruction), is similar to that performed when a BRK instruction is executed in the native mode. The imm8 operand specifies an interrupt vector type. The contents of PS, PC, and PSW are pushed on the stack and an MD flag value of 0 is saved. The mode flag is set to 1 and the interrupt vector specified by the operand is loaded into PS and PC.

**RETI [no operand]**

This is a general-purpose instruction used to return from interrupt routines entered by the BRK instruction or by an external interrupt in the native mode. When this instruction is executed at the end of a subroutine entered by the execution of the CALLN instruction, the operation that restores PS, PC, and PSW is exactly the same as the native mode execution. When PSW is restored, however, the 8080 emulation mode value of the mode flag (MD) is restored, the CPU is set in emulation mode, and all subsequent instructions are interpreted and executed as μPD8080AF instructions.

RETI is also used to return from an interrupt procedure initiated by an NMI or INT interrupt in the emulation mode.

**Floating Point Operation Chip Instructions**

- FPO1 fp-op, mem
- FPO2 fp-op, mem

These instructions are used for the external floating point processor. The floating point operation is passed to the floating point processor when the CPU fetches one of these instructions. From this point the CPU performs only the necessary auxiliary processing (effective address calculation, generation of physical addresses, and start-up of the memory read cycle).

The floating point processor always monitors the instructions fetched by the CPU. When it interprets one as an instruction to itself, it performs the appropriate processing. At this time, the floating point processor chip uses either the address alone or both the address and read data of the memory read cycle executed by the CPU. This difference in the data used depends on which of these instructions is executed.

**Note:** During the memory read cycle initiated by the CPU for FPO1 or FPO2 execution, the CPU does not accept any read data on the data bus from memory. Although the CPU generates the memory address, the data is used by the floating point processor.

## Interrupt Operation

The interrupts used in the μPD70116 can be divided into two types: interrupts generated by external interrupt requests and interrupts generated by software processing. These are the classifications.

### External interrupts

- (a) NMI input (nonmaskable)
- (b) INT input (maskable)

### Software processing

As the result of instruction execution

- When a divide error occurs during execution of the DIV or DIVU instruction
- When a memory-boundary-over error is detected by the CHKIND instruction

Conditional break instruction

- When V = 1 during execution of the BRKV instruction

Unconditional break instructions

- 1-byte break instruction: BRK3
- 2-byte break instruction: BRK imm8

Flag processing

- When stack operations are used to set the BRK flag

8080 Emulation mode instructions

- BRKEM imm8
- CALLN imm8

Starting addresses for interrupt processing routines are either determined automatically by a single location of the interrupt vector table or selected each time interrupt processing is entered.

The interrupt vector table is shown in figure 8. The table uses 1K bytes of memory addresses 000H to 3FFH and can store starting address data for a maximum of 256 vectors (4 bytes per vector).

The corresponding interrupt sources for vectors 0 to 5 are predetermined and vectors 6 to 31 are reserved. These vectors consequently cannot be used for general applications.

The BRKEM instruction and CALLN instruction (in the emulation mode) and the INT input are available for general applications for vectors 32 to 255.

A single interrupt vector is made up of 4 bytes (figure 9). The 2 bytes in the low addresses of memory are loaded into PC as the offset, and the high 2 bytes are loaded into PS as the base address. The bytes are combined in reverse order. The lower-order bytes in the vector become the most significant bytes in the PC and PS, and the higher-order bytes become the least significant bytes.

**Figure 8. Interrupt Vector Table**

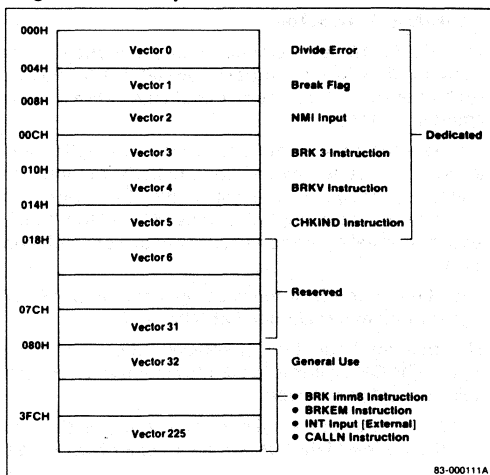
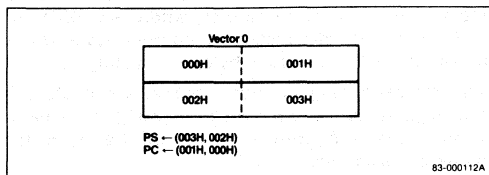


Figure 9. Interrupt Vector 0



Based on this format, the contents of each vector should be initialized at the beginning of the program.

The basic steps to jump to an interrupt processing routine are now shown.

- (SP - 1, SP - 2) ← PSW
- (SP - 3, SP - 4) ← PS
- (SP - 5, SP - 6) ← PC
- SP ← SP - 6
- IE ← 0, BRK ← 0, MD ← 1
- PS ← vector high bytes
- PC ← vector low bytes

### Standby Function

The μPD70116 has a standby mode to reduce power consumption during program wait states. This mode is set by the HALT instruction in both the native and the emulation mode.

In the standby mode, the internal clock is supplied only to those circuits related to functions required to release this mode and bus hold control functions. As a result, power consumption can be reduced to 1/10 the level of normal operation in either native or emulation mode.

The standby mode is released by inputting a RESET signal or an external interrupt (NMI, INT).

The bus hold function is effective during standby mode. The CPU returns to standby mode when the bus hold request is removed.

During standby mode, all control outputs are disabled and the address/data bus will be at either high or low levels.

## Instruction Set

### Symbols

Preceding the instruction set, several tables explain symbols, abbreviations, and codes.

### Clocks

In the Clocks column of the instruction set, the numbers cover these operations: instruction decoding, effective address calculation, operand fetch, and instruction execution.

Clock timings assume the instruction has been pre-fetched and is present in the six-byte instruction queue. Otherwise, add four clocks for each pair of bytes not present.

Add four clocks to the numbers given for each word transfer to an odd address.

For instructions that reference memory operands, the number on the left side of the slash (/) is for byte operands and the number on the right side is for word operands.

For conditional control transfer or branch instructions, the number on the left side of the slash is applicable if the transfer or branch takes place. The number on the right side is applicable if it does not take place.

If a range of numbers is given, the execution time depends on the operands involved.

### Symbols

Symbol	Meaning
acc	Accumulator (AW or AL)
disp	Displacement (8 or 16 bits)
dmem	Direct memory address
dst	Destination operand or address
ext-disp8	16-bit displacement (sign-extension byte + 8-bit displacement)
far_label	Label within a different program segment
far_proc	Procedure within a different program segment
fp_op	Floating point instruction operation
imm	8- or 16-bit immediate operand

### Symbols (cont)

Symbol	Meaning
imm3/4	3/4-bit immediate bit offset
imm8	8-bit immediate operand
imm16	16-bit immediate operand
mem	Memory field (000 to 111); 8- or 16-bit memory location
mem8	8-bit memory location
mem16	16-bit memory location
mem32	32-bit memory location
memptr16	Word containing the destination address within the current segment
memptr32	Double word containing a destination address in another segment
mod	Mode field (00 to 10)
near_label	Label within the current segment
near_proc	Procedure within the current segment
offset	Immediate offset data (16 bits)
pop_value	Number of bytes to discard from the stack
reg	Register field (000 to 111); 8- or 16-bit general-purpose register
reg8	8-bit general-purpose register
reg16	16-bit general-purpose register
regptr	16-bit register containing a destination address within the current segment
regptr16	Register containing a destination address within the current segment
seg	Immediate segment data (16 bits)
short_label	Label between -128 and +127 bytes from the end of the current instruction
sr	Segment register
src	Source operand or address
temp	Temporary register (8/16/32 bits)
tmpcy	Temporary carry flag (1 bit)
AC	Auxiliary carry flag
AH	Accumulator (high byte)
AL	Accumulator (low byte)
AND	Logical product
AW	Accumulator (16 bits)
BH	BW register (high byte)
BL	BW register (low byte)
BP	BP register
BRK	Break flag
BW	BW register (16 bits)
CH	CW register (high byte)
CL	CW register (low byte)

Symbol	Meaning
CW	CW register (16 bits)
CY	Carry flag
DH	DW register (high byte)
DIR	Direction flag
DL	DW register (low byte)
DS0	Data segment 0 register (16 bits)
DS1	Data segment 1 register (16 bits)
DW	DW register (16 bits)
IE	Interrupt enable flag
IX	Index register (source) (16 bits)
IY	Index register (destination) (16 bits)
MD	Mode flag
OR	Logical sum
P	Parity flag
PC	Program counter (16 bits)
PS	Program segment register (16 bits)
PSW	Program status word (16 bits)
R	Register set
S	Sign extend operand field S = 0 No sign extension S = 1 Sign extend immediate byte operand
S	Sign flag
SP	Stack pointer (16 bits)
SS	Stack segment register (16 bits)
V	Overflow flag
W	Word/byte field (0 to 1)
X, XXX, YYY, ZZZ	Data to identify the instruction code of the external floating point arithmetic chip
XOR	Exclusive logical sum
XXH	Two-digit hexadecimal value
XXXXH	Four-digit hexadecimal value
Z	Zero flag
( )	Values in parentheses are memory contents
←	Transfer direction
+	Addition
-	Subtraction
x	Multiplication
÷	Division
%	Modulo

### Flag Operations

Symbol	Meaning
(blank)	No change
0	Cleared to 0
1	Set to 1
x	Set or cleared according to result
u	Undefined
R	Restored to previous state

### Memory Addressing Modes

mem	mod = 00	mod = 01	mod = 10
000	BW + IX	BW + IX + disp8	BW + IX + disp16
001	BW + IY	BW + IY + disp8	BW + IY + disp16
010	BP + IX	BP + IX + disp8	BP + IX + disp16
011	BP + IY	BP + IY + disp8	BP + IY + disp16
100	IX	IX + disp8	IX + disp16
101	IY	IY + disp8	IY + disp16
110	Direct	BP + disp8	BP + disp16
111	BW	BW + disp8	BW + disp16

### Register Selection (mod = 11)

reg	W = 0	W = 1
000	AL	AW
001	CL	CW
010	DL	DW
011	BL	BW
100	AH	SP
101	CH	BP
110	DH	IX
111	BH	IY

### Segment Register Selection

sr	Segment Register
00	DS1
01	PS
10	SS
11	DS0

### Instruction Set

Mnemonic	Operand	Opcode										Clocks	Bytes	Flags								
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V
MOV	reg, reg	1	0	0	0	1	0	1	W	1	1	reg	reg	2	2							
	mem, reg	1	0	0	0	1	0	0	W	mod	reg	mem	9	2-4								
	reg, mem	1	0	0	0	1	0	1	W	mod	reg	mem	11	2-4								
	mem, imm	1	1	0	0	0	1	1	W	mod	reg	mem	11	3-6								
	reg, imm	1	0	1	1	W	reg						4	2-3								
	acc, dmem	1	0	1	0	0	0	0	W					10	3							
	dmem, acc	1	0	1	0	0	0	1	W					9	3							
	sr, reg16	1	0	0	0	1	1	1	0	1	1	0	sr	reg	2	2						
	sr, mem16	1	0	0	0	1	1	1	0	mod	0	sr	mem	11	2-4							
	reg16, sr	1	0	0	0	1	1	0	0	1	1	0	sr	reg	2	2						
	mem16, sr	1	0	0	0	1	1	0	0	mod	0	sr	mem	10	2-4							
LDEA	DS0, reg16, mem32	1	1	0	0	0	1	0	1	mod	reg	mem	18	2-4								
	DS1, reg16, mem32	1	1	0	0	0	1	0	0	mod	reg	mem	18	2-4								
	AH, PSW	1	0	0	1	1	1	1	1					2	1							
	PSW, AH	1	0	0	1	1	1	1	0					3	1	x	x		x	x	x	
	reg16, mem16	1	0	0	0	1	0	1	0	mod	reg	mem	4	2-4								
	src_table	1	1	0	1	0	1	1	1					9	1							
	XCH	reg, reg	1	0	0	0	0	1	1	W	1	1	reg	reg	3	2						
mem, reg		1	0	0	0	0	1	1	W	mod	reg	mem	16	2-4								
AW, reg16		1	0	0	1	0	reg					3	1									

### Instruction Set (cont)

Mnemonic	Operand	Opcode																Clocks	Bytes	Flags				
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S
<b>Repeat Prefixes</b>																								
REPC		0	1	1	0	0	1	0	1									2	1					
REPNC		0	1	1	0	0	1	0	0									2	1					
REP		1	1	1	1	0	0	1	1									2	1					
REPE																								
REPZ																								
REPNE		1	1	1	1	0	0	1	0									2	1					
REPNZ																								
<b>Block Transfer Instructions</b>																								
MOVBK	dst, src	1	0	1	0	0	1	0	W									11 + 8n	1					
CMPBK	dst, src	1	0	1	0	0	1	1	W									7 + 14n	1	x	x	x	x	
CMPM	dst	1	0	1	0	1	1	1	W									7 + 10n	1	x	x	x	x	
LDM	src	1	0	1	0	1	1	0	W									7 + 9n	1					
STM	dst	1	0	1	0	1	0	1	W									7 + 4n	1					
n = number of transfers																								
<b>I/O Instructions</b>																								
IN	acc, imm8	1	1	1	0	0	1	0	W									9	2					
	acc, DW	1	1	1	0	1	1	0	W									8	1					
OUT	imm8, acc	1	1	1	0	0	1	1	W									8	2					
	DW, acc	1	1	1	0	1	1	1	W									8	1					
INM	dst, DW	0	1	1	0	1	1	0	W									9 + 8n	1					
OUTM	DW, src	0	1	1	0	1	1	1	W									9 + 8n	1					
n = number of transfers																								
<b>BCD Instructions</b>																								
ADJBA		0	0	1	1	0	1	1	1									3	1	x	x	u	u	
ADJ4A		0	0	1	0	0	1	1	1									3	1	x	x	u	x	
ADJBS		0	0	1	1	1	1	1	1									7	1	x	x	u	u	
ADJ4S		0	0	1	0	1	1	1	1									7	1	x	x	u	x	
ADD4S	dst, src	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	7 + 19n	2	u	x	u	u		
SUB4S	dst, src	0	0	0	0	1	1	1	0	0	1	0	0	0	1	0	7 + 19n	2	u	x	u	u		
CMP4S	dst, src	0	0	0	0	1	1	1	0	0	1	0	0	1	1	0	7 + 19n	2	u	x	u	u		
ROL4	reg8	0	0	0	0	1	1	1	0	0	1	0	1	0	0	0	25	3						
	mem8	1	1	0	0	0	reg																	
		0	0	0	0	1	1	1	0	0	1	0	1	0	0	0	28	3-5						
		mod	0	0	0	mem																		
ROR4	reg8	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	29	3						
	mem8	1	1	0	0	0	reg																	
		0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	33	3-5						
		mod	0	0	0	mem																		
n = number of BCD digits divided by 2																								

**Instruction Set (cont)**

Mnemonic	Operand	Opcode											Clocks	Bytes	Flags										
		7	6	5	4	3	2	1	0	7	6	5			4	3	2	1	0	AC	CY	V	P	S	Z
<b>Data Type Conversion Instructions</b>																									
CVTBD		1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	15	2		u	u	u	x	x
CVTDB		1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	7	2		u	u	u	x	x
CVTBW		1	0	0	1	1	0	0	0									2	1						
CVTWL		1	0	0	1	1	0	0	1									4-5	1						
<b>Arithmetic Instructions</b>																									
ADD	reg, reg	0	0	0	0	0	0	1	W	1	1	reg	reg	2	2		x	x	x	x	x	x			
	mem, reg	0	0	0	0	0	0	0	W	mod	reg	mem	16	2-4		x	x	x	x	x	x				
	reg, mem	0	0	0	0	0	0	1	W	mod	reg	mem	11	2-4		x	x	x	x	x	x				
	reg, imm	1	0	0	0	0	0	S	W	1	1	0	0	0	reg	4	3-4		x	x	x	x	x		
	mem, imm	1	0	0	0	0	0	S	W	mod	0	0	0	mem	18	3-6		x	x	x	x	x			
	acc, imm	0	0	0	0	0	1	0	W						4	2-3		x	x	x	x	x			
ADDC	reg, reg	0	0	0	1	0	0	1	W	1	1	reg	reg	2	2		x	x	x	x	x	x			
	mem, reg	0	0	0	1	0	0	0	W	mod	reg	mem	16	2-4		x	x	x	x	x	x				
	reg, mem	0	0	0	1	0	0	1	W	mod	reg	mem	11	2-4		x	x	x	x	x	x				
	reg, imm	1	0	0	0	0	0	S	W	1	1	0	1	0	reg	4	3-4		x	x	x	x	x		
	mem, imm	1	0	0	0	0	0	S	W	mod	0	1	0	mem	18	3-6		x	x	x	x	x			
	acc, imm	0	0	0	1	0	1	0	W						4	2-3		x	x	x	x	x			
SUB	reg, reg	0	0	1	0	1	0	1	W	1	1	reg	reg	2	2		x	x	x	x	x	x			
	mem, reg	0	0	1	0	1	0	0	W	mod	reg	mem	16	2-4		x	x	x	x	x	x				
	reg, mem	0	0	1	0	1	0	1	W	mod	reg	mem	11	2-4		x	x	x	x	x	x				
	reg, imm	1	0	0	0	0	0	S	W	1	1	1	0	1	reg	4	3-4		x	x	x	x	x		
	mem, imm	1	0	0	0	0	0	S	W	mod	1	0	1	mem	18	3-6		x	x	x	x	x			
	acc, imm	0	0	1	0	1	1	0	W						4	2-3		x	x	x	x	x			
SUBC	reg, reg	0	0	0	1	1	0	1	W	1	1	reg	reg	2	2		x	x	x	x	x	x			
	mem, reg	0	0	0	1	1	0	0	W	mod	reg	mem	16	2-4		x	x	x	x	x	x				
	reg, mem	0	0	0	1	1	0	1	W	mod	reg	mem	11	2-4		x	x	x	x	x	x				
	reg, imm	1	0	0	0	0	0	S	W	1	1	0	1	1	reg	4	3-4		x	x	x	x	x		
	mem, imm	1	0	0	0	0	0	S	W	mod	0	1	1	mem	18	3-6		x	x	x	x	x			
	acc, imm	0	0	0	1	1	1	0	W						4	2-3		x	x	x	x	x			
INC	reg8	1	1	1	1	1	1	1	0	1	1	0	0	0	reg	2	2		x	x	x	x	x		
	mem	1	1	1	1	1	1	1	W	mod	0	0	0	mem	16	2-4		x	x	x	x	x			
	reg16	0	1	0	0	0	reg							2	1		x	x	x	x	x				
DEC	reg8	1	1	1	1	1	1	1	0	1	1	0	0	1	reg	2	2		x	x	x	x	x		
	mem	1	1	1	1	1	1	1	W	mod	0	0	1	mem	16	2-4		x	x	x	x	x			
	reg16	0	1	0	0	1	reg							2	1		x	x	x	x	x				
MULU	reg	1	1	1	1	0	1	1	W	1	1	1	0	0	reg	21-30	2		u	x	x	u	u		
	mem	1	1	1	1	0	1	1	W	mod	1	0	0	mem	27-36	2-4		u	x	x	u	u			

## Instruction Set (cont)

Mnemonic	Operand	Opcode										Clocks	Bytes	Flags									
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P
<b>Arithmetic Instructions (cont)</b>																							
MUL	reg	1	1	1	1	0	1	1	W	1	1	1	0	1	reg	33-47	2	u	x	x	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	0	1	mem	39-53	2-4	u	x	x	u	u	u	
	reg16,reg16,imm8	0	1	1	0	1	0	1	1	1	1	reg	reg	28-34	3	u	x	x	u	u	u		
	reg16,mem16,imm8	0	1	1	0	1	0	1	1	mod	reg	mem	34-40	3-5	u	x	x	u	u	u			
	reg16,reg16,imm16	0	1	1	0	1	0	0	1	1	1	reg	reg	36-42	4	u	x	x	u	u	u		
	reg16,mem16,imm16	0	1	1	0	1	0	0	1	mod	reg	mem	46-48	4-6	u	x	x	u	u	u			
DIVU	reg	1	1	1	1	0	1	1	W	1	1	1	1	0	reg	19-25	2	u	u	u	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	1	0	mem	25-31	2-4	u	u	u	u	u	u	
DIV	reg	1	1	1	1	0	1	1	W	1	1	1	1	1	reg	29-43	2	u	u	u	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	1	1	mem	35-49	2-4	u	u	u	u	u	u	
<b>Comparison Instructions</b>																							
CMP	reg, reg	0	0	1	1	1	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x		
	mem, reg	0	0	1	1	1	0	0	W	mod	reg	mem	11	2-4	x	x	x	x	x	x			
	reg, mem	0	0	1	1	1	0	1	W	mod	reg	mem	11	2-4	x	x	x	x	x	x			
	reg, imm	1	0	0	0	0	0	S	W	1	1	1	1	1	reg	4	3-4	x	x	x	x	x	x
	mem, imm	1	0	0	0	0	0	S	W	mod	1	1	1	mem	13	3-6	x	x	x	x	x	x	
	acc, imm	0	0	1	1	1	1	0	W					4	2-3	x	x	x	x	x	x		
<b>Logical Instructions</b>																							
NOT	reg	1	1	1	1	0	1	1	W	1	1	0	1	0	reg	2	2						
	mem	1	1	1	1	0	1	1	W	mod	0	1	0	mem	16	2-4							
NEG	reg	1	1	1	1	0	1	1	W	1	1	0	1	1	reg	2	2	x	x	x	x	x	x
	mem	1	1	1	1	0	1	1	W	mod	0	1	1	mem	16	2-4	x	x	x	x	x	x	
TEST	reg, reg	1	0	0	0	0	1	0	W	1	1	reg	reg	2	2	u	0	0	x	x	x		
	mem, reg	1	0	0	0	0	1	0	W	mod	reg	mem	10	2-4	u	0	0	x	x	x			
	reg, imm	1	1	1	1	0	1	1	W	1	1	0	0	0	reg	4	3-4	u	0	0	x	x	x
	mem, imm	1	1	1	1	0	1	1	W	mod	0	0	0	mem	11	3-6	u	0	0	x	x	x	
	acc, imm	1	0	1	0	1	0	0	W					4	2-3	u	0	0	x	x	x		
AND	reg, reg	0	0	1	0	0	0	1	W	1	1	reg	reg	2	2	u	0	0	x	x	x		
	mem, reg	0	0	1	0	0	0	0	W	mod	reg	mem	16	2-4	u	0	0	x	x	x			
	reg, mem	0	0	1	0	0	0	1	W	mod	reg	mem	11	2-4	u	0	0	x	x	x			
	reg, imm	1	0	0	0	0	0	0	W	1	1	1	0	0	reg	4	3-4	u	0	0	x	x	x
	mem, imm	1	0	0	0	0	0	0	W	mod	1	0	0	mem	18	3-6	u	0	0	x	x	x	
	acc, imm	0	0	1	0	0	1	0	W					4	2-3	u	0	0	x	x	x		
OR	reg, reg	0	0	0	0	1	0	1	W	1	1	reg	reg	2	2	u	0	0	x	x	x		
	mem, reg	0	0	0	0	1	0	0	W	mod	reg	mem	16	2-4	u	0	0	x	x	x			
	reg, mem	0	0	0	0	1	0	1	W	mod	reg	mem	11	2-4	u	0	0	x	x	x			
	reg, imm	1	0	0	0	0	0	0	W	1	1	0	0	1	reg	4	3-4	u	0	0	x	x	x
	mem, imm	1	0	0	0	0	0	0	W	mod	0	0	1	mem	18	3-6	u	0	0	x	x	x	
	acc, imm	0	0	1	0	0	1	0	W					4	2-3	u	0	0	x	x	x		



**Instruction Set (cont)**

Mnemonic	Operand	Opcode											Clocks	Bytes	Flags										
		7	6	5	4	3	2	1	0	7	6	5			4	3	2	1	0	AC	CY	V	P	S	Z
<b>Logical Instructions (cont)</b>																									
XOR	reg, reg	0	0	1	1	0	0	1	W	1	1	reg	reg	2	2	u	0	0	x	x	x				
	mem, reg	0	0	1	1	0	0	0	W	mod	reg	mem	16	2-4	u	0	0	x	x	x					
	reg, mem	0	0	1	1	0	0	1	W	mod	reg	mem	11	2-4	u	0	0	x	x	x					
	reg, imm	1	0	0	0	0	0	0	W	1	1	1	0	reg	4	3-4	u	0	0	x	x	x			
	mem, imm	1	0	0	0	0	0	0	W	mod	1	1	0	mem	18	3-6	u	0	0	x	x	x			
	acc, imm	0	0	1	0	0	1	0	W					4	2-3	u	0	0	x	x	x				
<b>Bit Manipulation Instructions</b>																									
INS	reg8, reg8	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	1	31-117	3						
		1	1	reg	reg																				
	reg8, imm8	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	1	31-117	4						
		1	1	0	0	0	reg																		
EXT	reg8, reg8	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1	26-55	3						
		1	1	reg	reg																				
	reg8, imm8	0	0	0	0	1	1	1	1	0	0	1	1	1	0	1	1	26-55	4						
		1	1	0	0	0	reg																		
TEST1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	W	3	3	u	0	0	u	u	x
		1	1	0	0	0	reg																		
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	W	12	3-5	u	0	0	u	u	x
		mod	0	0	0	mem																			
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0	W	4	4	u	0	0	u	u	x
		1	1	0	0	0	reg																		
	mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0	W	13	4-6	u	0	0	u	u	x
		mod	0	0	0	mem																			
SET1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	0	W	4	3						
		1	1	0	0	0	reg																		
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	0	W	13	3-5						
		mod	0	0	0	mem																			
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0	W	5	4						
		1	1	0	0	0	reg																		
	mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0	W	14	4-6						
		mod	0	0	0	mem																			
	CY	1	1	1	1	1	0	0	1								2	1			1				
	DIR	1	1	1	1	1	1	0	1								2	1							
CLR1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	W	5	3						
		1	1	0	0	0	reg																		
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	W	14	3-5						
		mod	0	0	0	mem																			
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	W	6	4						
		1	1	0	0	0	reg																		
	mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	W	15	4-6						
		mod	0	0	0	mem																			
	CY	1	1	1	1	1	0	0	0								2	1			0				
	DIR	1	1	1	1	1	1	0	0								2	1							

## Instruction Set (cont)

Mnemonic	Operands	Opcode															Clocks	Bytes	Flags				
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1			0	AC	CY	V	P
<b>Bit Manipulation Instructions (cont)</b>																							
NOT1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	1	W	4	3				
		1	1	0	0	0			reg														
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	1	W	18	3-5				
		mod	0	0	0	0			mem														
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	W	5	4				
	1	1	0	0	0			reg															
mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	W	19	4-6					
	mod	0	0	0	0			mem															
CY		1	1	1	1	0	1	0	1								2	1			x		
<b>Shift/Rotate Instructions</b>																							
SHL	reg, 1	1	1	0	1	0	0	0	W	1	1	1	0	0		reg	2	2	u	x	x	x	x
	mem, 1	1	1	0	1	0	0	0	W	mod	1	0	0		mem	16	2-4	u	x	x	x	x	
	reg, CL	1	1	0	1	0	0	1	W	1	1	1	0	0	reg	7 + n	2	u	x	u	x	x	
	mem, CL	1	1	0	1	0	0	1	W	mod	1	0	0		mem	19 + n	2-4	u	x	u	x	x	
	reg, imm8	1	1	0	0	0	0	0	W	1	1	1	0	0	reg	7 + n	3	u	x	u	x	x	
	mem, imm8	1	1	0	0	0	0	0	W	mod	1	0	0		mem	19 + n	3-5	u	x	u	x	x	
SHR	reg, 1	1	1	0	1	0	0	0	W	1	1	1	0	1	reg	2	2	u	x	x	x	x	
	mem, 1	1	1	0	1	0	0	0	W	mod	1	0	1	mem	16	2-4	u	x	x	x	x		
	reg, CL	1	1	0	1	0	0	1	W	1	1	1	0	1	reg	7 + n	2	u	x	u	x	x	
	mem, CL	1	1	0	1	0	0	1	W	mod	1	0	1	mem	19 + n	2-4	u	x	u	x	x		
	reg, imm8	1	1	0	0	0	0	0	W	1	1	1	0	1	reg	7 + n	3	u	x	u	x	x	
	mem, imm8	1	1	0	0	0	0	0	W	mod	1	0	1	mem	19 + n	3-5	u	x	u	x	x		
n = number of shifts																							
SHRA	reg, 1	1	1	0	1	0	0	0	W	1	1	1	1	1	reg	2	2	u	x	0	x	x	
	mem, 1	1	1	0	1	0	0	0	W	mod	1	1	1	mem	16	2-4	u	x	0	x	x		
	reg, CL	1	1	0	1	0	0	1	W	1	1	1	1	1	reg	7 + n	2	u	x	u	x	x	
	mem, CL	1	1	0	1	0	0	1	W	mod	1	1	1	mem	19 + n	2-4	u	x	u	x	x		
	reg, imm8	1	1	0	0	0	0	0	W	1	1	1	1	1	reg	7 + n	3	u	x	u	x	x	
	mem, imm8	1	1	0	0	0	0	0	W	mod	1	1	1	mem	19 + n	3-5	u	x	u	x	x		
ROL	reg, 1	1	1	0	1	0	0	0	W	1	1	0	0	0	reg	2	2				x	x	
	mem, 1	1	1	0	1	0	0	0	W	mod	0	0	0	mem	16	2-4				x	x		
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	0	0	reg	7 + n	2				x	u	
	mem, CL	1	1	0	1	0	0	1	W	mod	0	0	0	mem	19 + n	2-4				x	u		
	reg, imm	1	1	0	0	0	0	0	W	1	1	0	0	0	reg	7 + n	3				x	u	
	mem, imm	1	1	0	0	0	0	0	W	mod	0	0	0	mem	19 + n	3-5				x	u		
ROR	reg, 1	1	1	0	1	0	0	0	W	1	1	0	0	1	reg	2	2				x	u	
	mem, 1	1	1	0	1	0	0	0	W	mod	0	0	1	mem	16	2-4				x	x		
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	0	1	reg	7 + n	2				x	u	
	mem, CL	1	1	0	1	0	0	1	W	mod	0	0	1	mem	19 + n	2-4				x	u		
	reg, imm8	1	1	0	0	0	0	0	W	1	1	0	0	1	reg	7 + n	3				x	u	
	mem, imm8	1	1	0	0	0	0	0	W	mod	0	0	1	mem	19 + n	3-5				x	u		

**Instruction Set (cont)**

Mnemonic	Operands	Opcode										Clocks	Bytes	Flags									
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P
<b>Shift/Rotate Instructions (cont)</b>																							
ROLC	reg, 1	1	1	0	1	0	0	0	W	1	1	0	1	0	reg	2	2	x	x				
	mem, 1	1	1	0	1	0	0	0	W	mod	0	1	0	mem	16	2-4	x	x					
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	1	0	reg	7 + n	2	x	u				
	mem, CL	1	1	0	1	0	0	1	W	mod	0	1	0	mem	19 + n	2-4	x	u					
	reg, imm8	1	1	0	0	0	0	0	W	1	1	0	1	0	reg	7 + n	3	x	u				
	mem, imm8	1	1	0	0	0	0	0	W	mod	0	1	0	mem	19 + n	3-5	x	u					
RORC	reg, 1	1	1	0	1	0	0	0	W	1	1	0	1	1	reg	2	2	x	x				
	mem, 1	1	1	0	1	0	0	0	W	mod	0	1	1	mem	16	2-4	x	x					
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	1	1	reg	7 + n	2	x	u				
	mem, CL	1	1	0	1	0	0	1	W	mod	0	1	1	mem	19 + n	2-4	x	u					
	reg, imm8	1	1	0	0	0	0	0	W	1	1	0	1	1	reg	7 + n	3	x	u				
	mem, imm8	1	1	0	0	0	0	0	W	mod	0	1	1	mem	19 + n	3-5	x	u					

n = number of shifts

**Stack Manipulation Instructions**

PUSH	mem16	1	1	1	1	1	1	1	1	mod	1	1	0	mem	18	2-4						
	reg16	0	1	0	1	0	reg	8	1													
	sr	0	0	0	sr	1	1	0	8	1												
	PSW	1	0	0	1	1	1	0	0	8	1											
	R	0	1	1	0	0	0	0	0	35	1											
	imm	0	1	1	0	1	0	S	7-8	2-3												
POP	mem16	1	0	0	0	1	1	1	1	mod	0	0	0	mem	17	2-4						
	reg16	0	1	0	1	1	reg	8	1													
	sr	0	0	0	sr	1	1	1	8	1												
	PSW	1	0	0	1	1	1	0	1	8	1							R	R	R	R	R
	R	0	1	1	0	0	0	0	1	43	1											
PREPARE	imm16, imm8	1	1	0	0	1	0	0	0	*imm8 = 0 : 12 imm8 > 1 : 19 + 8 (imm8 - 1)	*	4										
DISPOSE		1	1	0	0	1	0	0	1		6	1										

**Control Transfer Instructions**

CALL	near_proc	1	1	1	0	1	0	0	0	16	3												
	regptr	1	1	1	1	1	1	1	1	1	1	0	1	0	reg	14	1						
	memptr16	1	1	1	1	1	1	1	1	mod	0	1	0	mem	23	2-4							
	far_proc	1	0	0	1	1	0	1	0	21	5												
	memptr32	1	1	1	1	1	1	1	1	mod	0	1	1	mem	31	2-4							
RET		1	1	0	0	0	0	1	1	15	1												
	pop_value	1	1	0	0	0	0	1	0	20	3												
		1	1	0	0	1	0	1	1	21	1												
	pop_value	1	1	0	0	1	0	1	0	24	3												

### Instruction Set (cont)

Mnemonic	Operand	Opcode										Clocks	Bytes	Flags											
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P	S	Z
<b>Control Transfer Instructions (cont)</b>																									
BR	near_label	1	1	1	0	1	0	0	1									13	3						
	short_label	1	1	1	0	1	0	0	1									12	2						
	reg	1	1	1	1	1	1	1	1	1	1	1	0	0	reg		11	2							
	memptr16	1	1	1	1	1	1	1	1	mod	1	0	0	mem		20	2-4								
	far_label	1	1	1	0	1	0	1	0									15	5						
	memptr32	1	1	1	1	1	1	1	1	mod	1	0	1	mem		27	2-4								
BV	near_label	0	1	1	1	0	0	0	0									14/4	2						
BNV	near_label	0	1	1	1	0	0	0	1									14/4	2						
BC, BL	near_label	0	1	1	1	0	0	1	0									14/4	2						
BNC, BNL	near_label	0	1	1	1	0	0	1	1									14/4	2						
BE, BZ	near_label	0	1	1	1	0	1	0	0									14/4	2						
BNE, BNZ	near_label	0	1	1	1	0	1	0	1									14/4	2						
BNH	near_label	0	1	1	1	0	1	1	0									14/4	2						
BH	near_label	0	1	1	1	0	1	1	1									14/4	2						
BN	near_label	0	1	1	1	1	0	0	0									14/4	2						
BP	near_label	0	1	1	1	1	0	0	1									14/4	2						
BPE	near_label	0	1	1	1	1	0	1	0									14/4	2						
BPO	near_label	0	1	1	1	1	0	1	1									14/4	2						
BLT	near_label	0	1	1	1	1	1	0	0									14/4	2						
BGE	near_label	0	1	1	1	1	1	0	1									14/4	2						
BLE	near_label	0	1	1	1	1	1	1	0									14/4	2						
BGT	near_label	0	1	1	1	1	1	1	1									14/4	2						
DBNZNE	near_label	1	1	1	0	0	0	0	0									14/5	2						
DBNZE	near_label	1	1	1	0	0	0	0	1									14/5	2						
DBNZ	near_label	1	1	1	0	0	0	1	0									13/5	2						
BCWZ	near_label	1	1	1	0	0	0	1	1									13/5	2						
<b>Interrupt Instructions</b>																									
BRK	3	1	1	0	0	1	1	0	0									38	1						
	imm8	1	1	0	0	1	1	0	1									38	2						
BRKV	imm8	1	1	0	0	1	1	1	1									40/3	1						
RETI		1	1	0	0	1	1	1	0									27	1	R	R	R	R	R	R
CHKIND	reg16, mem32	0	1	1	0	0	0	1	0	mod		reg	mem			53-56/18	2-4								
BRKEM	imm8	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1		38	3						



### Description

The μPD70208 (V40™) is a high-performance, low-power 16-bit microprocessor integrating a number of commonly used peripherals to dramatically reduce the size of microprocessor systems. The CMOS construction makes the μPD70208 ideal for the design of portable computers, instrumentation, and process control equipment.

The μPD70208 contains a powerful instruction set that is compatible with the μPD70108/μPD70116 (V20™/V30™) and μPD8086/μPD8088 instruction sets. Instruction set support includes a wide range of arithmetic, logical, and control operations as well as bit manipulation, BCD arithmetic, and high-speed block transfer instructions. The μPD70208 can also execute the entire μPD8080AF instruction set using the 8080 emulation mode. Also available is the μPD70216 (V50™), identical to the μPD70208 but with a 16-bit external data bus.

### Features

- Low-power CMOS technology
- V20/V30 instruction set compatible
- Minimum instruction execution time: 250 ns (at 8 MHz)
- Direct addressing of 1M bytes of memory
- Powerful set of addressing modes
- Fourteen 16-bit CPU registers
- On-chip peripherals including
  - Clock generator
  - Bus interface
  - Bus arbitration
  - Programmable wait state generator
  - DRAM refresh controller
  - Three 16-bit timer/counters
  - Asynchronous serial I/O controller
  - Eight-input interrupt controller
  - Four-channel DMA controller
- Hardware effective address calculation logic
- Maskable and nonmaskable interrupts
- IEEE 796 compatible bus interface
- Low-power standby mode

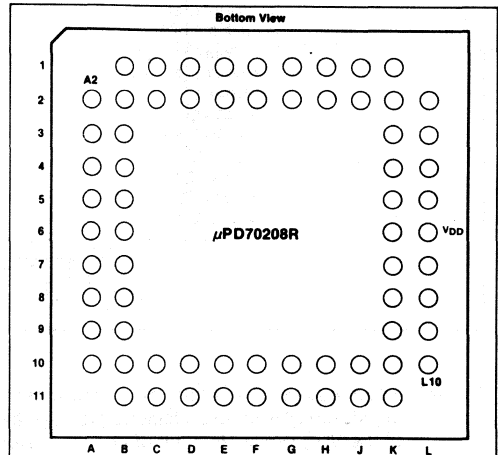
V20, V30, V40, and V50 are trademarks of NEC Corporation.

### Ordering Information

Part Number	Package	Maximum Frequency
μPD70208R-8	68-pin ceramic PGA	8 MHz
μPD70208L-8	68-pin PLCC	8 MHz
μPD70208GF-8	80-pin plastic flat pack	8 MHz
μPD70208R-10	68-pin ceramic PGA	10 MHz
μPD70208L-10	68-pin PLCC	10 MHz
μPD70208GF-10	80-pin plastic flat pack	10 MHz

### Pin Configurations

#### 68-Pin Ceramic PGA



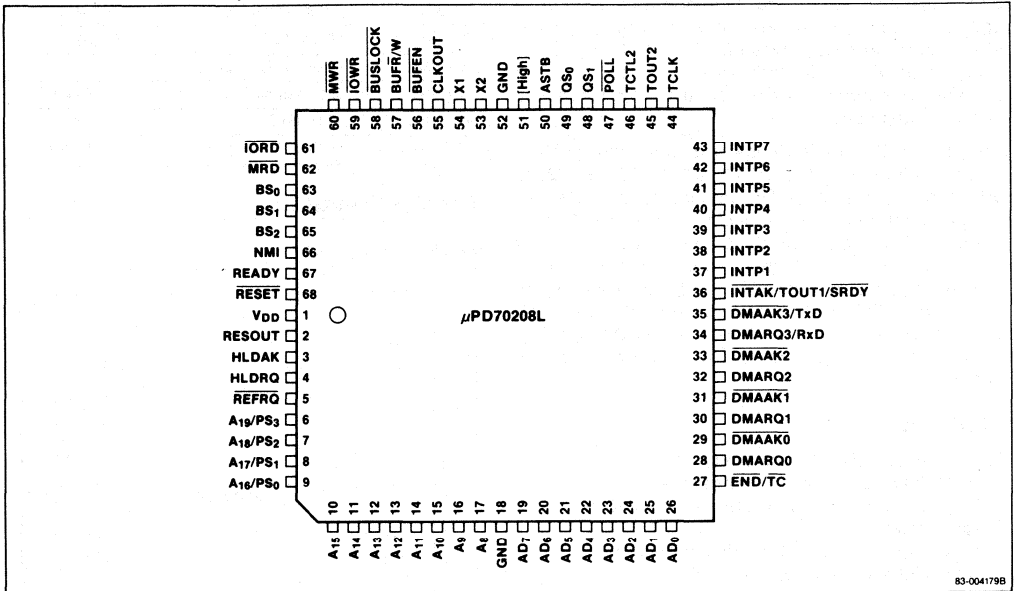
Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol
A2	INTP7	B9	DMARQ1	F10	AD7	K4	NMI
A3	INTP5	B10	DMARQ0	F11	GND	K5	RESET
A4	INTP3	B11	AD <sub>0</sub>	G1	X1	K6	RESOUT
A5	INTP1	C1	TCTL2	G2	CLKOUT	K7	HLDRQ
A6	DMAAK3/TxD	C2	POLL	G10	A <sub>8</sub>	K8	A <sub>16</sub> /P <sub>8</sub> <sub>2</sub>
A7	DMAAK2	C10	AD <sub>1</sub>	G11	A <sub>9</sub>	K9	A <sub>17</sub> /P <sub>8</sub> <sub>1</sub>
A8	DMAAK1	C11	AD <sub>2</sub>	H1	BUFEN	K10	A <sub>14</sub>
A9	DMAAK0	D1	Q <sub>8</sub> 1	H2	BUF <sub>R</sub> /W	K11	A <sub>15</sub>
A10	END/TC	D2	Q <sub>8</sub> 0	H10	A <sub>10</sub>	L2	IOR <sub>D</sub>
B1	TCLK	D10	AD <sub>3</sub>	H11	A <sub>11</sub>	L3	BS <sub>0</sub>
B2	TOUT2	D11	AD <sub>4</sub>	J1	SUBLOCK	L4	BS <sub>2</sub>
B3	INTP6	E1	ASTB	J2	IOWR	L5	READY
B4	INTP4	E2	[High]	J10	A <sub>12</sub>	L6	VDD
B5	INTP2	E10	AD <sub>5</sub>	J11	A <sub>13</sub>	L7	HLDAK
B6	INTAR/TOUT1/SRDY	E11	AD <sub>6</sub>	K1	MWR	L8	REFRQ
B7	DMARQ3/RxD	F1	GND	K2	M <sub>RD</sub>	L9	A <sub>16</sub> /P <sub>8</sub> <sub>2</sub>
B8	DMARQ2	F2	X2	K3	BS <sub>1</sub>	L10	A <sub>16</sub> /P <sub>8</sub> <sub>0</sub>

83-0027168

## μPD70208 (V40)

### Pin Configurations (cont)

#### 68-Pin Plastic Leaded Chip Carrier (PLCC) (Top View)

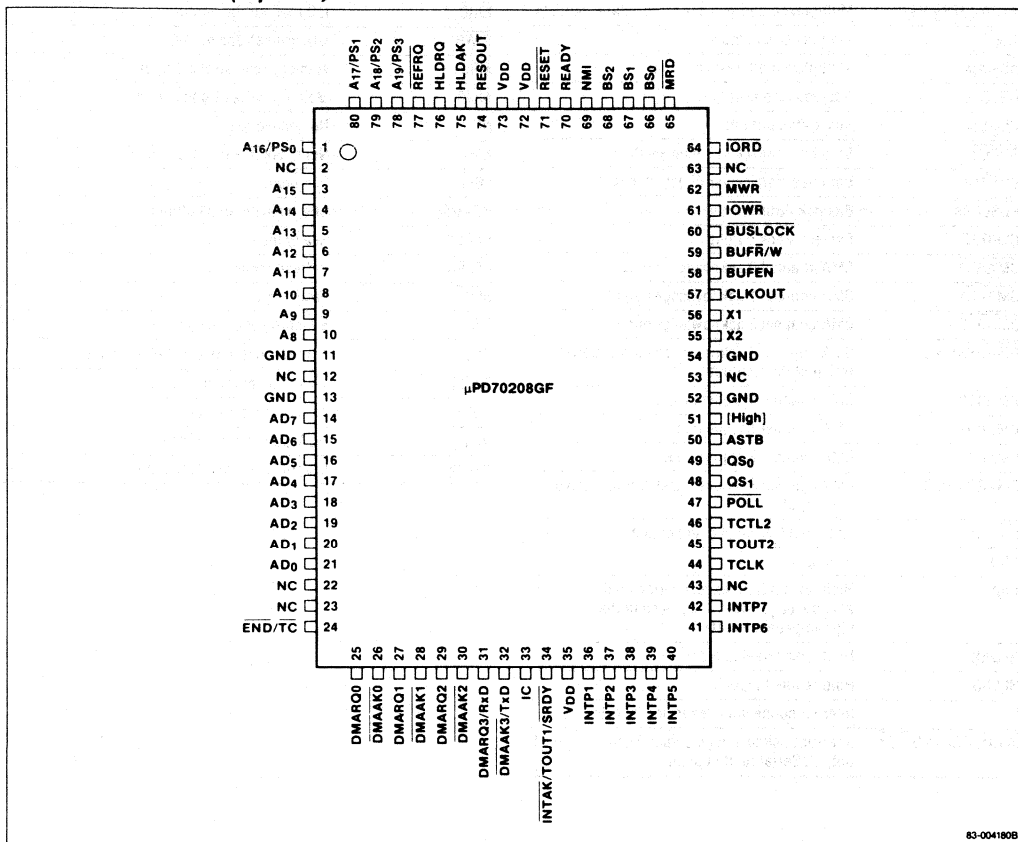


83-004179B

**Notes:** If one plugs a μPD70208L into a PGA socket, by use of an adaptor, it should be remembered that the location of the VDD pin is 180° out of phase on these two package types.

### Pin Configurations (cont)

#### 80-Pin Plastic Flat Pack (Top View)



83-004180B



**Pin Identification**

Symbol	Function
A <sub>19</sub> -A <sub>16</sub> /PS <sub>3</sub> -PS <sub>0</sub>	Multiplexed address/processor status outputs
A <sub>15</sub> -A <sub>8</sub>	Address bus outputs
AD <sub>7</sub> -AD <sub>0</sub>	Multiplexed address/data bus
ASTB	Address strobe output
BS <sub>2</sub> -BS <sub>0</sub>	Bus status outputs
BUFEN	Data bus transceiver enable output
BUFR/W	Data bus transceiver direction output
BUSLOCK	Buslock output
CLKOUT	System clock output
DMAAK0	DMA channel 0 acknowledge output
DMAAK1	DMA channel 1 acknowledge output
DMAAK2	DMA channel 2 acknowledge output
DMAAK3/TxD	DMA channel 3 acknowledge output/Serial transmit data output
DMARQ0	DMA channel 0 request input
DMARQ1	DMA channel 1 request input
DMARQ2	DMA channel 2 request input
DMARQ3/RxD	DMA channel 3 request input/Serial receive data input
END/TC	End input/Terminal count output
GND	Ground
High	High-level output except during hold acknowledge when it is placed in the high-impedance state
HLDK	Hold acknowledge output
HLDRQ	Hold request input
IC	Internal connection; leave unconnected
INTAK/TOUT1/SRDY	Interrupt acknowledge output/Timer/counter 1 output/Serial ready output

Symbol	Function
INTP1-INTP7	Interrupt request inputs
IORD	I/O read strobe output
IOWR	I/O write strobe output
MRD	Memory read strobe output
MWR	Memory write strobe output
NC	No connection
NMI	Nonmaskable interrupt input
POLL	Poll input
QS <sub>1</sub> -QS <sub>0</sub>	CPU queue status outputs
READY	Ready input
REFRQ	Refresh request output
RESET	Reset input
RESOUT	Synchronized reset output
TCLK	Timer/counter external clock input
TCTL2	Timer/counter 2 control input
TOUT2	Timer/counter 2 output
V <sub>DD</sub>	+5 V power supply input
X1, X2	Crystal/external clock inputs

## Pin Functions

### A<sub>19</sub>-A<sub>16</sub>/PS<sub>3</sub>-PS<sub>0</sub> [Address/Status Bus]

These three-state output pins contain the upper 4 bits of the 20-bit address during T1 and processor status information during the T2, T3, TW, and T4 states of a bus cycle. During T1 of a memory read or write cycle, these pins contain the upper 4 bits of the 20-bit address. These pins are forced low during T1 of an I/O bus cycle.

Processor status is output during T2, T3, TW, and T4 of both memory and I/O bus cycles. PS<sub>3</sub> is zero during any CPU native mode bus cycle. During any DMA, refresh, or 8080 emulation mode bus cycle, PS<sub>3</sub> outputs a high level. PS<sub>2</sub> outputs the contents of the interrupt enable (IE) flag in the CPU PSW register. PS<sub>1</sub> and PS<sub>0</sub> indicate the segment register used to form the physical address of a CPU bus cycle as follows:

PS <sub>1</sub>	PS <sub>0</sub>	Segment
0	0	Data segment 1 (DS1)
0	1	Stack segment (SS)
1	0	Program segment (PS)
1	1	Data segment 0 (DS0)

These pins are in the high-impedance state during hold acknowledge.

### A<sub>15</sub>-A<sub>8</sub> [Address Bus]

These three-state pins form the middle byte of the active-high address bus. During any CPU, DMA, or refresh bus cycle, A<sub>15</sub>-A<sub>8</sub> output the middle 8 bits of the 20-bit memory or I/O address. The A<sub>15</sub>-A<sub>8</sub> pins enter the high-impedance state during hold acknowledge or an internal interrupt acknowledge bus cycle. During a slave interrupt acknowledge bus cycle, A<sub>10</sub>-A<sub>8</sub> contain the address of the selected slave interrupt controller.

### AD<sub>7</sub>-AD<sub>0</sub> [Address/Data Bus]

These three-state pins form the active-high, time-multiplexed address/data bus. During T1 of a bus cycle, AD<sub>7</sub>-AD<sub>0</sub> output the lower 8 bits of the 20-bit memory or I/O address. During the T2, T3, TW, and T4 states, AD<sub>7</sub>-AD<sub>0</sub> form the 8-bit bidirectional data bus.

The AD<sub>7</sub>-AD<sub>0</sub> pins enter the high-impedance state during hold acknowledge or internal interrupt acknowledge bus cycles or while RESET is asserted.

### ASTB [Address Strobe]

This active-high output is used to latch the address from the multiplexed address bus in an external address latch during T1 of a bus cycle. ASTB is held at a low level during hold acknowledge.

### BS<sub>2</sub>-BS<sub>0</sub> [Bus Status]

Outputs BS<sub>2</sub>-BS<sub>0</sub> indicate the type of bus cycle being performed as follows. BS<sub>2</sub>-BS<sub>0</sub> become active during the state preceding T1 and return to the passive state during the bus state preceding T4.

BS <sub>2</sub>	BS <sub>1</sub>	BS <sub>0</sub>	Bus Cycle
0	0	0	Interrupt acknowledge
0	0	1	I/O read
0	1	0	I/O write
0	1	1	Halt (Note 1)
1	0	0	Instruction fetch
1	0	1	Memory read (Note 2)
1	1	0	Memory write (Note 3)
1	1	1	Passive state

#### Note:

- (1) BS<sub>2</sub>-BS<sub>0</sub> in a halt bus cycle returns to the passive state one clock earlier than normal CPU bus cycles.
- (2) Memory read bus cycles include CPU, DMA read, DMA verify, and refresh bus cycles.
- (3) Memory write bus cycles include CPU and DMA write bus cycles.

BS<sub>2</sub>-BS<sub>0</sub> are three-state outputs and are high impedance during hold acknowledge.

**BUFEN [Buffer Enable]**

**BUFEN** is an active-low output for enabling an external data bus transceiver during a bus cycle. **BUFEN** is asserted during T2 through T3 of a read cycle, T2 through T3 of a slave interrupt acknowledge cycle, and T1 through T4 of a write cycle. **BUFEN** is not asserted when the bus cycle corresponds to an internal peripheral, DMA, refresh, or internal interrupt acknowledge cycle. **BUFEN** enters the high-impedance state during hold acknowledge.

**BUFR/W [Buffer Read/Write]**

**BUFR/W** is a three-state, active-low output used to control the direction of an external data bus transceiver during CPU bus cycles. A high level indicates the μPD70208 will perform a write cycle and a low level indicates a read cycle. **BUFR/W** enters the high-impedance state during hold acknowledge.

**BUSLOCK**

This active-low output provides a means for the CPU to indicate to an external bus arbiter that the bus cycles of the next instruction are to be kept contiguous. **BUSLOCK** is asserted for the duration of the instruction following the **BUSLOCK** prefix. **BUSLOCK** is also asserted during interrupt acknowledge cycles and enters the high-impedance state during hold acknowledge. While **BUSLOCK** is asserted, DMAU, RCU, and external bus requests are ignored.

**CLKOUT**

**CLKOUT** is a buffered clock output used as a reference for all timing. **CLKOUT** has a 50-percent duty cycle at half the frequency of the input clock source.

**DMAAK0-DMAAK2 [DMA Acknowledge]**

This set of outputs contains the DMA acknowledge signals for channels 0-2 from the internal DMA controller and indicate that the peripheral can perform the requested transfer.

**DMAAK3/TxD [DMA Acknowledge 3]/[Serial Transmit Data]**

Two output signals multiplexed on this pin are selected by the PF field of the on-chip peripheral connection register.

- **DMAAK3** is an active-low output and enables an external DMA peripheral to perform the requested DMA transfer for channel 3.
- **TxD** is the serial data output from the serial control unit.

**DMARQ0-DMARQ2 [DMA Request]**

These synchronized inputs are used by external peripherals to request DMA service for channels 0-2 from the internal DMA controller.

**DMARQ3/RxD [DMA Request 3]/[Serial Receive Data]**

Two input signals multiplexed on this pin are selected by the PF field of the on-chip peripheral connection register.

- **DMARQ3** is used by an external peripheral to request a DMA transfer cycle for channel 3.
- **RxD** is the serial data input to the serial control unit.

**END/TC [End/Terminal Count]**

This active-low bidirectional pin controls the termination of a DMA service. Assertion of **END** by external hardware during DMA service causes the service to terminate. When a DMA channel reaches its terminal count, the DMAU asserts **TC**, indicating the programmed operation has completed.

**END/TC** is an open-drain I/O pin, and requires an external 2.2-kΩ pull-up resistor.

### HLD $\overline{\text{AK}}$ [Hold Acknowledge]

When an external bus requester has become the highest priority requester, the internal bus arbiter will assert the HLD $\overline{\text{AK}}$  output indicating the address, data, and control buses have entered a high-impedance state and are available for use by the external bus master.

Should the internal DMAU or RCU (demand mode) request the bus, the bus arbiter will drive HLD $\overline{\text{AK}}$  low. When this occurs, the external bus master should complete the current bus cycle and negate the HLDRQ signal. This allows the bus arbiter to reassign the bus to the higher priority requester.

If a higher priority internal bus master subsequently requests the bus, the high-level width of HLD $\overline{\text{AK}}$  is guaranteed to be a minimum of one CLKOUT period.

### HLDRQ [Hold Request]

This active-high signal is asserted by an external bus master requesting to use the local address, data, and control buses. The HLDRQ input is used by the internal bus arbiter, which gives control of the buses to the highest priority bus requester in the following order.

Bus Master	Priority
RCU	Highest (demand mode)
DMAU	•
HLDRQ	•
CPU	•
RCU	Lowest (normal operation)

### INTAK/TOUT1/SRDY [Interrupt Acknowledge]/[Timer 1 Output]/[Serial Ready]

Three output signals multiplexed on this pin are selected by the PF field of the on-chip peripheral connection register.

- INTAK is an interrupt acknowledge signal used to cascade external slave μPD71059 Interrupt Controllers. INTAK is asserted during T2, T3, and TW states of an interrupt acknowledge cycle.
- TOUT1 is the output of timer/counter 1.
- SRDY is an active-low output and indicates that the serial control unit is ready to receive the next character.

### INTP1-INTP7 [Peripheral Interrupts]

INTP1-INTP7 accept either rising-edge or high-level triggered asynchronous interrupt requests from external peripherals. These INTP1-INTP7 inputs are internally synchronized and prioritized by the interrupt control unit, which requests the CPU to perform an interrupt acknowledge bus cycle. External interrupt controllers such as the μPD71059 can be cascaded to increase the number of vectored interrupts.

These interrupt inputs cause the CPU to exit both the standby and 8080 emulation modes.

The INTP1-INTP7 inputs contain internal pull-up resistors and may be left unconnected.

### IORD [I/O Read]

This three-state pin outputs an active-low I/O read strobe during T2, T3, and TW of an I/O read bus cycle. Both CPU I/O read and DMA write bus cycles assert IORD. IORD is not asserted when the bus cycle corresponds to an internal peripheral or register. It enters the high-impedance state during hold acknowledge.

### IOWR [I/O Write]

This three-state pin outputs an active-low I/O write strobe during T2, T3, and TW of a CPU I/O write or an extended DMA read cycle and during T3 and TW of a DMA read bus cycle. IOWR is not asserted when the bus cycle corresponds to an internal peripheral or register. It enters the high-impedance state during hold acknowledge.

### **MRD [Memory Read Strobe]**

This three-state pin outputs an active-low memory read strobe during T2, T3, and TW of a memory read bus cycle. CPU memory read, DMA read, and refresh bus cycles all assert MRD. MRD enters the high-impedance state during hold acknowledge.

### **MWR [Memory Write Strobe]**

This three-state pin outputs an active-low memory write strobe during T2, T3, and TW of a CPU memory write or DMA extended write bus cycle and during T3 and TW of a DMA normal write bus cycle. MWR enters the high-impedance state during hold acknowledge.

### **NMI [Nonmaskable Interrupt]**

The NMI pin is a rising-edge-triggered interrupt input that cannot be masked by software. NMI is sampled by CPU logic each clock cycle and when found valid for one or more CLKOUT cycles, the NMI interrupt is accepted. The CPU will process the NMI interrupt immediately after the current instruction finishes execution by fetching the segment and offset of the NMI handler from interrupt vector 2. The NMI interrupt causes the CPU to exit both the standby and 8080 emulation modes. The NMI input takes precedence over the maskable interrupt inputs.

### **POLL [Poll]**

The active-low POLL input is used to synchronize the operation of external devices with the CPU. During execution of the POLL instruction, the CPU checks the POLL input state every five clocks until POLL is once again asserted.

### **QS<sub>1</sub>-QS<sub>0</sub> [Queue Status]**

The QS<sub>1</sub> and QS<sub>0</sub> outputs maintain instruction synchronization between the μPD70208 CPU and external devices such as the μPD72191 Floating Point Processor. These outputs are interpreted as follows.

QS <sub>1</sub>	QS <sub>0</sub>	Instruction Queue Status
0	0	No operation
0	1	First byte of instruction fetched
1	0	Flush queue contents
1	1	Subsequent byte of instruction fetched

Queue status is valid for one clock cycle after the CPU has accessed the instruction queue.

### **READY [Ready]**

This active-high input synchronizes external memory and peripheral devices with the μPD70208. Slow memory and I/O devices can lengthen a bus cycle by negating the READY input and forcing the BIU to insert TW states. READY must be negated prior to the rising edge of CLKOUT during the T2 state or by the last internally generated TW state to guarantee recognition. When READY is once again asserted and recognized by the BIU, the BIU will proceed to the T4 state.

The READY input operates in parallel with the internal μPD70208 wait control unit and can be used to insert more than three wait states into a bus cycle.

### **REFRQ [Refresh Request]**

REFRQ is an active-low output indicating the current bus cycle is a memory refresh operation. REFRQ is used to disable memory address decode logic and refresh dynamic memories. The 9-bit refresh row address is placed on A<sub>8</sub>-A<sub>0</sub> during a refresh bus cycle.

### **RESET [Reset]**

RESET is a Schmitt trigger input used to force the μPD70208 to a known state by resetting the CPU and on-chip peripherals. RESET must be asserted for a minimum of four clocks to guarantee recognition. To ensure proper rise time of the Reset pulse it is recommended to use an external schmitt trigger gate. After RESET has been released, the CPU will start program cution from address FFFF0H in the native mode.

RESET will release the CPU from the low-power standby mode and force it to the native mode.

### **RESOUT [Reset Output]**

This active-high output is available to perform a system-wide reset function. RESET is internally synchronized with CLKOUT and output on the RESOUT pin.

### **TCLK**

TCLK is an external clock source for the timer control unit. The three timer/counters can be programmed to operate with either the TCLK input or a prescaled CLKOUT input.

### TCTL2

TCTL2 is the control input for timer/counter 2.

### TOUT2

TOUT2 is the output of timer/counter 2.

### X1, X2 [Clock Inputs]

These pins accept either a parallel resonant, fundamental mode crystal or an external oscillator input with a frequency twice the desired operating frequency.

In the case of an external clock generator, the X2 pin can be either left unconnected or be driven by the complement of the X1 pin clock source.

### Pin States

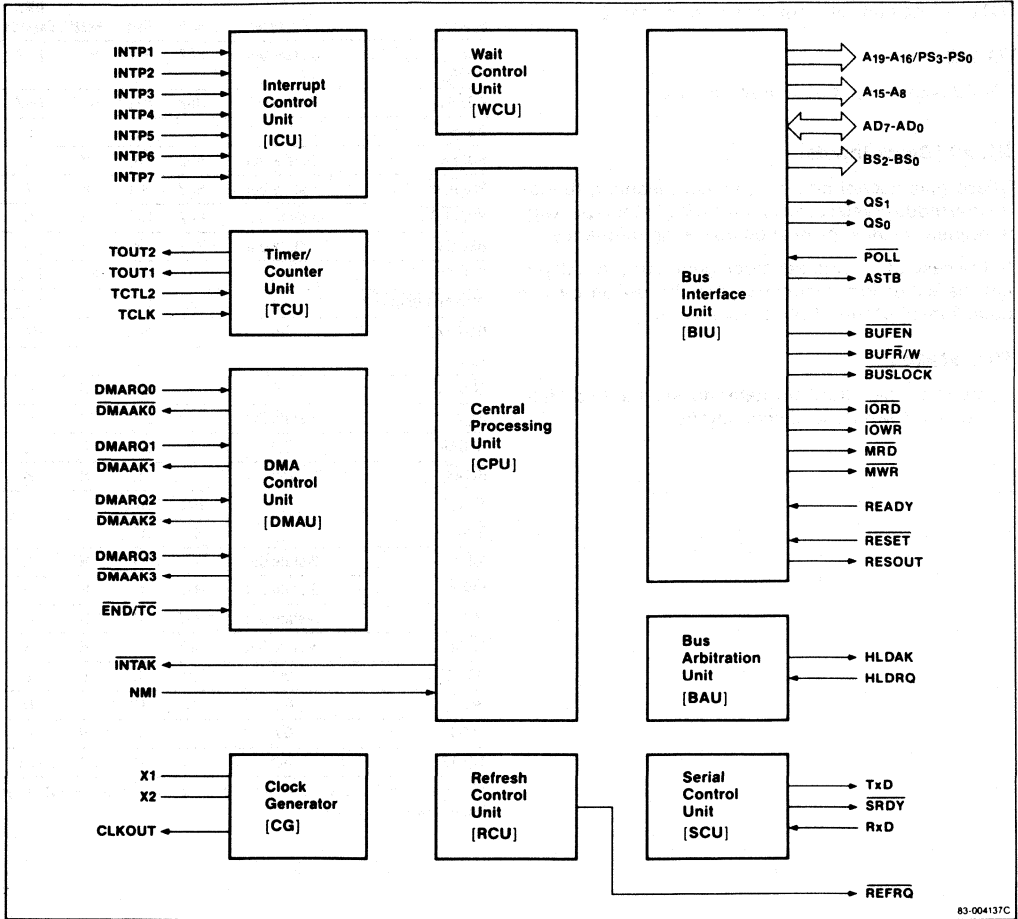
Table 1 lists the output pin states during the Hold, Halt, Reset, and DMA Cascade conditions.

**Table 1. Output Pin States**

Symbol	Pin Type	Hold	Halt	Reset	DMA Cascade
A <sub>19</sub> -A <sub>16</sub> /PS <sub>3</sub> -PS <sub>0</sub> , A <sub>15</sub> -A <sub>8</sub>	3-state Out	Hi-Z	H/L	H/L	Hi-Z
AD <sub>7</sub> -AD <sub>0</sub>	3-state I/O	Hi-Z	H/L	Hi-Z	Hi-Z
ASTB	Out	L	L	L	L
BUFEN	3-state Out	Hi-Z	H	H	H
BUFR/W	3-state Out	Hi-Z	H/L	H	H/L
BUSLOCK	3-state Out	Hi-Z	H/L	H	H
BS <sub>2</sub> -BS <sub>0</sub>	3-state Out	Hi-Z	H	H	H
CLKOUT	Out	H/L	H/L	H/L	H/L
DMAAK0-DMAAK2	Out	H	H/L	H	H/L
DMAAK3	Out	H	H/L	H	H/L
TxD		H/L	H/L		
END/TC	I/O	H	H/L	H	H
High	3-state Out	Hi-Z	H	H	H
HLDAK	Out	H	H/L	L	L
INTAK	Out	H	H	H	H
TOUT1		H/L	H/L		
SRDY		H/L	H/L		
IORD	3-state Out	Hi-Z	H	H	H
IOWR	3-state Out	Hi-Z	H	H	H
MRD	3-state Out	Hi-Z	H	H	H
MWR	3-state Out	Hi-Z	H	H	H
QS <sub>1</sub> -QS <sub>0</sub>	Out	H/L	L	L	L
REFRQ	Out	H	H/L	H	H
RESOUT	Out	L	L	H	H
TOUT2	Out	H/L	H/L	H/L	H/L

H: high level; L: low level; H/L: high or low level; Hi-Z: high impedance.

Block Diagram



83-004137C

### Absolute Maximum Ratings

T<sub>A</sub> = +25°C

Power supply voltage, V <sub>DD</sub>	-0.5 to +7.0 V
Input voltage, V <sub>I</sub>	-0.5 to V <sub>DD</sub> + 0.3 V
CLK input voltage, V <sub>K</sub>	-0.5 to V <sub>DD</sub> + 1.0 V
Output voltage, V <sub>O</sub>	-0.5 to V <sub>DD</sub> + 0.3 V
Operating temperature, T <sub>OP</sub>	-10 to +70°C
Storage temperature, T <sub>STG</sub>	-65 to +150°C

**Comment:** Exposure to Absolute Maximum Ratings for extended periods may affect device reliability; exceeding the ratings could cause permanent damage. The device should be operated within the limits specified under DC and AC Characteristics.

### Capacitance

T<sub>A</sub> = +25°C, V<sub>DD</sub> = 0 V

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input capacitance	C <sub>I</sub>		15	pF	f <sub>C</sub> = 1 MHz;
Output capacitance	C <sub>O</sub>		15	pF	unmeasured pins are returned to 0 V.

### DC Characteristics

T<sub>A</sub> = -10 to +70°C; V<sub>DD</sub> = +5 V ±10%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input voltage, high	V <sub>IH</sub>	2.2	V <sub>DD</sub> + 0.3	V	
Input voltage, low	V <sub>IL</sub>	-0.5	0.8	V	
X1, X2 input voltage, high	V <sub>KH</sub>	3.9	V <sub>DD</sub> + 1.0	V	
X1, X2 input voltage, low	V <sub>KL</sub>	-0.5	0.6	V	
Output voltage, high	V <sub>OH</sub>	0.7 V <sub>DD</sub>		V	I <sub>OH</sub> = -400 μA
Output voltage, low	V <sub>OL</sub>		0.4	V	I <sub>OL</sub> = 2.5 mA
Input leakage current, high	I <sub>LIH</sub>		10	μA	V <sub>I</sub> = V <sub>DD</sub>
Input leakage current, low	I <sub>L IPL</sub>		-300	μA	V <sub>I</sub> = 0 V, INTP input pins
	I <sub>L I L</sub>		-10	μA	V <sub>I</sub> = 0 V, other input pins
Output leakage current, high	I <sub>LOH</sub>		10	μA	V <sub>O</sub> = V <sub>DD</sub>
Output leakage current, low	I <sub>LOL</sub>		-10	μA	V <sub>O</sub> = 0 V
Supply current	I <sub>DD</sub>	μPD70208-8 (8 MHz)	90	mA	Normal mode
			20	mA	Standby mode
		μPD70208-10 (10 MHz)	120	mA	Normal mode
			25	mA	Standby mode



**AC Characteristics**

μPD70208-8:  $T_A = -10$  to  $+70^\circ\text{C}$ ;  $V_{DD} = +5\text{ V} \pm 10\%$ ;  $C_L = 100\text{ pF}$

μPD70208-10:  $T_A = -10$  to  $+70^\circ\text{C}$ ;  $V_{DD} = +5\text{ V} \pm 5\%$ ;  $C_L = 100\text{ pF}$

Parameter	Symbol	μPD70208-8		μPD70208-10		Unit	Test Condition
		Min	Max	Min	Max		
External clock input cycle time	t <sub>CYX</sub>	62	250	50	250	ns	
External clock pulse width, high	t <sub>XH</sub>	20		19		ns	V <sub>KH</sub> = 3.0 V
External clock pulse width, low	t <sub>XL</sub>	20		19		ns	V <sub>KL</sub> = 1.5 V
External clock rise time	t <sub>XR</sub>		10		5	ns	1.5 → 3.0 V
External clock fall time	t <sub>XF</sub>		10		5	ns	3.0 → 1.5 V
CLKOUT cycle time	t <sub>CYK</sub>	124	500	100	500	ns	
CLKOUT pulse width, high	t <sub>KH</sub>	0.5 t <sub>CYK-7</sub>		0.5 t <sub>CYK-5</sub>		ns	V <sub>KH</sub> = 3.0 V
CLKOUT puls width, low	t <sub>KL</sub>	0.5 t <sub>CYK-7</sub>		0.5 t <sub>CYK-5</sub>		ns	V <sub>KL</sub> = 1.5 V
CLKOUT rise time	t <sub>KR</sub>		7		5	ns	1.5 → 3.0 V
CLKOUT fall time	t <sub>KF</sub>		7		5	ns	3.0 → 1.5 V
CLKOUT delay time from external clock	t <sub>DXK</sub>		55		40	ns	
Input rise time (incl. RESET, but except external clock)	t <sub>IR</sub>		20		15	ns	0.8 → 2.2 V
Input fall time (incl. RESET, but external clock)	t <sub>IF</sub>		12		10	ns	2.2 → 0.8 V
Output rise time (except CLKOUT)	t <sub>OR</sub>		20		15	ns	0.8 → 2.2 V
Output fall time (except CLKOUT)	t <sub>OF</sub>		12		10	ns	2.2 → 0.8 V
RESET setup time to CLKOUT↑	t <sub>SRESK</sub>	25		20		ns	
RESET hold time after CLKOUT↓	t <sub>HKRES</sub>	35		25		ns	
RESOUT delay time from CLKOUT↓	t <sub>DKRES</sub>	5	60	5	50	ns	
READY inactive setup time to CLKOUT↑	t <sub>SRYLK</sub>	15		15		ns	
READY inactive hold time after CLKOUT↑	t <sub>HKRYL</sub>	25		20		ns	
READY active setup time to CLKOUT↑	t <sub>SRYHK</sub>	15		15		ns	
READY active hold time after CLKOUT↑	t <sub>HKRYH</sub>	25		20		ns	
NMI, POLL setup time to CLKOUT↑	t <sub>SIK</sub>	15		15		ns	

### AC Characteristics (cont)

Parameter	Symbol	μPD70208-8		μPD70208-10		Unit	Test Condition
		Min	Max	Min	Max		
Data setup time to CLKOUT↓	t <sub>SDK</sub>	15		15		ns	
Data hold time after CLKOUT↓	t <sub>HKD</sub>	10		10		ns	
Address delay time from CLKOUT↓	t <sub>DKA</sub>	10	55	10	50	ns	
Address hold time after CLKOUT↓	t <sub>HKA</sub>	10		10		ns	
I/O recovery time	t <sub>AI</sub>	2t <sub>CYK</sub> -50		2t <sub>CYK</sub> -40		ns	(Note 4)
PS delay time from CLKOUT↓	t <sub>DKP</sub>	10	60	10	50	ns	
PS float delay time from CLKOUT↓	t <sub>FKP</sub>	10	60	10	50	ns	
Address setup time to ASTB↓	t <sub>SAST</sub>	t <sub>KKL</sub> -30		t <sub>KKL</sub> -20		ns	
Address float delay time from CLKOUT↓	t <sub>FKA</sub>	t <sub>HKA</sub>	60	t <sub>HKA</sub>	50	ns	
ASTB↑ delay time from CLKOUT↓	t <sub>DKSTH</sub>		45		40	ns	
ASTB↓ delay time from CLKOUT↑	t <sub>DKSTL</sub>		50		45	ns	
ASTB pulse width, high	t <sub>STST</sub>	t <sub>KKL</sub> -10		t <sub>KKL</sub> -10		ns	
Address hold time after ASTB↓	t <sub>HSTA</sub>	t <sub>KKH</sub> -20		t <sub>KKH</sub> -20		ns	
Control delay time from CLKOUT	t <sub>DKCT1</sub>	10	70	10	60	ns	(Note 1)
	t <sub>DKCT2</sub>	10	60	10	55	ns	(Note 2)
RD↓ delay time from address float	t <sub>DAFRL</sub>	0		0		ns	(Note 3)
RD↓ delay time from CLKOUT↓	t <sub>DKRL</sub>	10	75	10	65	ns	
RD↑ delay time from CLKOUT↓	t <sub>DKRH</sub>	10	70	10	60	ns	
REFRQ↑ delay from MRD↑	t <sub>DRQHRH</sub>	t <sub>KKL</sub> -30		t <sub>KKL</sub> -30		ns	(Note 5)
Address delay time from RD	t <sub>DRHA</sub>	t <sub>CYK</sub> -40		t <sub>CYK</sub> -40		ns	
RD pulse width, low	t <sub>RR</sub>	2t <sub>CYK</sub> -50		2t <sub>CYK</sub> -40		ns	
BUF <sub>R</sub> /W delay from BUF <sub>E</sub> ↑	t <sub>DBECT</sub>	t <sub>KKL</sub> -20		t <sub>KKL</sub> -20		ns	Read cycle
	t <sub>DWCT</sub>	t <sub>KKL</sub> -20		t <sub>KKL</sub> -20		ns	Write cycle
Data output delay time from CLKOUT↓	t <sub>DKD</sub>	10	60	10	55	ns	
Data float delay time from CLKOUT↓	t <sub>FKD</sub>	10	60	10	55	ns	
WR pulse width, low	t <sub>WW</sub>	2t <sub>CYK</sub> -40		2t <sub>CYK</sub> -40		ns	(Note 3)

**AC Characteristics (cont)**

Parameter	Symbol	μPD70208-8		μPD70208-10		Unit	Test Condition
		Min	Max	Min	Max		
BS↓ delay time from CLKOUT↑	tDKBL	10	60	10	55	ns	
BS↑ delay time from CLKOUT↓	tDKBH	10	60	10	55	ns	
HLDRQ setup time to CLKOUT↑	tSHQK	20		15		ns	
HLDAK delay time from CLKOUT↓	tDKHA	10	70	10	60	ns	
Address drive delay time from CLKOUT↓	tDKA2	tCYK		tCYK		ns	
DMAAK↓ delay time from CLKOUT↑	tDKHDA	10	60	10	55	ns	
DMAAK↓ delay time from CLKOUT↓	tDKLDA	10	90	10	80	ns	Cascade mode
WR pulse width, low (DMA cycle)	tWW1	2tCYK -40		2tCYK -40		ns	DMA extended write cycle
WR pulse width, low (DMA cycle)	tWW2	tCYK -40		tCYK -40		ns	DMA normal write cycle
RD↓, WR↓ delay from DMAAK↓	tDDARW	tKXH -30		tKXH -30		ns	
DMAAK↑ delay from RD↑	tDRHDAH	tKXL -30		tKXL -30		ns	
RD↑ delay from WR↑	tDWRH	5		5		ns	
TC output delay time from CLKOUT↑	tDKTCL		60		55	ns	
TC off delay time from CLKOUT↑	tDKTCF		60		55	ns	
TC pulse width, low	tTCTCL	tCYK -15		tCYK -15		ns	
TC pullup delay time from CLKOUT↑	tDKTCH	tKXH + tCYK -10		tKXH + tCYK -10		ns	
END setup time to CLKOUT↑	tSEDK	35		30		ns	
END pulse width, low	tEEDL	100		80		ns	
DMARQ setup time to CLKOUT↑	tSDQK	35		30		ns	
INTPn pulse width, low	tPIPL	100		80		ns	
RxD setup time to SCU internal clock↓	tSRX	1		0.5		μs	
RxD hold time after SCU internal clock↓	tHRX	1		0.5		μs	
SRDY delay time from CLKOUT↓	tDKSR		150		100	ns	

### AC Characteristics (cont)

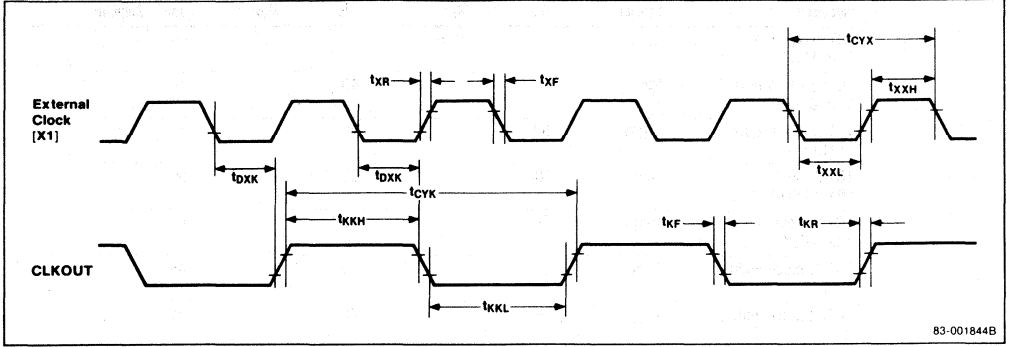
Parameter	Symbol	μPD70208-8		μPD70208-10		Unit	Test Condition
		Min	Max	Min	Max		
TxD delay time from TOUT1 <sup>+</sup>	tDTX		500		200	ns	
TCTL2 setup time from CLKOUT <sup>+</sup>	tSGK	50		40		ns	
TCTL2 setup time to TCLK <sup>†</sup>	tSGTK	50		40		ns	
TCTL2 hold time after CLKOUT <sup>+</sup>	tHKG	100		80		ns	
TCTL2 hold time after TCLK <sup>†</sup>	tHTKG	50		40		ns	
TCTL2 pulse width, high	tGGH	50		40		ns	
TCTL2 pulse width, low	tGGL	50		40		ns	
TOUT output delay time from CLKOUT <sup>+</sup>	tDKTO		200		150	ns	
TOUT output delay time from TOUT <sup>+</sup>	tDTKTO		150		100	ns	
TOUT output delay time from TCTL2 <sup>+</sup>	tDGTO		120		90	ns	
TCLK rise time	tTKR		25		25	ns	
TCLK fall time	tTKF		25		25	ns	
TCLK pulse width, high	tTKTKH	50		45		ns	
TCLK pulse width, low	tTKTKL	50		45		ns	
TCLK cycle time	tCYTK	124	Dc	100	Dc	ns	
RESET pulse width low	tRESET1	50		50		μs	After power on
	tRESET2	4 <sup>†</sup> CYK		4 <sup>†</sup> CYK			During operation

#### Notes:

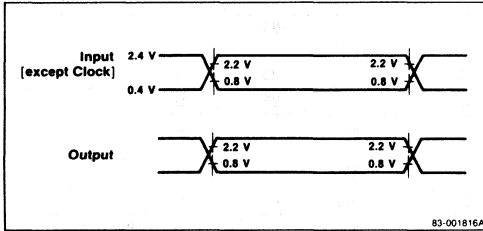
- (1) Delay from CLKOUT to DMA cycle MWR/IOWR outputs.
- (2) Delay from CLKOUT to BUFR/W, BUFEN, INTAK, REFRQ outputs and CPU cycle MWR/IOWR outputs.
- (3) RD represents IORD and MRD. WR represents IOWR and MWR.
- (4) This is specified to guarantee a read/write recovery time for I/O device.
- (5) This is specified to guarantee that REFRQ<sup>†</sup> is delayed from MRD<sup>†</sup> at all times.

**Timing Waveforms**

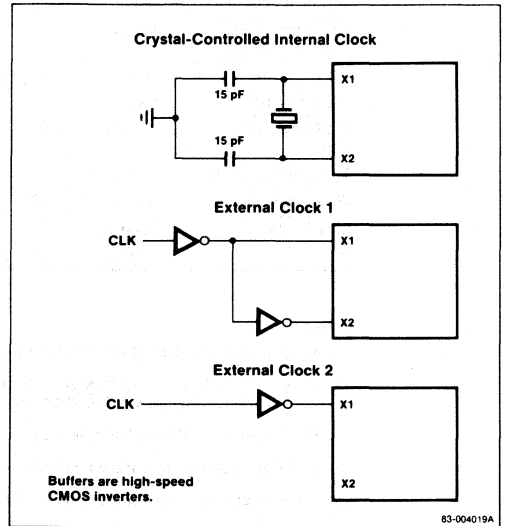
**Clock Timing**



**Timing Measurement Points**

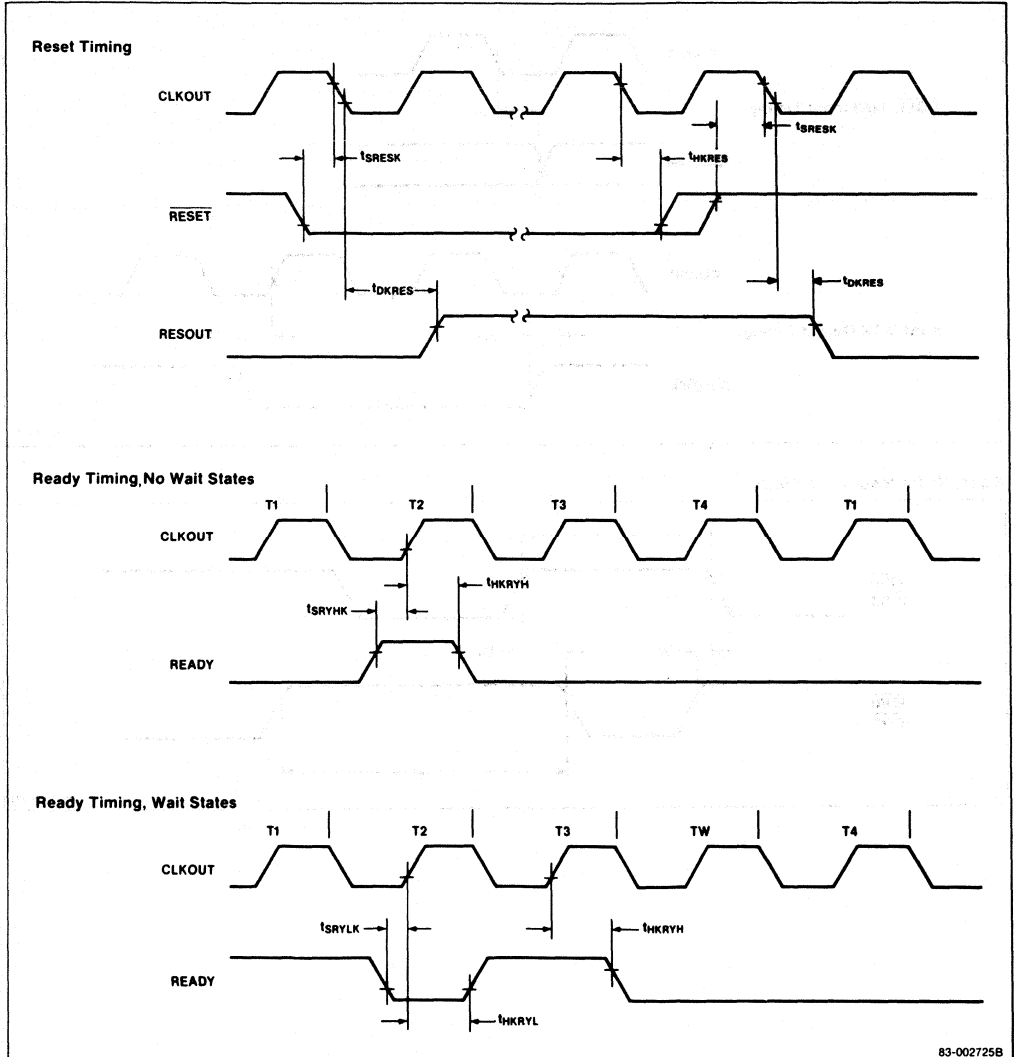


**Clock Input Configurations**



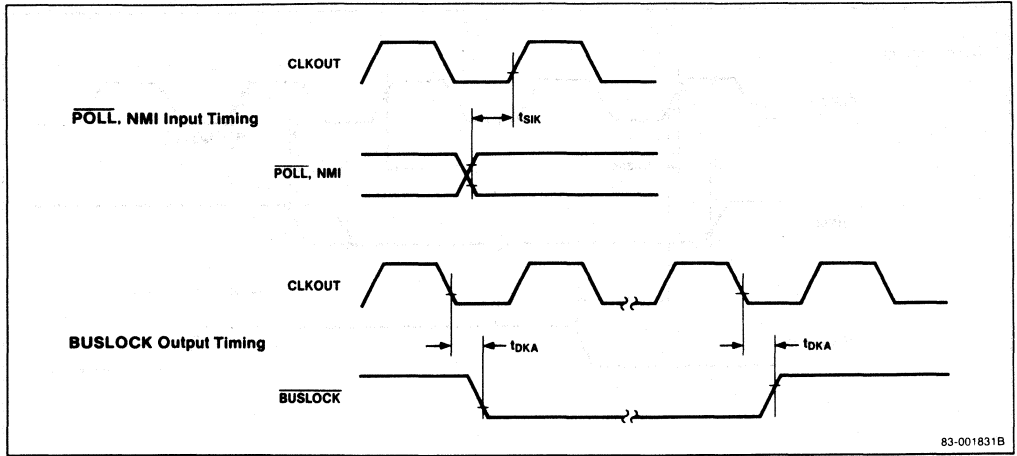
## Timing Waveforms (cont)

### Reset and Ready Timing



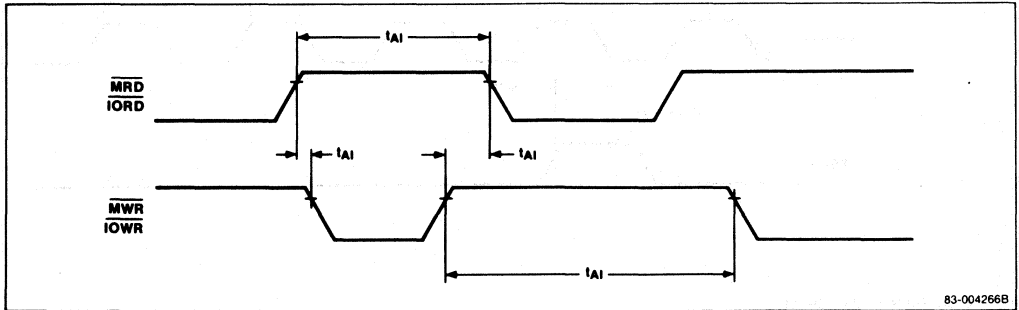
**Timing Waveforms (cont)**

**Poll, NMI, and Buslock Timing**



83-001831B

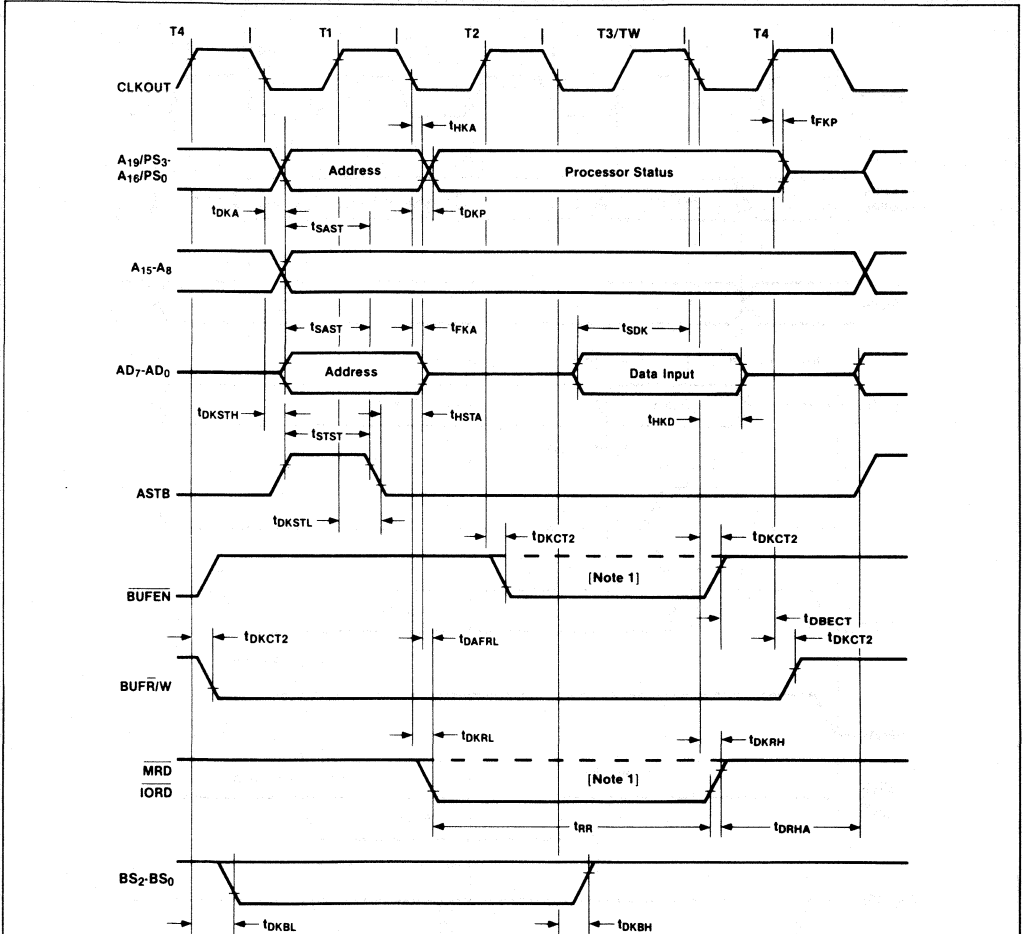
**Read/Write Recovery Time**



83-004266B

### Timing Waveforms (cont)

#### Read Timing

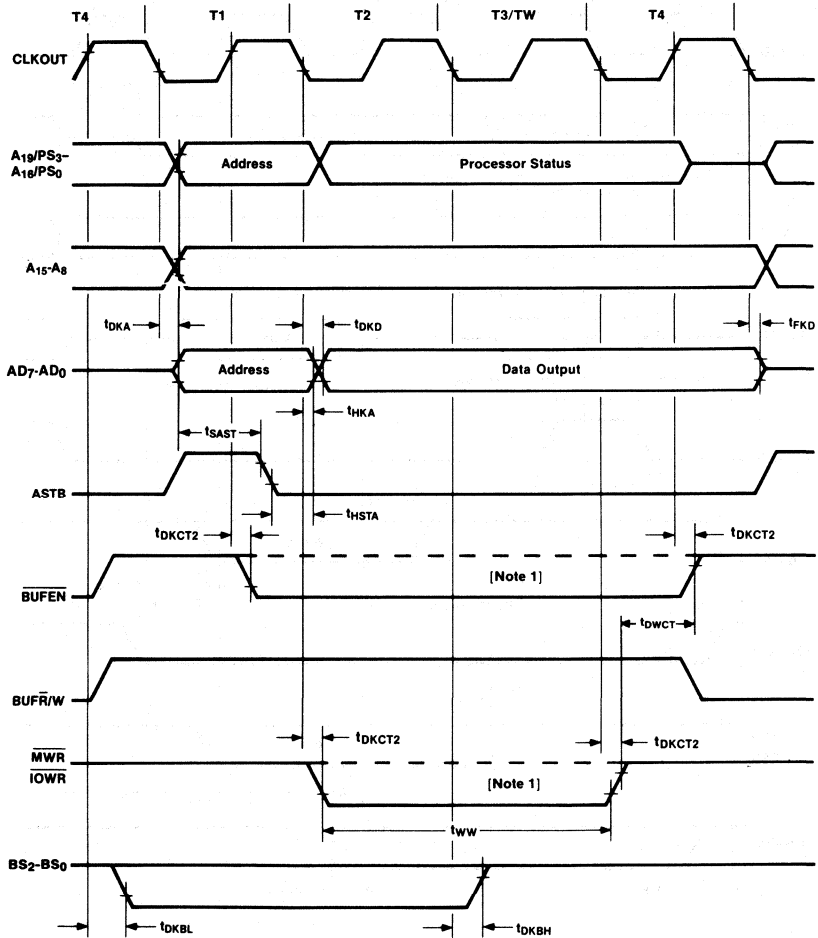


Note:  
 [1] Except internal I/O accesses.



**Timing Waveforms (cont)**

**Write Timing**

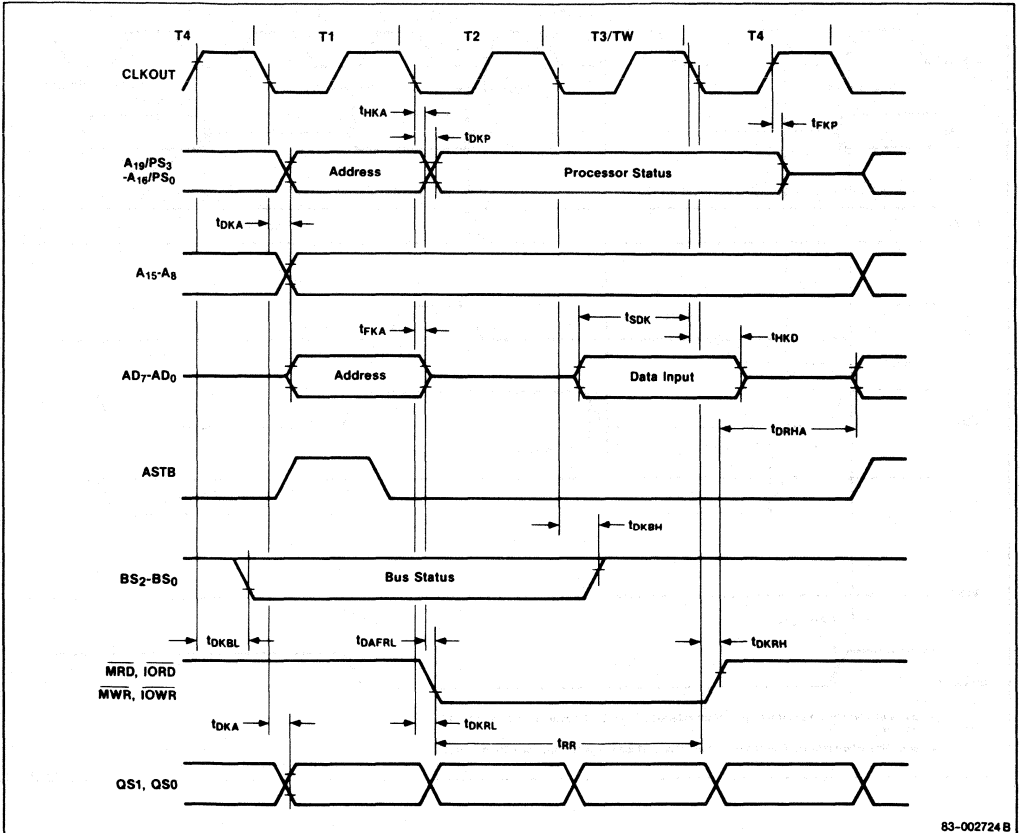


Note:

[1] Except internal I/O accesses.

### Timing Waveforms (cont)

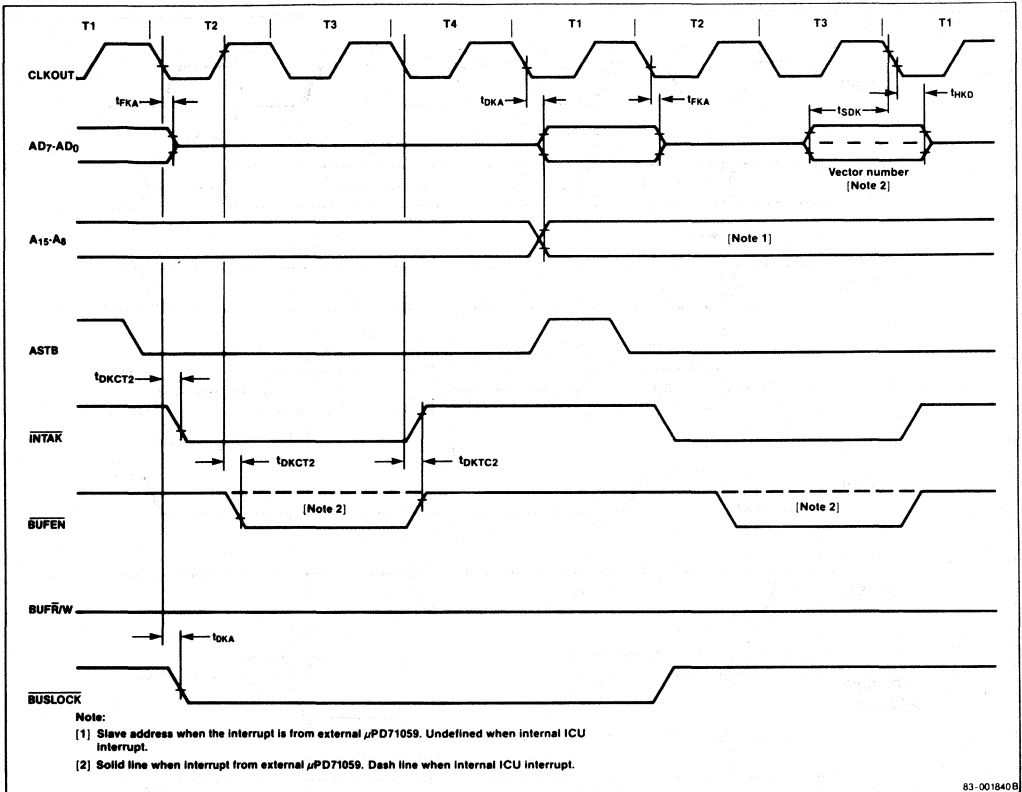
#### Status Timing



83-002724 B

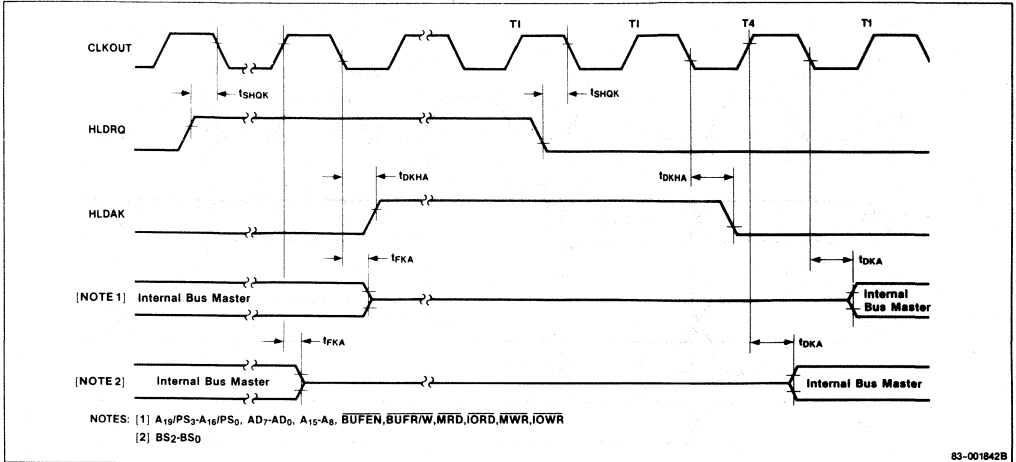
Timing Waveforms (cont)

Interrupt Acknowledge Timing

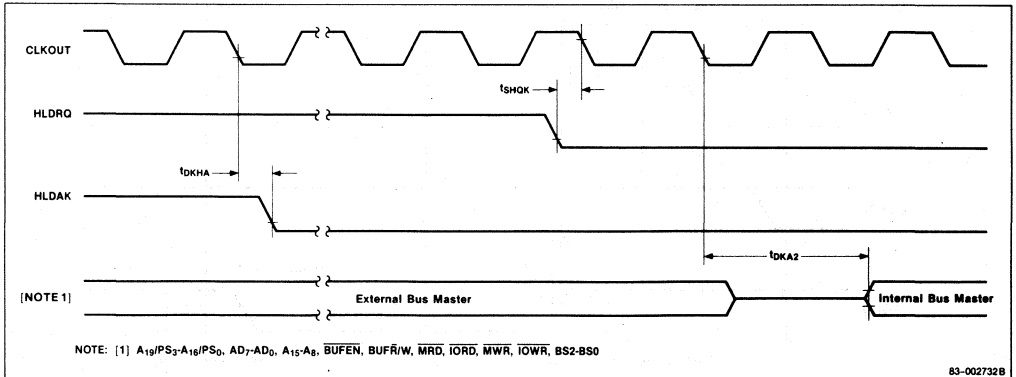


## Timing Waveforms (cont)

### HLDRQ/HLDAK Timing, Normal Operation

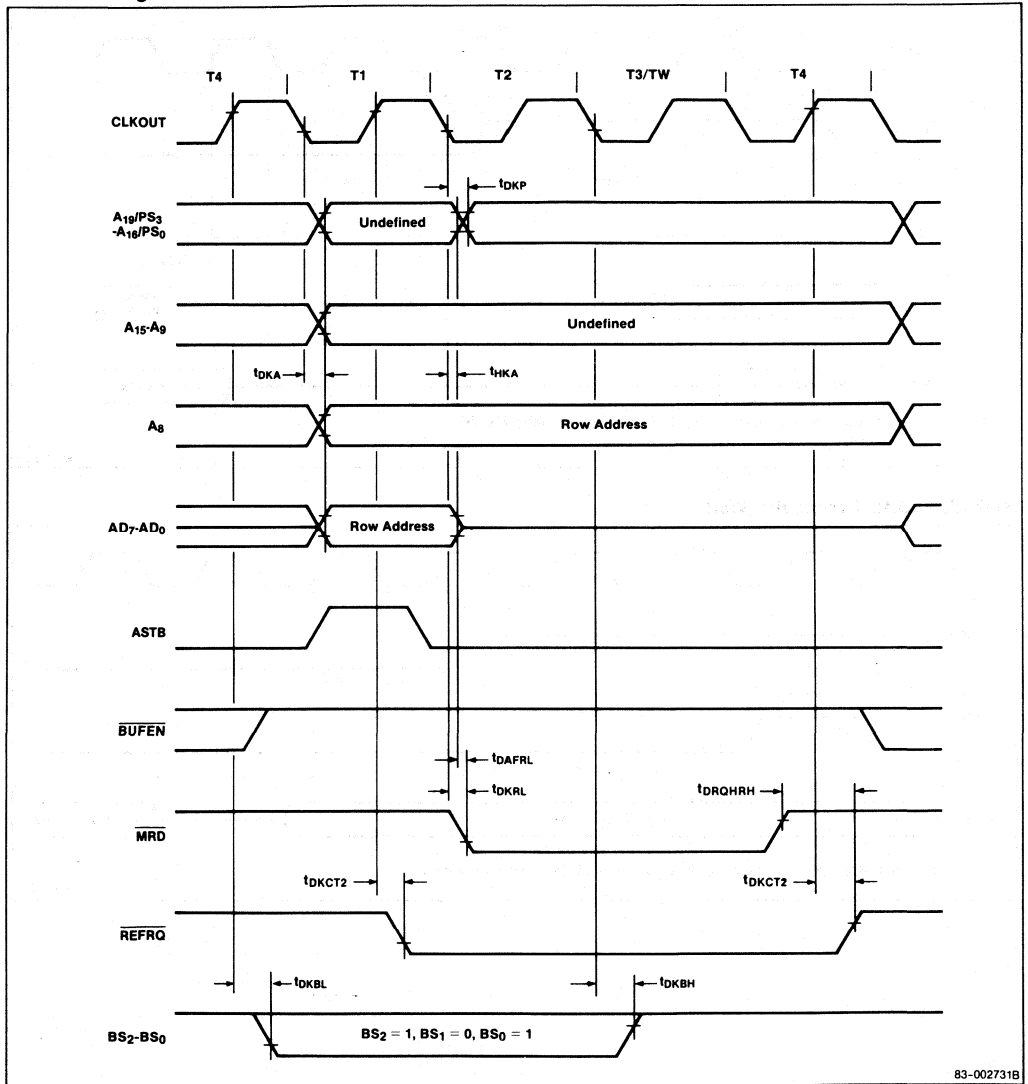


### HLDRQ/HLDAK Timing, Bus Wait



**Timing Waveforms (cont)**

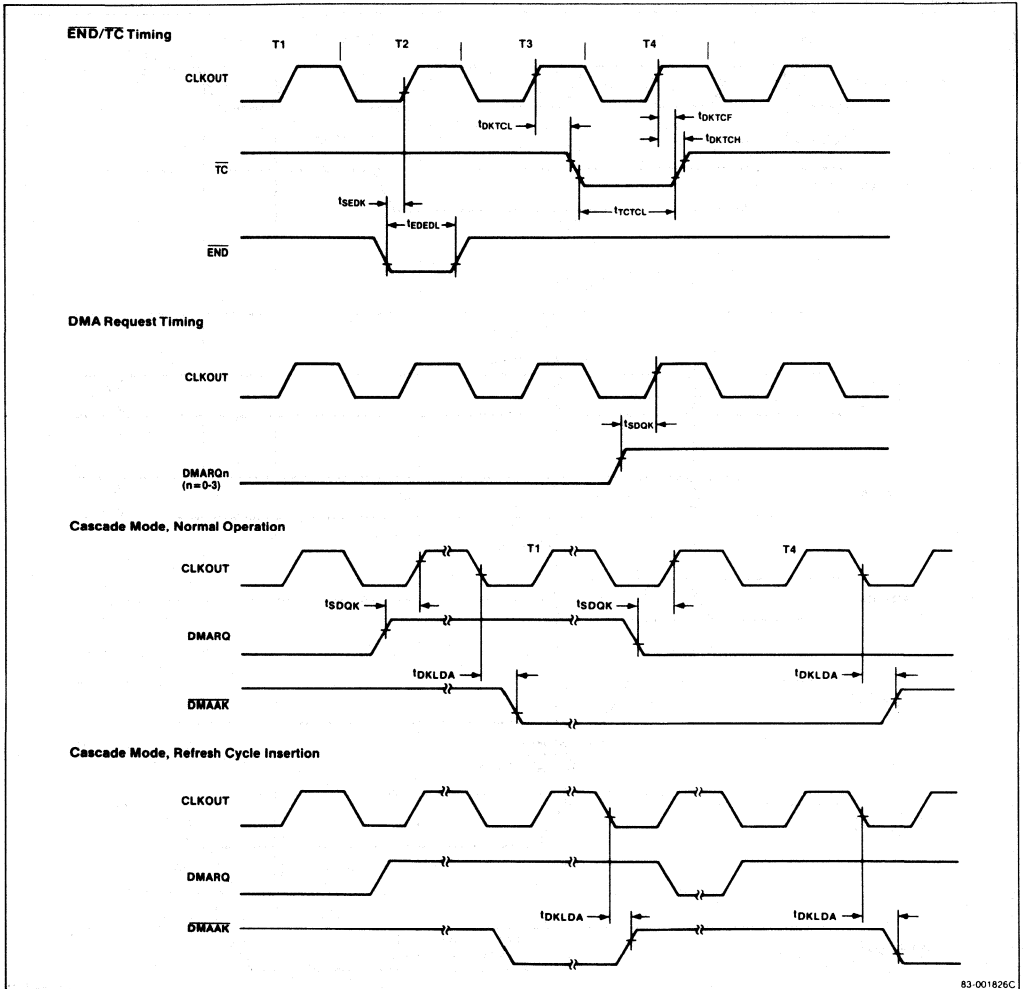
**Refresh Timing**





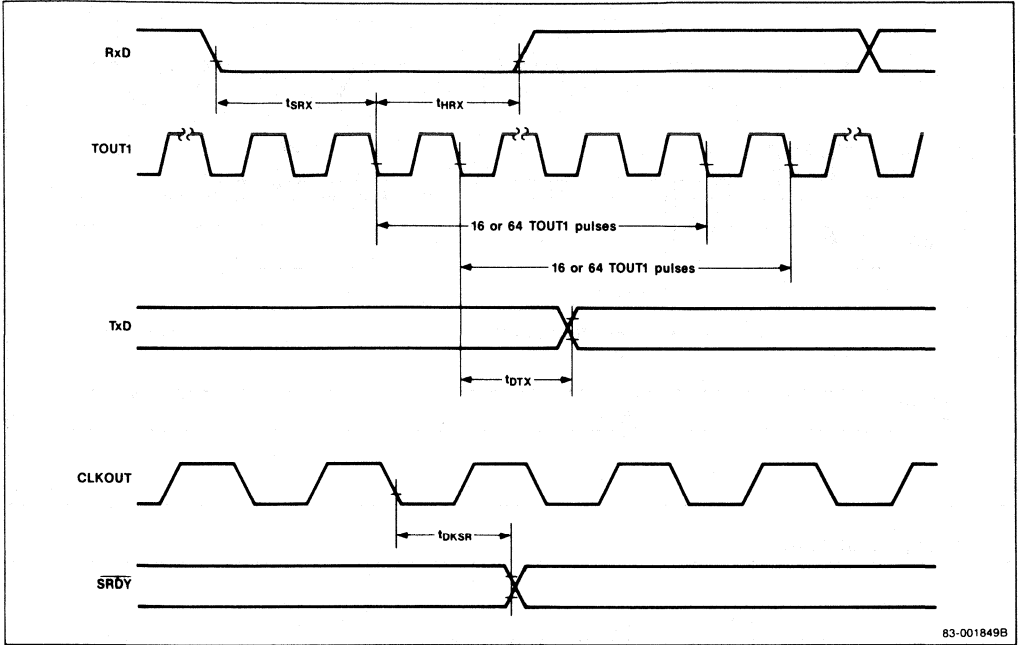
Timing Waveforms (cont)

DMA Timing

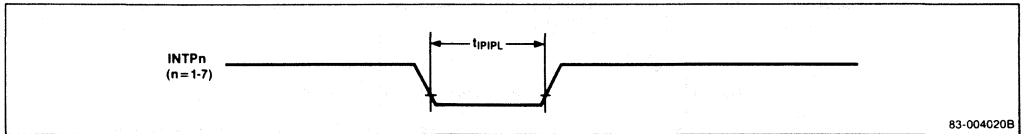


## Timing Waveforms (cont)

### SCU Timing



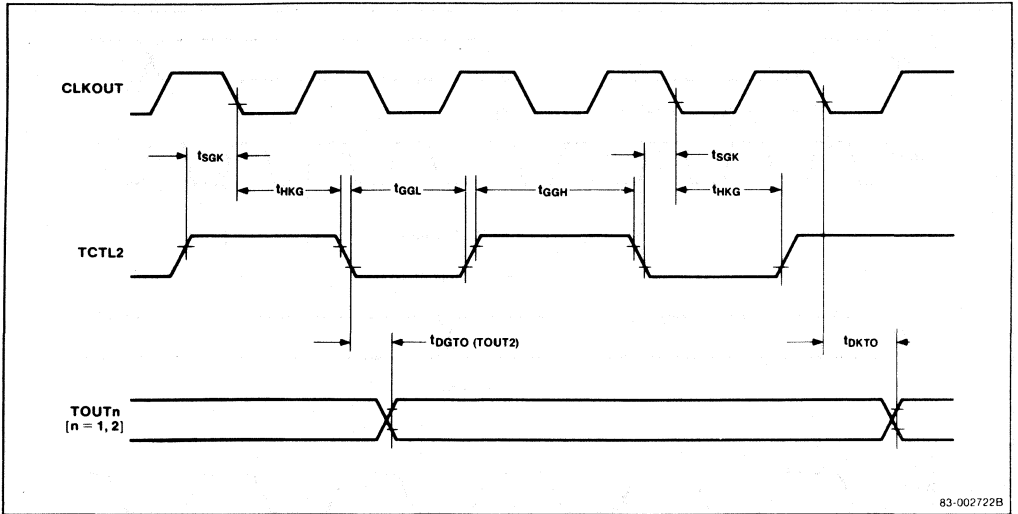
### ICU Timing



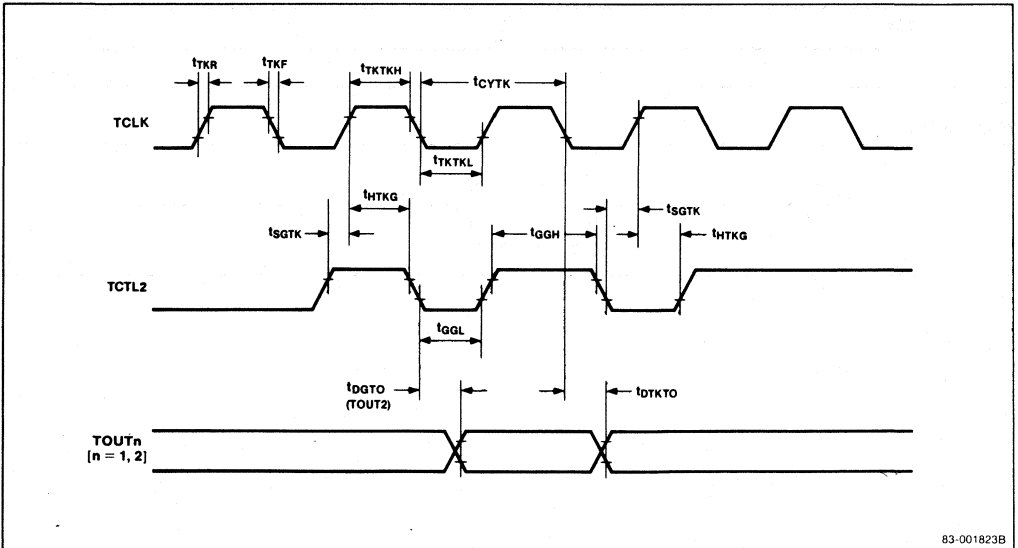


Timing Waveforms (cont)

TCU Timing, Internal Clock Source



TCU Timing, TCLK Source



### Functional Description

Refer to the μPD70208 block diagram for an overview of the ten major functional blocks listed below.

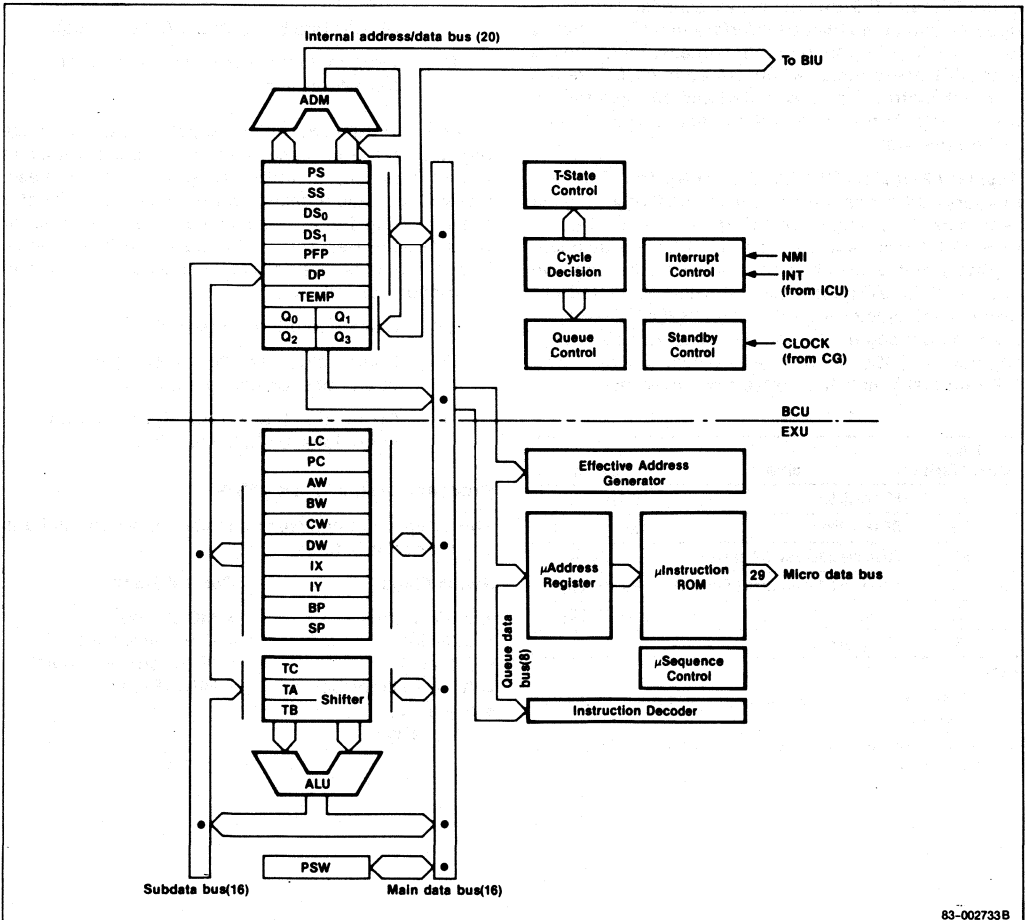
- Central processing unit (CPU)
- Clock generator (CG)
- Bus interface unit (BIU)
- Bus arbitration unit (BAU)
- Refresh control unit (RCU)
- Wait control unit (WCU)
- Timer/counter unit (TCU)
- Serial control unit (SCU)
- Interrupt control unit (ICU)
- DMA control unit (DMAU)

### Central Processing Unit

The μPD70208 CPU functions similarly to the CPU of the μPD70108 CMOS microprocessor. However, because the μPD70208 has internal peripheral devices, its bus architecture has been modified to permit sharing the bus with internal peripherals. The μPD70208 CPU is object code compatible with both the μPD70108/μPD70116 and the μPD8086/μPD8088 microprocessors.

Figure 1 is the μPD70208 CPU block diagram. A listing of the μPD70208 instruction set is in the final sections of this data sheet.

Figure 1. μPD70208 CPU Block Diagram



**Register Configuration**

**Program Counter [PC].** The program counter is a 16-bit binary counter that contains the program segment offset of the next instruction to be executed. The PC is incremented each time the microprogram fetches an instruction from the instruction queue. The contents of the PC are replaced whenever a branch, call, return, or break instruction is executed and during interrupt processing. At this time, the contents of the PC are the same as the prefetch pointer (PFP).

**Prefetch Pointer [PFP].** The prefetch pointer is a 16-bit binary counter that contains the program segment offset of the next instruction to be fetched for the instruction queue. Because instruction queue prefetch is independent of instruction execution, the contents of the PFP and PC are not always identical. The PFP is updated each time the bus interface unit (BIU) fetches an instruction for the instruction queue. The contents of the PFP are replaced whenever a branch, call, return or break instruction is executed and during interrupt processing. At this time, the contents of the PFP and PC are the same.

**Segment Registers [PS, SS, DS<sub>0</sub>, DS<sub>1</sub>].** The μPD70216 memory address space is divided into 64K-byte logical segments. A memory address is determined by the sum of a 20-bit base address (obtained from a segment register) and a 16-bit offset known as the effective address (EA). I/O address space is not segmented and no segment register is used. The four segment registers are program segment (PS), stack segment (SS), data segment 0 (DS<sub>0</sub>), and data segment 1 (DS<sub>1</sub>). The following table lists their offsets and overrides.

Default Segment Register	Offset	Override
PS	PFP register	None
SS	SP register	None
SS	Effective address (BP-based)	PS, DS <sub>0</sub> , DS <sub>1</sub>
DS <sub>0</sub>	Effective address (non BP-based)	PS, SS, DS <sub>1</sub>
DS <sub>0</sub>	IX register (1)	PS, SS, DS <sub>1</sub>
DS <sub>1</sub>	IY register (2)	None

**Note:**

- (1) Includes source block transfer, output, BCD string, and bit field extraction.  
 (2) Includes destination block transfer, input, BCD string, and bit field insertion.

**General-Purpose Registers.** The μPD70208 CPU contains four 16-bit, general-purpose registers (AW, BW, CW, DW), each of which can be used as a pair of 8-bit registers by dividing into upper and lower bytes (AH, AL, BH, BL, CH, CL, DH, DL). General-purpose registers may also be specified implicitly in an instruction. The implicit assignments are:

- AW Word multiplication/division, word I/O, data conversion
- AL Byte multiplication/division, byte I/O, BCD rotation, data conversion, translation
- AH Byte multiplication/division
- BW Translation
- CW Loop control, repeat prefix
- CL Shift/rotate bit counts, BCD operations
- DW Word multiplication/division, indirect I/O addressing

**Pointer [SP, BP] and Index Registers [IX, IY].** These registers serve as base pointers or index registers when accessing memory using one of the base, indexed, or base indexed addressing modes. Pointer and index registers can also be used as operands for word data transfer, arithmetic, and logical instructions. These registers are implicitly selected by certain instructions as follows.

- SP Stack operations, interrupts
- IX Source block transfer, BCD string operations, bit field extraction
- IY Destination block transfer, BCD string operations, bit field insertion

**Program Status Word [PSW]**

The program status word consists of six status flags and four control flags.

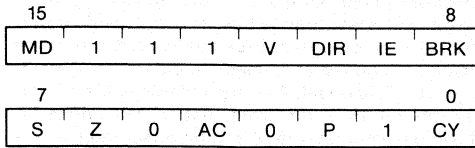
**Status Flags**

- V (Overflow)
- S (Sign)
- Z (Zero)
- AC (Auxiliary Carry)
- P (Parity)
- CY (Carry)

**Control Flags**

- MD (Mode)
- DIR (Direction)
- IE (Interrupt Enable)
- BRK (Break)

When pushed onto the stack, the word image of the PSW is as follows:



The status flags are set and cleared automatically depending upon the result of the previous instruction execution. Instructions are provided to set, clear, and complement certain status and control flags. Other flags can be manipulated by using the POP PSW instruction.

Between execution of the BRKEM and RETEM instructions, the native mode RETI and POP PSW instructions can modify the MD bit. Care must be exercised by emulation mode programs to prevent inadvertent alteration of this bit.

### CPU Architectural Features

The major architectural features of the μPD70208 CPU are:

- Dual data buses
- Effective address generator
- Loop counter
- PC and PFP

**Dual Data Buses.** To increase performance, dual data buses (figure 2) have been employed in the CPU to fetch operands in parallel and avoid the bottleneck of a single bus. For two-operand instructions and effective address calculations, the dual data bus approach is 30 percent faster than single-bus systems.

**Effective Address Generator.** Effective address (EA) calculation requires only two clocks regardless of the addressing mode complexity due to the hardware effective address generator (figure 3). When compared with microprogrammed methods, the hardware approach saves between 3 and 10 clock cycles during effective address calculation.

**Loop Counter and Shifters.** A dedicated loop counter is used to count the iterations of block transfer and multiple shift instructions. This logic offers a significant performance advantage over architectures that control block transfers and multiple shifts using microprogramming. Dedicated shift registers also speed up the execution of the multiply and divide instructions. Compared with microprogrammed methods, multiply and divide instructions execute approximately four times faster.

Figure 2. Dual Data Buses

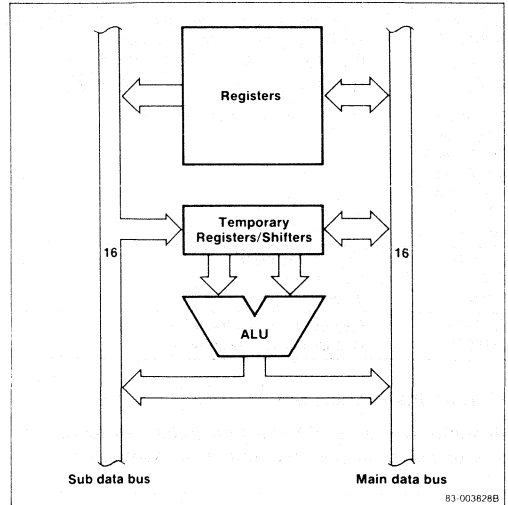
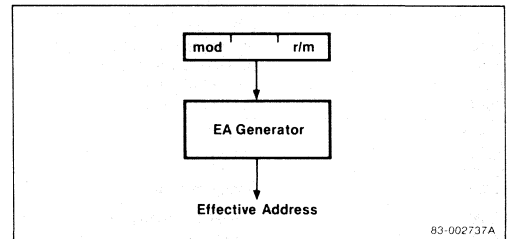


Figure 3. Effective Address Generator



**Program Counter and Prefetch Pointer.** The functions of instruction execution and queue prefetch are decoupled in the μPD70208. By avoiding a single instruction pointer and providing separate PC and PFP registers, the execution time of control transfers and the interrupt response latency can be minimized. Several clocks are saved by avoiding the need to readjust an instruction pointer to account for prefetching before computing the new destination address.

### Enhanced Instruction Set

In addition to the μPD8086/88 instruction set, the μPD70208 has added the following enhanced instructions.

Instruction	Function
PUSH imm	Push immediate data onto stack
PUSH R	Push all general registers onto stack
POP R	Pop all general registers from stack
MUL imm	Multiply register/memory by immediate data
SHL imm8	Shift/rotate by immediate count
SHR imm8	
SHRA imm8	
ROL imm8	
ROR imm8	
ROLC imm8	
RORC imm8	
CHKIND	Check array index
INM	Input multiple
OUTM	Output multiple
PREPARE	Prepare new stack frame
DISPOSE	Dispose current stack frame

### Unique Instruction Set

In addition to the μPD70208 enhanced instruction set, the following unique instructions are supported.

Instruction	Function
INS	Insert bit field
EXT	Extract bit field
ADD4S	BCD string addition BCD string subtraction BCD string comparison Rotate BCD digit left Rotate BCD digit right
SUB4S	
CMP4S	
ROL4	
ROR4	
TEST1	Test bit
SET1	Set bit
CLR1	Clear bit
NOT1	Complement bit
REPC	Repeat while carry set
REPNC	Repeat while carry cleared
FPO2	Floating point operation 2

**Bit Fields.** Bit fields are data structures that range in length from 1 to 16 bits. Two separate operations on bit fields, insertion and extraction, with no restrictions on the position of the bit field in memory are supported. Separate segment, byte offset, and bit offset registers are used for bit field insertion and extraction. Because of their power and flexibility, these instructions are highly effective for graphics, high-level languages, and data packing/unpacking applications.

Insert bit field (INS) copies the bit field of specified length (0 = 1 bit, 15 = 16 bits) from the AW register to the bit field addressed by DS1:IY:reg8 (figure 4). The bit field length can be located in any byte register or supplied as an immediate value. The value in reg8 is a bit field offset. A content of 0 selects bit 0 and 15 selects bit 15 of the word that DS0:IX points to. Following execution, the IY and bit offset register are updated to point to the start of the next bit field.

Bit field extraction (EXT) copies the bit field of specified length (0 = 1 bit, 15 = 16 bits) from the bit field addressed by DS0:IX:reg8 to the AW register (figure 5). If the bit field is less than 16 bits, it is right justified with a zero fill. The bit field length can be located in any byte register or supplied as immediate data. The value in reg8 is a bit field offset. A content of 0 selects bit 0 and 15 selects bit 15 of the word that DS0:IX points to. Following execution, the IX and bit offset register are updated to point to the start of the next bit field.

**Packed BCD Strings.** These instructions are provided to efficiently manipulate packed BCD data as strings (length from 1 to 254 digits) or as a byte data type with a single instruction.

BCD string arithmetic is supported by the ADD4S, SUB4S, and CMP4S instructions. These instructions allow the source string (addressed by DS0:IX) and the destination string (addressed by DS1:IY) to be manipulated with a single instruction. When the number of BCD digits is even, the Z and CY flags are set according to the result of the operation. If the number of digits is odd, the Z flag will not be correctly set unless the upper 4 bits of the result are zero. The CY flag will not be correctly set unless there is a carry out of the upper 4 bits of the result.

The two BCD rotate instructions (ROR4, ROL4) perform rotation of a single BCD digit in the lower half of the AL register through the register or memory operand.

**Bit Manipulation.** Four bit manipulation instructions have been added to the μPD70208 instruction set. The ability to test, set, clear, or complement a single bit in a register or memory operand increases code readability as well as performance over the logical operations traditionally used to manipulate bit data.

**Repeat Prefixes.** Two repeat prefixes (REPC, REPNC) allow conditional block transfer instructions to use the state of the CY flag as a terminating condition. The use of these prefixes allows inequalities to be used when working on ordered data, increasing the performance of searching and sorting algorithms.

**Floating Point Operation Instructions.** Two floating point operation (FPO) instruction types are recognized by the μPD70208 CPU. These instructions are detected by the CPU, which performs any auxiliary processing such as effective address calculation and the initial bus cycle if specified by the instruction. It is the responsibility of the external coprocessor to latch the address information and data (if a read cycle) from the bus and complete the execution of the instruction.

Figure 4. Bit Field Insertion

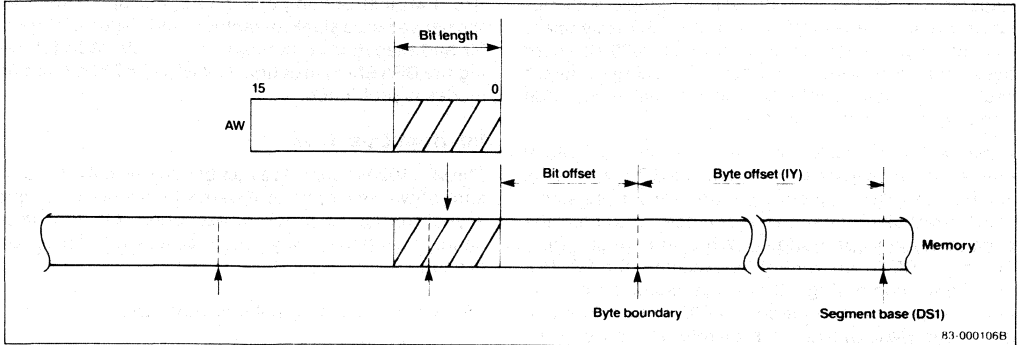
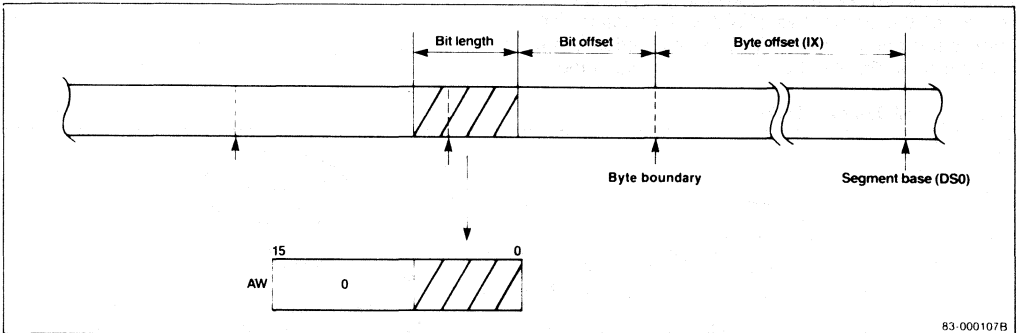


Figure 5. Bit Field Extraction



**8080 Emulation Mode.** The μPD70208 CPU can operate in either of two modes; see figure 6. Native mode allows the execution of the μPD8086/88, enhanced and unique instructions. The other operating mode is 8080 emulation mode, which allows the entire μPD8080AF instruction set to be executed. A mode (MD) flag is provided to distinguish between the two operating modes. Native mode is active when MD is 1 and 8080 emulation mode is active when MD is 0.

Two instructions are provided to switch from native to 8080 emulation mode and return back. Break for emulation (BRKEM) operates similarly to a BRK instruction, except that after the PSW has been pushed on the native mode stack, the MD flag becomes write-enabled and is cleared.

During 8080 emulation mode, the registers and flags of the 8080 are mapped onto the native mode registers and flags as shown below. Note that PS, SS, DS<sub>0</sub>, DS<sub>1</sub>, IX, IY, AH, and the upper half of the PSW registers are inaccessible to 8080 programs.

	μPD8080AF	μPD70208
Registers	A/PSW	AL/PSW (lower)
	B	CH
	C	CL
	D	DH
	E	DL
	H	BH
	L	BL
	SP PC	BP PC
Flags	C	CY
	Z	Z
	S	S
	P	P
	AC	AC

During 8080 emulation mode, the BP register functions as the 8080 stack pointer. The use of separate stack pointers prevents inadvertent damage to the native mode stack pointer by emulation mode programs.

The 8080 emulation mode PC is combined with the PS register to form the 20-bit physical address. All emulation mode data references use DS0 as the segment register. For compatibility with older 8080 software these registers must be equal. By using different segment register contents, separate 64K-byte code and data spaces are possible.

Either an NMI or maskable interrupt will cause the 8080 emulation mode to be suspended. The CPU pushes the PS, PC, and PSW registers on the native mode stack, sets the MD bit (indicating native mode), and enters the specified interrupt handler. When the return from interrupt (RETI) instruction is executed, the PS, PC, and PSW (containing MD=0) are popped from the native stack and execution in 8080 emulation mode continues. Reset will also force a return to native mode.

The 8080 emulation mode programs also have the capability to invoke native mode interrupt handlers by means of the call native (CALLN) instruction. This instruction operates like the BRK instruction except that the saved PSW indicates 8080 emulation mode.

To exit 8080 emulation mode, the return from emulation (RETEM) instruction pops the PS, PC, and PSW from the native mode stack, disables modification of the MD bit, and execution continues with the instruction following the BRKEM instruction. Nesting of 8080 emulation modes is prohibited.

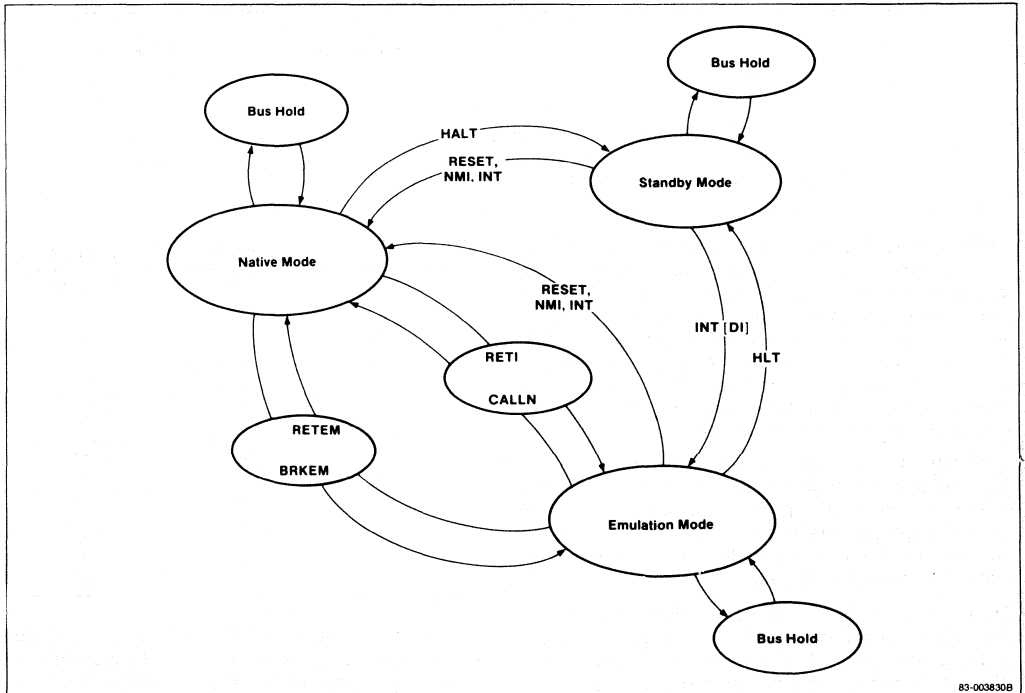
### Interrupt Operation

The μPD70208 supports a number of external interrupts and software exceptions. External interrupts are events asynchronous to program execution. On the other hand, exceptions always occur as a result of program execution.

The two types of external interrupts are:

- Nonmaskable interrupt (NMI)
- Maskable interrupt (INT)

Figure 6. μPD70208 Modes



The six software exceptions are:

- Divide error (DIV, DIVU instructions)
- Array bound error (CHKIND instruction)
- Break on overflow (BRKV instruction)
- Break (BRK, BRK3 instructions)
- Single step (BRK bit in PSW set)
- Mode switch (BRKEM, CALLN instructions)

Interrupt vectors are determined automatically for exceptions and the NMI interrupt or supplied by hardware for maskable interrupts. The 256 interrupt vectors are stored in a table (figure 7) located at address 00000H. Vectors 0 to 5 are predetermined and vectors 6 to 31 are reserved. Interrupt vectors 32 to 255 are available for use by application software.

Each vector is made up of two words. The word located at the lower address contains the new PC for the interrupt handler. The word at the next-higher address is the new PS value for the interrupt handler. These must be initialized by software at the start of a program.

Nonmaskable interrupts and maskable interrupts (when enabled) are normally serviced following the execution of the current instruction. However, the following cases are exceptions to this rule and the occurrence of the interrupt will be delayed until after the execution of the next instruction.

- Moves to/from segment registers
- POLL instruction
- Instruction prefixes
- EI instruction (maskable interrupts only)

Another special case is the block transfer instructions. These instructions are interruptable and resumable, but because of the asynchronous operation of the BIU, the actual occurrence of the interrupt may be delayed up to three bus cycles.

### Standby Mode

The μPD70208 CPU has a low-power standby mode, which can dramatically reduce power consumption during idle periods. Standby mode is entered by simply executing a native or 8080 emulation HALT instruction; no external hardware is required. All other peripherals such as the timer/counter unit, refresh control unit, and DMA control unit continue to operate as programmed.

During standby mode, the clock is distributed only to the circuits required to release the standby mode. When a RESET, NMI, or INT event is detected, the standby mode is released. Both NMI and unmaskable interrupts are processed before control returns to the instruction following the HALT. In the case of the INT input being masked, execution will begin with the instruction immediately following the HALT instruction without an intervening interrupt acknowledge bus cycle. When maskable interrupts are again enabled, the interrupt will be serviced.

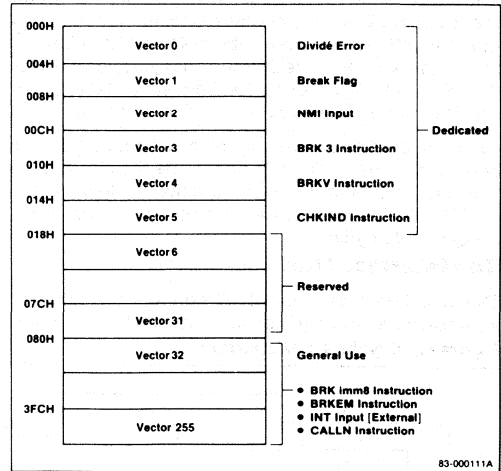
Output signal states in the standby mode are listed below.

Output Signal	Status in Standby Mode
INTAK, BUFEN, MRD, MWR, IOWR, IORD	High level
BS <sub>2</sub> -BS <sub>0</sub> (Note 2)	High level
QS <sub>1</sub> -QS <sub>0</sub> , ASTB	Low level
BUSLOCK	High level (low level if the HALT instruction follows the BUSLOCK prefix)
BUFR/W, A <sub>19</sub> -A <sub>16</sub> /PS <sub>3</sub> -PS <sub>0</sub> , A <sub>15</sub> -A <sub>8</sub> , AD <sub>7</sub> -AD <sub>0</sub>	High or low level

#### Note:

- (1) Output pin states during refresh and DMA bus cycles will be as defined for those operations.
- (2) Halt status is presented prior to entering the passive state.

Figure 7. Interrupt Vector Table



### Differences between μPD70208 and μPD70108

#### Clocks

The execution time of instructions with memory access differ between the μPD70208 and μPD70108 as given in the table below.

Instructions	Clocks	
	μPD70108	μPD70208
Instruction with memory read	n	n - 1
Instruction with memory write	n	n - 2
Instruction with memory read and memory write	n	n - 3



**Bus Status Signals**

The status signals BS2, BS1 and BS0 of μPD70208 have some different meaning compared to the μPD70108.

Regarding the μPD70108, these signals show the bus cycle state of the CPU, but in μPD70208 these signals indicate the bus cycle state of the following functional units:

- CPU
- DMAU
- REFU

Differences on bus status signals				
BS2	BS1	BS0	Bus cycle	
			μPD70108	μPD70208
1	0	1	Memory read	<ul style="list-style-type: none"> <li>● CPU's memory read</li> <li>● Refresh cycle</li> <li>● Memory to I/O DMA cycle</li> <li>● Verify cycle (DMAU)</li> </ul>
1	1	0	Memory write	<ul style="list-style-type: none"> <li>● CPU's memory write</li> <li>● I/O to Memory DMA cycle</li> </ul>

**Clock Generator**

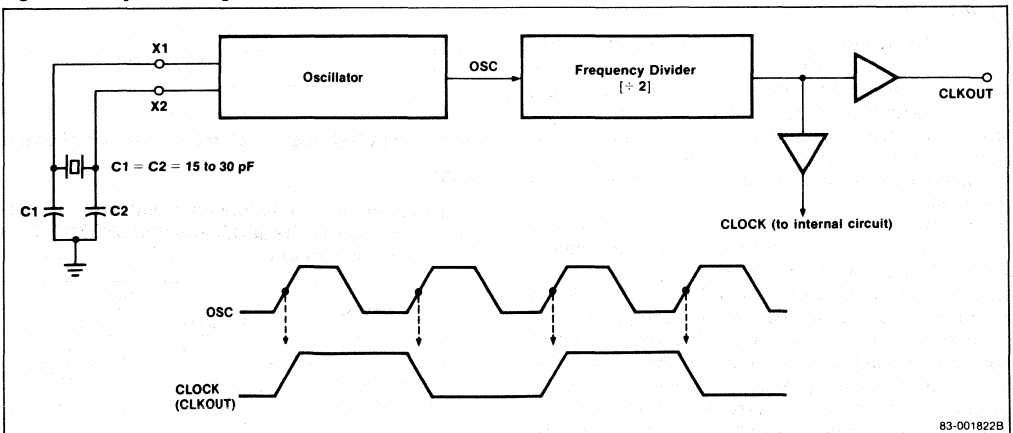
The clock generator (CG) generates a clock signal half the frequency of a parallel-resonant, fundamental mode crystal connected to pins X1 and X2. Figure 8 shows the recommended circuit configuration. Capacitors C1 and C2, required for frequency stability, are selected to match the crystal load capacitance.

External clock sources are also accommodated as shown in figure 9. The CG distributes the clock to the CLKOUT pin and to each functional block of the μPD70208. The generated clock signal has a 50-percent duty cycle.

**Bus Interface Unit**

The bus interface unit (BIU) controls the external address, data, and control buses for the three internal

**Figure 8. Crystal Configuration**

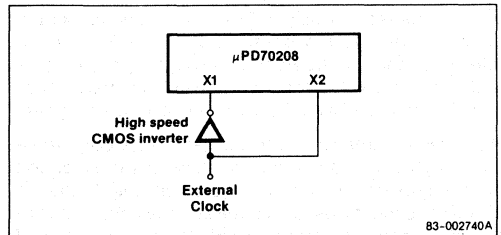


83-001822B

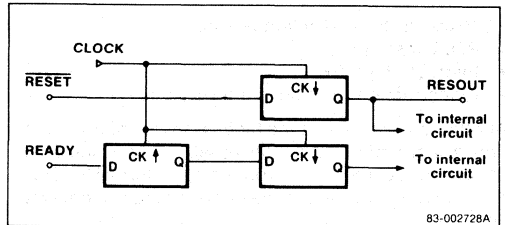
bus masters: CPU, DMA control unit (DMAU), and refresh control unit (RCU). The BIU is also responsible for synchronization of the RESET and READY inputs with the clock. The synchronized reset signal is used internally by the μPD70208 and provided externally at the RESOUT pin as a system-wide reset. The synchronized READY signal is combined with the output of the wait control unit (WCU) and is distributed internally to the CPU, DMAU, and RCU. Figure 10 shows the synchronization of RESET and READY.

The BIU also has the capability of overlapping the execution of the next instruction with memory write bus cycles. There is no overlap of instruction execution with read or I/O write bus cycles.

**Figure 9. External Oscillator Configuration**



**Figure 10. RESET/READY Synchronization**



### Bus Arbitration Unit

The bus arbitration unit (BAU) arbitrates the external address, data and control buses between the internal CPU, DMAU, and RCU bus requesters and an external bus master. The BAU bus priorities from the highest priority requester to the lowest are:

- RCU (Demand mode)
- DMAU
- HLDRQ
- CPU
- RCU (Normal mode)

Note that RCU requests the bus at either the highest or lowest priority depending on the status of the refresh request queue. Bus masters other than the CPU are prohibited from using the bus when the CPU is executing an instruction containing a BUSLOCK prefix. Therefore, caution should be exercised when using the BUSLOCK prefix with instructions having a long execution time.

If a bus master with higher priority than the current bus master requests the bus, the BAU inactivates the current bus master's acknowledge signal. When the BAU sees the bus request from the current master go inactive, the BAU gives control of the bus to the higher priority bus master. Whenever possible, the BAU performs bus switching between internal bus masters without the introduction of idle bus cycles, enhancing system throughput.

### System I/O Area

The I/O address space from addresses FF00H to FFFFH is reserved for use as the system I/O area. Located in this area are the 12 μPD70208 registers that

determine the I/O addressing, enable/disable peripherals, and control pin multiplexing. Byte I/O instructions must be used to access the system I/O area.

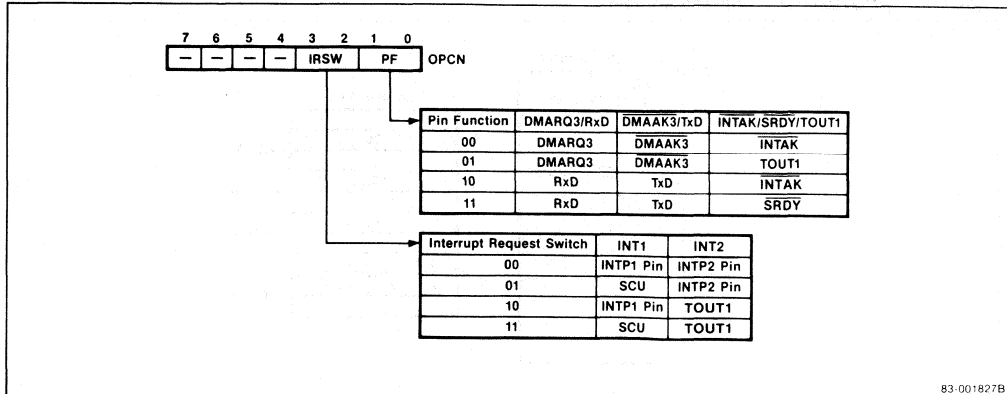
I/O Address	Register	Operation
FFFFH	Reserved	—
FFFEH	OPCN	Read/Write
FFFDH	OPSEL	Read/Write
FFFCH	OPHA	Read/Write
FFFBH	DULA	Read/Write
FFFAH	IULA	Read/Write
FFF9H	TULA	Read/Write
FFF8H	SULA	Read/Write
FFF7H	Reserved	—
FFF6H	WCY2	Read/Write
FFF5H	WCY1	Read/Write
FFF4H	WMB	Read/Write
FFF3H	Reserved	—
FFF2H	RFC	Read/Write
FFF1H	Reserved	—
FFF0H	TCKS	Read/Write

### On-Chip Peripheral Connection Register

The on-chip peripheral connection (OPCN) register controls multiplexing of the μPD70208 multiplexed pins. Figure 11 shows the format of the OPCN register. The interrupt request switch (IRSW) field controls multiplexing of ICU interrupt inputs INT1 and INT2. The output of an internal peripheral or an external interrupt source can be selected as the INT1 and INT2 inputs to the ICU.

The pin function (PF) field in the OPCN selects one of four possible states for the DMARQ3/RxD, DMAAK3/TxD, and INTAK/TOUT1/SRDY pins. Bit 0 of the

Figure 11. OPCN Register Format



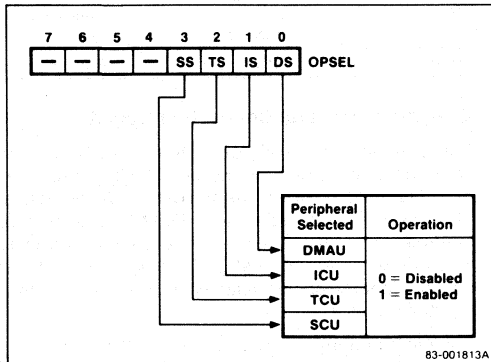
## μPD70208 (V40)

OPCN controls the function of the  $\overline{\text{INTAK}}$ /TOUT1/SRDY pin. If cleared,  $\overline{\text{INTAK}}$  will appear on this output pin. If bit 0 is set, either TOUT1 or SRDY will appear at the output depending on the state of bit 1. If bit 1 is cleared, DMA channel 3 I/O signals will appear on the DMARQ3/RxD and DMAAK3/TxD pins. If the SCU is to be used, bit 1 of the PF field must be set.

### On-Chip Peripheral Selection Register

The on-chip peripheral selection (OPSEL) register is used to enable or disable the μPD70208 internal peripherals. Figure 12 shows the format of the OPSEL register. Any of the four (DMAU, TCU, ICU, SCU) peripherals can be independently enabled or disabled by setting or clearing the appropriate OPSEL bit.

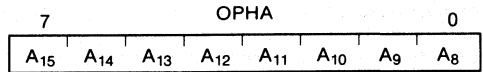
Figure 12. OPSEL Register Format



### Internal Peripheral Relocation Registers

The five internal peripheral relocation registers (figure 13) are used to fix the I/O addresses of the DMAU, ICU, TCU, and SCU. The on-chip peripheral high-address (OPHA) register is common to all four internal peripherals and fixes the high-order byte of the 16-bit I/O address. The individual DMAU low-address (DULA) register, ICU low-address (IULA) register, TCU low-address (TULA) register, and the SCU low-address (SULA) register select the low-order byte of the I/O addresses for the DMAU, ICU, TCU, and SCU peripherals.

The contents of the OPHA register are:



The formats for the individual internal peripheral registers appear below. Since address checking is not performed, do not overlap two peripheral I/O address spaces.

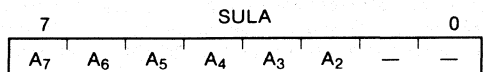
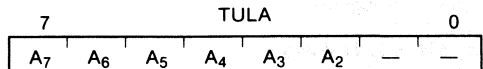
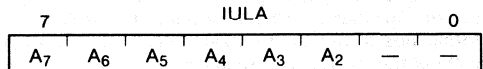
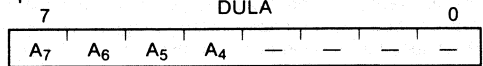
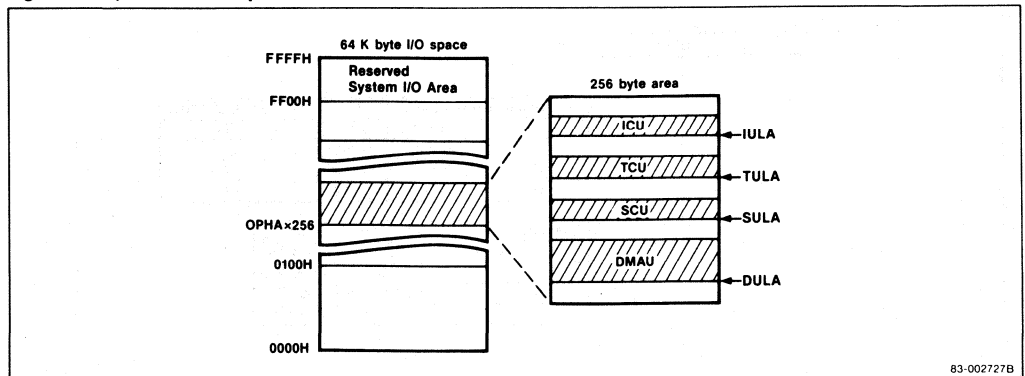


Figure 13. μPD70208 Peripheral Relocation



### Timer Clock Selection Register

The timer clock selection (TCKS) register selects the clock source for each of the timer/counters as well as the divisor for the internal clock prescaler. Figure 14 shows the format of the TCKS register. The clock source for each timer/counter is independently selected from either the prescaled internal CPU clock or from an external clock source (TCLK). The internal clock is derived from the CLKOUT signal and can be divided by 2, 4, 8, or 16 before being presented to the clock select logic.

### Refresh Control Unit

The refresh control unit (RCU) refreshes external dynamic RAM devices by outputting a 9-bit row address on address lines A<sub>8</sub>-A<sub>0</sub> and performing a memory read bus cycle. External logic can distinguish a refresh bus cycle by monitoring the refresh request (REFRQ) pin. Following each refresh bus cycle, the refresh row counter is incremented.

The refresh control (RFC) register in the system I/O area contains two fields. The refresh enable field enables or disables the refreshing function. The refresh timer (RTM) field selects a refresh interval to match the dynamic memory refresh requirements. Figure 15 shows the format for the RFC register.

To minimize the impact of refresh on the system bus bandwidth, the μPD70208 utilizes a refresh request queue to store refresh requests and perform refresh bus cycles in otherwise idle bus cycles.

The RCU normally requests the bus as the lowest-priority bus requester (normal mode). However, if seven refresh requests are allowed to accumulate in the RCU refresh request queue, the RCU will change to the highest-priority bus requester (demand mode).

The RCU will then perform back-to-back refresh cycles until three requests remain in the queue. This guarantees the integrity of the DRAM system while maximizing performance.

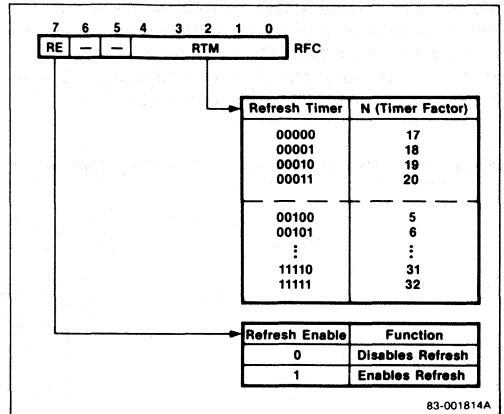
The refresh count interval can be calculated as follows:

$$\text{Refresh interval} = 8 \times N \times t_{\text{CYK}}$$

where N is the timer factor selected by the RTM field.

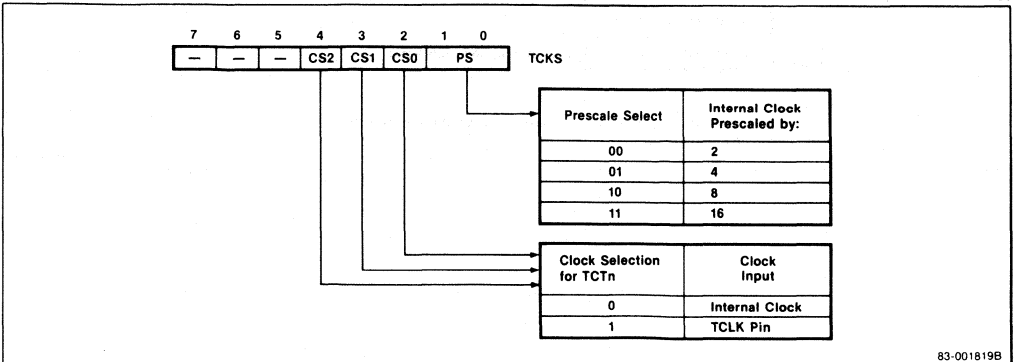
When the μPD70208 is reset, the RE field in the RFC register is unaffected and the RTM field is set to 01000 (N = 9). No refresh bus cycles occur while RESET is asserted.

Figure 15. Refresh Control Register



3

Figure 14. Timer Clock Selection Register



**Wait Control Unit**

The wait control unit (WCU) inserts from zero to three wait states into a bus cycle in order to compensate for the varying access times of memory and I/O devices. The number of wait states for CPU, DMAU, and RCU bus cycles is separately programmable. In addition, the memory address space is divided into three independent partitions to accommodate a wide range of system designs. RESET initializes the WCU to insert three wait states in all bus cycles. This allows operation with slow memory and peripheral devices before the initialization of the WCU registers.

The three system I/O area registers that control the WCU are wait cycle 1 (WCY1), wait cycle 2 (WCY2), and wait state memory boundary (WMB). The WCU always inserts wait states corresponding to the wait count programmed in WCY1 or WCY2 registers into a bus cycle, regardless of the state of the external READY input. After the programmed number of wait states occurs, the WCU will insert Tw states as long as

the READY pin remains inactive. When READY is again asserted, the bus cycle continues with T4 as the next cycle. The μPD70208 internal peripherals never require wait states; four clock cycles will terminate an internal peripheral bus cycle.

**CPU Wait States**

The WMB register divides the 1M-byte memory address space into three independent partitions: lower, middle, and upper. Figure 16 shows the WMB register format.

Initialization software can then set the number of wait states for each memory partition and the I/O partition via the WCY1 register (figure 17).

**DMA and Refresh Wait States**

The WCY2 register (figure 18) specifies the number of wait states to be automatically inserted in DMA and refresh bus cycles. DMA wait states must be set to the maximum of the DMA memory and I/O partitions. Refresh wait states should be set to the maximum value of all DRAM memory partitions.

**Figure 16. Wait State Memory Boundary Register**

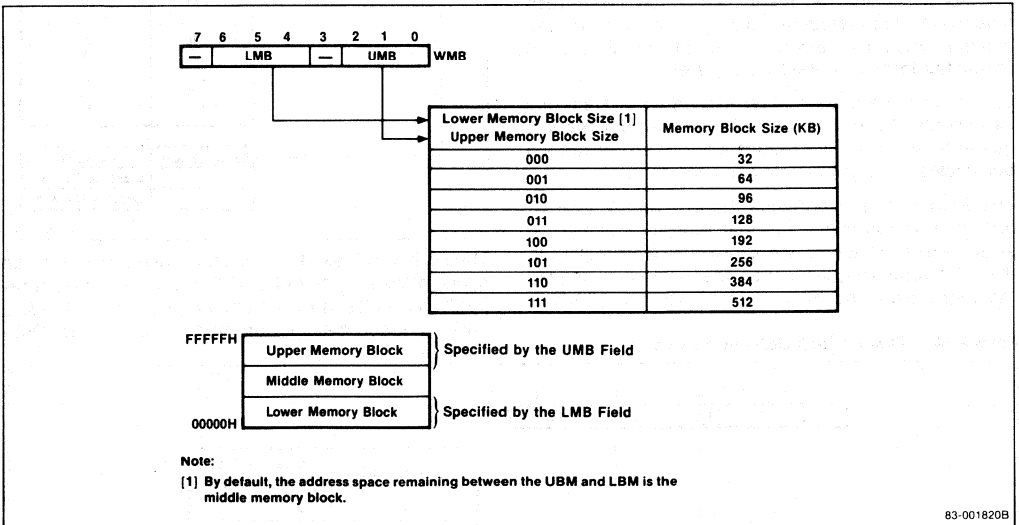


Figure 17. Wait Cycle 1 Register

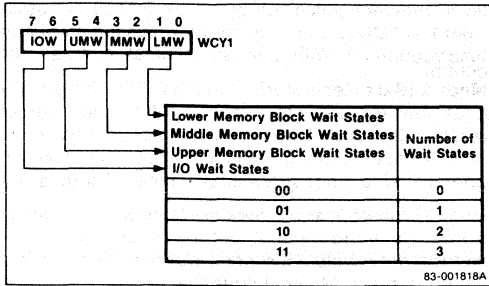
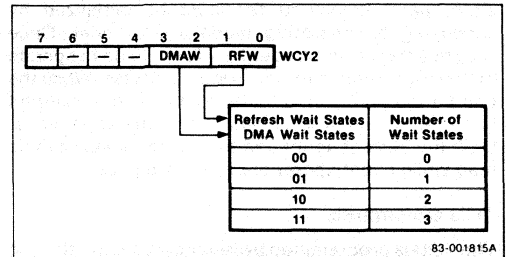


Figure 18. Wait Cycle 2 Register



## Timer/Counter Unit

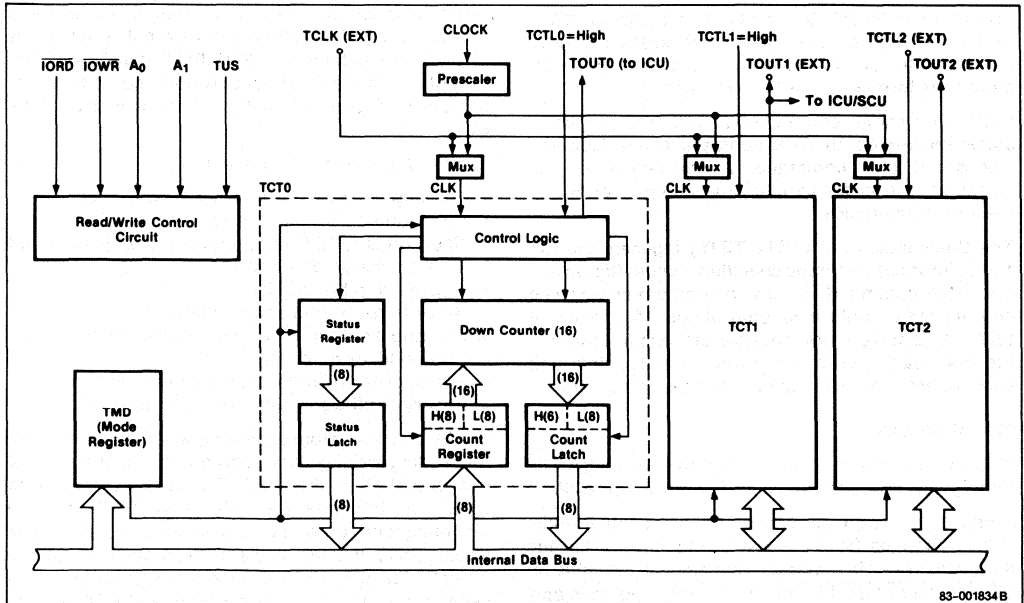
The timer/counter unit (TCU) provides a set of three independent 16-bit timer/counters. The output signal of timer/counter 0 is hardwired internally as an interrupt source. The output of timer/counter 1 is available internally as an interrupt source, used as a baud rate generator, or used as an external output. The timer/counter 2 output is available as an external output. Due to mode restrictions, the TCU is a subset of the

μPD71054 Programmable Timer/Counter. Figure 19 shows the internal block diagram of the TCU.

The TCU has the following features:

- Three 16-bit timer/counters
- Six programmable count modes
- Binary/BCD counting
- Multiple latch command
- Choice of two clock sources

Figure 19. TCU Block Diagram



Because RESET leaves the TCU in an uninitialized state, each timer/counter must be initialized by specifying an operating mode and a count. Once programmed, a timer/counter will continue to operate in that mode until another mode is selected. When the count has been written to the counter and transferred to the down counter, a new count operation starts. Both the current count and the counter status can be read while count operations are in progress.

TCU Commands

The TCU is programmed by issuing I/O instructions to the I/O port addresses programmed in the OPHA and TULA registers. The individual TCU registers are selected by address bits A1 and A0 as follows.

Table with 4 columns: A1, A0, Register, Operation. It lists four register operations: TCT0/TST0 (Read/Write Read), TCT1/TST1 (Read/Write Read), TCT2/TST2 (Read/Write Read), and TMD (Write).

The timer mode (TMD) register selects the operating mode for each timer/counter and issues the latch command for one or more timer/counters. Figure 20 shows the format for the TMD register.

Writes to the timer/counter 2-0 (TCT2-TCT0) registers stores the new count in the appropriate timer/counter. The count latch command is used before reading count data in order to latch the current count and prevent inaccuracies.

The timer status 2-0 (TST2-TST0) registers contain status information for the specified counter (figure 21). The latch command is used to latch the appropriate counter status before reading status information. If both status and counter data are latched for a counter, the first read operation returns the status data and subsequent read operations obtain the count data.

Count Modes

There are six programmable timer/counter modes. The timing waveforms for these modes are in figure 22.

Mode 0 [Interrupt on End of Count]. In this mode, TOUT changes from the low to high level when the specified count is reached. This mode is available on all timer/counters. Minimal number of counts is one.

Mode 1 [Retriggerable One-Shot]. In mode 1, a low-level one-shot pulse, triggered by TCTL2 is output from the TOUT2 pin. This mode is available only on timer/counter 2. With one as minimal number of counts.

Mode 2 [Rate Generator]. In mode 2, TOUT cyclically goes low for one clock period when the counter reaches the 0001H count. A counter in this mode operates as a frequency divider. All timer/counters can operate using mode 2. Minimal number of counts is 2.

Mode 3 [Square-Wave Generator]. Mode 3 is a frequency divider similar to mode 2, but the output has a symmetrical duty cycle. This mode is available on all three timer/counters. Minimal count value is 2.

Mode 4 [Software-Triggered Strobe]. In mode 4, when the specified count is reached, TOUT goes low for the duration of one clock pulse. Mode 4 is available on all timer/counters. Minimal number of counts is 1.

Mode 5 [Hardware-Triggered Strobe]. Mode 5 is similar to mode 4 except that operation is triggered by the TCTL2 input and can be retrIGGERED. This mode is available only on timer/counter 2.

Serial Control Unit

The serial control unit (SCU) is a single asynchronous serial channel that performs serial communication between the μPD70208 and an external serial device. The SCU is similar to the μPD71051 Serial Control Unit except for the lack of synchronous communication protocols. Figure 23 is the block diagram of the SCU.

The SCU has the following features.

- Full-duplex asynchronous serial controller
• Clock rate divisor (x16, x64)
• Baud rates to 250 kb/s supported (with ext. clock)
• 7-, 8-bit character lengths
• 1-, 2-bit stop bit lengths
• Break transmission and detection
• Full-duplex, double-buffered transmitter/receiver
• Even, odd, or no parity
• Parity, overrun, and framing error detection
• Receiver-full/transmitter-empty interrupt

The SCU contains four separately addressable registers for reading/writing data, reading status, and controlling operation of the SCU. The serial receive buffer (SRB) and the serial transmit buffer (STB) store the incoming and outgoing character data. The serial status (SST) register allows software to determine the current state of both the transmitter and receiver. The serial command (SCM) and serial mode (SMD) registers determine the operating mode of the SCU while the serial interrupt mask (SIMK) register allows software control of the SCU receive and transmit interrupts.

Figure 20. Timer Mode Register

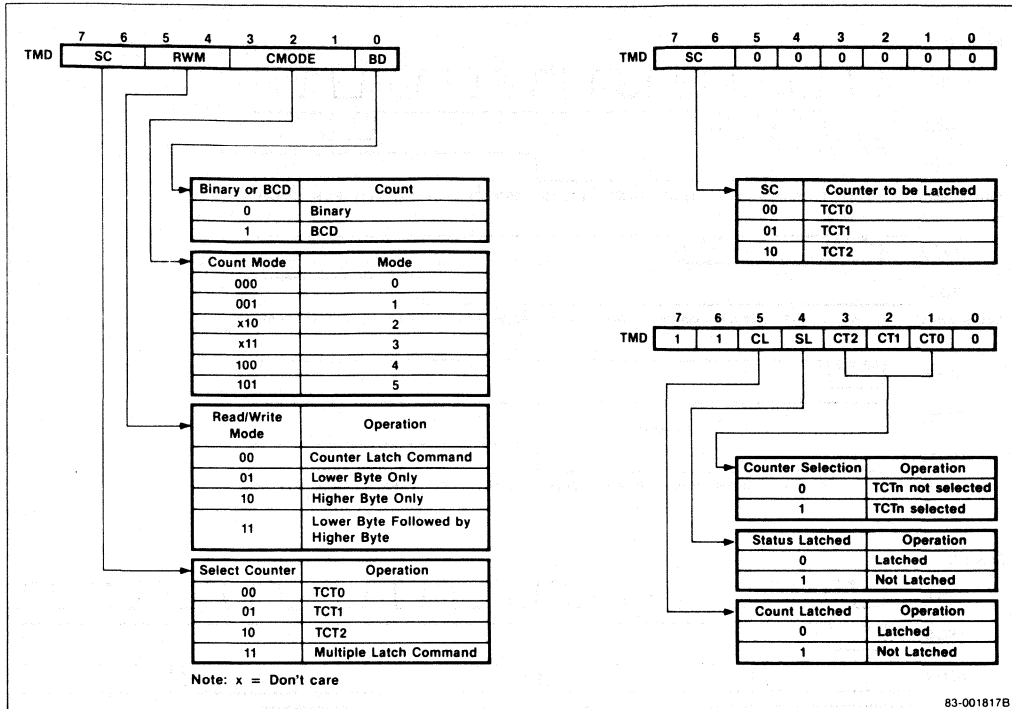


Figure 21. TCU Status Register

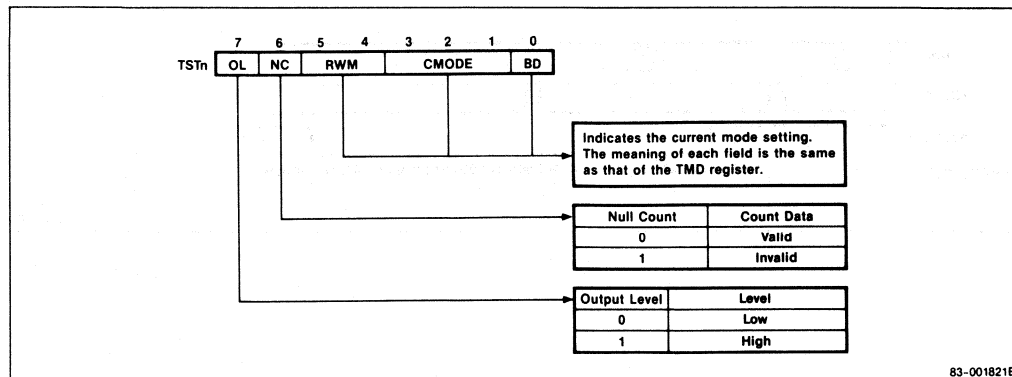




Figure 22. TCU Waveforms (Sheet 1 of 3)

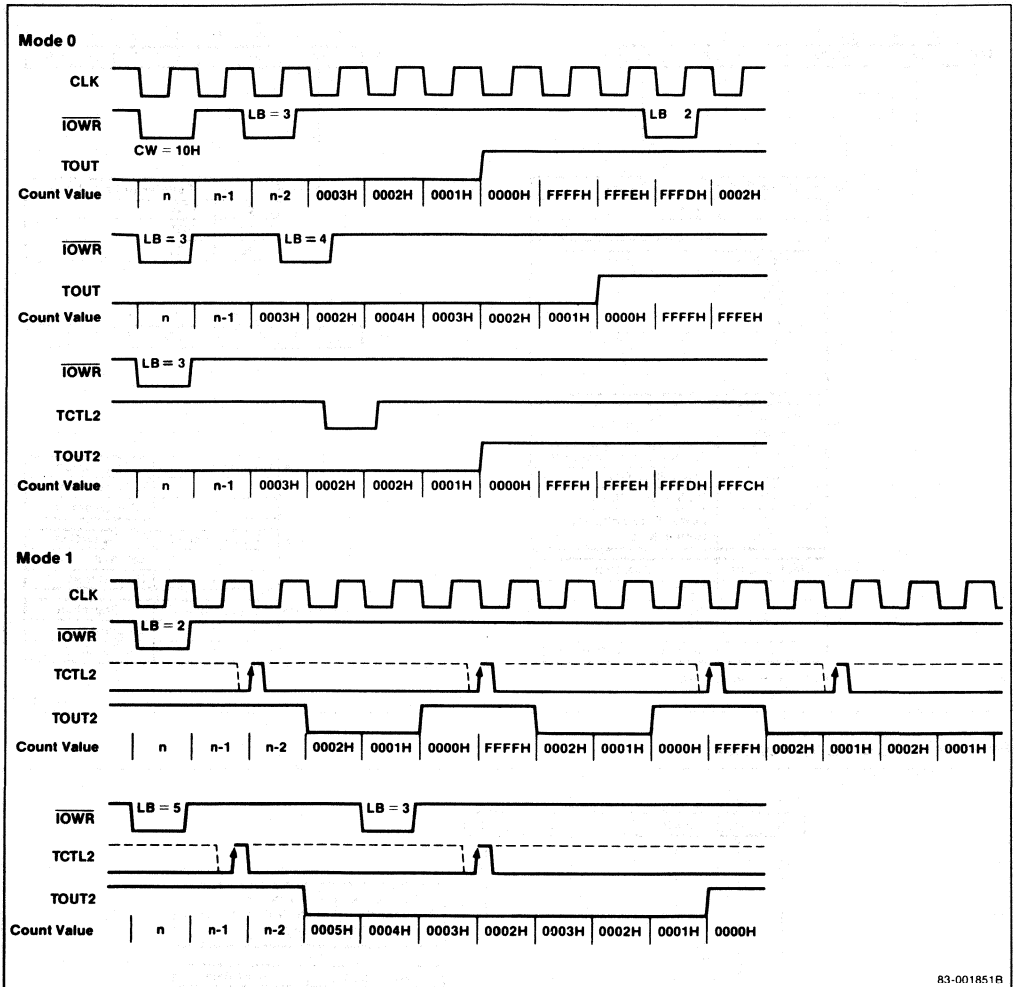


Figure 22. TCU Waveforms (Sheet 2 of 3)

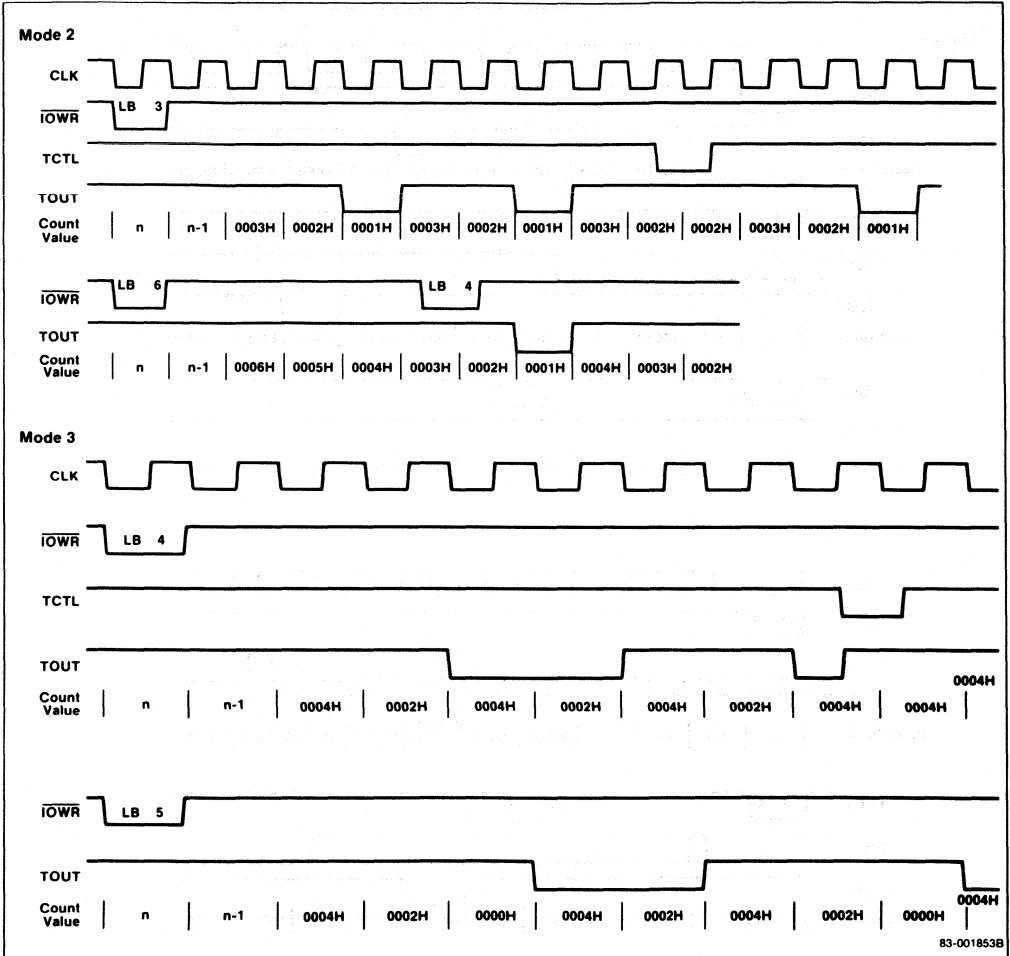


Figure 22. TCU Waveforms (Sheet 3 of 3)

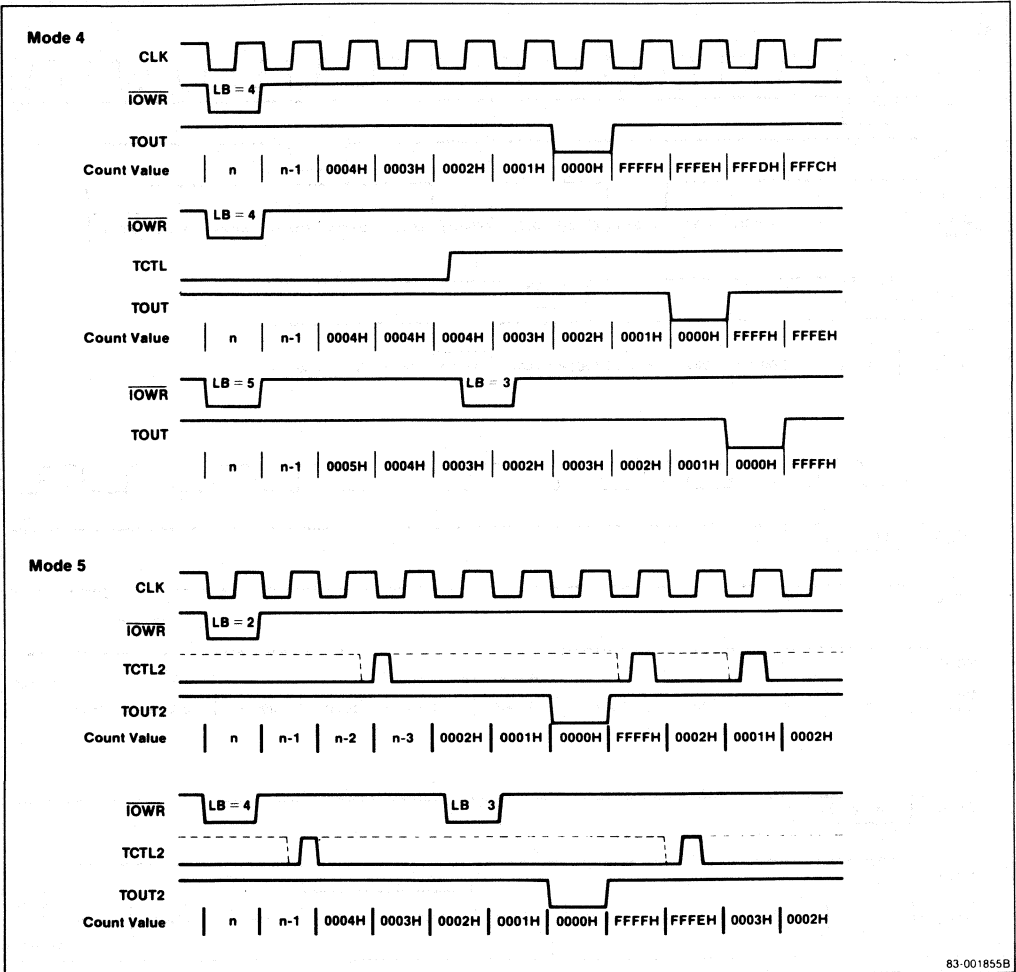
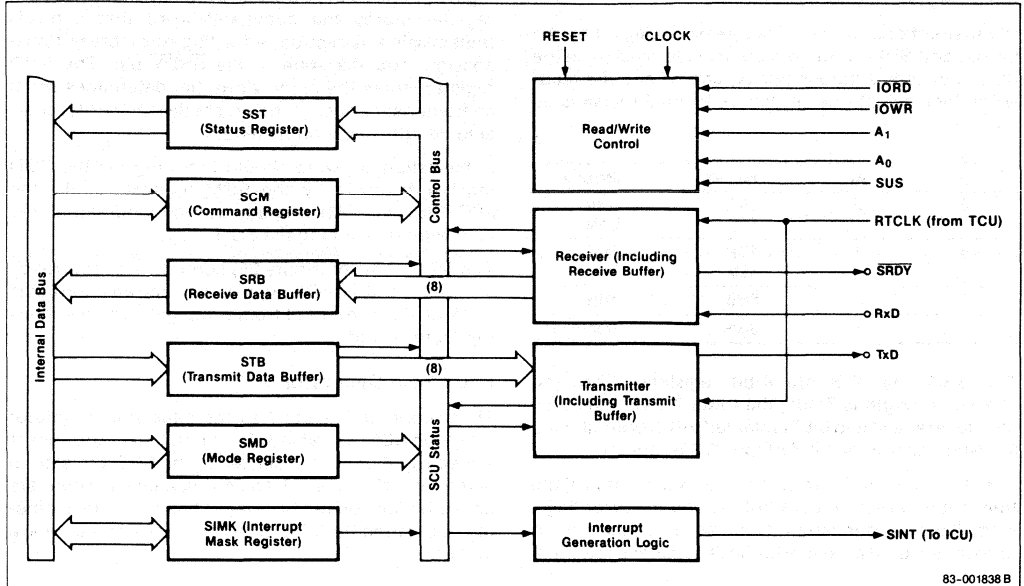


Figure 23. SCU Block Diagram



3

### Receiver Operation

While the RxD pin is high, the receiver is in an idle state. A transition on RxD from high to low indicates the start of new serial data. When a complete character has been received, it is transferred to the SRB; the receive buffer ready (RBRDY) bit in the SST register is set and (if unmasked) an interrupt is generated. The SST also latches any parity, overrun, or framing errors at this time.

The receiver detects a break condition when a null character with zero parity is received. The BRK bit is set for as long as the subsequent receive data is low and resets when RxD returns to a high level. The MRDY bit (SCM) and RBRDY (SST) are gated to form the output SRDY. SRDY prevents overruns from occurring when the program is unable to process the input data. Software can control MRDY to prevent data from being sent from the remote transmitter while RBRDY can prevent the immediate overrun of a received character.

### Transmitter Operation

TxD is kept high while the STB register is empty. When the transmitter is enabled and a character is written to the STB register, the data is converted to serial format and output on the TxD pin. The start bit indicates the start of the transmission and is followed by the character

stream (LSB to MSB) and an optional parity bit. One or two stop bits are then appended, depending on the programmed mode. When the character has been transferred from the STB, the TRBDY bit in the SST is set and if unmasked, a transmit buffer empty interrupt is generated.

Serial data can be transmitted and received by polling the SST register and checking the TBRDY or RBRDY flags. Data can also be transmitted and received by SCU-generated interrupts to the interrupt control unit. The SCU generates an interrupt in either of these conditions:

- (1) The receiver is enabled, the SRB is full, and receive interrupts are unmasked.
- (2) The transmitter is enabled, the STB is empty, and transmit interrupts are unmasked.

**SCU Registers and Commands**

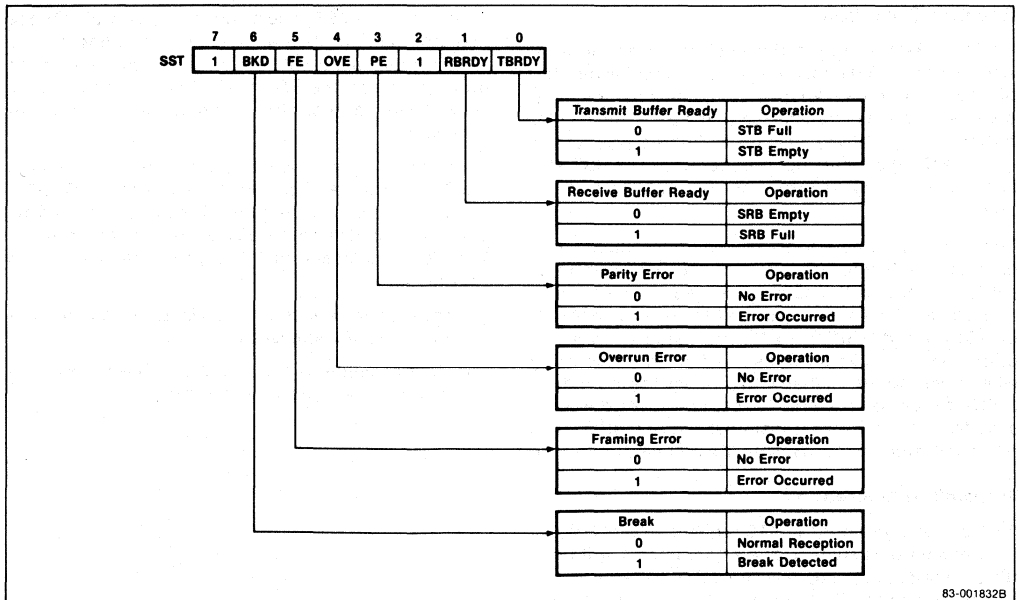
I/O instructions to the I/O addresses selected by the OPHA and SULA registers are used to read/write the SCU registers. Address bits A<sub>1</sub> and A<sub>0</sub> and the read/write lines select one of the six internal registers as follows:

A <sub>1</sub>	A <sub>0</sub>	Register	Operation
0	0	SRB STB	Read Write
0	1	SST SCM	Read Write
1	0	SMD	Write
1	1	SIMK	Read/Write

The SRB and STB are 8-bit registers. When the character length is 7 bits, the lower 7 bits of the SRB register are valid and bit 7 is cleared to 0. If programmed for 7-bit characters, bit 7 of the STB is ignored.

The SST register (figure 24) contains the status of the transmit and receive data buffers and the error flags. Error flags are persistent. Once an error flag is set, it remains set until a clear error flags command is issued.

**Figure 24. SST Register**



83-001832B

Figure 25 shows the SCM and SMD registers. The SCM register stores the command word that controls transmission, reception, error flag reset, break transmission, and the state of the SRDY pin. The SMD register stores the mode word that determines serial characteristics such as baud rate divisor, parity, character length, and stop bit length.

Initialization software should first program the SMD register followed by the SCM register. Unlike the μPD71051, the SMD register can be modified at any time without resetting the SCU.

The SIMK register (figure 26) controls the occurrence of RBRDY and TBRDY interrupts. When an interrupt is masked, it is prevented from propagating to the interrupt control unit.

**Baud Rate Generator**

Timer/counter 1 is used as the baud rate generator when the SCU is enabled. The input baud rate clock is scaled by 16 or 64, as selected in the SMD register, to determine the receive/transmit data clock. There are no restrictions on the SCU input baud rate clock other than operating the TCU in mode 3 with a square-wave output.

Figure 25. SCM and SMD Registers

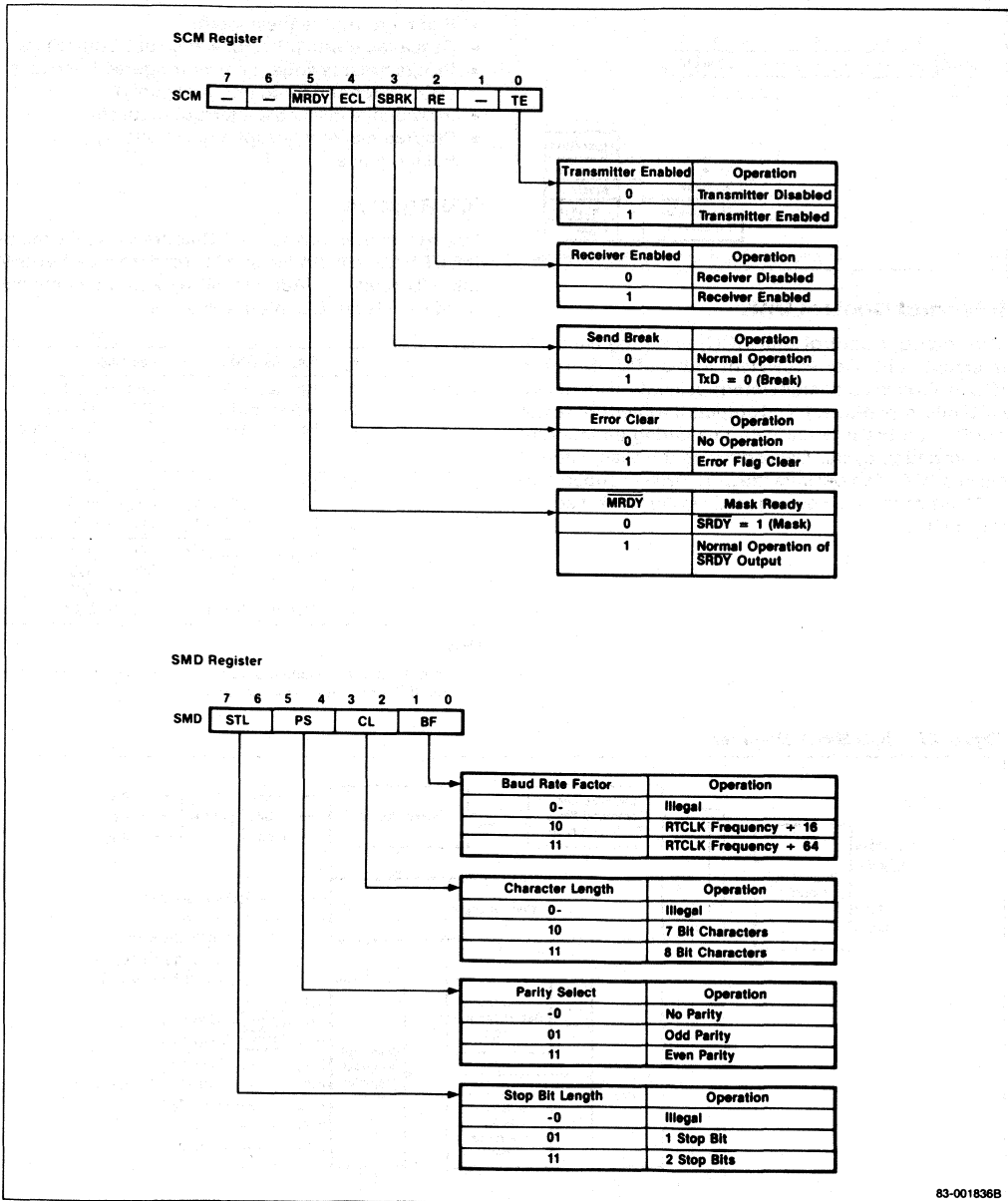
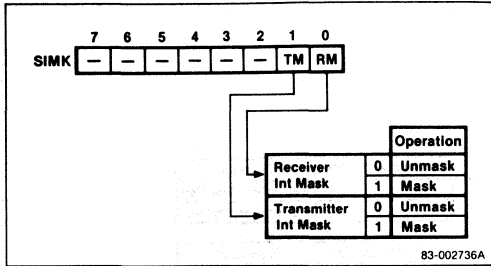


Figure 26. SIMK Register



**Interrupt Control Unit**

The interrupt control unit (ICU) is a programmable interrupt controller equivalent to the μPD71059. The ICU arbitrates up to eight interrupt inputs, generates a CPU interrupt request, and outputs the interrupt vector number on the internal data bus during an interrupt acknowledge cycle. Cascading up to seven external slave μPD71059s permits the μPD70208 to support up to 56 interrupt sources. Figure 27 is the block diagram for the ICU.

The ICU has the following features.

- Eight interrupt request inputs
- Cascadable with μPD71059 Interrupt Controllers
- Programmable edge- or level-triggered interrupts (TCU, edge-triggered interrupts only)
- Individually maskable interrupt requests
- Programmable interrupt request priority
- Polling mode

**ICU Registers**

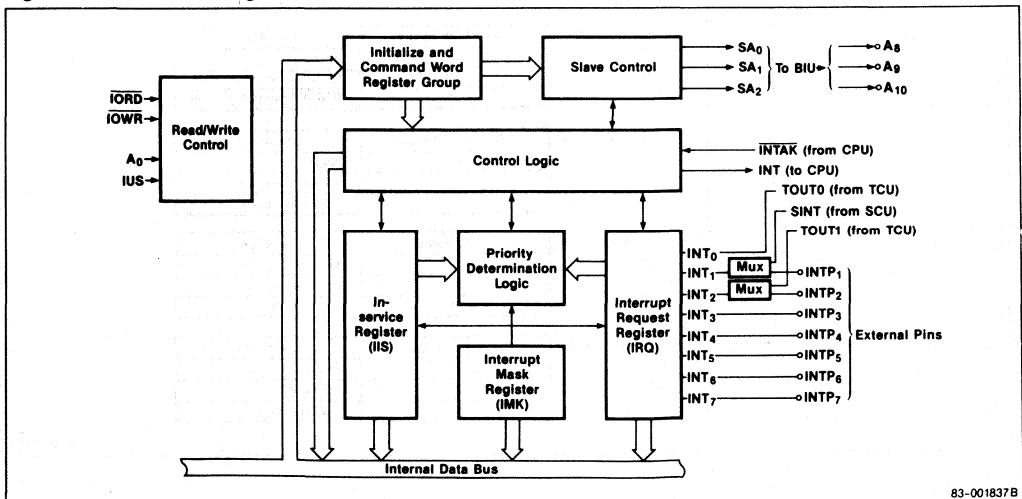
Use I/O instructions to the I/O addresses selected by the OPHA and IULA registers to read from and write to the ICU registers. Address bit A<sub>0</sub> and the command word selects an ICU internal register.

	A <sub>0</sub>	Other Condition	Operation
Read	0	IMD selects IRQ	CPU ← IRQ data
	0	IMD selects IIS	CPU ← IIS data
	0	Polling phase	CPU ← Polling data
	1	—	CPU ← IMKW
Write	0	D4 = 1	CPU → IIW1
	0	D4 = 0 and D3 = 0	CPU → IPFW
	0	D4 = 0 and D3 = 1	CPU → IMDW
	1	During initialization	CPU → IIW2
	1		CPU → IIW3
	1	After initialization	CPU → IIW4
	1		CPU → IMKW

**Note:**

- (1) In polling phase, polling data has priority over the contents of the IRQ or IIS register when read.

Figure 27. ICU Block Diagram



### Initializing the ICU

The ICU is always used to service maskable interrupts in a μPD70208 system. Prior to accepting maskable interrupts, the ICU must first be initialized (figure 28). Following initialization, command words from the CPU can change the interrupt request priorities, mask/un-mask interrupt requests, and select the polling mode. Figures 29 and 30 list the ICU initialization and command words.

Interrupt initialization words 1-4 (IIW1-IIW4) initialize the ICU, indicate whether external μPD71059s are connected as slaves, select the base interrupt vector, and select edge- or level-triggered inputs for INT1-INT7. Interrupt sources from the TCU are fixed as edge-triggering. INT0 is internally connected to TOUT0, and INT2 may be connected to TOUT1 by the IRSW field in the OPCN.

The interrupt mask word (IMKW) contains programmable mask bits for each of the eight interrupt inputs. The interrupt priority and finish word (IPFW) is used by the interrupt handler to terminate processing of an interrupt or change interrupt priorities. The interrupt mode word (IMDW) selects the polling register, interrupt request (IRQ) or interrupt in service (IIS) register, and the nesting mode.

The initialization words are written in consecutive order starting with IIW1. IIW2 sets the interrupt vector. IIW3 specifies which interrupts are connected to slaves. IIW4 is only written if II4 = 1 (bit D<sub>1</sub> of IIW1). IIW4 is only written if II4 = 1 (bit D<sub>0</sub> of IIW1).

### μPD71059 Cascade Connection

To increase the number of maskable interrupts, up to seven slave μPD71059 Interrupt Controllers can be cascaded. During cascade operation (figure 31), each

slave μPD71059 INT output is routed to one of the μPD70208 INTP inputs. During the second interrupt acknowledge bus cycle, the ICU places the slave address on address lines A<sub>10</sub>-A<sub>8</sub>. Each slave compares this address with the slave address programmed using interrupt initialization word 3 (IIW3). If the same, the slave will place the interrupt vector on pins AD<sub>7</sub>-AD<sub>0</sub> during the second interrupt acknowledge bus cycle.

Figure 29. Interrupt Initialization Words 1-4

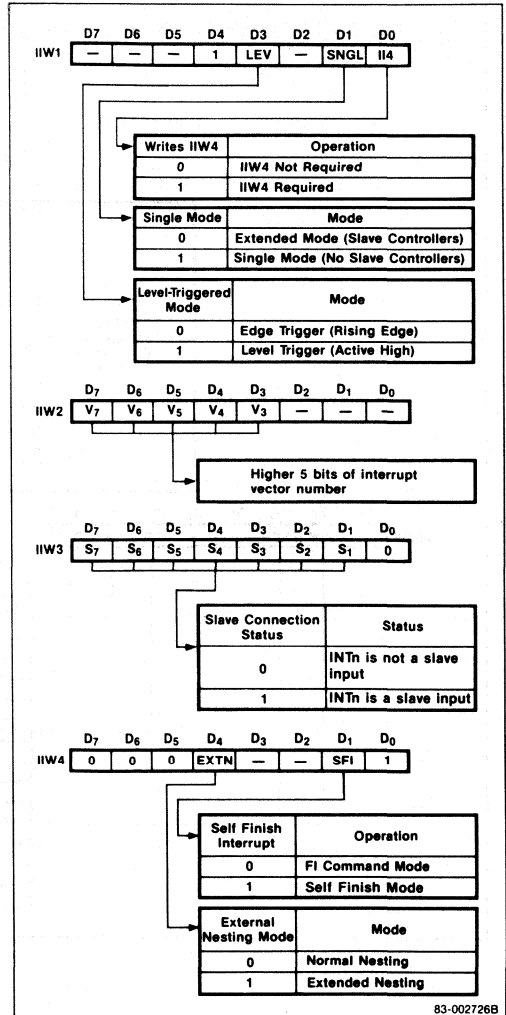


Figure 28. Initialization Sequence

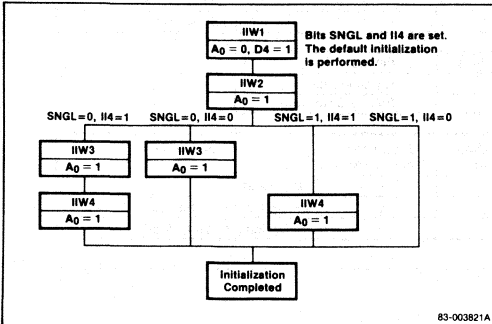




Figure 30. Command Words

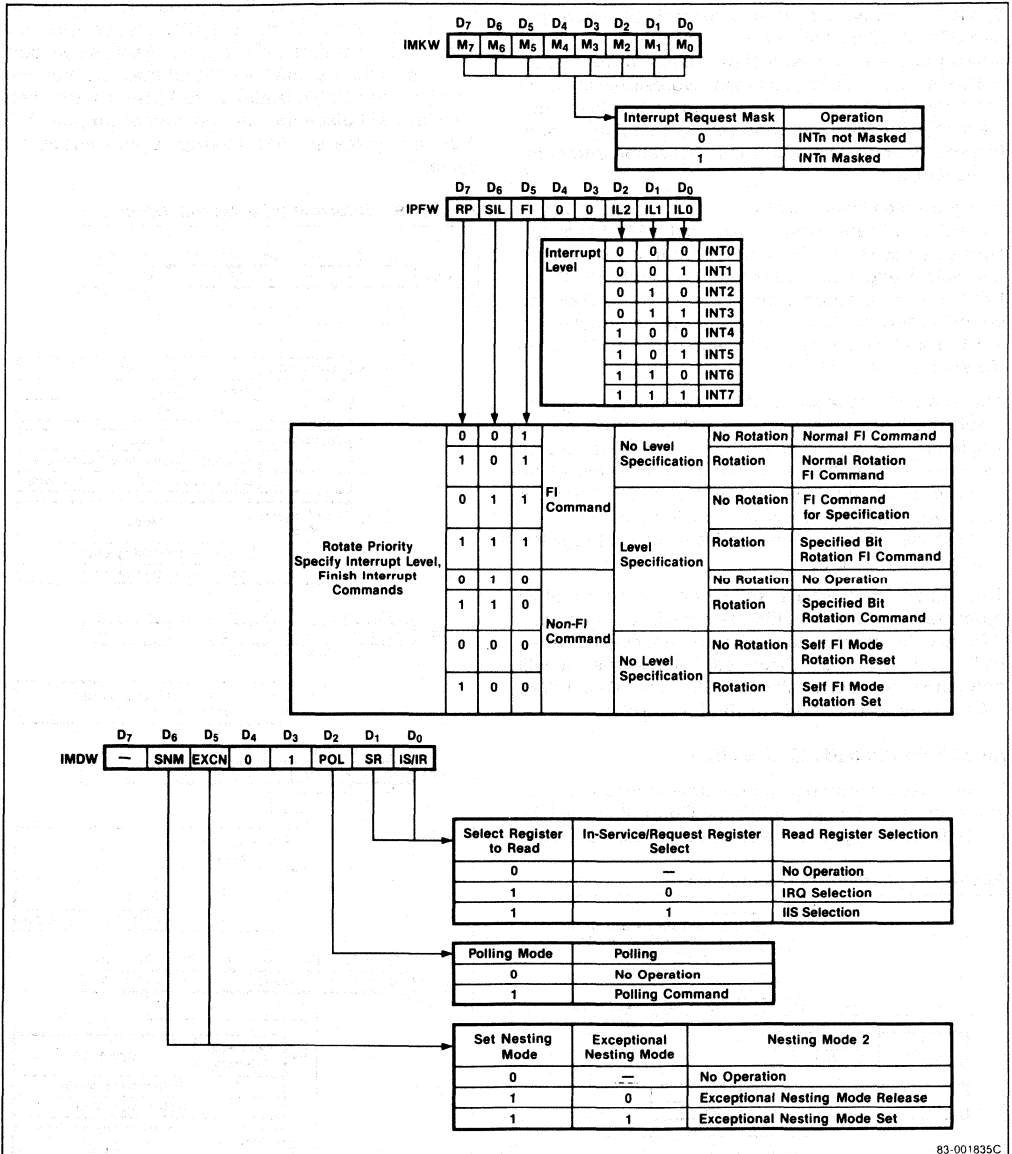
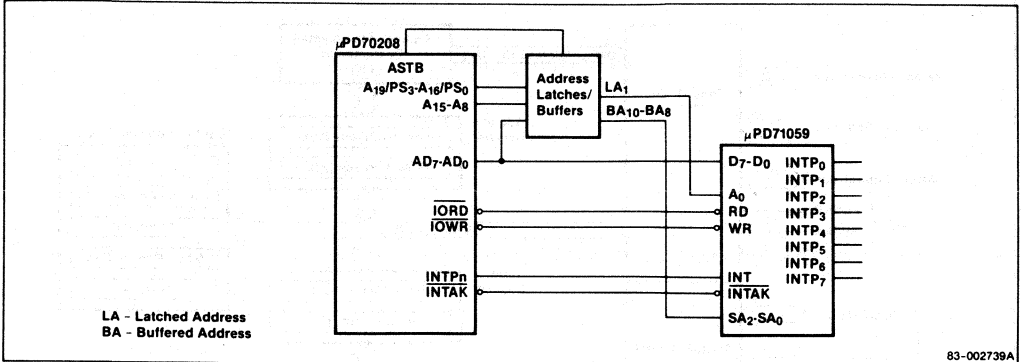


Figure 31. μPD71059 Cascade Connection



## DMA Control Unit

The DMA Control Unit (DMAU) is a high-speed DMA controller compatible with the μPD71071 DMA Controller. The DMAU has four independent DMA channels and performs high-speed data transfers between memory and external peripheral devices at speeds as high as 2 megabytes/second in an 8-MHz system. Figure 32 is the block diagram for the DMAU.

The DMAU has the following features.

- Four independent DMA channels
- Cascade mode for slave μPD71071 DMA controllers
- 20-bit address registers
- 16-bit transfer count registers
- Single, demand, and block transfer modes
- Bus release and bus hold modes
- Autoinitialization
- Address increment/decrement
- Fixed/rotating channel priorities
- TC output at transfer end
- Forced termination of service by  $\overline{\text{END}}$  input

## DMAU Basic Operation

The DMAU operates in either a slave or master mode. In the slave mode, the DMAU samples the four DMARQ input pins every clock. If one or more inputs are active, the corresponding DMA request bits are set and the DMAU sends a bus request to the BAU while continuing to sample the DMA request inputs. After the BAU returns the DMA bus acknowledge signal, the DMAU stops DMA request sampling, selects the DMA channel with the highest priority, and enters the bus master mode to perform the DMA transfer. While in the bus master mode, the DMAU controls the external bus and performs DMA transfers based on the preprogrammed channel information.

## Terminal Count

The DMAU ends DMA service when the terminal count condition is generated or when the  $\overline{\text{END}}$  input is asserted. A terminal count (TC) is produced when the contents of the current count register becomes zero. If autoinitialization is not enabled when DMA service terminates, the mask bit of the channel is set and the DMARQ input of that channel is masked. Otherwise, the current count and address registers are reloaded from the base registers and new DMA transfers are again enabled.

## DMA Transfer Type

The type of transfer the DMAU performs depends on the following conditions.

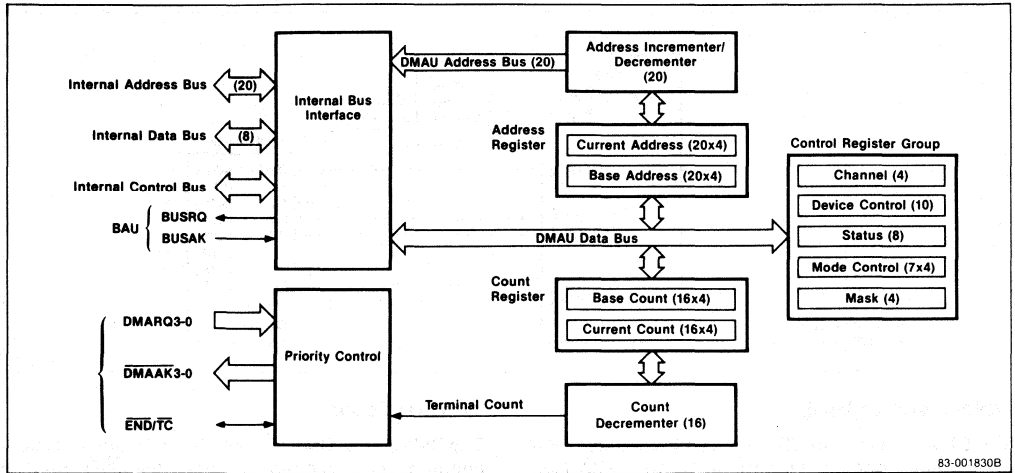
- Direction of the transfer (each channel)
- Transfer mode (each channel)
- Bus mode

## Transfer Direction

All DMA transfers use memory as a reference point. Therefore, a DMA read operation transfers data from memory to an I/O port. A DMA write operation reads an I/O port and writes the data into memory. During memory-to-I/O transfer, the DMA mode (DMD) register is used to select the transfer directions for each channel and activate the appropriate control signals.

Operation	Transfer Direction	Activated Signals
DMA read	Memory → I/O	$\overline{\text{IOWR}}$ , MRD
DMA write	I/O → Memory	$\overline{\text{IORD}}$ , MWR
DMA verify		Addresses only; no transfer performed

Figure 32. DMAU Block Diagram



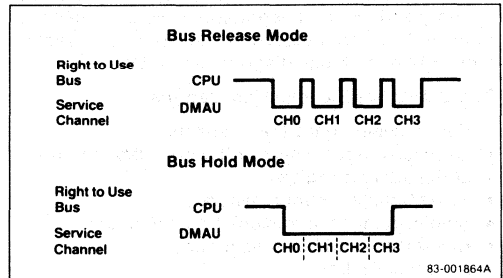
83-001830B

**Bus Mode**

The DMA device control (DDC) register selects operation in either the bus release or bus hold mode. The selected bus mode determines the DMAU conditions for return of the bus to the BAU. Figure 33 shows that in bus release mode, only a single channel is serviced after the DMAU obtains the bus. When DMA service ends (termination conditions depend on the transfer mode), the DMAU returns the bus to the BAU regardless of the state of other DMA requests, and the DMAU reenters the slave mode. When the DMAU regains use of the bus, a new DMA operation can begin.

In bus hold mode, several channels can receive contiguous service without releasing the bus. If there is another valid DMA request when a channel's DMA service is finished, the new DMA service can begin immediately after the previous service without returning the bus to the BAU.

Figure 33. Bus Modes



83-001864A

**Transfer Modes**

The DMD register also selects either single, demand, or block transfer mode for each channel. The conditions for the termination of each transfer characterize each transfer mode. The following table shows the various transfer modes and termination conditions.

Transfer Mode	Termination Conditions
Single	After each byte/word transfer
Demand	END input Terminal count Inactive DMARQ DMARQ of a higher priority channel becomes active (bus hold mode)
Block	END input Terminal count

The operation of single, demand, and block mode transfers depends on whether the DMAU is in bus release or bus hold mode. Figure 34 shows the operation flow for the six possible transfer and bus mode operations in DMA transfer.

**Single-Mode Transfer.** In bus release mode, when a channel completes transfer of a single byte, the DMAU enters the slave mode regardless of the state of DMA request inputs. In this manner, other lower-priority bus masters will be able to access the bus.

In bus hold mode, when a channel completes transfer of a single byte, the DMAU terminates the channel's service even if the DMARQ request signal is asserted. The DMAU will then service any other requesting channel. If there are no requests from any other DMA channels, the DMAU releases the bus and enters the slave state.

**Demand-Mode Transfer.** In bus release mode, the currently active channel continues to transfer data as long as the DMA request of that channel is active, even though other DMA channels are issuing higher-priority requests. When the DMA request of the serviced channel becomes inactive, the DMAU releases the bus and enters the slave state.

In bus hold mode, when the active channel completes a single transfer, the DMAU checks the other DMA request lines without ending the current service. If there is a higher-priority DMA request, the DMAU stops the service of the current channel and starts servicing the highest-priority channel requesting service. If there is no higher request than the current one, the DMAU continues to service the currently active channel. Lower-priority DMA requests are honored without releasing the bus after the current channel service is complete.

**Block-Mode Transfer.** In bus release mode, the current channel continues DMA transfers until a terminal count or the external END input becomes active. During this time, the DMAU ignores all other DMA requests. After completion of the block transfer, the DMAU releases the bus and enters the slave state, even if DMA requests from other channels are active.

In bus hold mode, the current channel transfers data until an internal or external END signal becomes active. When the service is complete, the DMAU checks all DMA requests without releasing the bus. If there is an active request, the DMAU immediately begins servicing the request. The DMAU releases the bus after it honors all DMA requests or a higher-priority bus master requests the bus.

## Byte Transfer

The DMD register can specify only byte DMA transfers for each channel. Depending on the mode selected, the address register can either increment or decrement, whereas the count register is always decremented.

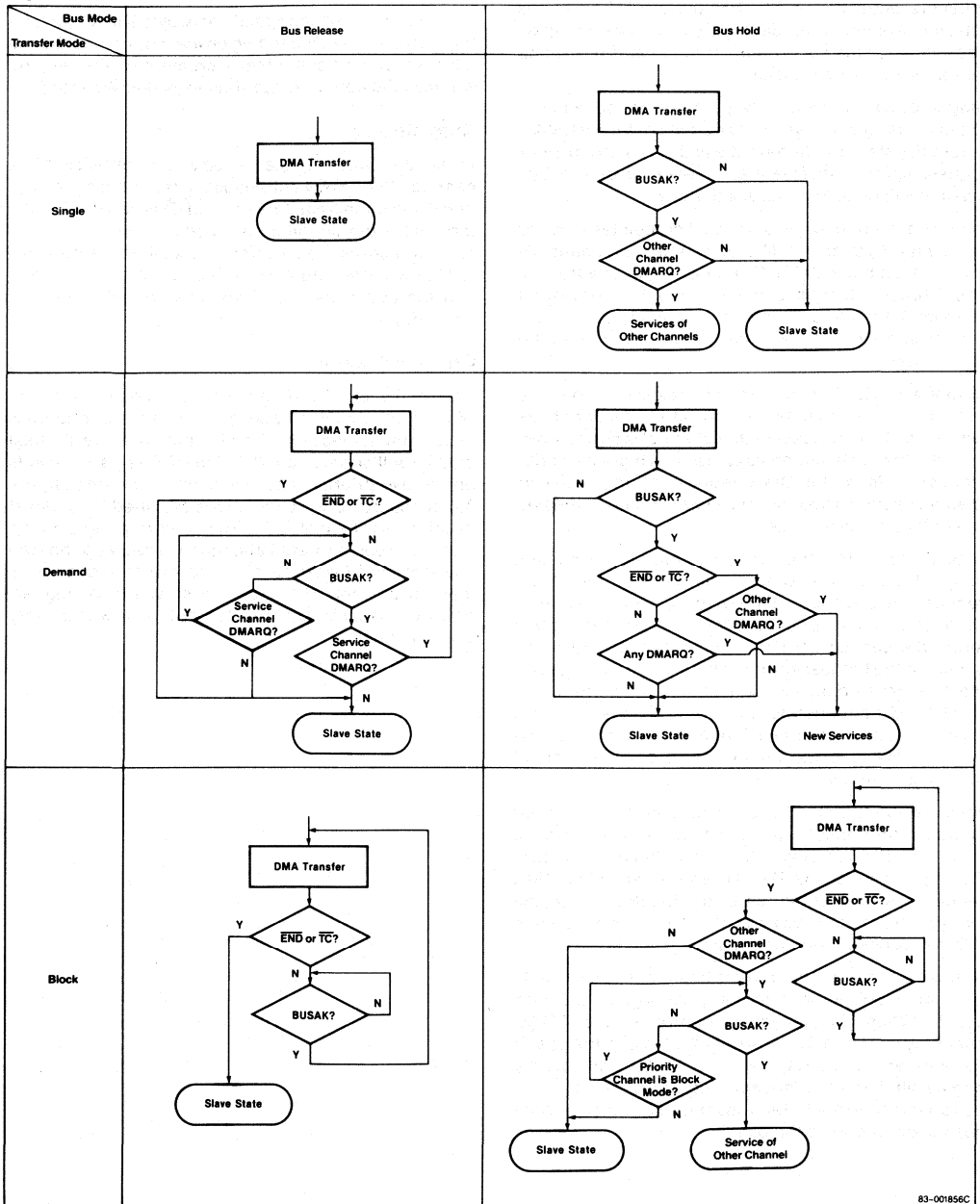
## Autoinitialize

When the DMD register selects autoinitialize for a channel, the DMAU automatically reinitializes the address and count registers when END is asserted or the terminal count condition is reached. The contents of the base address and base count registers are transferred to the current address and current count registers, and the applicable bit of the mask register remains cleared.

## Channel Priority

Each of the four DMAU channels is assigned a priority. When multiple DMA requests from several channels occur simultaneously, the channel with the highest priority will be serviced first. The DDC register selects one of two priority schemes: fixed or rotating (figure 35). In fixed priority, channel 0 is assigned the highest priority and channel 3, the lowest. In rotating priority, priority order is rotated after each service so that the channel last serviced receives the lowest priority. This method prevents the exclusive servicing of higher-priority channels and the lockout of lower-priority DMA channels.

Figure 34. Transfer Modes



## Cascade Connection

Slave μPD71071 DMA Controllers can be cascaded to easily expand the system DMA channel capacity to 16 DMA channels. Figure 36 shows an example of cascade connection. During cascade operation, the DMAU acts as a mediator between the BAU and the slave μPD71071s. During DMA cascade mode operation, it is the responsibility of external logic to isolate the cascade bus master from the μPD70208 control outputs. These outputs are listed in a table at the front of this data sheet.

The DMAU always operates in the bus hold mode while a cascade channel is in service, even when the bus release mode is programmed. Other DMA requests are held pending while a slave μPD71071 channel is in service. When the cascaded μPD71071 ends service and moves into the slave state, the DMAU also moves to the slave state and releases the bus. At this time, all bits of the DMAU request register are cleared. The DMAU continues to operate normally with the other noncascaded channels.

Figure 35. Priority Order

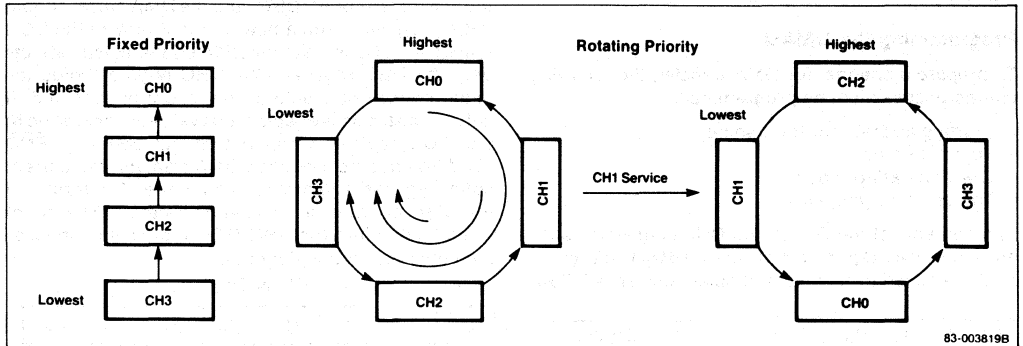
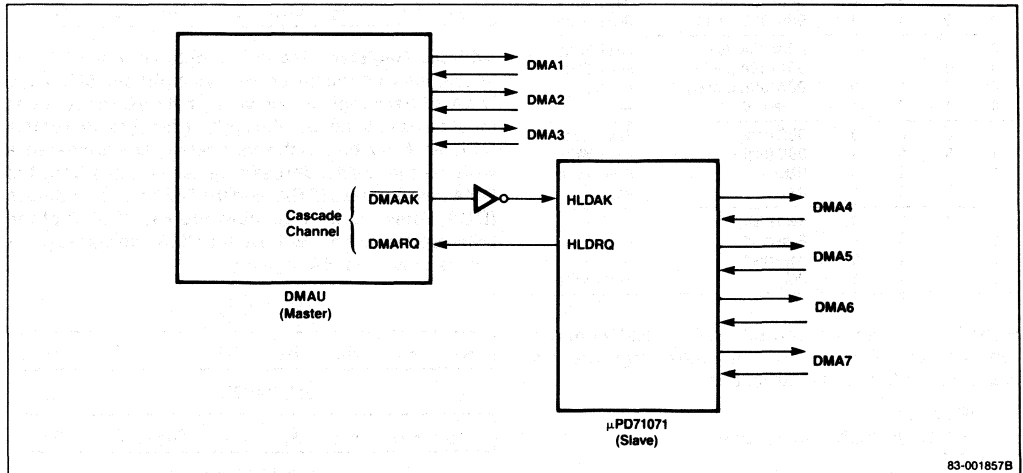


Figure 36. μPD71071 Cascade Example



**Bus Waiting Operation**

The DMAU will automatically perform a bus waiting operation (figure 37) whenever the RCU refresh request queue fills. When the DMA bus acknowledge goes inactive, the DMAU enters the bus waiting mode and inactivates the DMA bus request signal. Control of the bus is then transferred to the higher-priority RCU by the BAU.

Two clocks later, the DMAU reasserts its internal DMA bus request. The bus waiting mode is continued until the DMA bus acknowledge signal again becomes active and the interrupted DMA service is immediately restarted.

**Programming the DMAU**

To prepare a channel for DMA transfer, the following characteristics must be programmed.

- Starting address for the transfer
- Transfer count
- DMA operating mode
- Transfer size (byte/word)

The contents of the OPHA and DULA registers determine the base I/O port address of the DMAU. Addresses A<sub>3</sub>-A<sub>0</sub> are used to select a particular register as follow:

A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Register	Operation
0	0	0	0	DICM	Write
0	0	0	1	DCH	Read/Write
0	0	1	0	DBC/DCC (low)	Read/Write
0	0	1	1	DBC/DCC (high)	Read/Write
0	1	0	0	DBA/DCA (low)	Read/Write
0	1	0	1	DBA/DCA (high)	Read/Write
0	1	1	0	DBA/DCA (upper)	Read/Write
0	1	1	1	Reserved	—
1	0	0	0	DDC (low)	Read/Write
1	0	0	1	DDC (high)	Read/Write
1	0	1	0	DMD	Read/Write
1	0	1	1	DST	Read
1	1	0	0	Reserved	—
1	1	0	1	Reserved	—
1	1	1	0	Reserved	—
1	1	1	1	DMK	Read/Write

Word I/O instructions can be used to read/write the register pairs listed below. All other registers are accessed via byte I/O instructions.

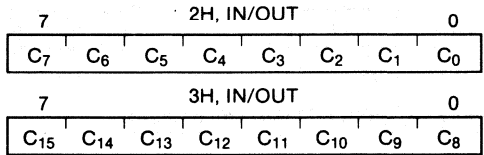
- DBC/DCC
- DBA/DCA (higher/lower only)
- DDC

**DMAU Registers**

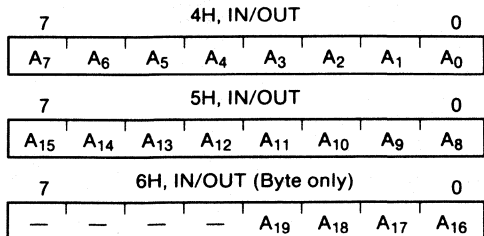
**Initialize.** The DMA initialize command (DICM) register (figure 38) is used to perform a software reset of the DMAU. The DICM is accessed using the byte OUT instruction.

**Channel Register.** Writes to the DMA channel (DCH) register (figure 39) select one of the four DMA channels for programming and also the base/current registers. Reads of the DCH register return the currently-selected channel and the register access mode.

**Count Registers.** When bit 2 of the DCH register is cleared, a write to the DMA count register updates both the DMA base count (DBC) and the DMA current count (DCC) registers with a new count. If bit 2 of the DCH register is set, a write to the DMA count register affects only the DBC register. The DBC register holds the initial count value until a new count is specified. If autoinitialization is enabled, this value is transferred to the DCC register when a terminal count or END condition occurs. For each DMA transfer, the current count register is decremented by one. The format of the DMA count register is shown below. The count value loaded into the DBC/DCC registers is one less than the desired transfer count.



**Address Register.** Use either byte or word I/O instructions with the lower two bytes (4H and 5H) of the DMA address register. However, byte I/O instructions must be used to access the high-order byte (6H) of this register. When bit 2 of the channel register is cleared, a write to the DMA address register updates both the DMA base address (DBA) and the DMA current address (DCA) registers with the new address. If bit 2 of the DCH register is set, a write to the DMA address register affects only the DBA register.

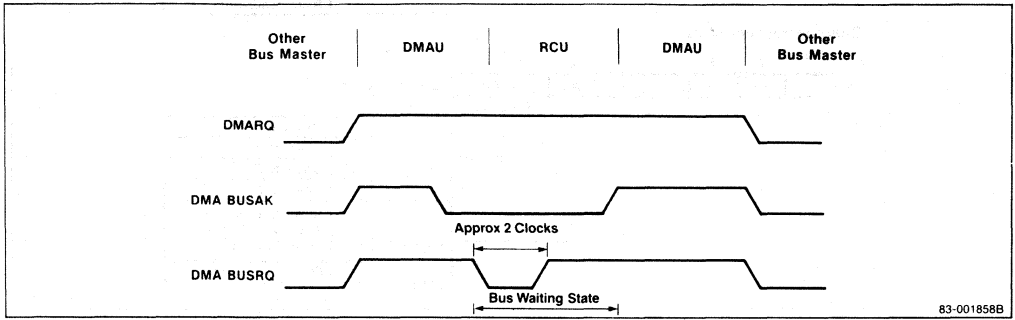


The DBA register holds the starting address value until a new address is specified. This value is transferred to the DCA register automatically if autoinitialization is selected. For each DMA transfer, the current address register is incremented/decremented by one.

**Device Control Register.** The DMA device control (DDC) register (figure 40) is used to program the DMA transfer characteristics common to all DMA channels. It controls the bus mode, write timing, priority logic, and enable/disable of the DMAU.

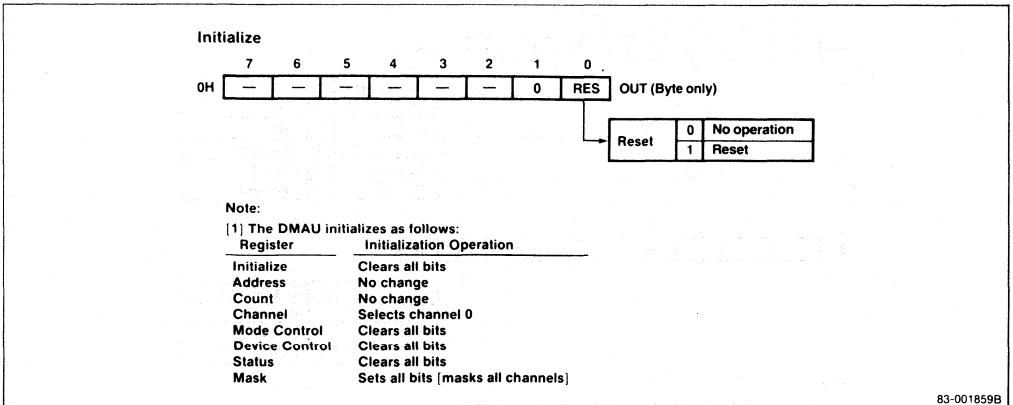
**Status Register.** The DMA status (DST) register (figure 41) contains information about the current state of each DMA channel. Software can determine if a termination condition has been reached ( $TC_3$ - $TC_0$ ) or if a DMA service request is present ( $RQ_3$ - $RQ_0$ ). The byte IN instruction must be used to read this register.

**Figure 37. Bus Waiting Operation**



83-001858B

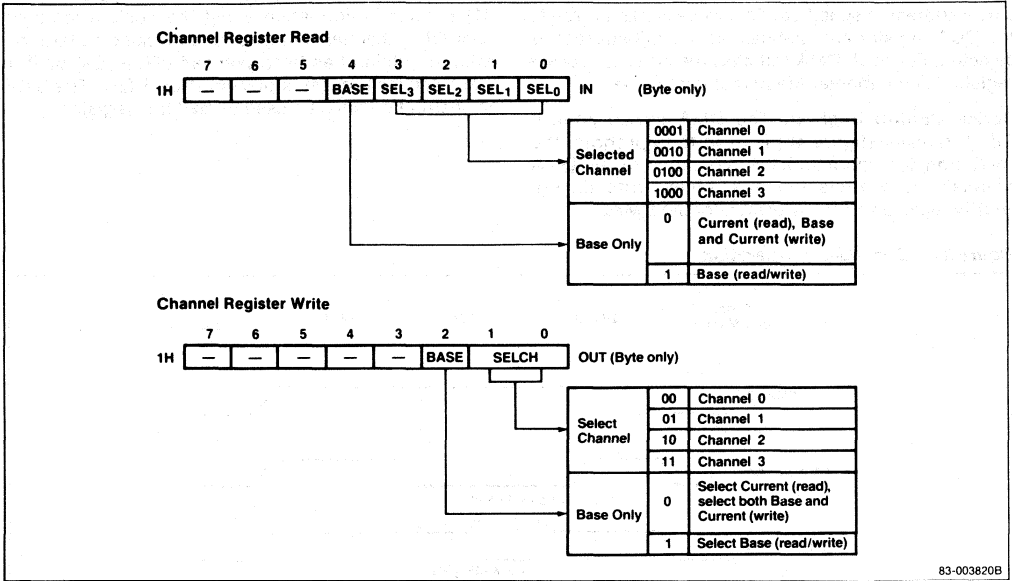
**Figure 38. DMA Initialize Command Register**



83-001859B

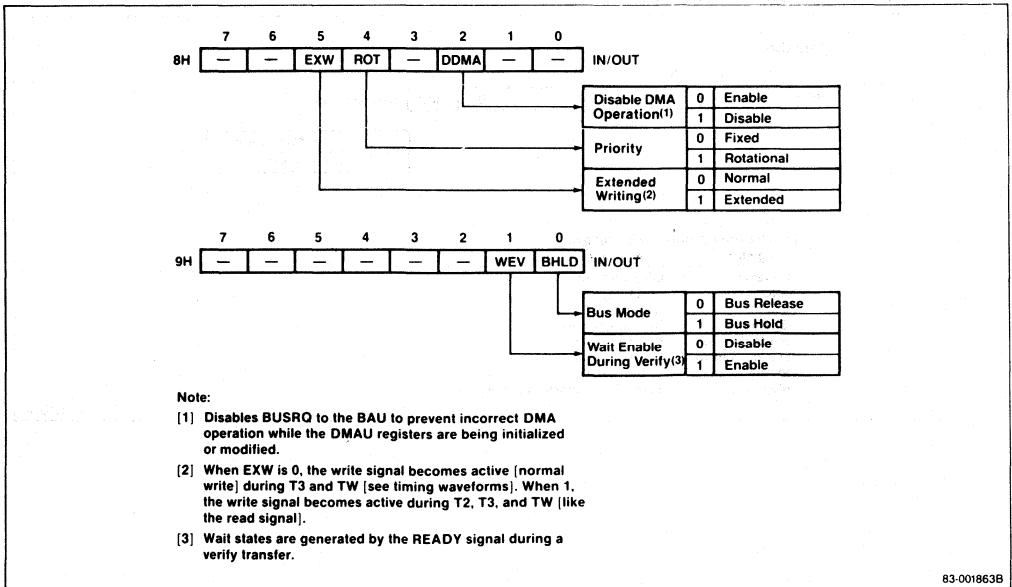


Figure 39. DMA Channel Register



83-003820B

Figure 40. DMA Device Control Register

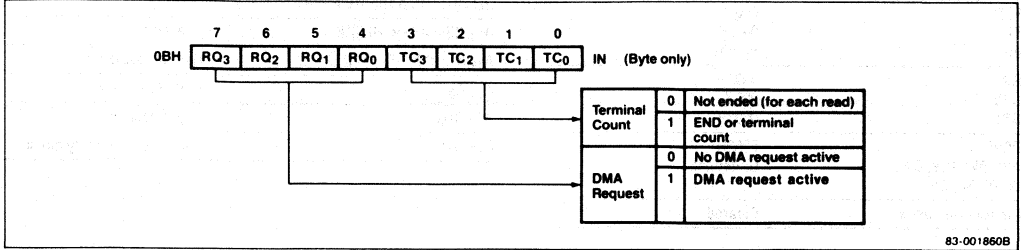


83-001863B

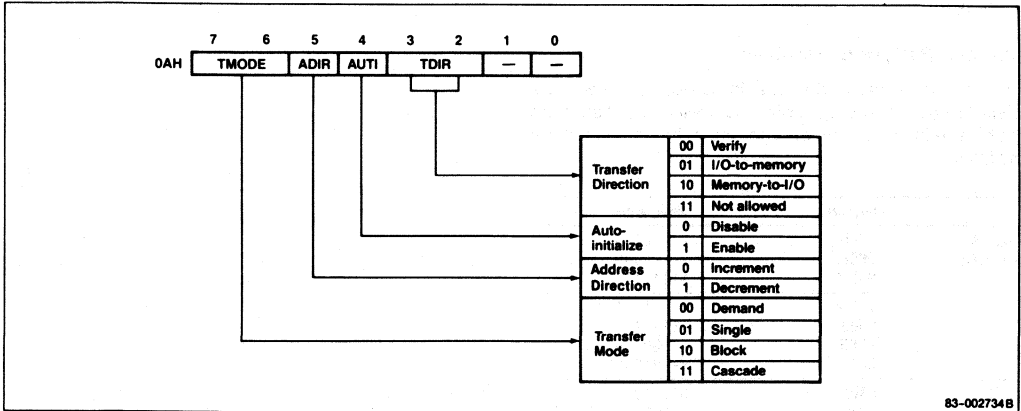
**Mode Control Register.** The DMA mode (DMD) register (figure 42) selects the operating mode for each DMA channel. The DCH register selects which DMD register will be accessed. A byte IN/OUT instruction must be used to access this register.

**Mask Register Read/Write.** The DMA mask (DMK) register (figure 43) allows software to individually enable and disable DMA channels. The DMK register can only be accessed via byte I/O instructions.

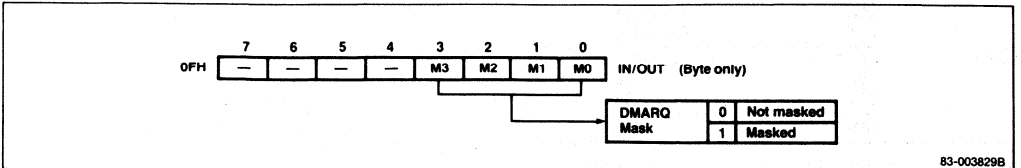
**Figure 41. DMA Status Register**



**Figure 42. DMA Mode Register**



**Figure 43. DMA Mask Register**



## μPD70208 (V40)

### Reset

The falling edge of the  $\overline{\text{RESET}}$  signal resets the μPD70208. The signal must be held low for at least four clock cycles to be recognized as valid.

CPU Reset State Register	Reset Value
PFP	0000H
PC	0000H
PS	FFFFH
SS	0000H
DS0	0000H
DS1	0000H
PSW AW, BW, CW, DW, IX, IY, BP, SP	F002H Undefined
Instruction queue	Cleared

When RESET returns to the high level, the CPU will start fetching instructions from physical address FFFF0H.

### Internal Peripheral Registers

Internal peripheral devices initialized on reset are listed in the following table. I/O devices not listed are not initialized on reset and must be initialized by software.

	Register	Reset Value
System I/O area	OPCN	----0000
	OPSEL	----0000
	WCY1	11111111
	WCY2	----1111
	WMB	-111-111
	TCKS	---00000
SCU	RFC	x--01000
	SMD	01001011
	SCM	--0000-0
	SIMK	-----11
	SST	10000100
	DCH	---00001
DMAU	DMD	000000-0
	DDC (low)	--0-0--
	DDC (high)	-----00
	DST	xxxx0000
	DMK	----1111

**Symbols:** x = unaffected; 0 = cleared; 1 = set; (-) = unused.

### Output Pin Status

The following table lists output pin status during reset.

Signal	Status
$\overline{\text{INTAK}}$ , $\overline{\text{BUFEN}}$ , $\overline{\text{BUFR/W}}$ , $\overline{\text{MRD}}$ , $\overline{\text{MWR}}$ , $\overline{\text{END/TC}}$ , $\overline{\text{IOWR}}$ , $\overline{\text{IORD}}$ , $\overline{\text{REFRD}}$ , $\text{BS}_7$ - $\text{BS}_0$ , $\overline{\text{BUSLOCK}}$ , $\overline{\text{RESOUT}}$ , $\overline{\text{DMAAK3}}$ - $\overline{\text{DMAAK0}}$	High level
$\text{QS}_7$ - $\text{QS}_0$ , $\text{ASTB}$ , $\text{HLDAK}$	Low level
$\text{A}_{19}$ - $\text{A}_{16}$ / $\text{PS}_3$ - $\text{PS}_0$ , $\text{TOUT2}$	High or low level
$\text{A}_{15}$ - $\text{A}_8$	High or low level
$\text{AD}_7$ - $\text{AD}_0$	High impedance
CLKOUT	Continues to supply clock

## Instruction Set

### Symbols

Preceding the instruction set, several tables explain symbols, abbreviations, and codes.

### Clocks

In the Clocks column of the instruction set, the numbers cover these operations: instruction decoding, effective address calculation, operand fetch, and instruction execution.

Clock timings assume the instruction has been pre-fetched and is present in the four-byte instruction queue. Otherwise, add four clocks for each byte not present.

For instructions that reference memory operands, the number on the left side of the slash (/) is for byte operands and the number on the right side is for word operands.

For conditional control transfer or branch instructions, the number on the left side of the slash is applicable if the transfer or branch takes place. The number on the right side is applicable if it does not take place.

If a range of numbers is given, the execution time depends on the operands involved.

### Symbols

Symbol	Meaning
acc	Accumulator (AW or AL)
disp	Displacement (8 or 16 bits)
dmem	Direct memory address
dst	Destination operand or address
ext-disp8	16-bit displacement (sign-extension byte + 8-bit displacement)
far_label	Label within a different program segment
far_proc	Procedure within a different program segment
fp_op	Floating point instruction operation
imm	8- or 16-bit immediate operand
imm3/4	3- or 4-bit immediate bit offset
imm8	8-bit immediate operand
imm16	16-bit immediate operand
mem	Memory field (000 to 111); 8- or 16-bit memory location

### Symbols

Symbol	Meaning
mem8	8-bit memory location
mem16	16-bit memory location
mem32	32-bit memory location
memptr16	Word containing the destination address within the current segment
memptr32	Double word containing a destination address in another segment
mod	Mode field (00 to 10)
near_label	Label within the current segment
near_proc	Procedure within the current segment
offset	Immediate offset data (16 bits)
pop_value	Number of bytes to discard from the stack
reg	Register field (000 to 111); 8- or 16-bit general-purpose register
reg8	8-bit general-purpose register
reg16	16-bit general-purpose register
regptr	16-bit register containing a destination address within the current segment
regptr16	Register containing a destination address within the current segment
seg	Immediate segment data (16 bits)
short_label	Label between -128 and +127 bytes from the end of the current instruction
sr	Segment register
src	Source operand or address
temp	Temporary register (8/16/32 bits)
AC	Auxiliary carry flag
AH	Accumulator (high byte)
AL	Accumulator (low byte)
AW	Accumulator (16 bits)
BH	BW register (high byte)
BL	BW register (low byte)
BP	BP register
BRK	Break flag
BW	BW register (16 bits)
CH	CW register (high byte)
CL	CW register (low byte)
CW	CW register (16 bits)
CY	Carry flag
DH	DW register (high byte)
DIR	Direction flag
DL	DW register (low byte)

**Symbols (cont)**

Symbol	Meaning
DS0	Data segment 0 register (16 bits)
DS1	Data segment 1 register (16 bits)
DW	DW register (16 bits)
IE	Interrupt enable flag
IX	Index register (source) (16 bits)
IY	Index register (destination) (16 bits)
MD	Mode flag
P	Parity flag
PC	Program counter (16 bits)
PS	Program segment register (16 bits)
PSW	Program status word (16 bits)
R	Register set
S	Sign extend operand field S = 0 No sign extension S = 1 Sign extend immediate byte operand
S	Sign flag
SP	Stack pointer (16 bits)
SS	Stack segment register (16 bits)
V	Overflow flag
W	Word/byte field (0 to 1)
X, XXX, YYY, ZZZ	Data to identify the instruction code of the external floating point arithmetic chip
XOR	Exclusive logical sum
XXH	Two-digit hexadecimal value
XXXXH	Four-digit hexadecimal value
Z	Zero flag

**Flag Operations**

Symbol	Meaning
(blank)	No change
0	Cleared to 0
1	Set to 1
x	Set or cleared according to result
u	Undefined
R	Restored to previous state

**Memory Addressing Modes**

mem	mod = 00	mod = 01	mod = 10
000	BW + IX	BW + IX + disp8	BW + IX + disp16
001	BW + IY	BW + IY + disp8	BW + IY + disp16
010	BP + IX	BP + IX + disp8	BP + IX + disp16
011	BP + IY	BP + IY + disp8	BP + IY + disp16
100	IX	IX + disp8	IX + disp16
101	IY	IY + disp8	IY + disp16
110	Direct	BP + disp8	BP + disp16
111	BW	BW + disp8	BW + disp16

**Register Selection (mod = 11)**

reg	W = 0	W = 1
000	AL	AW
001	CL	CW
010	DL	DW
011	BL	BW
100	AH	SP
101	CH	BP
110	DH	IX
111	BH	IY

**Segment Register Selection**

sr	Segment Register
00	DS1
01	PS
10	SS
11	DS0

### Instruction Set

Mnemonic	Operand	Opcode											Clocks	Bytes	Flags							
		7	6	5	4	3	2	1	0	7	6	5			4	3	2	1	0	AC	CY	V
<b>Data Transfer Instructions</b>																						
MOV	reg, reg	1	0	0	0	1	0	1	W	1	1	reg	reg	2	2							
	mem, reg	1	0	0	0	1	0	0	W	mod	reg	mem	7/11	2-4								
	reg, mem	1	0	0	0	1	0	1	W	mod	reg	mem	10/14	2-4								
	mem, imm	1	1	0	0	0	1	1	W	mod	reg	mem	9/13	3-6								
	reg, imm	1	0	1	1	W	reg					4	2-3									
	acc, dmem	1	0	1	0	0	0	0	W				10/14	3								
	dmem, acc	1	0	1	0	0	0	1	W				9/13	3								
	sr, reg16	1	0	0	0	1	1	1	0	1	1	0	sr	reg	2	2						
	sr, mem16	1	0	0	0	1	1	1	0	mod	0	sr	mem	14	2-4							
	reg16, sr	1	0	0	0	1	1	0	0	1	1	0	sr	reg	2	2						
	mem16, sr	1	0	0	0	1	1	0	0	mod	0	sr	mem	12	2-4							
	DS0, reg16, mem32	1	1	0	0	0	1	0	1	mod	reg	mem	25	2-4								
	DS1, reg16, mem32	1	1	0	0	0	1	0	0	mod	reg	mem	25	2-4								
	AH, PSW	1	0	0	1	1	1	1	1				2	1								
	PSW, AH	1	0	0	1	1	1	1	0				3	1			x	x		x	x	x
LDEA	reg16, mem16	1	0	0	0	1	1	0	1	mod	reg	mem	4	2-4								
TRANS	src_table	1	1	0	1	0	1	1	1				9	1								
XCH	reg, reg	1	0	0	0	0	1	1	W	1	1	reg	reg	3	2							
	mem, reg	1	0	0	0	0	1	1	W	mod	reg	mem	13/21	2-4								
	AW, reg16	1	0	0	1	0	reg					3	1									
<b>Repeat Prefixes</b>																						
REPC		0	1	1	0	0	1	0	1				2	1								
REPNC		0	1	1	0	0	1	0	0				2	1								
REP		1	1	1	1	0	0	1	1				2	1								
REPE																						
REPZ																						
REPNE		1	1	1	1	0	0	1	0				2	1								
REPZ																						
<b>Block Transfer Instructions</b>																						
MOVBK	dst, src	1	0	1	0	0	1	0	W				1									
													9 (9) + 8n (W = 0)									
													9 (17) + 16n (W = 1)									
CMPBK	dst, src	1	0	1	0	0	1	1	W				1				x	x	x	x	x	x
													7 (13) + 14n (W = 0)									
													7 (21) + 22n (W = 1)									
CMPM	dst	1	0	1	0	1	1	1	W				1				x	x	x	x	x	x
													7 (7) + 10n (W = 0)									
													7 (11) + 14n (W = 1)									
LDM	src	1	0	1	0	1	1	0	W				1									
													7 (7) + 9n (W = 0)									
													7 (11) + 13n (W = 1)									
STM	dst	1	0	1	0	1	0	1	W				1									
													5 (5) + 4n (W = 0)									
													5 (9) + 8n (W = 1)									

n = number of transfers

String instruction execution clocks for a single instruction execution are in parentheses.

**Instruction Set (cont)**

Mnemonic	Operand	Opcode										Clocks	Bytes	Flags											
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P	S	Z
<b>I/O Instructions</b>																									
IN	acc, imm8	1	1	1	0	0	1	0	W									9/13	2						
	acc, DW	1	1	1	0	1	1	0	W									8/12	1						
OUT	imm8, acc	1	1	1	0	0	1	1	W									8/12	2						
	DW, acc	1	1	1	0	1	1	1	W									8/12	1						
INM	dst, DW	0	1	1	0	1	1	0	W									1							
																		9 (10) + 8n (W = 0) 9 (18) + 16n (W = 1)							
OUTM	DW, src	0	1	1	0	1	1	1	W									1							
																		9 (10) + 8n (W = 0) 9 (18) + 16n (W = 1)							

n = number of transfers

String instruction execution clocks for a single instruction execution are in parentheses.

**BCD Instructions**

ADJBA		0	0	1	1	0	1	1	1									7	1	x	x	u	u	u	u
ADJ4A		0	0	1	0	0	1	1	1									3	1	x	x	u	x	x	x
ADJBS		0	0	1	1	1	1	1	1									7	1	x	x	u	u	u	u
ADJ4S		0	0	1	0	1	1	1	1									3	1	x	x	u	x	x	x
ADD4S	dst, src	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	7 + 19n	2	u	x	u	u	u	x
SUB4S	dst, src	0	0	0	0	1	1	1	1	0	0	1	0	0	0	1	0	7 + 19n	2	u	x	u	u	u	x
CMP4S	dst, src	0	0	0	0	1	1	1	1	0	0	1	0	0	1	1	0	7 + 19n	2	u	x	u	u	u	x
ROL4	reg8	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	13	3						
		1	1	0	0	0			reg																
	mem8	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	25	3-5						
		mod	0	0	0			mem																	
ROR4	reg8	0	0	0	0	1	1	1	1	0	0	1	0	1	0	1	0	17	3						
		1	1	0	0	0			reg																
	mem8	0	0	0	0	1	1	1	1	0	0	1	0	1	0	1	0	29	3-5						
		mod	0	0	0			mem																	

n = number of BCD digits divided by 2

**Data Type Conversion Instructions**

CVTBD		1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	15	2	u	u	u	x	x	x
CVTDB		1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	7	2	u	u	u	x	x	x
CVTBW		1	0	0	1	1	0	0	0									2	1						
CVTWL		1	0	0	1	1	0	0	1									4/5	1						

**Arithmetic Instructions**

ADD	reg, reg	0	0	0	0	0	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x		
	mem, reg	0	0	0	0	0	0	0	W	mod	reg	mem	13/21	2-4	x	x	x	x	x	x			
	reg, mem	0	0	0	0	0	0	1	W	mod	reg	mem	10/14	2-4	x	x	x	x	x	x			
	reg, imm	1	0	0	0	0	0	S	W	1	1	0	0	0	reg	4	3-4	x	x	x	x	x	x
	mem, imm	1	0	0	0	0	0	S	W	mod	0	0	0	mem	15/23	3-6	x	x	x	x	x	x	
	acc, imm	0	0	0	0	0	1	0	W							4	2-3	x	x	x	x	x	x

## Instruction Set (cont)

Mnemonic	Operand	Opcode										Clocks	Bytes	Flags									
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P
<b>Arithmetic Instructions (cont)</b>																							
ADDC	reg, reg	0	0	0	1	0	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x		
	mem, reg	0	0	0	1	0	0	0	W	mod	reg	mem	mem	13/21	2-4	x	x	x	x	x	x		
	reg, mem	0	0	0	1	0	0	1	W	mod	reg	mem	mem	10/14	2-4	x	x	x	x	x	x		
	reg, imm	1	0	0	0	0	0	S	W	1	1	0	1	0	reg	4	3-4	x	x	x	x	x	x
	mem, imm	1	0	0	0	0	0	S	W	mod	0	1	0	mem	15/23	3-6	x	x	x	x	x	x	
	acc, imm	0	0	0	1	0	1	0	W					4	2-3	x	x	x	x	x	x		
SUB	reg, reg	0	0	1	0	1	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x		
	mem, reg	0	0	1	0	1	0	0	W	mod	reg	mem	mem	13/21	2-4	x	x	x	x	x	x		
	reg, mem	0	0	1	0	1	0	1	W	mod	reg	mem	mem	10/14	2-4	x	x	x	x	x	x		
	reg, imm	1	0	0	0	0	0	S	W	1	1	1	0	1	reg	4	3-4	x	x	x	x	x	x
	mem, imm	1	0	0	0	0	0	S	W	mod	1	0	1	mem	15/23	3-6	x	x	x	x	x	x	
	acc, imm	0	0	1	0	1	1	0	W					4	2-3	x	x	x	x	x	x		
SUBC	reg, reg	0	0	0	1	1	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x		
	mem, reg	0	0	0	1	1	0	0	W	mod	reg	mem	mem	13/21	2-4	x	x	x	x	x	x		
	reg, mem	0	0	0	1	1	0	1	W	mod	reg	mem	mem	10/14	2-4	x	x	x	x	x	x		
	reg, imm	1	0	0	0	0	0	S	W	1	1	0	1	1	reg	4	3-4	x	x	x	x	x	x
	mem, imm	1	0	0	0	0	0	S	W	mod	0	1	1	mem	15/23	3-6	x	x	x	x	x	x	
	acc, imm	0	0	0	1	1	1	0	W					4	2-3	x	x	x	x	x	x		
INC	reg8	1	1	1	1	1	1	1	0	1	1	0	0	0	reg	2	2	x	x	x	x	x	x
	mem	1	1	1	1	1	1	1	W	mod	0	0	0	mem	13/21	2-4	x	x	x	x	x	x	
	reg16	0	1	0	0	0			reg					2	1	x	x	x	x	x	x		
DEC	reg8	1	1	1	1	1	1	1	0	1	1	0	0	1	reg	2	2	x	x	x	x	x	x
	mem	1	1	1	1	1	1	1	W	mod	0	0	1	mem	13/21	2-4	x	x	x	x	x	x	
	reg16	0	1	0	0	1			reg					2	1	x	x	x	x	x	x		
MULU	reg	1	1	1	1	0	1	1	W	1	1	1	0	0	reg	21-30	2	u	x	x	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	0	0	mem	26-39	2-4	u	x	x	u	u	u	
MUL	reg	1	1	1	1	0	1	1	W	1	1	1	0	1	reg	33-47	2	u	x	x	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	0	1	mem	38-56	2-4	u	x	x	u	u	u	
	reg16, reg16, imm8	0	1	1	0	1	0	1	1	1	1	reg	reg	28-34	3	u	x	x	u	u	u		
	reg16, mem16, imm8	0	1	1	0	1	0	1	1	mod	reg	mem	mem	37-43	3-5	u	x	x	u	u	u		
	reg16, reg16, imm16	0	1	1	0	1	0	0	1	1	1	reg	reg	36-42	4	u	x	x	u	u	u		
	reg16, mem16, imm16	0	1	1	0	1	0	0	1	mod	reg	mem	mem	45-51	4-6	u	x	x	u	u	u		
DIVU	reg	1	1	1	1	0	1	1	W	1	1	1	1	0	reg	19-25	2	u	u	u	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	1	0	mem	24-34	2-4	u	u	u	u	u	u	
DIV	reg	1	1	1	1	0	1	1	W	1	1	1	1	1	reg	29-43	2	u	u	u	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	1	1	mem	34-52	2-4	u	u	u	u	u	u	



**Instruction Set (cont)**

Mnemonic	Operand	Opcode											Clocks	Bytes	Flags								
		7	6	5	4	3	2	1	0	7	6	5			4	3	2	1	0	AC	CY	V	P
<b>Comparison Instructions</b>																							
CMP	reg, reg	0	0	1	1	1	0	1	W	1	1		reg	reg	2	2	x	x	x	x	x	x	
	mem, reg	0	0	1	1	1	0	0	W	mod		reg	mem	10/14	2-4	x	x	x	x	x	x		
	reg, mem	0	0	1	1	1	0	1	W	mod		reg	mem	10/14	2-4	x	x	x	x	x	x		
	reg, imm	1	0	0	0	0	0	S	W	1	1	1	1	1	reg	4	3-4	x	x	x	x	x	x
	mem, imm	1	0	0	0	0	0	S	W	mod	1	1	1	mem	12/16	3-6	x	x	x	x	x	x	
	acc, imm	0	0	1	1	1	1	0	W						4	2-3	x	x	x	x	x	x	
<b>Logical Instructions</b>																							
NOT	reg	1	1	1	1	0	1	1	W	1	1	0	1	0	reg	2	2						
	mem	1	1	1	1	0	1	1	W	mod	0	1	0	mem	13/21	2-4							
NEG	reg	1	1	1	1	0	1	1	W	1	1	0	1	1	reg	2	2	x	x	x	x	x	x
	mem	1	1	1	1	0	1	1	W	mod	0	1	1	mem	13/21	2-4	x	x	x	x	x	x	
TEST	reg, reg	1	0	0	0	0	1	0	W	1	1		reg	reg	2	2	u	0	0	x	x	x	
	mem, reg	1	0	0	0	0	1	0	W	mod		reg	mem	9/13	2-4	u	0	0	x	x	x		
	reg, imm	1	1	1	1	0	1	1	W	1	1	0	0	0	reg	4	3-4	u	0	0	x	x	x
	mem, imm	1	1	1	1	0	1	1	W	mod	0	0	0	mem	10/14	3-6	u	0	0	x	x	x	
	acc, imm	1	0	1	0	1	0	0	W						4	2-3	u	0	0	x	x	x	
AND	reg, reg	0	0	1	0	0	0	1	W	1	1		reg	reg	2	2	u	0	0	x	x	x	
	mem, reg	0	0	1	0	0	0	0	W	mod		reg	mem	13/21	2-4	u	0	0	x	x	x		
	reg, mem	0	0	1	0	0	0	1	W	mod		reg	mem	10/14	2-4	u	0	0	x	x	x		
	reg, imm	1	0	0	0	0	0	0	W	1	1	1	0	0	reg	4	3-4	u	0	0	x	x	x
	mem, imm	1	0	0	0	0	0	0	W	mod	1	0	0	mem	15/23	3-6	u	0	0	x	x	x	
	acc, imm	0	0	1	0	0	1	0	W						4	2-3	u	0	0	x	x	x	
OR	reg, reg	0	0	0	0	1	0	1	W	1	1		reg	reg	2	2	u	0	0	x	x	x	
	mem, reg	0	0	0	0	1	0	0	W	mod		reg	mem	13/21	2-4	u	0	0	x	x	x		
	reg, mem	0	0	0	0	1	0	1	W	mod		reg	mem	10/14	2-4	u	0	0	x	x	x		
	reg, imm	1	0	0	0	0	0	0	W	1	1	0	0	1	reg	4	3-4	u	0	0	x	x	x
	mem, imm	1	0	0	0	0	0	0	W	mod	0	0	1	mem	15/23	3-6	u	0	0	x	x	x	
	acc, imm	0	0	1	0	0	1	0	W						4	2-3	u	0	0	x	x	x	
XOR	reg, reg	0	0	1	1	0	0	1	W	1	1		reg	reg	2	2	u	0	0	x	x	x	
	mem, reg	0	0	1	1	0	0	0	W	mod		reg	mem	13/21	2-4	u	0	0	x	x	x		
	reg, mem	0	0	1	1	0	0	1	W	mod		reg	mem	10/14	2-4	u	0	0	x	x	x		
	reg, imm	1	0	0	0	0	0	0	W	1	1	1	1	0	reg	4	3-4	u	0	0	x	x	x
	mem, imm	1	0	0	0	0	0	0	W	mod	1	1	0	mem	15/23	3-6	u	0	0	x	x	x	
	acc, imm	0	0	1	0	0	1	0	W						4	2-3	u	0	0	x	x	x	

### Instruction Set (cont)

Mnemonic	Operand	Opcode														Clocks	Bytes	Flags							
		7	6	5	4	3	2	1	0	7	6	5	4	3	2			1	0	AC	CY	V	P	S	Z
<b>Bit Manipulation Instructions</b>																									
INS	reg8, reg8	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	1	35-133	3						
		1	1	reg			reg																		
	reg8, imm8	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	1	35-133	4						
		1	1	0	0	0		reg																	
EXT	reg8, reg8	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1	34-59	3						
		1	1	reg			reg																		
	reg8, imm8	0	0	0	0	1	1	1	1	0	0	1	1	1	0	1	1	34-59	4						
		1	1	0	0	0		reg																	
TEST1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	W	3	3	u	0	0	u	u	x
		1	1	0	0	0		reg																	
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	W	7/11	3-5	u	0	0	u	u	x
		mod	0	0	0	0		mem																	
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0	W	4	4	u	0	0	u	u	x
		1	1	0	0	0		reg																	
	mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0	W	8/12	4-6	u	0	0	u	u	x
		mod	0	0	0	0		mem																	
SET1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	0	W	4	3						
		1	1	0	0	0		reg																	
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	0	W	10/18	3-5						
		mod	0	0	0	0		mem																	
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0	W	5	4						
		1	1	0	0	0		reg																	
CLR1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	W	5	3						
		1	1	0	0	0		reg																	
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	W	11/19	3-5						
		mod	0	0	0	0		mem																	
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	W	6	4						
		1	1	0	0	0		reg																	
NOT1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	1	W	4	3						
		1	1	0	0	0		reg																	
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	1	W	10/18	3-5						
		mod	0	0	0	0		mem																	
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	W	5	4						
		1	1	0	0	0		reg																	
	mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	W	11/19	4-6						
		mod	0	0	0	0		mem																	
	CY	1	1	1	1	0	1	0	1									2	1						
	DIR	1	1	1	1	1	0	1										2	1						

3

**Instruction Set (cont)**

Mnemonic	Operands	Opcode																Clocks	Bytes	Flags					
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
<b>Shift/Rotate Instructions</b>																									
SHL	reg, 1	1	1	0	1	0	0	0	W	1	1	1	0	0	reg	2	2	u	x	x	x	x	x		
	mem, 1	1	1	0	1	0	0	0	W	mod	1	0	0	mem	13/21	2-4	u	x	x	x	x	x			
	reg, CL	1	1	0	1	0	0	1	W	1	1	1	0	0	reg	7 + n	2	u	x	u	x	x	x		
	mem, CL	1	1	0	1	0	0	1	W	mod	1	0	0	mem	16/24 + n	2-4	u	x	u	x	x	x			
	reg, imm8	1	1	0	0	0	0	0	W	1	1	1	0	0	reg	7 + n	3	u	x	u	x	x	x		
	mem, imm8	1	1	0	0	0	0	0	W	mod	1	0	0	mem	16/24 + n	3-5	u	x	u	x	x	x			
SHR	reg, 1	1	1	0	1	0	0	0	W	1	1	1	0	1	reg	2	2	u	x	x	x	x	x		
	mem, 1	1	1	0	1	0	0	0	W	mod	1	0	1	mem	13/21	2-4	u	x	x	x	x	x			
	reg, CL	1	1	0	1	0	0	1	W	1	1	1	0	1	reg	7 + n	2	u	x	u	x	x	x		
	mem, CL	1	1	0	1	0	0	1	W	mod	1	0	1	mem	16/24 + n	2-4	u	x	u	x	x	x			
	reg, imm8	1	1	0	0	0	0	0	W	1	1	1	0	1	reg	7 + n	3	u	x	u	x	x	x		
	mem, imm8	1	1	0	0	0	0	0	W	mod	1	0	1	mem	16/24 + n	3-5	u	x	u	x	x	x			
SHRA	reg, 1	1	1	0	1	0	0	0	W	1	1	1	1	1	reg	2	2	u	x	0	x	x	x		
	mem, 1	1	1	0	1	0	0	0	W	mod	1	1	1	mem	13/21	2-4	u	x	0	x	x	x			
	reg, CL	1	1	0	1	0	0	1	W	1	1	1	1	1	reg	7 + n	2	u	x	u	x	x	x		
	mem, CL	1	1	0	1	0	0	1	W	mod	1	1	1	mem	16/24 + n	2-4	u	x	u	x	x	x			
	reg, imm8	1	1	0	0	0	0	0	W	1	1	1	1	1	reg	7 + n	3	u	x	u	x	x	x		
	mem, imm8	1	1	0	0	0	0	0	W	mod	1	1	1	mem	16/24 + n	3-5	u	x	u	x	x	x			
ROL	reg, 1	1	1	0	1	0	0	0	W	1	1	0	0	0	reg	2	2	x	x						
	mem, 1	1	1	0	1	0	0	0	W	mod	0	0	0	mem	13/21	2-4	x	x							
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	0	0	reg	7 + n	2	x	u						
	mem, CL	1	1	0	1	0	0	1	W	mod	0	0	0	mem	16/24 + n	2-4	x	u							
	reg, imm	1	1	0	0	0	0	0	W	1	1	0	0	0	reg	7 + n	3	x	u						
	mem, imm	1	1	0	0	0	0	0	W	mod	0	0	0	mem	16/24 + n	3-5	x	u							
ROR	reg, 1	1	1	0	1	0	0	0	W	1	1	0	0	1	reg	2	2	x	u						
	mem, 1	1	1	0	1	0	0	0	W	mod	0	0	1	mem	13/21	2-4	x	x							
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	0	1	reg	7 + n	2	x	u						
	mem, CL	1	1	0	1	0	0	1	W	mod	0	0	1	mem	16/24 + n	2-4	x	u							
	reg, imm8	1	1	0	0	0	0	0	W	1	1	0	0	1	reg	7 + n	3	x	u						
	mem, imm8	1	1	0	0	0	0	0	W	mod	0	0	1	mem	16/24 + n	3-5	x	u							
ROLC	reg, 1	1	1	0	1	0	0	0	W	1	1	0	1	0	reg	2	2	x	x						
	mem, 1	1	1	0	1	0	0	0	W	mod	0	1	0	mem	13/21	2-4	x	x							
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	1	0	reg	7 + n	2	x	u						
	mem, CL	1	1	0	1	0	0	1	W	mod	0	1	0	mem	16/24 + n	2-4	x	u							
	reg, imm8	1	1	0	0	0	0	0	W	1	1	0	1	0	reg	7 + n	3	x	u						
	mem, imm8	1	1	0	0	0	0	0	W	mod	0	1	0	mem	16/24 + n	3-5	x	u							

n = number of shifts

## Instruction Set (cont)

Mnemonic	Operands	Opcode														Clocks	Bytes	Flags					
		7	6	5	4	3	2	1	0	7	6	5	4	3	2			1	0	AC	CY	V	P
<b>Shift Rotate Instructions (cont)</b>																							
RORC	reg, 1	1	1	0	1	0	0	0	W	1	1	0	1	1	reg	2	2			x	x		
	mem, 1	1	1	0	1	0	0	0	W	mod	0	1	1	mem	13/21	2-4			x	x			
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	1	1	reg	7 + n	2			x	u		
	mem, CL	1	1	0	1	0	0	1	W	mod	0	1	1	mem	16/24 + n	2-4			x	u			
	reg, imm8	1	1	0	0	0	0	0	W	1	1	0	1	1	reg	7 + n	3			x	u		
	mem, imm8	1	1	0	0	0	0	0	W	mod	0	1	1	mem	16/24 + n	3-5			x	u			

n = number of shifts

## Stack Manipulation Instructions

PUSH	mem16	1	1	1	1	1	1	1	1	mod	1	1	0	mem	23	2-4								
	reg16	0	1	0	1	0			reg						10	1								
	sr	0	0	0	sr	1	1	0							10	1								
	PSW	1	0	0	1	1	1	0	0						10	1								
	R	0	1	1	0	0	0	0	0						65	1								
	imm	0	1	1	0	1	0	S	0						9-10	2-3								
POP	mem16	1	0	0	0	1	1	1	1	mod	0	0	0	mem	25	2-4								
	reg16	0	1	0	1	1			reg						12	1								
	sr	0	0	0	sr	1	1	1							12	1								
	PSW	1	0	0	1	1	1	0	1						12	1			R	R	R	R	R	R
	R	0	1	1	0	0	0	0	1						75	1								
PREPARE	imm16, imm8	1	1	0	0	1	0	0	0						*	4								

\*imm8 = 0 : 16  
imm8 > 1 : 21 + 16 (imm8 - 1)

DISPOSE		1	1	0	0	1	0	0	1						10	1						
---------	--	---	---	---	---	---	---	---	---	--	--	--	--	--	----	---	--	--	--	--	--	--

## Control Transfer Instructions

CALL	near_proc	1	1	1	0	1	0	0	0						20	3							
	regptr	1	1	1	1	1	1	1	1	1	1	0	1	0	reg	18	1						
	memptr16	1	1	1	1	1	1	1	1	mod	0	1	0	mem	31	2-4							
	far_proc	1	0	0	1	1	0	1	0						29	5							
	memptr32	1	1	1	1	1	1	1	1	mod	0	1	1	mem	47	2-4							
RET		1	1	0	0	0	0	1	1						19	1							
	pop_value	1	1	0	0	0	0	1	0						24	3							
		1	1	0	0	1	0	1	1						29	1							
	pop_value	1	1	0	0	1	0	1	0						32	3							
BR	near_label	1	1	1	0	1	0	0	1						13	3							
	short_label	1	1	1	0	1	0	0	1						12	2							
	reg	1	1	1	1	1	1	1	1	1	1	1	0	0	reg	11	2						
	memptr16	1	1	1	1	1	1	1	1	mod	1	0	0	mem	23	2-4							
	far_label	1	1	1	0	1	0	1	0						15	5							
	memptr32	1	1	1	1	1	1	1	1	mod	1	0	1	mem	34	2-4							
BV	near_label	0	1	1	1	0	0	0	0						14/4	2							
BNV	near_label	0	1	1	1	0	0	0	1						14/4	2							

**Instruction Set (cont)**

Mnemonic	Operands	Opcode										Clocks	Bytes	Flags											
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P	S	Z
<b>Control Transfer Instructions (cont)</b>																									
BC, BL	near_Label	0	1	1	1	0	0	1	0								14/4	2							
BNC, BNL	near_Label	0	1	1	1	0	0	1	1								14/4	2							
BE, BZ	near_Label	0	1	1	1	0	1	0	0								14/4	2							
BNE, BNZ	near_Label	0	1	1	1	0	1	0	1								14/4	2							
BNH	near_Label	0	1	1	1	0	1	1	0								14/4	2							
BH	near_Label	0	1	1	1	0	1	1	1								14/4	2							
BN	near_Label	0	1	1	1	1	0	0	0								14/4	2							
BP	near_Label	0	1	1	1	1	0	0	1								14/4	2							
BPE	near_Label	0	1	1	1	1	0	1	0								14/4	2							
BPO	near_Label	0	1	1	1	1	0	1	1								14/4	2							
BLT	near_Label	0	1	1	1	1	1	0	0								14/4	2							
BGE	near_Label	0	1	1	1	1	1	0	1								14/4	2							
BLE	near_Label	0	1	1	1	1	1	1	0								14/4	2							
BGT	near_Label	0	1	1	1	1	1	1	1								14/4	2							
DBNZNE	near_Label	1	1	1	0	0	0	0	0								14/5	2							
DBNZE	near_Label	1	1	1	0	0	0	0	1								14/5	2							
DBNZ	near_Label	1	1	1	0	0	0	1	0								13/5	2							
BCWZ	near_Label	1	1	1	0	0	0	1	1								13/5	2							
<b>Interrupt Instructions</b>																									
BRK	3	1	1	0	0	1	1	0	0								50	1							
	imm8	1	1	0	0	1	1	0	1								50	2							
BRKV	imm8	1	1	0	0	1	1	1	0								52/3	1							
RETI		1	1	0	0	1	1	1	1								39	1	R	R	R	R	R	R	
CHKIND	reg16, mem32	0	1	1	0	0	0	1	0	mod	reg	mem					25/72-75	2-4							
BRKEM	imm8	0	0	0	0	1	1	1	1	1	1	1	1	1	1		50	3							
<b>CPU Control Instructions</b>																									
HALT		1	1	1	1	0	1	0	0								2	1							
BUSLOCK		1	1	1	1	0	0	0	0								2	1							
FP01	fp_op	1	1	0	1	1	X	X	X	1	1	Y	Y	Y	Z	Z	2	2							
	fp_op, mem	1	1	0	1	1	X	X	X	mod	Y	Y	Y	mem			14	2-4							
FP02	fp_op	0	1	1	0	0	1	1	X	1	1	Y	Y	Y	Z	Z	2	2							
	fp_op, mem	0	1	1	0	0	1	1	X	mod	Y	Y	Y	mem			14	2-4							
POLL		1	0	0	1	1	0	1	1								2 + 5n	1							
n = number of times POLL pin is sampled.																									
NOP		1	0	0	1	0	0	0	0								3	1							
DI		1	1	1	1	1	0	1	0								2	1							
EI		1	1	1	1	1	0	1	1								2	1							
DS0, DS1, PS, and SS: (segment override prefixes)		0	0	1	seg	1	1	0								2	1								
<b>8080 Instruction Set Enhancements</b>																									
RETEM		1	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	39	2	R	R	R	R	R	R
CALLN	imm8	1	1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	58	3						

### Description

The μPD70216 (V50™) is a high-performance, low-power 16-bit microprocessor integrating a number of commonly-used peripherals to dramatically reduce the size of microprocessor systems. The CMOS construction makes the μPD70216 ideal for the design of portable computers, instrumentation, and process control equipment.

The μPD70216 contains a powerful instruction set that is compatible with the μPD70108/μPD70116 (V20™/V30™) and μPD8086/μPD8088 instruction sets. Instruction set support includes a wide range of arithmetic, logical, and control operations as well as bit manipulation, BCD arithmetic, and high-speed block transfer instructions. The μPD70216 can also execute the entire μPD8080AF instruction set using the 8080 emulation mode. Also available is the μPD70208 (V40™), identical to the μPD70216 but with an 8-bit external data bus.

### Features

- Low-power CMOS technology
- V20/V30 instruction set compatible
- Minimum instruction execution time: 250 ns (at 8 MHz)
- Direct addressing of 1M bytes of memory
- Powerful set of addressing modes
- Fourteen 16-bit CPU registers
- On-chip peripherals including
  - Clock generator
  - Bus interface
  - Bus arbitration
  - Programmable wait state generator
  - DRAM refresh controller
  - Three 16-bit timer/counters
  - Asynchronous serial I/O controller
  - Eight-input interrupt controller
  - Four-channel DMA controller
- Hardware effective address calculation logic
- Maskable and nonmaskable interrupt inputs
- IEEE 796 compatible bus interface
- Low-power standby mode

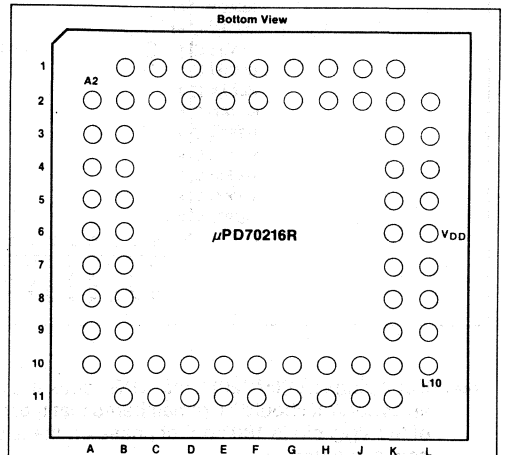
V20, V30, V40, and V50 are trademarks of NEC Corporation.

### Ordering Information

Part Number	Package	Maximum Frequency
μPD70216R-8	68-pin ceramic PGA	8 MHz
μPD70216L-8	68-pin PLCC	8 MHz
μPD70216GF-8	80-pin plastic flat pack	8 MHz
μPD70216R-10	68-pin ceramic PGA	10 MHz
μPD70216L-10	68-pin PLCC	10 MHz
μPD70216GF-10	80-pin plastic flat pack	10 MHz

### Pin Configurations

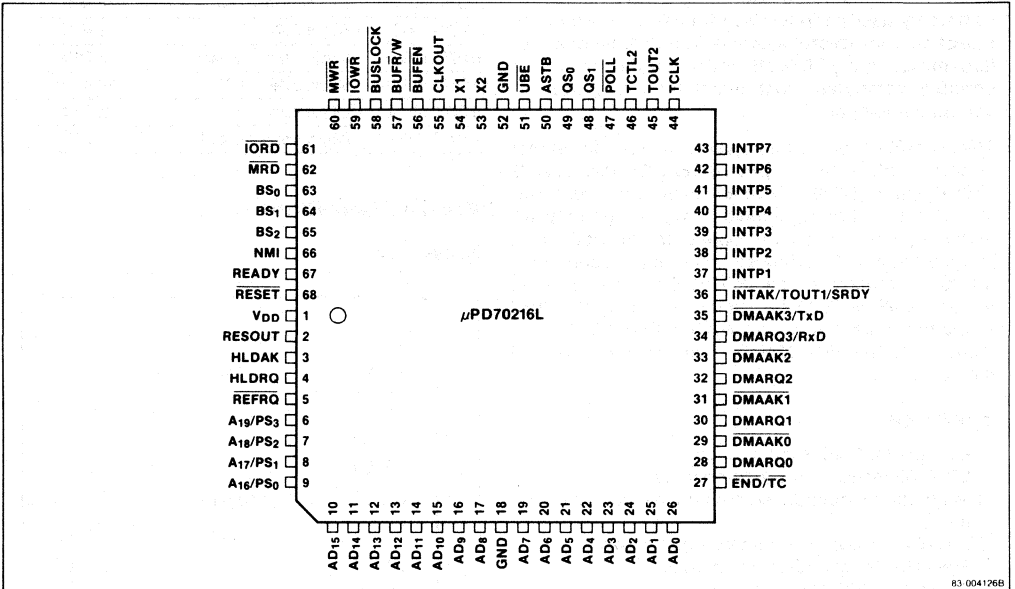
#### 68-Pin PGA



Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol
A2	INTP7	B9	DMARQ1	F10	AD7	K4	NMI
A3	INTP5	B10	DMARQ0	F11	GND	K5	RESET
A4	INTP3	B11	AD0	G1	X1	K6	RESOUT
A5	INTP1	C1	TCTL2	G2	CLKOUT	K7	HLDRQ
A6	DMAAK3/TxD	C2	POLL	G10	AD8	K8	A19/PS3
A7	DMAAK2	C10	AD1	G11	AD9	K9	A17/PS1
A8	DMAAK1	C11	AD2	H1	BUFEN	K10	AD14
A9	DMAAK0	D1	QS1	H2	BUFEN/W	K11	AD15
A10	END/TC	D2	QS0	H10	AD10	L2	IORD
B1	TCLK	D10	AD3	H11	AD11	L3	BS0
B2	TOUT2	D11	AD4	J1	BUSLOCK	L4	BS2
B3	INTP6	E1	ASTB	J2	IOWR	L5	READY
B4	INTP4	E2	UBE	J10	AD12	L6	VDD
B5	INTP2	E10	AD5	J11	AD13	L7	HLDAK
B6	INTAK/TOUT1/SRDY	E11	AD6	K1	MWR	L8	REFRQ
B7	DMARQ3/RxD	F1	GND	K2	MWD	L9	A18/PS2
B8	DMARQ2	F2	X2	K3	BS1	L10	A16/PS0

**Pin Configurations (cont)**

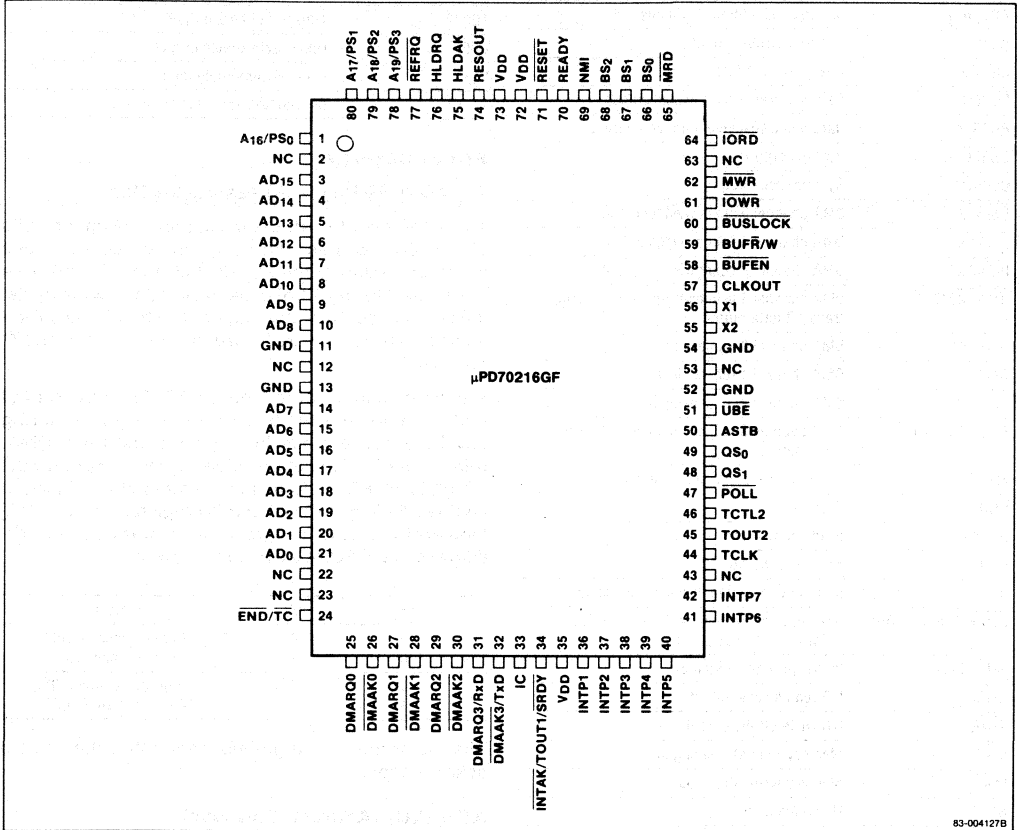
**68-Pin Plastic Leaded Chip Carrier (PLCC) (Top View)**



**Notes:** If one plugs a μPD70208L into a PGA socket, by use of an adaptor, it should be remembered that the location of the V<sub>DD</sub> pin is 180° out of phase on these two package types.

### Pin Configurations (cont)

#### 80-Pin Plastic Flat Pack (Top View)



83-004127B



**Pin Identification**

Symbol	Function
A <sub>19</sub> -A <sub>16</sub> /PS <sub>3</sub> -PS <sub>0</sub>	Multiplexed address/processor status outputs
AD <sub>15</sub> -AD <sub>0</sub>	Multiplexed address/data bus
ASTB	Address strobe output
BS <sub>2</sub> -BS <sub>0</sub>	Bus status outputs
BUFEN	Data bus transceiver enable output
BUFR/W	Data bus transceiver direction output
BUSLOCK	Buslock output
CLKOUT	System clock output
DMAAK0	DMA channel 0 acknowledge output
DMAAK1	DMA channel 1 acknowledge output
DMAAK2	DMA channel 2 acknowledge output
DMAAK3/TxD	DMA channel 3 acknowledge output/Serial transmit data output
DMARQ0	DMA channel 0 request input
DMARQ1	DMA channel 1 request input
DMARQ2	DMA channel 2 request input
DMARQ3/RxD	DMA channel 3 request input/Serial receive data input
END/TC	End input/Terminal count output
GND	Ground
HLDAK	Hold acknowledge output
HLDRQ	Hold request input
IC	Internal connection; leave unconnected
INTAK/TOUT1/SRDY	Interrupt acknowledge output/Timer/counter 1 output/Serial ready output
INTP1-INTP7	Interrupt request inputs
IORD	I/O read strobe output
IOWR	I/O write strobe output
MRD	Memory read strobe output
MWR	Memory write strobe output
NC	No connection
NMI	Nonmaskable interrupt input
POLL	Poll input
QS <sub>1</sub> -QS <sub>0</sub>	CPU queue status outputs
READY	Ready input
REFRQ	Refresh request output
RESET	Reset input
RESOUT	Synchronized reset output

Symbol	Function
TCLK	Timer/counter external clock input
TCTL2	Timer/counter 2 control input
TOUT2	Timer/counter 2 output
UBE	Upper byte enable output
V <sub>DD</sub>	+5 V power supply input
X1, X2	Crystal/external clock inputs

**Pin Functions**

**A<sub>19</sub>-A<sub>16</sub>/PS<sub>3</sub>-PS<sub>0</sub> [Address/Status Bus]**

These three-state output pins contain the upper 4 bits of the 20-bit address during T1 and processor status information during the T2, T3, TW, and T4 states of a bus cycle. During T1 of a memory read or write cycle, these pins contain the upper 4 bits of the 20-bit address. These pins are forced low during T1 of an I/O bus cycle.

Processor status is output during T2, T3, TW, and T4 of both memory and I/O bus cycles. PS<sub>3</sub> is zero during any CPU native mode bus cycle. During any DMA, refresh, or 8080 emulation mode bus cycle, PS<sub>3</sub> outputs a high level. PS<sub>2</sub> outputs the contents of the interrupt enable (IE) flag in the CPU PSW register. PS<sub>1</sub> and PS<sub>0</sub> indicate the segment register used to form the physical address of a CPU bus cycle as follows:

PS <sub>1</sub>	PS <sub>0</sub>	Segment
0	0	Data segment 1 (DS1)
0	1	Stack segment (SS)
1	0	Program segment (PS)
1	1	Data segment 0 (DS0)

These pins are in the high-impedance state during hold acknowledge.

**AD<sub>15</sub>-AD<sub>0</sub> [Address/Data Bus]**

These three-state pins form the middle byte of the active-high, time-multiplexed address/data bus. During T1 of a bus cycle, AD<sub>15</sub>-AD<sub>0</sub> output the lower 16 bits of the 20-bit memory or I/O address. During the T2, T3, TW, and T4 states, AD<sub>15</sub>-AD<sub>0</sub> form the 16-bit bidirectional data bus.

The memory and I/O address spaces are organized into a pair of byte-wide banks. The even bank is accessed whenever AD<sub>0</sub> = 0 during T1 of a bus cycle. Access to the odd bank is controlled by the UBE pin.

The AD<sub>15</sub>-AD<sub>0</sub> pins enter the high-impedance state during hold acknowledge or internal interrupt acknowledge bus cycles or while RESET is asserted. Pins AD<sub>10</sub>-AD<sub>8</sub> contain the slave address of an external interrupt controller during the second interrupt acknowledge bus cycle.

## ASTB [Address Strobe]

This active-high output is used to latch the address from the multiplexed address bus in an external address latch during T1 of a bus cycle. ASTB is held at a low level during hold acknowledge.

## BS<sub>2</sub>-BS<sub>0</sub> [Bus Status]

Outputs BS<sub>2</sub>-BS<sub>0</sub> indicate the type of bus cycle being performed as shown below. BS<sub>2</sub>-BS<sub>0</sub> become active during the state preceding T1 and return to the passive state during the bus state preceding T4.

BS <sub>2</sub>	BS <sub>1</sub>	BS <sub>0</sub>	Bus Cycle
0	0	0	Interrupt acknowledge
0	0	1	I/O read
0	1	0	I/O write
0	1	1	Halt
1	0	0	Instruction fetch
1	0	1	Memory read (1)
1	1	0	Memory write (2)
1	1	1	Passive state

### Note:

- (1) Memory read bus cycles include CPU, DMA read, DMA verify, and refresh bus cycles.
- (2) Memory write bus cycles include CPU and DMA write bus cycles.

BS<sub>2</sub>-BS<sub>0</sub> are three-state outputs and are high impedance during hold acknowledge.

## BUFEN [Buffer Enable]

BUFEN is an active-low output for enabling an external data bus transceiver during a bus cycle. BUFEN is asserted during T2 through T3 of a read cycle, T2 through T3 of a slave interrupt acknowledge cycle, and T1 through T4 of a write cycle. BUFEN is not asserted when the bus cycle corresponds to an internal peripheral, DMA, refresh, or internal interrupt acknowledge cycle. BUFEN enters the high-impedance state during hold acknowledge.

## BUFR/W [Buffer Read/Write]

BUFR/W is a three-state, active-low output used to control the direction of an external data bus transceiver during CPU bus cycles. A high level indicates the μPD70216 will perform a write cycle and a low level indicates a read cycle. BUFR/W enters the high-impedance state during hold acknowledge.

## BUSLOCK

This active-low output provides a means for the CPU to indicate to an external bus arbiter that the bus cycles of the next instruction are to be kept contiguous. BUSLOCK is asserted for the duration of the instruction following the BUSLOCK prefix. BUSLOCK is also asserted during interrupt acknowledge cycles and enters the high-impedance state during hold acknowledge. While BUSLOCK is asserted, DMAU, RCU, and external bus requests are ignored.

## CLKOUT

CLKOUT is a buffered clock output used as a reference for all timing. CLKOUT has a 50-percent duty cycle at half the frequency of the input clock source.

## DMAAK0-DMAAK2 [DMA Acknowledge]

This set of outputs contains the DMA acknowledge signals for channels 0-2 from the internal DMA controller and indicate that the peripheral can perform the requested transfer.

## DMAAK3/TxD [DMA Acknowledge 3]/[Serial Transmit Data]

Two output signals multiplexed on this pin are selected by the PF field of the on-chip peripheral connection register.

- DMAAK3 is an active-low output and enables an external DMA peripheral to perform the requested DMA transfer for channel 3.
- TxD is the serial data output from the serial control unit.

## DMARQ0-DMARQ2 [DMA Request]

These synchronized inputs are used by external peripherals to request DMA service for channels 0-2 from the internal DMA controller.

## DMARQ3/RxD [DMA Request 3]/[Serial Receive Data]

Two input signals multiplexed on this pin are selected by the PF field of the on-chip peripheral connection register.

- DMARQ3 is used by an external peripheral to request a DMA transfer cycle for channel 3.
- RxD is the serial data input to the serial control unit.

**END/TC [End/Terminal Count]**

This active-low bidirectional pin controls the termination of a DMA service. Assertion of **END** by external hardware during DMA service causes the service to terminate. When a DMA channel reaches its terminal count, the DMAU asserts **TC**, indicating the programmed operation has completed.

**END/TC** is an open-drain I/O pin, and requires an external 2.2-kΩ pull-up resistor.

**HLDK [Hold Acknowledge]**

When an external bus requester has become the highest priority requester, the internal bus arbiter will assert the **HLDK** output indicating the address, data, and control buses have entered a high-impedance state and are available for use by the external bus master.

Should the internal DMAU or RCU (demand mode) request the bus, the bus arbiter will drive **HLDK** low. When this occurs, the external bus master should complete the current bus cycle and negate the **HLDK** signal. This allows the bus arbiter to reassign the bus to the higher priority requester.

If a higher priority internal bus master subsequently requests the bus, the high-level width of **HLDK** is guaranteed to be a minimum of one **CLKOUT** period.

**HLDK [Hold Request]**

This active-high signal is asserted by an external bus master requesting to use the local address, data, and control buses. The **HLDK** input is used by the internal bus arbiter, which gives control of the buses to the highest priority bus requester in the following order.

Bus Master	Priority
RCU	Highest (demand mode)
DMAU	•
HLDK	•
CPU	•
RCU	Lowest (normal operation)

**INTAK/TOUT1/SRDY [Interrupt Acknowledge]/  
[Timer 1 Output]/[Serial Ready]**

Three output signals multiplexed on this pin are selected by the PF field of the on-chip peripheral connection register.

- **INTAK** is an interrupt acknowledge signal used to cascade external slave μPD71059 Interrupt Controllers. **INTAK** is asserted during T2, T3, and TW states of an interrupt acknowledge cycle.

- **TOUT1** is the output of timer/counter 1.
- **SRDY** is an active-low output and indicates that the serial control unit is ready to receive the next character.

**INTP1-INTP7 [Peripheral Interrupts]**

**INTP1-INTP7** accept either rising-edge or high-level triggered asynchronous interrupt requests from external peripherals. These **INTP1-INTP7** inputs are internally synchronized and prioritized by the interrupt control unit, which requests the CPU to perform an interrupt acknowledge bus cycle. External interrupt controllers such as the μPD71059 can be cascaded to increase the number of vectored interrupts.

These interrupt inputs cause the CPU to exit both the standby and 8080 emulation modes.

The **INTP1-INTP7** inputs contain internal pull-up resistors and may be left unconnected.

**IOR [I/O Read]**

This three-state pin outputs an active-low I/O read strobe during T2, T3, and Tw of an I/O read bus cycle. Both CPU I/O read and DMA write bus cycles assert **IOR**. **IOR** is not asserted when the bus cycle corresponds to an internal peripheral or register. It enters the high-impedance state during hold acknowledge.

**IOW [I/O Write]**

This three-state pin outputs an active-low I/O write strobe during T2, T3, and Tw of a CPU I/O write or an extended DMA read cycle and during T3 and TW of a DMA read bus cycle. **IOW** is not asserted when the bus cycle corresponds to an internal peripheral or register. It enters the high-impedance state during hold acknowledge.

**MR [Memory Read Strobe]**

This three-state pin outputs an active-low memory read strobe during T2, T3, and TW of a memory read bus cycle. CPU memory read, DMA read, and refresh bus cycles all assert **MR**. **MR** enters the high-impedance state during hold acknowledge.

**MWR [Memory Write Strobe]**

This three-state pin outputs an active-low memory write strobe during T2, T3, and TW of a CPU memory write or DMA extended write bus cycle and during T3 and TW of a DMA normal write bus cycle. **MWR** enters the high-impedance state during hold acknowledge.

### NMI [Nonmaskable Interrupt]

The NMI pin is a rising-edge-triggered interrupt input that cannot be masked by software. NMI is sampled by CPU logic each clock cycle and when found valid for one or more CLKOUT cycles, the NMI interrupt is accepted. The CPU will process the NMI interrupt immediately after the current instruction finishes execution by fetching the segment and offset of the NMI handler from interrupt vector 2. The NMI interrupt causes the CPU to exit both the standby and 8080 emulation modes. The NMI input takes precedence over the maskable interrupt inputs.

### POLL [Poll]

The active-low  $\overline{\text{POLL}}$  input is used to synchronize the operation of external devices with the CPU. During execution of the POLL instruction, the CPU checks the  $\overline{\text{POLL}}$  input state every five clocks until  $\overline{\text{POLL}}$  is once again asserted.

### QS<sub>1</sub>-QS<sub>0</sub> [Queue Status]

The QS<sub>1</sub> and QS<sub>0</sub> outputs maintain instruction synchronization between the μPD70216 CPU and external devices such as the μPD72191 Floating Point Processor. These outputs are interpreted as follows.

QS <sub>1</sub>	QS <sub>0</sub>	Instruction Queue Status
0	0	No operation
0	1	First byte of instruction fetched
1	0	Flush queue contents
1	1	Subsequent byte of instruction fetched

Queue status is valid for one clock cycle after the CPU has accessed the instruction queue.

### READY [Ready]

This active-high input synchronizes external memory and peripheral devices with the μPD70216. Slow memory and I/O devices can lengthen a bus cycle by negating the READY input and forcing the BIU to insert TW states. READY must be negated prior to the rising edge of CLKOUT during the T2 or by the last internally-generated TW state to guarantee recognition. When READY is once again asserted and recognized by the BIU, the BIU will proceed to the T4 state.

The READY input operates in parallel with the internal μPD70216 wait control unit and can be used to insert more than three wait states into a bus cycle.

### REFRQ [Refresh Request]

REFRQ is an active-low output indicating the current bus cycle is a memory refresh operation. REFRQ is used to disable memory address decode logic and refresh dynamic memories. The 8-bit refresh row address is placed on A<sub>8</sub>-A<sub>1</sub> during a refresh bus cycle.

### RESET [Reset]

RESET is a Schmitt trigger input used to force the μPD70216 to a known state by resetting the CPU and on-chip peripherals. RESET must be asserted for a minimum of four clocks to guarantee recognition. To ensure proper rise time of the RESET pulse it is recommended to use an external schmitt trigger gate. After RESET has been released, the CPU will start program execution from address FFFF0H in the native mode.

RESET will release the CPU from the low-power standby mode and force it to the native mode.

### RESOUT [Reset Output]

This active-high output is available to perform a system-wide reset function. RESOUT is internally synchronized with CLKOUT and output on the RESOUT pin.

### TCLK

TCLK is an external clock source for the timer control unit. The three timer/counters can be programmed to operate with either the TCLK input or a prescaled CLKOUT input.

### TCTL2

TCTL2 is the control input for timer/counter 2.

### TOUT2

TOUT2 is the output of timer/counter 2.

**UBE [Upper Byte Enable]**

UBE is an active-low output, asserted when the upper byte of the 16-bit data bus contains valid data. UBE is used along with A0 by the memory decoding logic to select the even/odd banks as follows.

Operation	UBE	A0	Bus Cycles
Word, even address	0	0	1
Word, odd address	0	1 (1st bus cycle)	2
	1	0 (2nd bus cycle)	
Byte, even address	1	0	1
Byte, odd address	0	1	1

UBE is a three-state output and enters the high-impedance state during hold acknowledge.

**X1, X2 [Clock Inputs]**

These pins accept either a parallel resonant, fundamental mode crystal or an external oscillator input with a frequency twice the desired operating frequency.

In the case of an external clock generator, the X2 pin can either be left unconnected or be driven by the complement of the X1 pin clock source.

**Pin States**

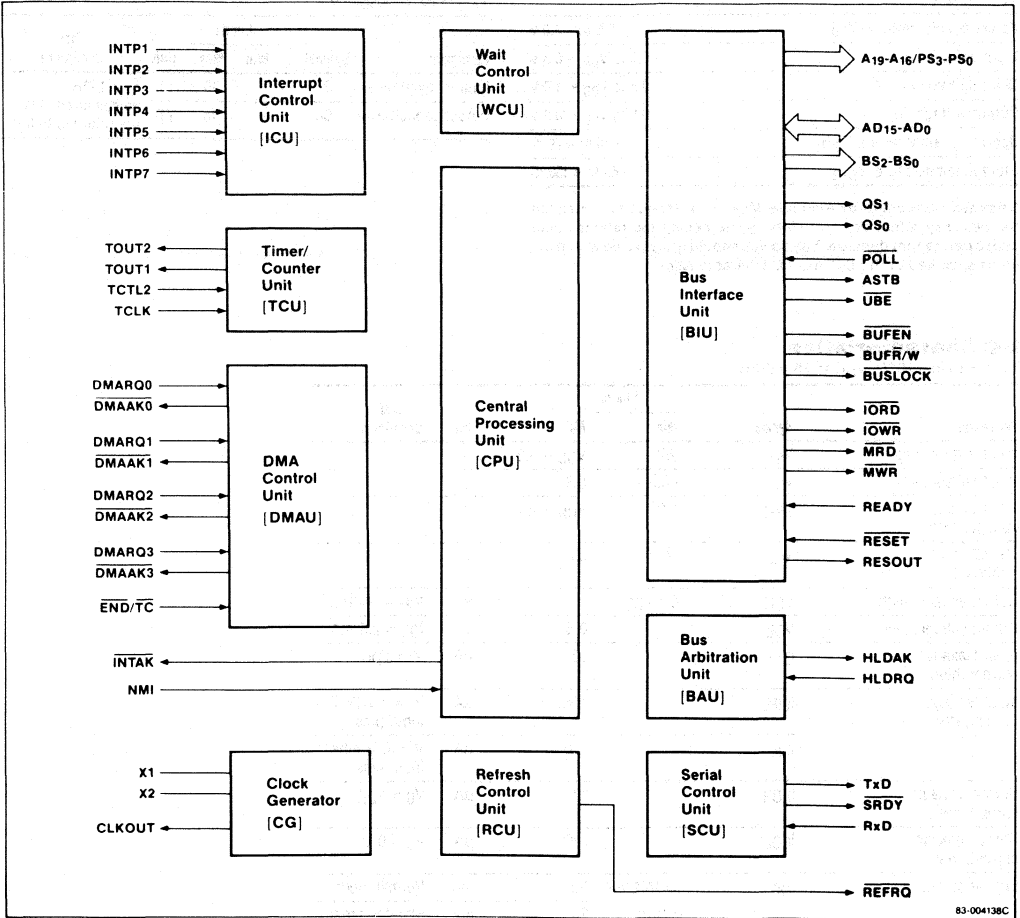
Table 1 lists the output pin states during the Hold, Halt, Reset and DMA Cascade conditions.

**Table 1. Output Pin States**

Symbol	Input/Output	Hold	Halt	Reset	DMA Cascade
A19-A16/PS3-PS0	3-state Out	Hi-Z	H/L	H/L	Hi-Z
AD15-AD0	3-state I/O	Hi-Z	H/L	Hi-Z	Hi-Z
ASTB	Out	L	L	L	L
BS2-BS0	3-state Out	Hi-Z	H	H	H
BUFEN	3-state Out	Hi-Z	H	H	H
BUFR/W	3-state Out	Hi-Z	H/L	H	H/L
BUSLOCK	3-state Out	Hi-Z	H/L	H	H
CLKOUT	Out	H/L	H/L	H/L	H/L
DMAAK0-DMAAK2	Out	H	H/L	H	H/L
DMAAK3/	Out	H	H/L	H	H/L
		H/L	H/L		
TxD		H/L	H/L		
END/TC	I/O	H	H/L	H	H
HLDK	Out	H	H/L	L	L
INTAK	Out	H	H	H	H
TOUT1		H/L	H/L		
SRDY		H/L	H/L		
IOR	3-state Out	Hi-Z	H	H	H
IOWR	3-state Out	Hi-Z	H	H	H
MRD	3-state Out	Hi-Z	H	H	H
MWR	3-state Out	Hi-Z	H	H	H
QS1-QS0	Out	H/L	L	L	L
REFRQ	Out	H	H/L	H	H
RESOUT	Out	L	L	H	H
TOUT2	Out	H/L	H/L	H/L	H/L
UBE	3-state Out	Hi-Z	H	H	H

H: high level; L: low level; H/L: high or low level; Hi-Z: high impedance.

## Block Diagram



3

## μPD70216 (V50)

### Absolute Maximum Ratings

T<sub>A</sub> = +25°C

Power supply voltage, V <sub>DD</sub>	-0.5 to +7.0 V
Input voltage, V <sub>I</sub>	-0.5 to V <sub>DD</sub> + 0.3 V
CLK input voltage, V <sub>K</sub>	-0.5 to V <sub>DD</sub> + 1.0 V
Output voltage, V <sub>O</sub>	-0.5 to V <sub>DD</sub> + 0.3 V
Operating temperature, T <sub>OP</sub> T	-10 to +70°C
Storage temperature, T <sub>STG</sub>	-65 to +150°C

**Comment:** Exposure to Absolute Maximum Ratings for extended periods may affect device reliability; exceeding the ratings could cause permanent damage. The device should be operated within the limits specified under DC and AC Characteristics.

### Capacitance

T<sub>A</sub> = +25°C, V<sub>DD</sub> = 0 V

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input capacitance	C <sub>I</sub>		15	pF	f <sub>C</sub> = 1 MHz; unmeasured pins are returned to 0 V.
Output capacitance	C <sub>O</sub>		15	pF	

### DC Characteristics

T<sub>A</sub> = -10 to +70°C, V<sub>DD</sub> = +5 V ±10%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input voltage, high	V <sub>IH</sub>	2.2	V <sub>DD</sub> + 0.3	V	
Input voltage, low	V <sub>IL</sub>	-0.5	0.8	V	
X1, X2 input voltage, high	V <sub>KH</sub>	3.9	V <sub>DD</sub> + 1.0	V	
X1, X2 input voltage, low	V <sub>KL</sub>	-0.5	0.6	V	
Output voltage, high	V <sub>OH</sub>	0.7 V <sub>DD</sub>		V	I <sub>OH</sub> = -400 μA
Output voltage, low	V <sub>OL</sub>		0.4	V	I <sub>OL</sub> = 2.5 mA
Input leakage current, high	I <sub>LIH</sub>		10	μA	V <sub>I</sub> = V <sub>DD</sub>
Input leakage current, low	I <sub>LIPL</sub>		-300	μA	V <sub>I</sub> = 0 V, INTP input pins
	I <sub>LJL</sub>		-10	μA	V <sub>I</sub> = 0 V, other input pins
Output leakage current, high	I <sub>LOH</sub>		10	μA	V <sub>O</sub> = V <sub>DD</sub>
Output leakage current, low	I <sub>LOL</sub>		-10	μA	V <sub>O</sub> = 0 V
Supply current	I <sub>DD</sub>	μPD70216-8 (8 MHz)	90	mA	Normal mode
			20	mA	Standby mode
		μPD70216-10 (10 MHz)	120	mA	Normal mode
			25	mA	Standby mode

### AC Characteristics

μPD70208-8: T<sub>A</sub> = -10 to +70°C; V<sub>DD</sub> = +5 V ± 10%; C<sub>L</sub> = 100 pF

μPD70208-10: T<sub>A</sub> = -10 to +70°C; V<sub>DD</sub> = +5 V ± 5%; C<sub>L</sub> = 100 pF

Parameter	Symbol	μPD70216-8		μPD70216-10		Unit	Test Condition
		Min	Max	Min	Max		
External clock input cycle time	t <sub>CYX</sub>	62	250	50	250	ns	
External clock pulse width, high	t <sub>XXH</sub>	20		19		ns	V <sub>KH</sub> = 3.0 V
External clock pulse width, low	t <sub>XXL</sub>	20		19		ns	V <sub>KL</sub> = 1.5 V
External clock rise time	t <sub>XR</sub>		10		5	ns	1.5 → 3.0 V
External clock fall time	t <sub>XF</sub>		10		5	ns	3.0 → 1.5 V
CLKOUT cycle time	t <sub>CYK</sub>	124	500	100	500	ns	
CLKOUT pulse width, high	t <sub>KKH</sub>	0.5 t <sub>CYK</sub> -7		0.5 t <sub>CYK</sub> -5		ns	V <sub>KH</sub> = 3.0 V
CLKOUT puls width, low	t <sub>KKL</sub>	0.5 t <sub>CYK</sub> -7		0.5 t <sub>CYK</sub> -5		ns	V <sub>KL</sub> = 1.5 V
CLKOUT rise time	t <sub>KR</sub>		7		5	ns	1.5 → 3.0 V
CLKOUT fall time	t <sub>KF</sub>		7		5	ns	3.0 → 1.5 V
CLKOUT delay time from external clock	t <sub>DXK</sub>		55		40	ns	
Input rise time (incl. RESET, but except external clock)	t <sub>IR</sub>		20		15	ns	0.8 → 2.2 V
Input fall time (incl. RESET, but external clock)	t <sub>IF</sub>		12		10	ns	2.2 → 0.8 V
Output rise time (except CLKOUT)	t <sub>OR</sub>		20		15	ns	0.8 → 2.2 V
Output fall time (except CLKOUT)	t <sub>OF</sub>		12		10	ns	2.2 → 0.8 V
RESET setup time to CLKOUT <sup>†</sup>	t <sub>SRESK</sub>	25		20		ns	
RESET hold time after CLKOUT <sup>†</sup>	t <sub>HKRES</sub>	35		25		ns	
RESOUT delay time from CLKOUT <sup>†</sup>	t <sub>DKRES</sub>	5	60	5	50	ns	
READY inactive setup time to CLKOUT <sup>†</sup>	t <sub>SRYLK</sub>	15		15		ns	
READY inactive hold time after CLKOUT <sup>†</sup>	t <sub>HKRYL</sub>	25		20		ns	
READY active setup time to CLKOUT <sup>†</sup>	t <sub>SRYHK</sub>	15		15		ns	
READY active hold time after CLKOUT <sup>†</sup>	t <sub>HKRYH</sub>	25		20		ns	
NMI, POLL setup time to CLKOUT <sup>†</sup>	t <sub>SIK</sub>	15		15		ns	



**AC Characteristics (cont)**

Parameter	Symbol	μPD70216-8		μPD70216-10		Unit	Test Condition
		Min	Max	Min	Max		
Data setup time to CLKOUT↓	tSDK	15		15		ns	
Data hold time after CLKOUT↓	tHKD	10		10		ns	
Address delay time from CLKOUT↓	tDKA	10	55	10	50	ns	
Address hold time after CLKOUT↓	tHKA	10		10		ns	
I/O recovery time	tAI	2tCYK -50		2tCYK -40		ns	(Note 4)
PS delay time from CLKOUT↓	tDKP	10	60	10	50	ns	
PS float delay time from CLKOUT↑	tFKP	10	60	10	50	ns	
Address setup time to ASTB↓	tSAST	tKKL -30		tKKL -20		ns	
Address float delay time from CLKOUT↓	tFKA	tHKA	60	tHKA	50	ns	
ASTB↑ delay time from CLKOUT↓	tDKSTH		45		40	ns	
ASTB↓ delay time from CLKOUT↑	tDKSTL		50		45	ns	
ASTB pulse width, high	tSTST	tKKL -10		tKKL -10		ns	
Address hold time after ASTB↓	tHSTA	tKKH -20		tKKH -20		ns	
Control delay time from CLKOUT	tDKCT1	10	70	10	60	ns	(Note 1)
	tDKCT2	10	60	10	55		(Note 2)
R <sub>D</sub> ↓ delay time from address float	tDAFRL	0		0		ns	(Note 3)
R <sub>D</sub> ↓ delay time from CLKOUT↓	tDKRL	10	75	10	65	ns	
R <sub>D</sub> ↑ delay time from CLKOUT↓	tDKRH	10	70	10	60	ns	
REFRQ↑ delay from MRD↑	tDRQHRH	tKKL -30		tKKL -30		ns	(Note 5)
Address delay time from R <sub>D</sub>	tDRHA	tCYK -40		tCYK -40		ns	
R <sub>D</sub> pulse width, low	tRR	2tCYK -50		2tCYK -40		ns	
BUFR/W delay from BUFEN↑	tDBECT	tKKL -20		tKKL -20		ns	Read cycle
	tDWCT	tKKL -20		tKKL -20		ns	Write cycle
Data output delay time from CLKOUT↓	tDKD	10	60	10	55	ns	
Data float delay time from CLKOUT↓	tFKD	10	60	10	55	ns	
WR pulse width, low	tWW	2tCYK -40		2tCYK -40		ns	(Note 3)

### AC Characteristics (cont)

Parameter	Symbol	μPD70216-8		μPD70216-10		Unit	Test Condition
		Min	Max	Min	Max		
BS↓ delay time from CLKOUT↑	tDKBL	10	60	10	55	ns	
BS↑ delay time from CLKOUT↓	tDKBH	10	60	10	55	ns	
HLDRO setup time to CLKOUT↑	tSHQK	20		15		ns	
HLEDAK delay time from CLKOUT↓	tDKHA	10	70	10	60	ns	
Address drive delay time from CLKOUT↓	tDKA2	tCYK		tCYK		ns	
DMAAK↓ delay time from CLKOUT↑	tDKHDA	10	60	10	55	ns	
DMAAK↓ delay time from CLKOUT↓	tDKLDA	10	90	10	80	ns	Cascade mode
WR pulse width, low (DMA cycle)	tWW1	2tCYK - 40		2tCYK - 40		ns	DMA extended write cycle
WR pulse width, low (DMA cycle)	tWW2	tCYK - 40		tCYK - 40		ns	DMA normal write cycle
RD↓, WR↓ delay from DMAAK↓	tDDARW	tKKH - 30		tKKH - 30		ns	
DMAAK↑ delay from RD↑	tDRHDAH	tKKL - 30		tKKL - 30		ns	
RD↑ delay from WR↑	tDWHRH	5		5		ns	
TC output delay time from CLKOUT↑	tDKTCL		60		55	ns	
TC off delay time from CLKOUT↑	tDKTCF		60		55	ns	
TC pulse width, low	tTCTCL	tCYK - 15		tCYK - 15		ns	
TC pullup delay time from CLKOUT↑	tDKTCH	tKKH + tCYK - 10		tKKH + tCYK - 10		ns	
END setup time to CLKOUT↑	tSEDK	35		30		ns	
END pulse width, low	tEEDL	100		80		ns	
DMARQ setup time to CLKOUT↑	tSDQK	35		30		ns	
INTPh pulse width, low	tPIPL	100		80		ns	
RxD setup time to SCU internal clock↓	tSRX	1		0.5		μs	
RxD hold time after SCU internal clock↓	tHRX	1		0.5		μs	
SRDY delay time from CLKOUT↓	tDKSR		150		100	ns	

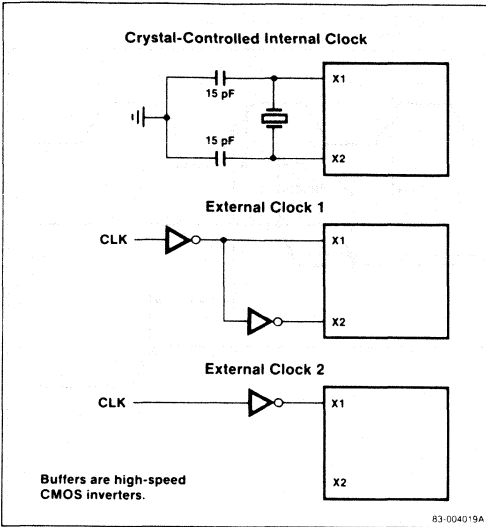
**AC Characteristics (cont)**

Parameter	Symbol	μPD70216-8		μPD70216-10		Unit	Test Condition
		Min	Max	Min	Max		
TxD delay time from TOUT1†	tDTX		500		200	ns	
TCTL2 setup time from CLKOUT‡	tSGK	50		40		ns	
TCTL2 setup time to TCLK†	tSGTK	50		40		ns	
TCTL2 hold time after CLKOUT‡	tHKG	100		80		ns	
TCTL2 hold time after TCLK†	tHTKG	50		40		ns	
TCTL2 pulse width, high	tGGH	50		40		ns	
TCTL2 pulse width, low	tGGL	50		40		ns	
TOUT output delay time from CLKOUT‡	tDKTO		200		150	ns	
TOUT output delay time from TOUT‡	tDTKTO		150		100	ns	
TOUT output delay time from TCTL2‡	tDGTO		120		90	ns	
TCLK rise time	tTKR		25		25	ns	
TCLK fall time	tTKF		25		25	ns	
TCLK pulse width, high	tTKTKH	50		45		ns	
TCLK pulse width, low	tTKTKL	50		45		ns	
TCLK cycle time	tCYTK	124	Dc	100	Dc	ns	
RESET pulse width low	tRESET1	50		50		μs	After power on
	tRESET2	4*tCYK		4*tCYK			During operation

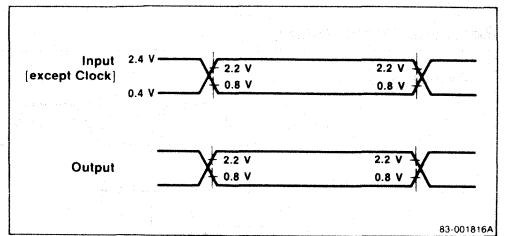
**Notes:**

- (1) Delay from CLKOUT to DMA cycle MWR/IOWR outputs.
- (2) Delay from CLKOUT to BUFR/W, BUFEN, INTAK, REFRQ outputs and CPU cycle MWR/IOWR outputs.
- (3) RD represents IORD and MRD. WR represents IOWR and MWR.
- (4) This is specified to guarantee a read/write recovery time for I/O device.
- (5) This is specified to guarantee that REFRQ† is delayed from MRD† at all times.

## μPD70216 Clock Input Configurations

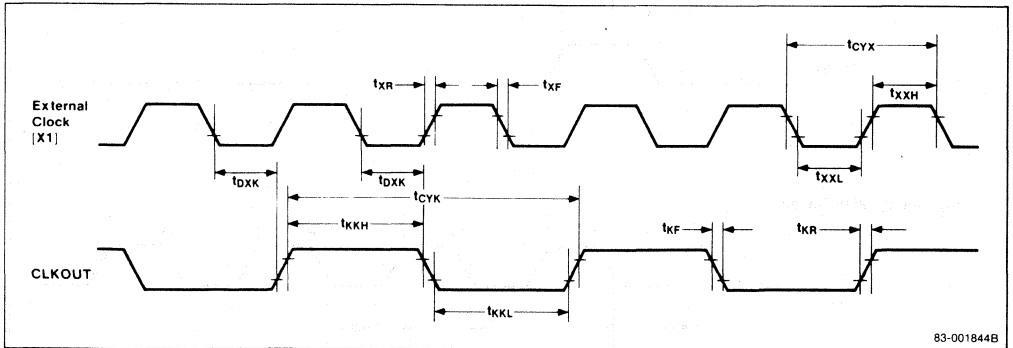


## Timing Measurement Points



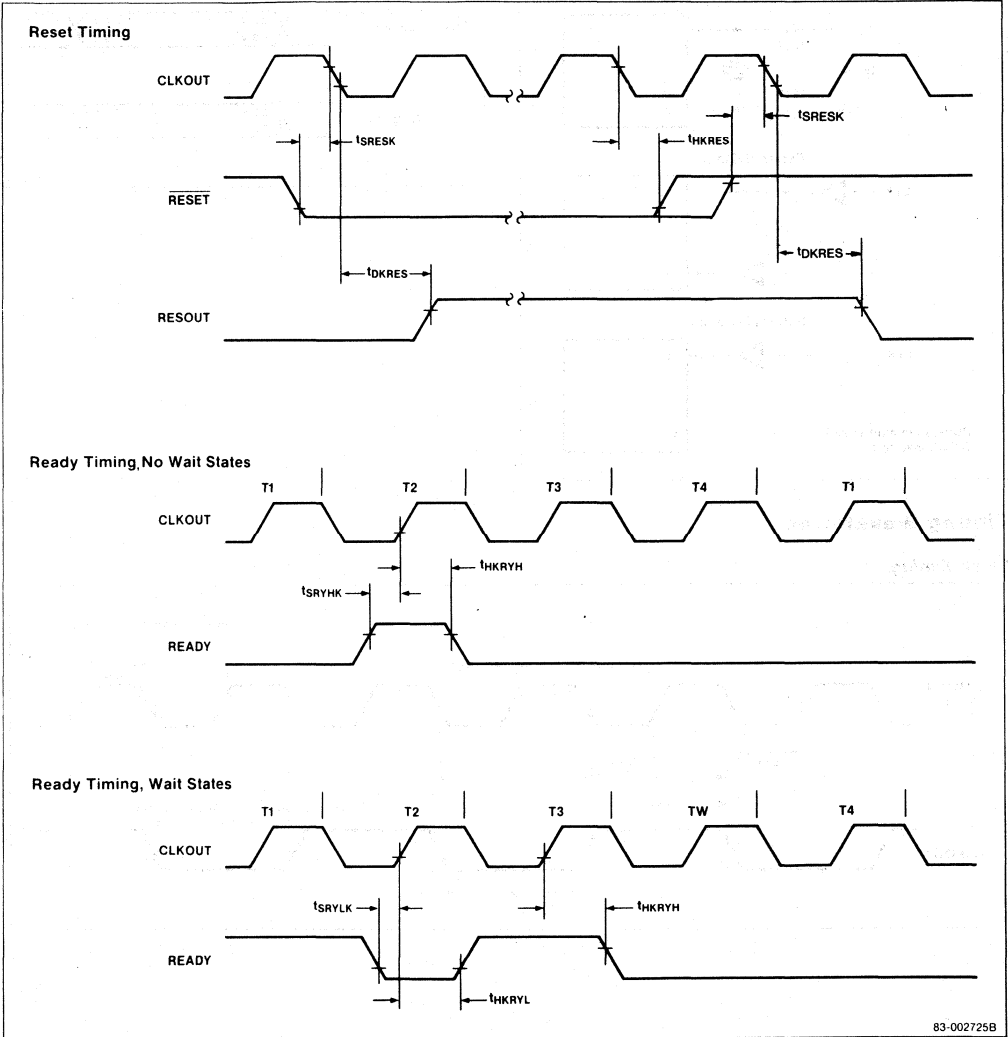
## Timing Waveforms

### Clock Timing



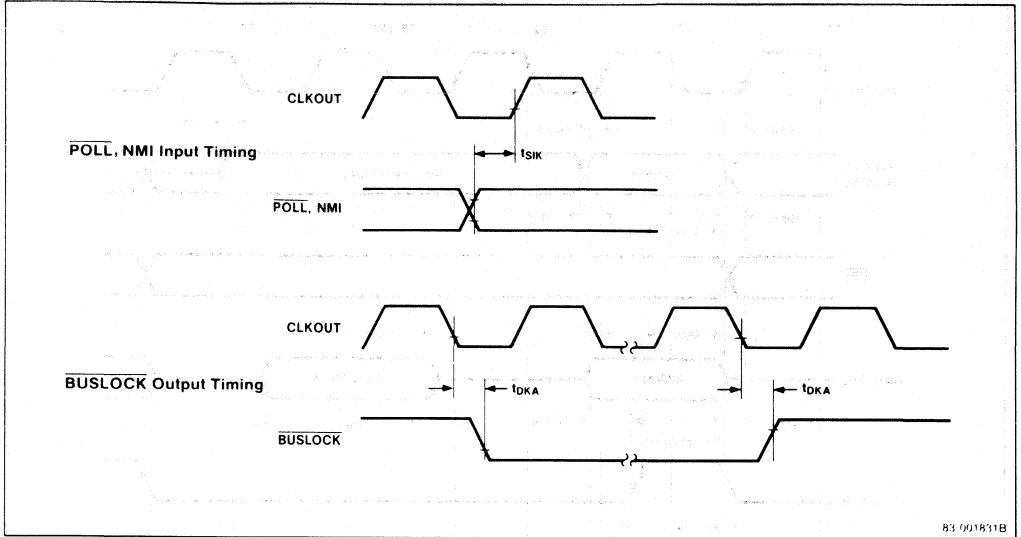
Timing Waveforms (cont)

Reset and Ready Timing

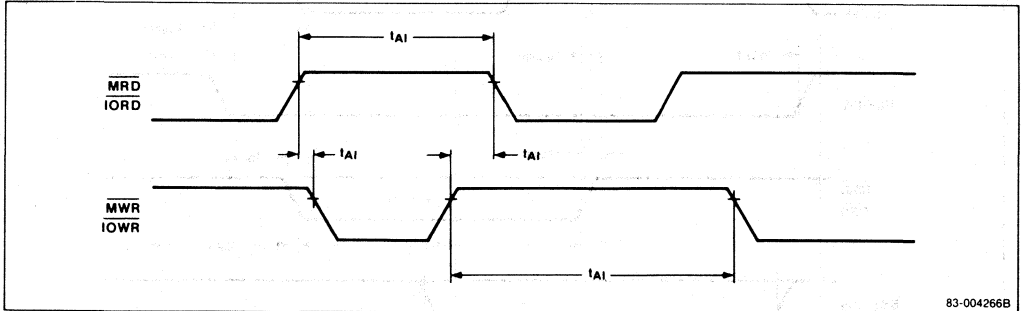


### Timing Waveforms (cont)

#### Poll, NMI, and Buslock Timing

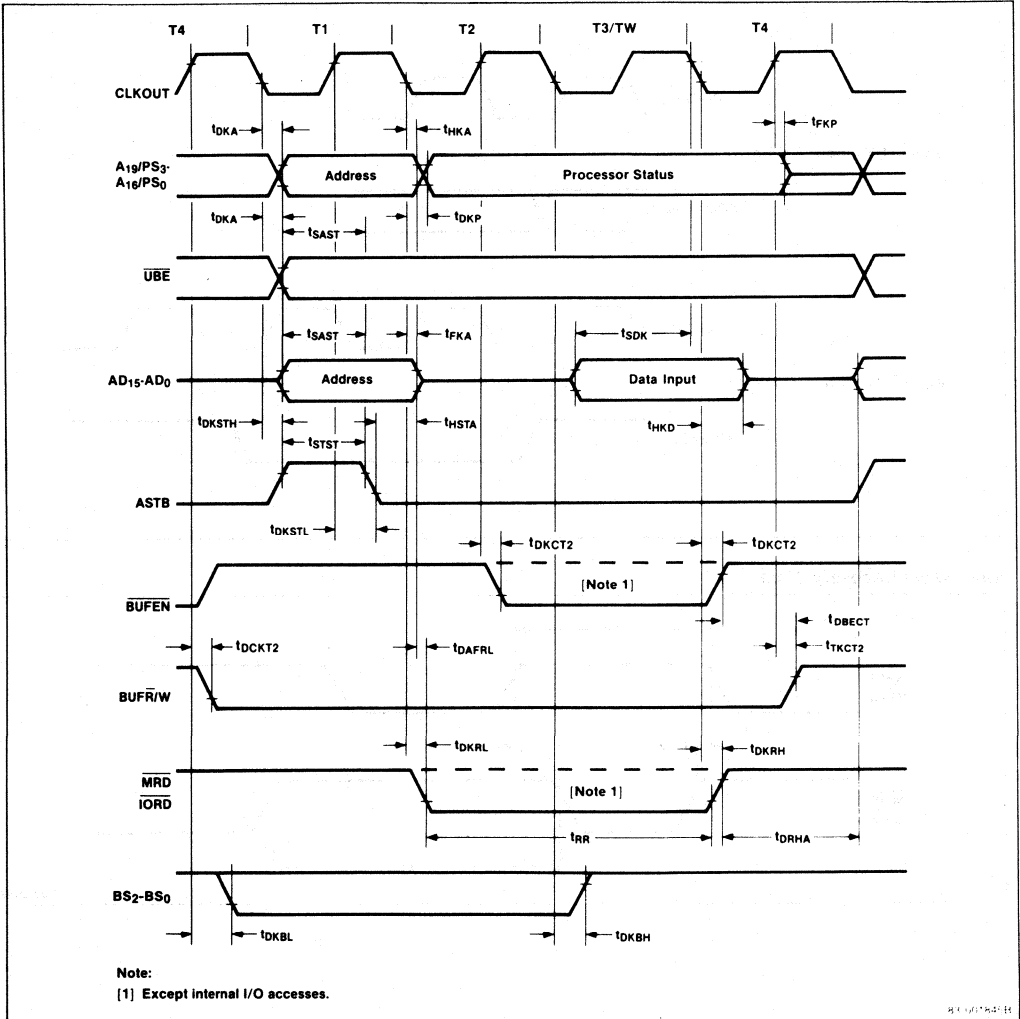


#### Read/Write Recovery Time



Timing Waveforms (cont)

Read Timing

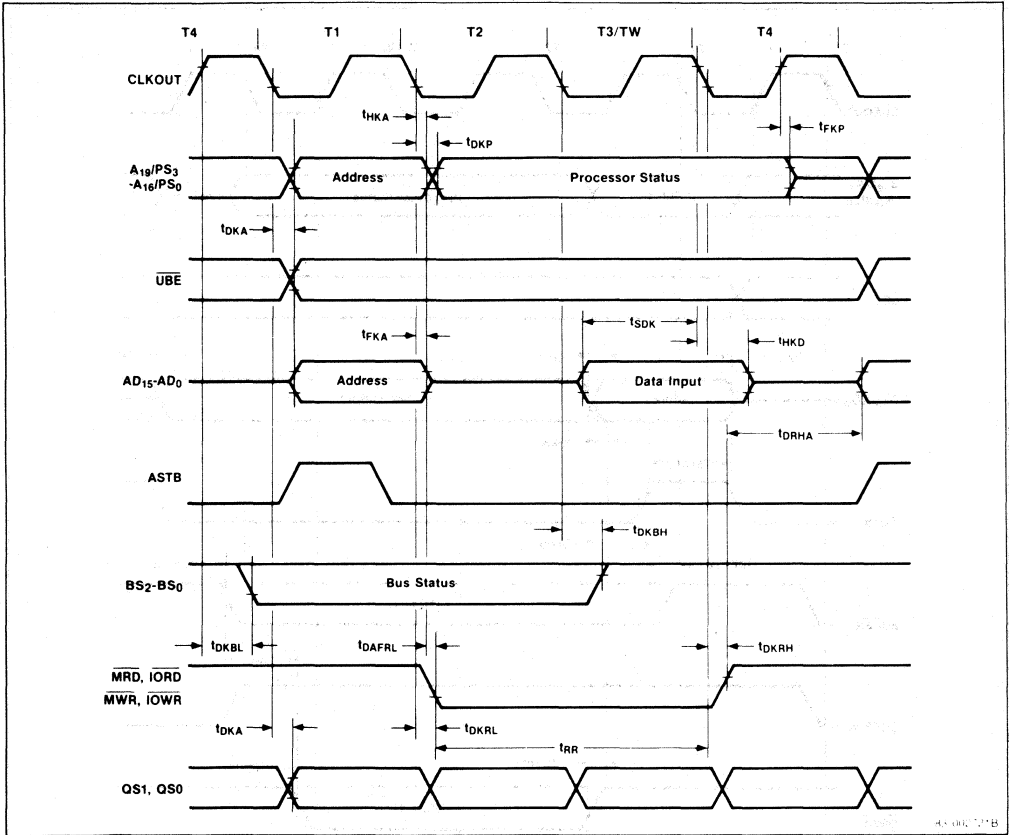






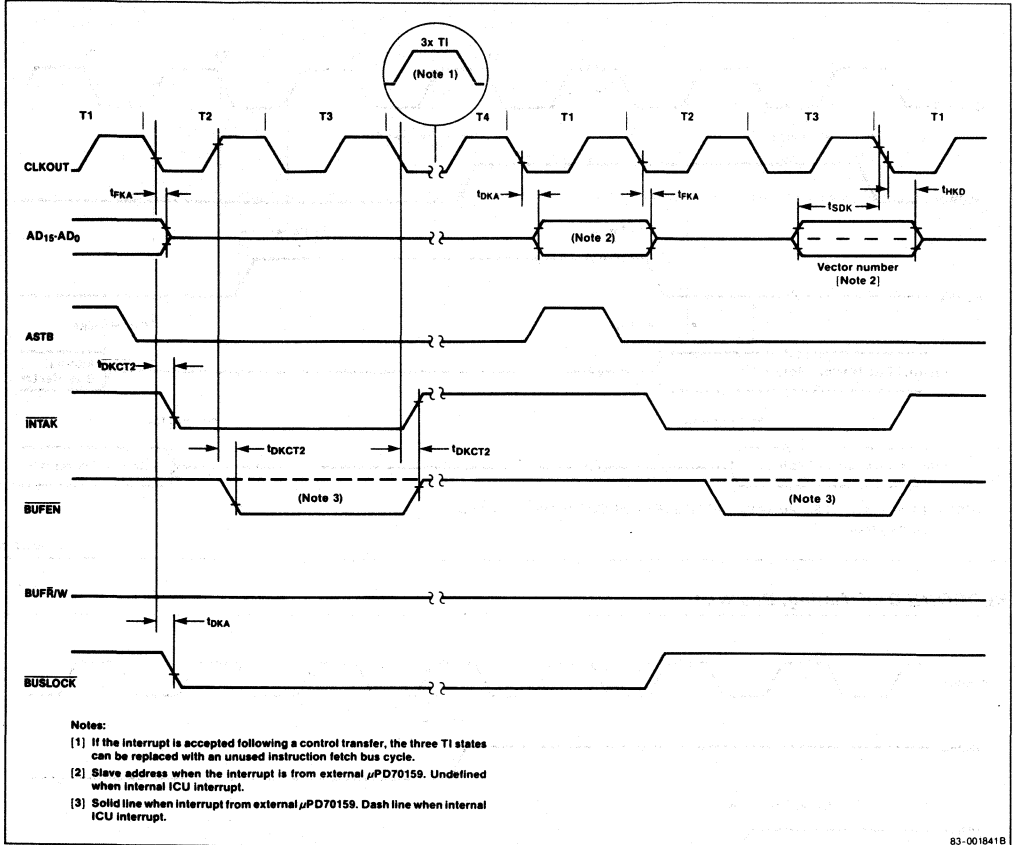
Timing Waveforms (cont)

Status Timing



### Timing Waveforms (cont)

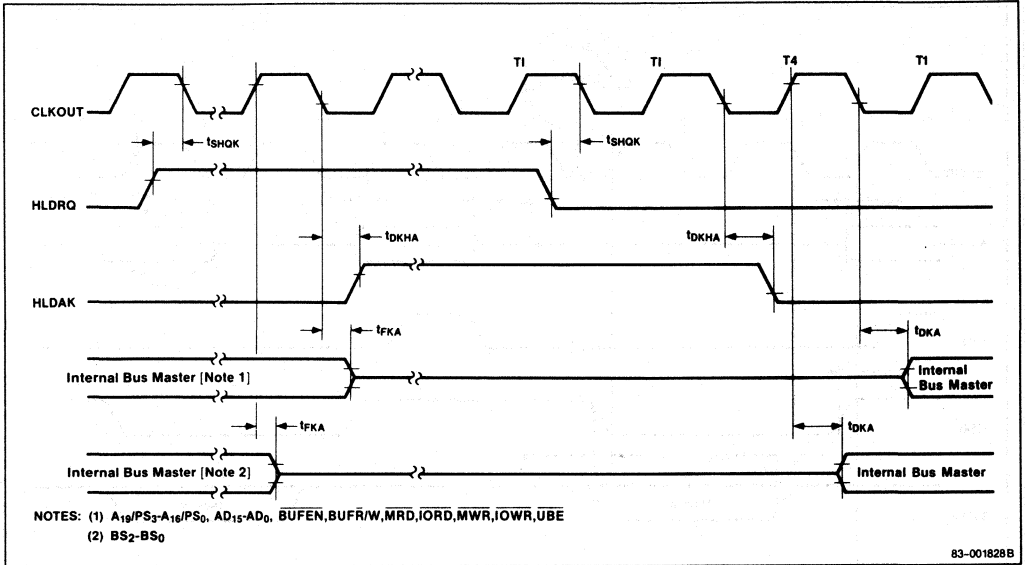
#### Interrupt Acknowledge Timing



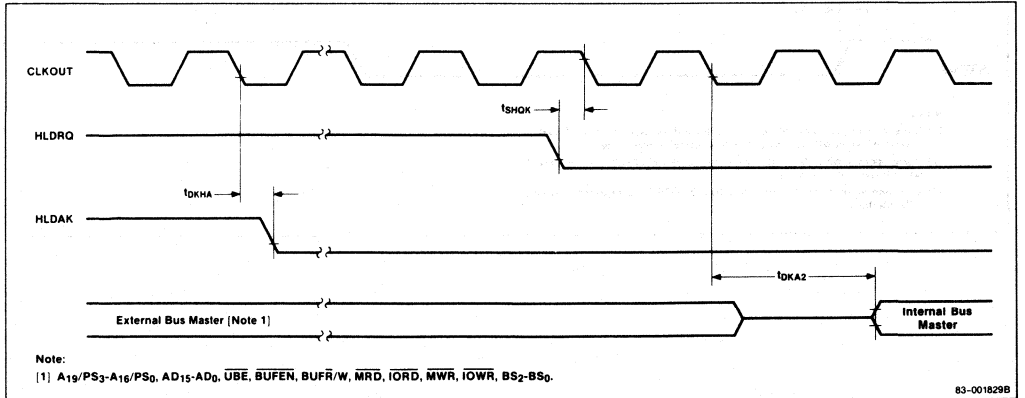
3

Timing Waveforms (cont)

HLD<sub>RQ</sub>/HLD<sub>AK</sub> Timing, Normal Operation

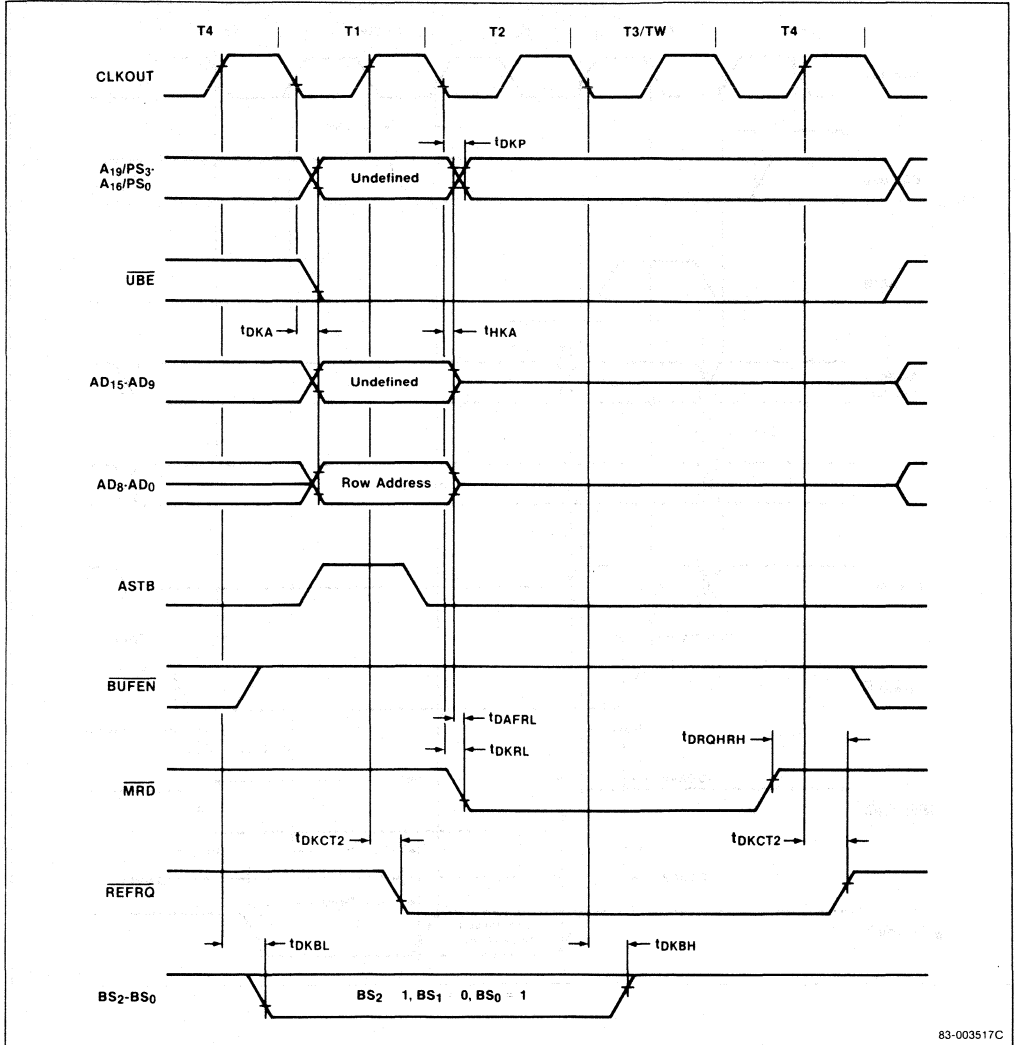


HLD<sub>RQ</sub>/HLD<sub>AK</sub> Timing, Bus Wait



### Timing Waveforms (cont)

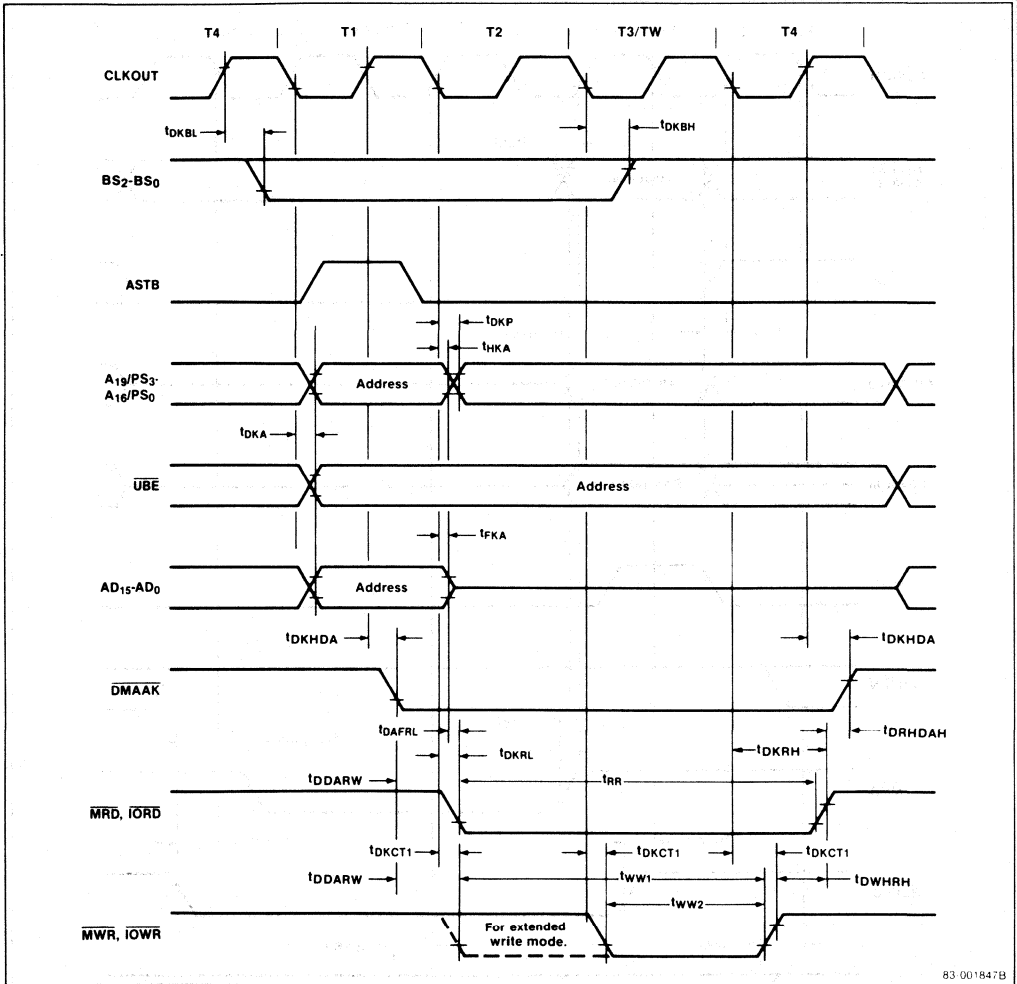
#### Refresh Timing



83-003517C

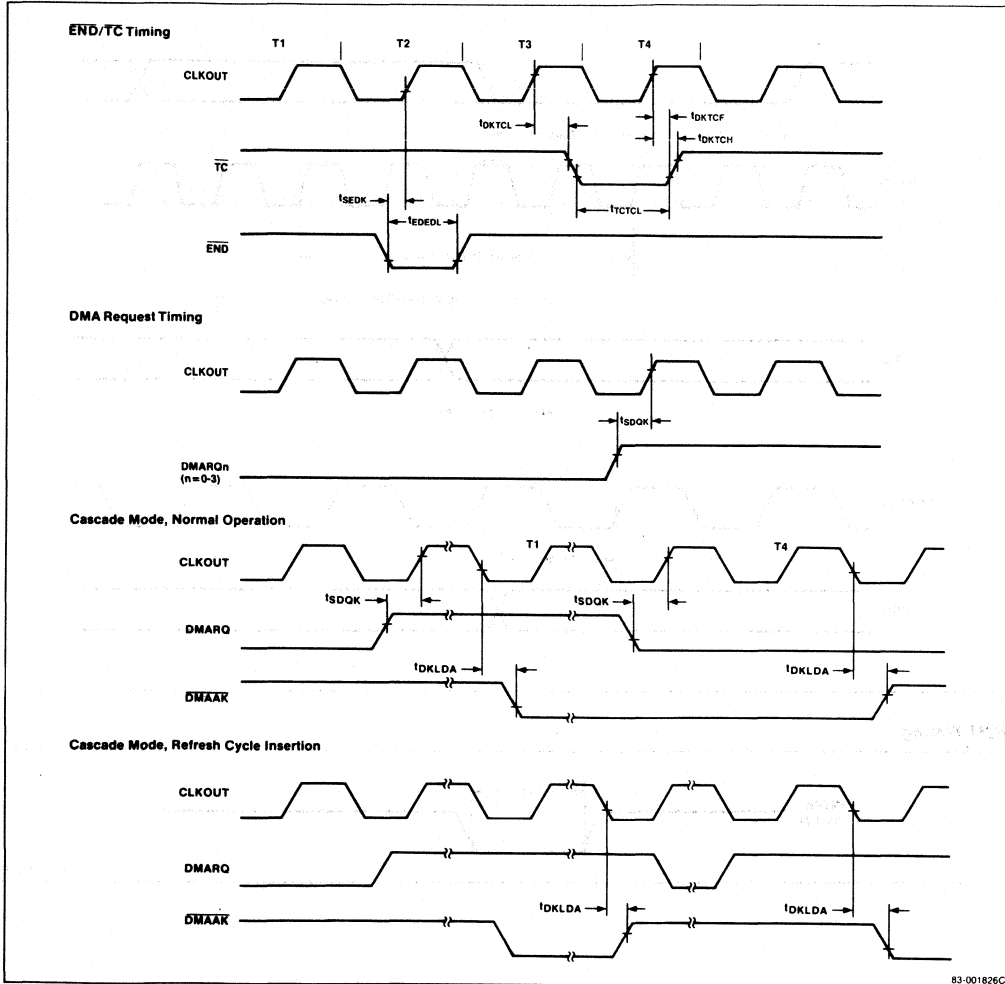
Timing Waveforms (cont)

DMAU, DMA Transfer Timing



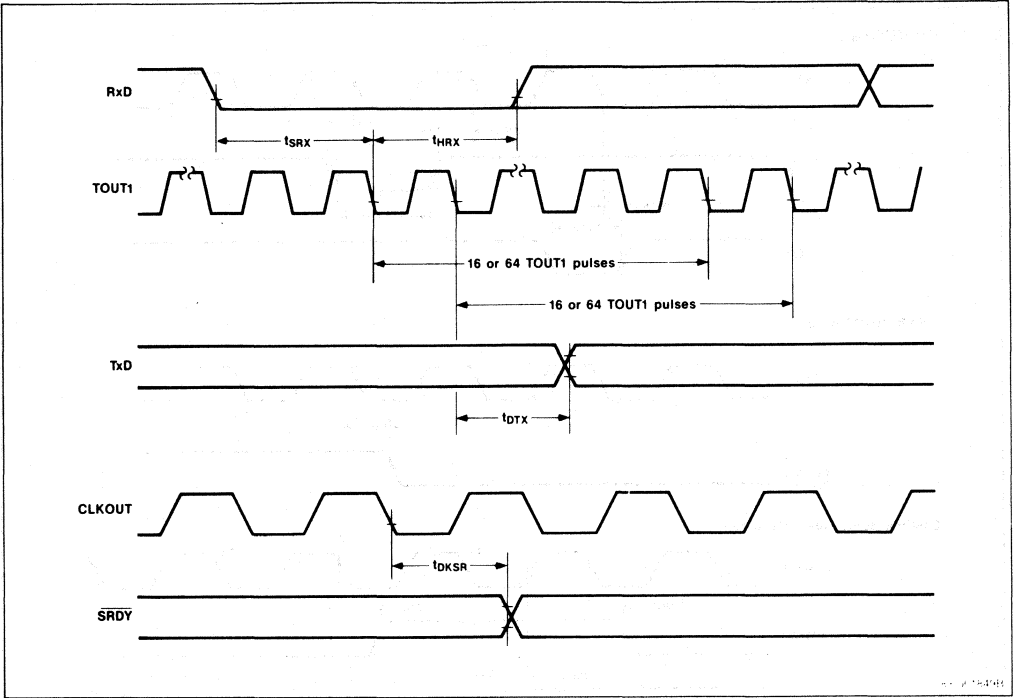
## Timing Waveforms (cont)

### DMA Timing

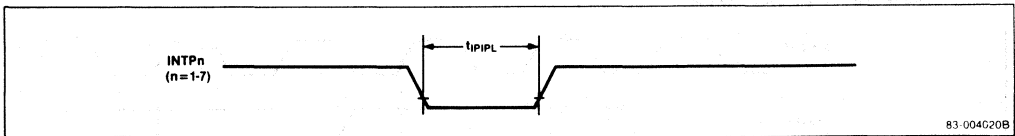


**Timing Waveforms (cont)**

**SCU Timing**

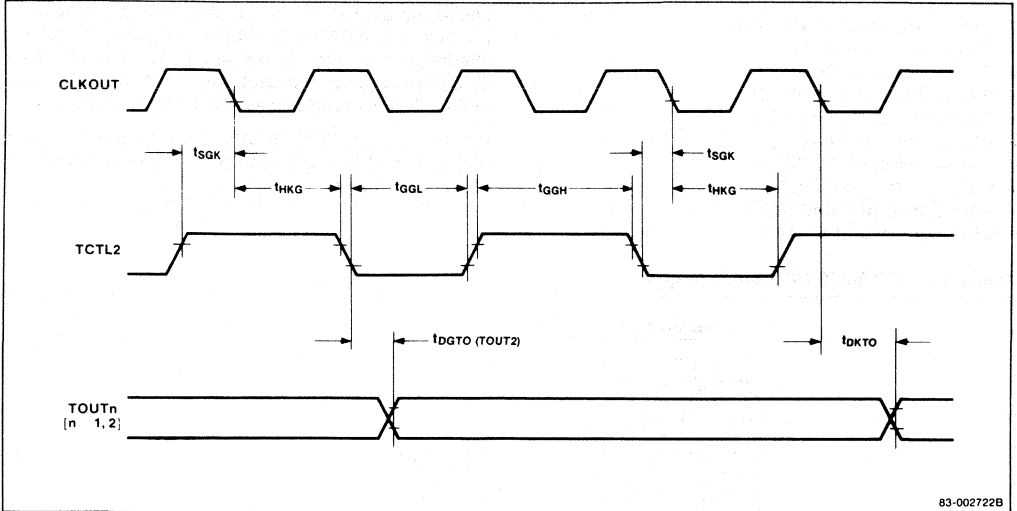


**ICU Timing**

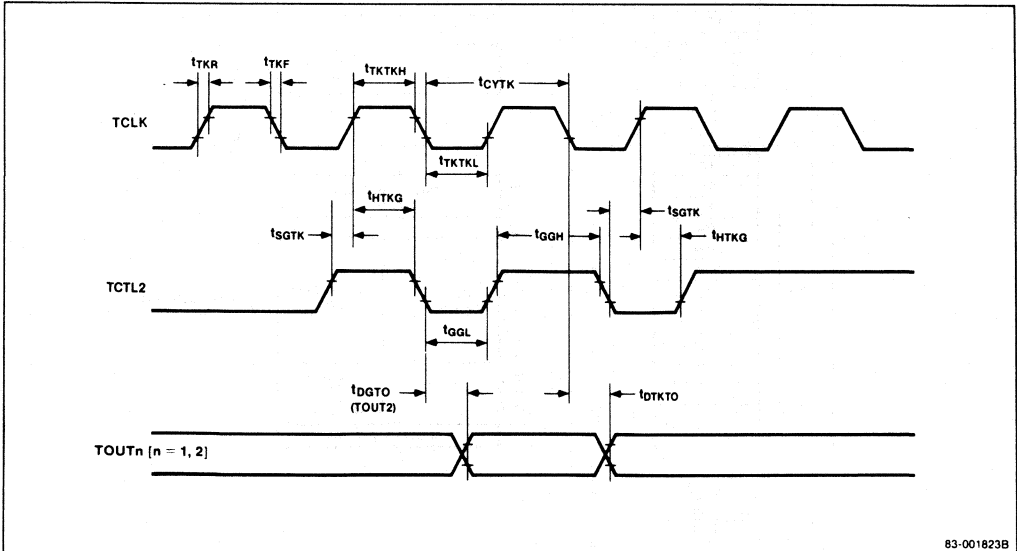


### Timing Waveforms (cont)

#### TCU Timing, Internal Clock Source



#### TCU Timing, TCLK Source





### Functional Description

Refer to the μPD70216 block diagram for an overview of the ten major functional blocks listed below.

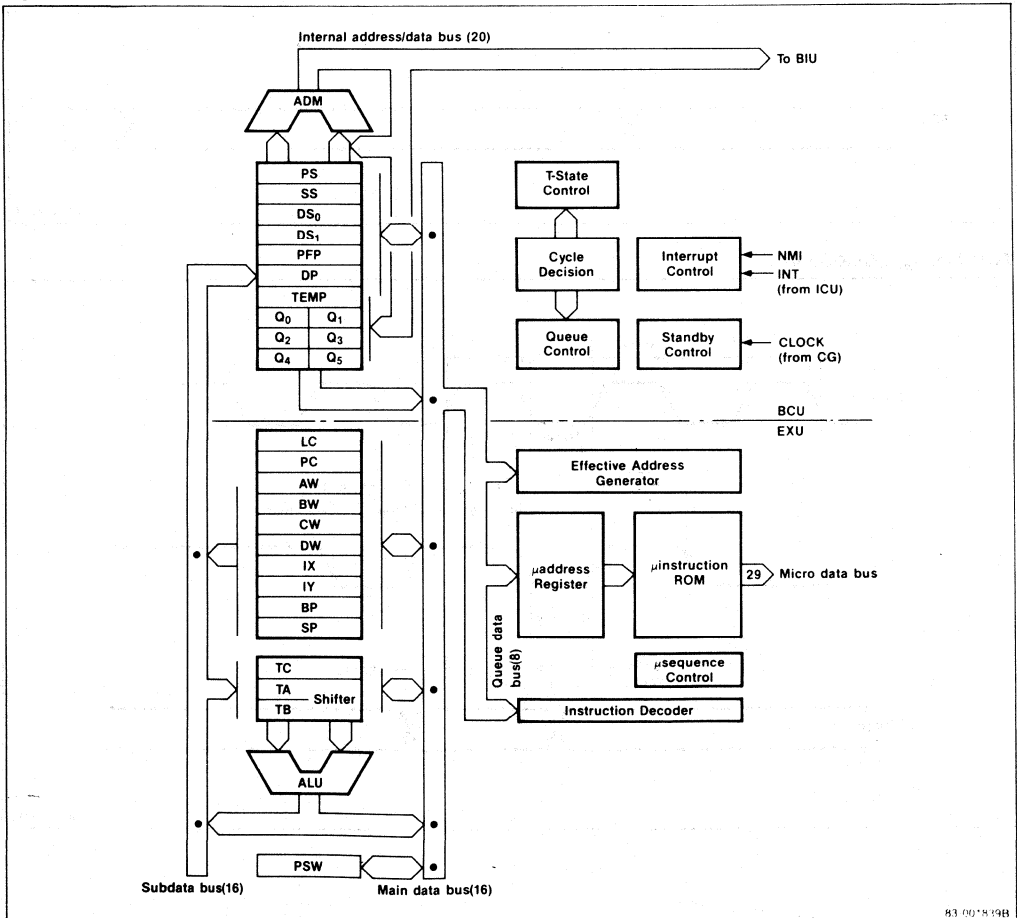
- Central processing unit (CPU)
- Clock generator (CG)
- Bus interface unit (BIU)
- Bus arbitration unit (BAU)
- Refresh control unit (RCU)
- Wait control unit (WCU)
- Timer/counter unit (TCU)
- Serial control unit (SCU)
- Interrupt control unit (ICU)
- DMA control unit (DMAU)

### Central Processing Unit

The μPD70216 CPU functions similarly to the CPU of the μPD70116 CMOS microprocessor. However, because the μPD70216 has internal peripheral devices, its bus architecture has been modified to permit sharing the bus with internal peripherals. The μPD70216 CPU is object code compatible with both the μPD70108/μPD70116 and the μPD8086/μPD8088 microprocessors.

Figure 1 is the μPD70216 CPU block diagram. A listing of the μPD70216 instruction set is in the final sections of this data sheet.

Figure 1. μPD70216 CPU Block Diagram



## Register Configuration

**Program Counter [PC].** The program counter is a 16-bit binary counter that contains the program segment offset of the next instruction to be executed. The PC is incremented each time the microprogram fetches an instruction from the instruction queue. The contents of the PC are replaced whenever a branch, call, return, or break instruction is executed and during interrupt processing. At this time, the contents of the PC are the same as the prefetch pointer (PFP).

**Prefetch Pointer [PFP].** The prefetch pointer is a 16-bit binary counter that contains the program segment offset of the next instruction to be fetched for the instruction queue. Because instruction queue prefetch is independent of instruction execution, the contents of the PFP and PC are not always identical. The PFP is updated each time the bus interface unit (BIU) fetches an instruction for the instruction queue. The contents of the PFP are replaced whenever a branch, call, return or break instruction is executed and during interrupt processing. At this time, the contents of the PFP and PC are the same.

**Segment Registers [PS, SS, DS<sub>0</sub>, DS<sub>1</sub>].** The μPD70216 memory address space is divided into 64K-byte logical segments. A memory address is determined by the sum of a 20-bit base address (obtained from a segment register) and a 16-bit offset known as the effective address (EA). I/O address space is not segmented and no segment register is used. The four segment registers are program segment (PS), stack segment (SS), data segment 0 (DS<sub>0</sub>), and data segment 1 (DS<sub>1</sub>). The following table lists their offsets and overrides.

Default Segment Register	Offset	Override
PS	PFP register	None
SS	SP register	None
SS	Effective address (BP-based)	PS, DS <sub>0</sub> , DS <sub>1</sub>
DS <sub>0</sub>	Effective address (non BP-based)	PS, SS, DS <sub>1</sub>
DS <sub>0</sub>	IX register (1)	PS, SS, DS <sub>1</sub>
DS <sub>1</sub>	IY register (2)	None

**Note:**

- (1) Includes source block transfer, output, BCD string, and bit field extraction.
- (2) Includes destination block transfer, input, BCD string, and bit field insertion.

**General-Purpose Registers.** The μPD70216 CPU contains four 16-bit general-purpose registers (AW, BW, CW, DW), each of which can be used as a pair of 8-bit registers by dividing into upper and lower bytes (AH, AL, BH, BL, CH, CL, DH, DL). General purpose registers may also be specified implicitly in an instruction. The implicit assignments are:

- AW Word multiplication/division, word I/O, data conversion
- AL Byte multiplication/division, byte I/O, BCD rotation, data conversion, translation
- AH Byte multiplication/division
- BW Translation
- CW Loop control, repeat prefix
- CL Shift/rotate bit counts, BCD operations
- DW Word multiplication/division, indirect I/O addressing

**Pointer [SP, BP] and Index Registers [IX, IY].** These registers serve as base pointers or index registers when accessing memory using one of the base, indexed, or base indexed addressing modes. Pointer and index registers can also be used as operands for word data transfer, arithmetic, and logical instructions. These registers are implicitly selected by certain instructions as follows.

- SP Stack operations, interrupts
- IX Source block transfer, BCD string operations, bit field extraction
- IY Destination block transfer, BCD string operations, bit field insertion

## Program Status Word [PSW]

The program status word consists of six status flags and four control flags.

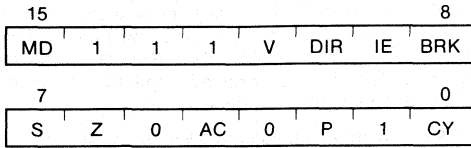
### Status Flags

- V (Overflow)
- S (Sign)
- Z (Zero)
- AC (Auxiliary Carry)
- P (Parity)
- CY (Carry)

### Control Flags

- MD (Mode)
- DIR (Direction)
- IE (Interrupt Enable)
- BRK (Break)

When pushed onto the stack, the word image of the PSW is as follows:



The status flags are set and cleared automatically depending upon the result of the previous instruction execution. Instructions are provided to set, clear, and complement certain status and control flags. Other flags can be manipulated by using the POP PSW instruction.

Between execution of the BRKEM and RETEM instructions, the native mode RETI and POP PSW instructions can modify the MD bit. Care must be exercised by emulation mode programs to prevent inadvertent alteration of this bit.

**CPU Architectural Features**

The major architectural features of the μPD70216 CPU are:

- Dual data buses
- Effective address generator
- Loop counter
- PC and PFP

**Dual Data Buses.** To increase performance, dual data buses (figure 2) have been employed in the CPU to fetch operands in parallel and avoid the bottleneck of a single bus. For two-operand instructions and effective address calculations, the dual data bus approach is 30 percent faster than single-bus systems.

**Effective Address Generator.** Effective address (EA) calculation requires only two clocks regardless of the addressing mode complexity due to the hardware effective address generator (figure 3). When compared with microprogrammed methods, the hardware approach saves between 3 and 10 clock cycles during effective address calculation.

**Loop Counter and Shifters.** A dedicated loop counter is used to count the iterations of block transfer and multiple shift instructions. This logic offers a significant performance advantage over architectures that control block transfers and multiple shifts using microprogramming. Dedicated shift registers also speed up the execution of the multiply and divide instructions. Compared with microprogrammed methods, multiply and divide instructions execute approximately four times faster.

Figure 2. Dual Data Buses

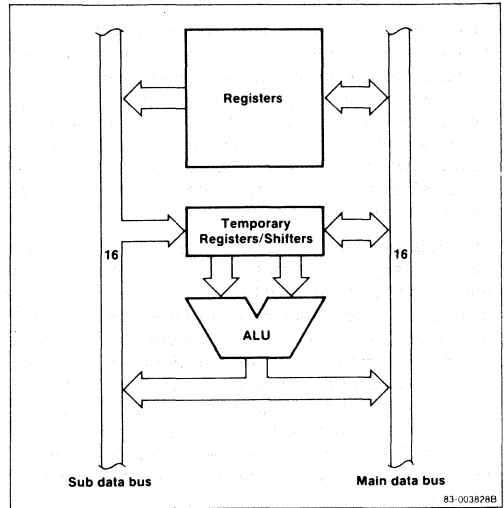
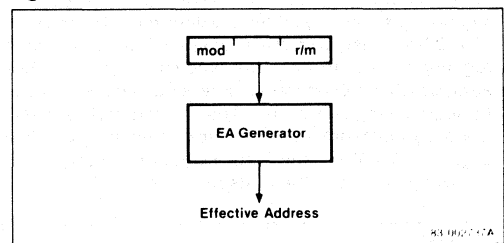


Figure 3. Effective Address Generator



**Program Counter and Prefetch Pointer.** The functions of instruction execution and queue prefetch are decoupled in the μPD70216. By avoiding a single instruction pointer and providing separate PC and PFP registers, the execution time of control transfers and the interrupt response latency can be minimized. Several clocks are saved by avoiding the need to readjust an instruction pointer to account for prefetching before computing the new destination address.

## Enhanced Instruction Set

In addition to the μPD8086/88 instruction set, the μPD70216 has added the following enhanced instructions.

Instruction	Function
PUSH imm	Push immediate data onto stack
PUSH R	Push all general registers onto stack
POP R	Pop all general registers from stack
MUL imm	Multiply register/memory by immediate data
SHL imm8	Shift/rotate by immediate count
SHR imm8	
SHRA imm8	
ROL imm8	
ROR imm8	
ROLC imm8	
RORC imm8	
CHKIND	Check array index
INM	Input multiple
OUTM	Output multiple
PREPARE	Prepare new stack frame
DISPOSE	Dispose current stack frame

## Unique Instruction Set

In addition to the μPD70216 enhanced instruction set, the following unique instructions are supported.

Instruction	Function
INS	Insert bit field
EXT	Extract bit field
ADD4S	BCD string addition
SUB4S	BCD string subtraction
CMP4S	BCD string comparison
ROL4	Rotate BCD digit left
ROR4	Rotate BCD digit right
TEST1	Test bit
SET1	Set bit
CLR1	Clear bit
NOT1	Complement bit
REPC	Repeat while carry set
REPNC	Repeat while carry cleared
FP02	Floating point operation 2

**Bit Fields.** Bit fields are data structures that range in length from 1 to 16 bits. Two separate operations on bit fields, insertion and extraction, with no restrictions on the position of the bit field in memory are supported. Separate segment, byte offset, and bit offset registers are used for bit field insertion and extraction. Because of their power and flexibility, these instructions are highly effective for graphics, high-level languages, and data packing/unpacking applications.

Insert bit field (INS) copies the bit field of specified length (0 = 1 bit, 15 = 16 bits) from the AW register to the bit field addressed by DS1:Y:reg8 (figure 4). The bit field length can be located in any byte register or supplied as an immediate value. The value in reg8 is a bit field offset. A content of 0 selects bit 0 and 15 selects bit 15 of the word that DS0:IX points to. Following execution, the IY and bit offset register are updated to point to the start of the next bit field.

Bit field extraction (EXT) copies the bit field of specified length (0 = 1 bit, 15 = 16 bits) from the bit field addressed by DS0:IX:reg8 to the AW register (figure 5). If the bit field is less than 16 bits, it is right justified with a zero fill. The bit field length can be located in any byte register or supplied as immediate data. The value in reg8 is a bit field offset. A content of 0 selects bit 0 and 15 selects bit 15 of the word that DS0:IX points to. Following execution, the IX and bit offset register are updated to point to the start of the next bit field.

**Packed BCD Strings.** These instructions are provided to efficiently manipulate packed BCD data as strings (length from 1 to 254 digits) or as a byte data type with a single instruction.

BCD string arithmetic is supported by the ADD4S, SUB4S, and CMP4S instructions. These instructions allow the source string (addressed by DS0:IX) and the destination string (addressed by DS1:Y) to be manipulated with a single instruction. When the number of BCD digits is even, the Z and CY flags are set according to the result of the operation. If the number of digits is odd, the Z flag will not be correctly set unless the upper 4 bits of the result are zero. The CY flag will not be correctly set unless there is a carry out of the upper 4 bits of the result.

The two BCD rotate instructions (ROR4, ROL4) perform rotation of a single BCD digit in the lower half of the AL register through the register or memory operand.

**Bit Manipulation.** Four bit manipulation instructions have been added to the μPD70216 instruction set. The ability to test, set, clear, or complement a single bit in a register or memory operand increases code readability as well as performance over the logical operations traditionally used to manipulate bit data.

**Repeat Prefixes.** Two repeat prefixes (REPC, REPNC) allow conditional block transfer instructions to use the state of the CY flag as a terminating condition. The use of these prefixes allows inequalities to be used when working on ordered data, increasing the performance of searching and sorting algorithms.

Figure 4. Bit Field Insertion

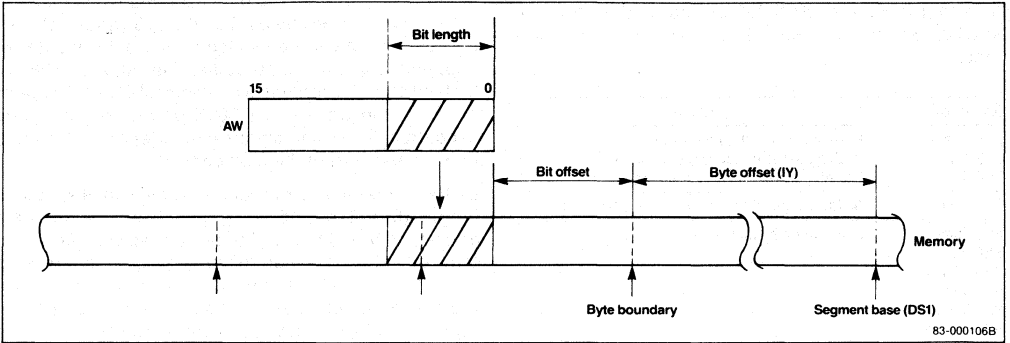
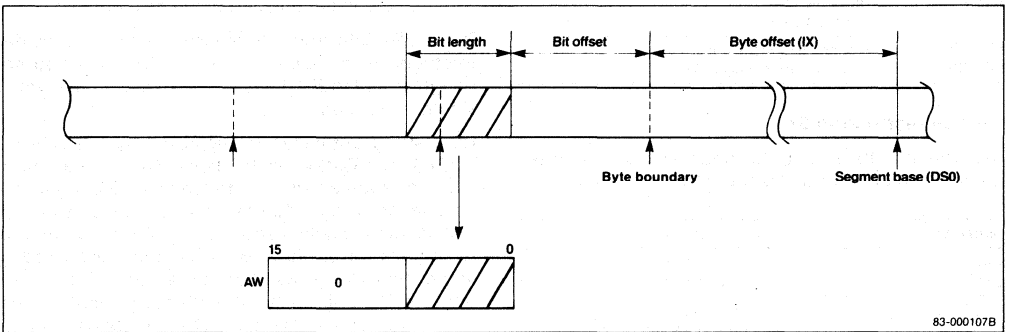


Figure 5. Bit Field Extraction



**Floating Point Operation Instructions.** Two floating point operation (FPO) instruction types are recognized by the μPD70216 CPU. These instructions are detected by the CPU, which performs any auxiliary processing such as effective address calculation and the initial bus cycle if specified by the instruction. It is the responsibility of the external coprocessor to latch the address information and data (if a read cycle) from the bus and complete the execution of the instruction.

**8080 Emulation Mode.** The μPD70216 CPU can operate in either of two modes; see figure 6. Native mode allows the execution of the μPD8086/88, enhanced and unique instructions. The other operating mode is 8080 emulation mode, which allows the entire μPD8080AF instruction set to be executed. A mode (MD) flag is provided to distinguish between the two operating modes. Native mode is active when MD is 1 and 8080 emulation mode is active when MD is 0.

Two instructions are provided to switch from native to 8080 emulation mode and return back. Break for emulation (BRKEM) operates similarly to a BRK

instruction, except that after the PSW has been pushed on the native mode stack, the MD flag becomes write-enabled and is cleared. During 8080 emulation mode, the registers and flags of the 8080 are mapped onto the native mode registers and flags as shown below. Note that PS, SS, DS<sub>0</sub>, DS<sub>1</sub>, IX, IY, AH and the upper half of the PSW registers are inaccessible to 8080 programs.

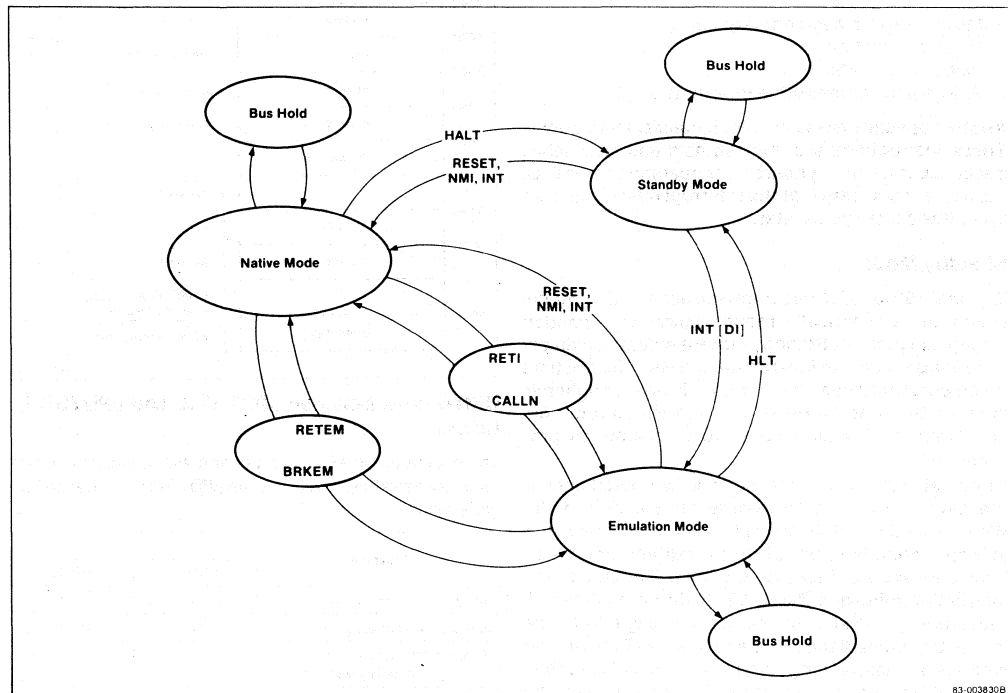
During 8080 emulation mode, the BP register functions as the 8080 stack pointer. The use of separate stack pointers prevents inadvertent damage to the native mode stack pointer by emulation mode programs.

The 8080 emulation mode PC is combined with the PS register to form the 20-bit physical address. All emulation mode data references use DS<sub>0</sub> as the segment register. For compatibility with older 8080 software these registers must be equal. By using different segment register contents, separate 64K-byte code and data spaces are possible.

	μPD8080AF	μPD70216
Registers	A/PSW	AL/PSW (lower)
	B	CH
	C	CL
	D	DH
	E	DL
	H	BH
	L	BL
	SP	BP
	PC	PC
Flags	C	CY
	Z	Z
	S	S
	P	P
	AC	AC

Either an NMI or maskable interrupt will cause the 8080 emulation mode to be suspended. The CPU pushes the PS, PC, and PSW registers on the native mode stack, sets the MD bit (indicating native mode), and enters the specified interrupt handler. When the return from interrupt (RETI) instruction is executed, the PS, PC, and PSW (containing MD=0) are popped from the native stack and execution in 8080 emulation mode continues. Reset will also force a return to native mode.

Figure 6. μPD70216 Modes.



The 8080 emulation mode programs also have the capability to invoke native mode interrupt handlers by means of the call native (CALLN) instruction. This instruction operates like the BRK instruction except that the saved PSW indicates 8080 emulation mode.

To exit 8080 emulation mode, the return from emulation (RETEM) instruction pops the PS, PC, and PSW from the native mode stack, disables modification of the MD bit, and execution continues with the instruction following the BRKEM instruction. Nesting of 8080 emulation modes is prohibited.

### Interrupt Operation

The μPD70216 supports a number of external interrupts and software exceptions. External interrupts are events asynchronous to program execution. On the other hand, exceptions always occur as a result of program execution.

The two types of external interrupts are:

- Nonmaskable interrupt (NMI)
- Maskable interrupt (INT)

The six software exceptions are:

- Divide error (DIV, DIVU instructions)
- Array bound error (CHKIND instruction)
- Break on overflow (BRKV instruction)
- Break (BRK, BRK3 instructions)
- Single step (BRK bit in PSW set)
- Mode switch (BRKEM, CALLN instructions)

Interrupt vectors are determined automatically for exceptions and the NMI interrupt or supplied by hardware for maskable interrupts. The 256 interrupt vectors are stored in a table (figure 7) located at address 00000H. Vectors 0 to 5 are predetermined and vectors 6 to 31 are reserved. Interrupt vectors 32 to 255 are available for use by application software.

Each vector is made up of two words. The word located at the lower address contains the new PC for the interrupt handler. The word at the next-higher address is the new PS value for the interrupt handler. These must be initialized by software at the start of a program.

Nonmaskable interrupts and maskable interrupts (when enabled) are normally serviced following the execution of the current instruction. However, the following cases are exceptions to this rule and the occurrence of the interrupt will be delayed until after the execution of the next instruction.

- Moves to/from segment registers
- POLL instruction
- Instruction prefixes
- EI instruction (maskable interrupts only)

Another special case is the block transfer instructions. These instructions are interruptable and resumable, but because of the asynchronous operation of the BIU, the actual occurrence of the interrupt may be delayed up to three bus cycles later.

**Standby Mode**

The μPD70216 CPU has a low-power standby mode, which can dramatically reduce power consumption during idle periods. Standby mode is entered by simply executing a native or 8080 emulation HALT instruction; no external hardware is required. All other peripherals such as the timer/counter unit, refresh control unit, and DMA control unit continue to operate as programmed.

During standby mode, the clock is distributed only to the circuits required to release the standby mode. When a RESET, NMI, or INT event is detected, the standby mode is released. Both NMI and unmasked interrupts are processed before control returns to the instruction following the HALT. In the case of the INT input being masked, execution will begin with the instruction immediately following the HALT instruction without an intervening interrupt acknowledge bus cycle. When maskable interrupts are again enabled, the interrupt will be serviced.

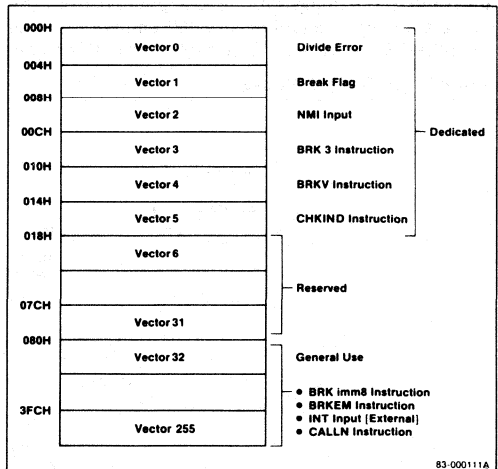
Output signal states in the standby mode are listed below.

Output Signal	Status in Standby Mode
INTAK, BUFEN, MRD, MWR, IOWR, IORD, UBE	High level
BS <sub>2</sub> -BS <sub>0</sub> (Note 2)	Sends halt status (011), then remains high (111)
QS <sub>1</sub> -QS <sub>0</sub> , ASTB	Low level
BUSLOCK	High level (low level if the HALT instruction follows the BUSLOCK prefix)
BUFR/W, A <sub>19</sub> -A <sub>16</sub> /PS <sub>3</sub> -PS <sub>0</sub> , AD <sub>15</sub> -AD <sub>0</sub>	High or low level

Note:

- (1) Output pin states during refresh and DMA bus cycles will be as defined for those operations.
- (2) Halt status is presented prior to entering the passive state.

Figure 7. Interrupt Vector Table



**Differences between μPD70216 and μPD70116 Clocks**

The execution times of instructions with memory access differ between the μPD70216 and μPD70116 as given in the table below.

Instructions	Clocks	
	μPD70116	μPD70216
Instruction with memory read	n	n - 1
Instruction with memory write	n	n - 2
Instruction with memory read and memory write	n	n - 3

## Bus Status Signals

The status signals BS2, BS1 and BS0 of μPD70216 have some different meaning compared to the μPD70116.

Regarding the μPD70116, these signals show the bus cycle state of the CPU, but in μPD70216 these signals indicate the bus cycle state of the following functional units:

- CPU
- DMAU
- REFU

Differences on bus status signals				
BS2	BS1	BS0	Bus cycle	
			μPD70116	μPD70216
1	0	1	Memory read	<ul style="list-style-type: none"> <li>● CPU's memory read</li> <li>● Refresh cycle</li> <li>● Memory to I/O DMA cycle</li> <li>● Verify cycle (DMAU)</li> </ul>
1	1	0	Memory write	<ul style="list-style-type: none"> <li>● CPU's memory write</li> <li>● I/O to Memory DMA cycle</li> </ul>

## Clock Generator

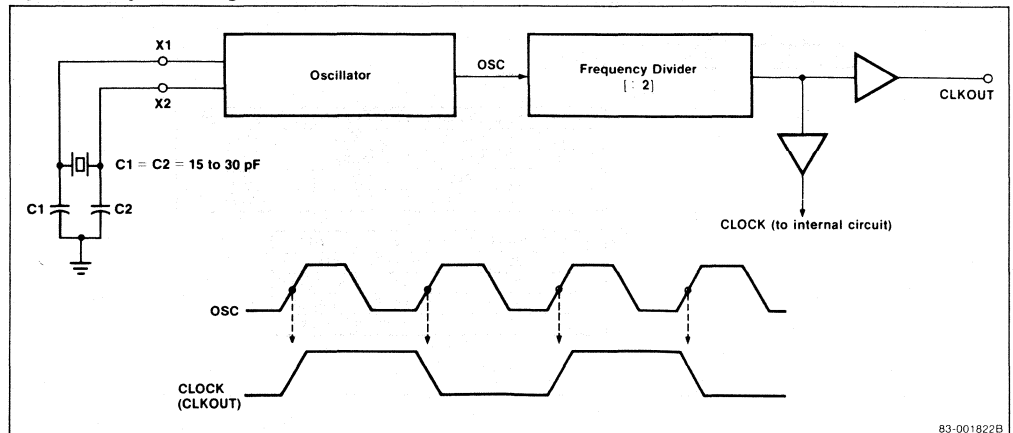
The clock generator (CG) generates a clock signal half the frequency of a parallel-resonant, fundamental mode crystal connected to pins X1 and X2. Figure 8 shows the recommended circuit configuration. Capacitors C1 and C2, required for frequency stability, are selected to match the crystal load capacitance.

External clock sources are also accommodated as shown in figure 9. The CG distributes the clock to the CLKOUT pin and to each functional block of the μPD70216. The generated clock signal has a 50-percent duty cycle.

## Bus Interface Unit

The bus interface unit (BIU) controls the external address, data, and control buses for the three internal bus masters: CPU, DMA control unit (DMAU), and refresh control unit (RCU). The BIU is also responsible

**Figure 8. Crystal Configuration**

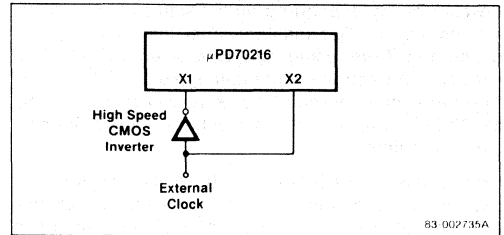


83-001822B

for synchronization of the  $\overline{\text{RESET}}$  and READY inputs with the clock. The synchronized reset signal is used internally by the μPD70216 and provided externally at the RESOUT pin as a system-wide reset. The synchronized READY signal is combined with the output of the wait control unit (WCU) and is distributed internally to the CPU, DMAU, and RCU. Figure 10 shows the synchronization of RESET and READY.

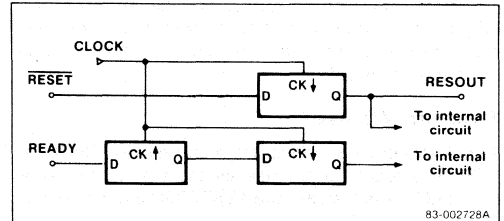
The BIU also has the capability of overlapping the execution of the next instruction with memory write bus cycles. There is no overlap of instruction execution with read or I/O write bus cycles.

**Figure 9. External Oscillator Configuration**



83-002735A

**Figure 10. RESET/READY Synchronization**



83-002728A



### Bus Arbitration Unit

The bus arbitration unit (BAU) arbitrates the external address, data, and control buses between the internal CPU, DMAU, and RCU bus requesters and an external bus master. The BAU bus priorities from the highest priority requester to the lowest are:

- RCU (Demand mode)
- DMAU
- HLDRQ
- CPU
- RCU (Normal mode)

Note that RCU requests the bus at either the highest or lowest priority depending on the status of the refresh request queue. Bus masters other than the CPU are prohibited from using the bus when the CPU is executing an instruction containing a BUSLOCK prefix. Therefore, caution should be exercised when using the BUSLOCK prefix with instructions having a long execution time.

If a bus master with higher priority than the current bus master requests the bus, the BAU inactivates the current bus master's acknowledge signal. When the BAU sees the bus request from the current master go inactive, the BAU gives control of the bus to the higher priority bus master. Whenever possible, the BAU performs bus switching between internal bus masters without the introduction of idle bus cycles, enhancing system throughput.

### System I/O Area

The I/O address space from addresses FF00H to FFFFH is reserved for use as the system I/O area. Located in this area are the 12 μPD70216 registers that

determine the I/O addressing, enable/disable peripherals, and control pin multiplexing. Byte I/O instructions must be used to access the system I/O area.

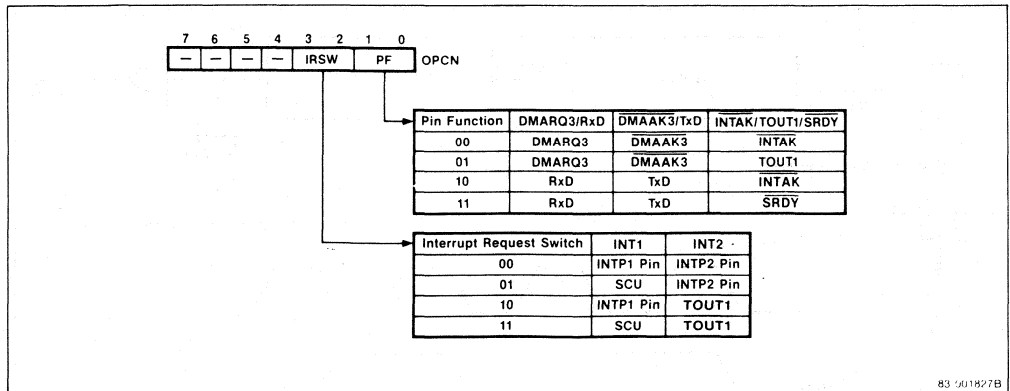
I/O Address	Register	Operation
FFFFH	Reserved	—
FFFEH	OPCN	Read/Write
FFFDH	OPSEL	Read/Write
FFFC	OPHA	Read/Write
FFFBH	DULA	Read/Write
FFFAH	IULA	Read/Write
FFF9H	TULA	Read/Write
FFF8H	SULA	Read/Write
FFF7H	Reserved	—
FFF6H	WCY2	Read/Write
FFF5H	WCY1	Read/Write
FFF4H	WMB	Read/Write
FFF3H	Reserved	—
FFF2H	RFC	Read/Write
FFF1H	Reserved	—
FFF0H	TCKS	Read/Write

### On-Chip Peripheral Connection Register

The on-chip peripheral connection (OPCN) register controls multiplexing of the μPD70216 multiplexed pins. Figure 11 shows the format of the OPCN register. The interrupt request switch (IRSW) field controls multiplexing of ICU interrupt inputs INT1 and INT2. The output of an internal peripheral or an external interrupt source can be selected as the INT1 and INT2 inputs to the ICU.

The pin function (PF) field in the OPCN selects one of four possible states for the DMARQ3/RxD, DMAAK3/TxD, and INTAK/TOUT1/SRDY pins. Bit 0 of the

Figure 11. OPCN Register Format

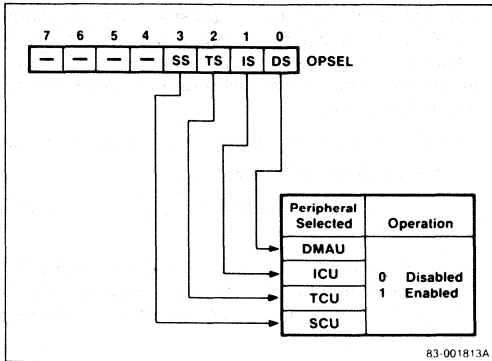


OPCN controls the function of the  $\overline{\text{INTAK}}/\overline{\text{SRDY}}/\text{TOUT1}$  pin. If cleared,  $\overline{\text{INTAK}}$  will appear on this output pin. If bit 0 is set, either TOUT1 or  $\overline{\text{SRDY}}$  will appear at the output depending on the state of bit 1. If bit 1 is cleared, DMA channel 3 I/O signals will appear on the DMARQ3/RxD and DMAAK3/TxD pins. If the SCU is to be used, bit 1 of the PF field must be set.

### On-Chip Peripheral Selection Register

The on-chip peripheral selection (OPSEL) register is used to enable or disable the μPD70216 internal peripherals. Figure 12 shows the format of the OPSEL register. Any of the four (DMAU, TCU, ICU, SCU) peripherals can be independently enabled or disabled by setting or clearing the appropriate OPSEL bit.

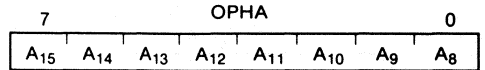
Figure 12. OPSEL Register Format



### Internal Peripheral Relocation Registers

The five internal peripheral relocation registers (figure 13) are used to fix the I/O addresses of the DMAU, ICU, TCU, and SCU. The on-chip peripheral high-address (OPHA) register is common to all four internal peripherals and fixes the high-order byte of the 16-bit I/O address. The individual DMAU low-address (DULA) register, ICU low-address (IULA) register, TCU low-address (TULA) register, and the SCU low-address (SULA) register select the low-order byte of the I/O addresses for the DMAU, ICU, TCU, and SCU peripherals.

The contents of the OPHA register are:



The formats for the individual internal peripheral registers appear below. Since address checking is not performed, do not overlap two peripheral I/O address spaces.

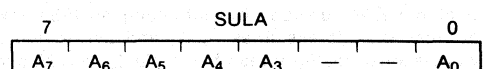
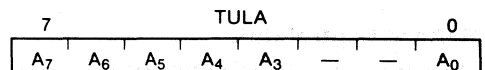
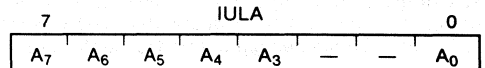
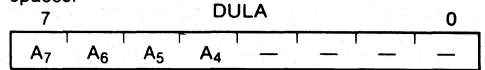


Figure 13. μPD70216 Peripheral Relocation

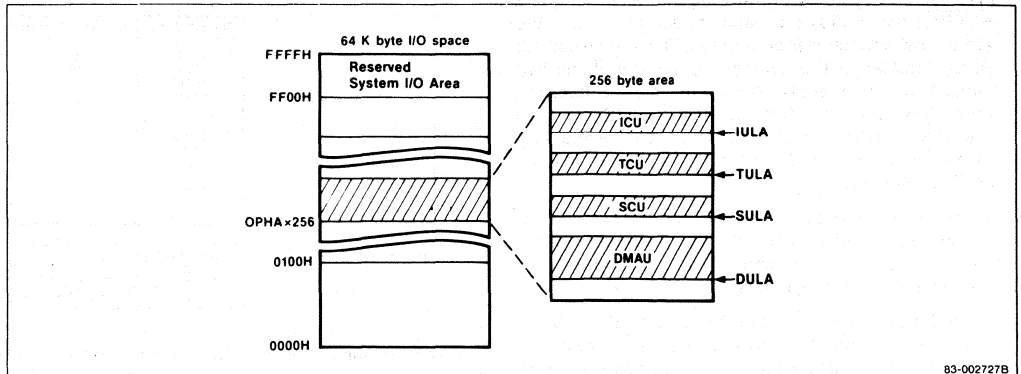
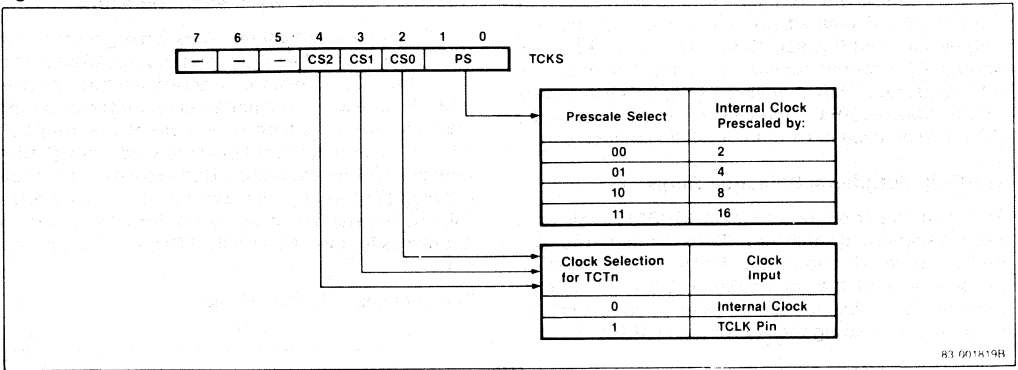


Figure 14. Timer Clock Selection Register



**Timer Clock Selection Register**

The timer clock selection (TCKS) register selects the clock source for each of the timer/counters as well as the divisor for the internal clock prescaler. Figure 14 shows the format of the TCKS register. The clock source for each timer/counter is independently selected from either the prescaled internal CPU clock or from an external clock source (TCLK). The internal clock is derived from the CLKOUT signal and can be divided by 2, 4, 8, or 16 before being presented to the clock select logic.

**Refresh Control Unit**

The refresh control unit (RCU) refreshes external dynamic RAM devices by outputting an 8-bit row address on address lines A<sub>8</sub>-A<sub>1</sub> and performing a word-aligned memory read bus cycle. Both  $\overline{UBE}$  and A<sub>0</sub> are asserted to allow the refresh of both the even and odd memory banks. External logic can distinguish a refresh bus cycle by monitoring the refresh request (REFRQ) pin. Following each refresh bus cycle, the refresh row counter is incremented. The refresh control (RFC) register in the system I/O area contains two fields. The refresh enable field enables or disables the refreshing function. The refresh timer (RTM) field selects a refresh interval to match the dynamic memory refresh requirements. Figure 15 shows the format for the RFC register.

To minimize the impact of refresh on the system bus bandwidth, the μPD70216 utilizes a refresh request queue to store refresh requests and perform refresh bus cycles in otherwise idle bus cycles.

The RCU normally requests the bus as the lowest-priority bus requester (normal mode). However, if seven refresh requests are allowed to accumulate in the RCU refresh request queue, the RCU will change to

the highest-priority bus requester (demand mode). The RCU will then perform back-to-back refresh cycles until three requests remain in the queue. This guarantees the integrity of the DRAM system while maximizing performance.

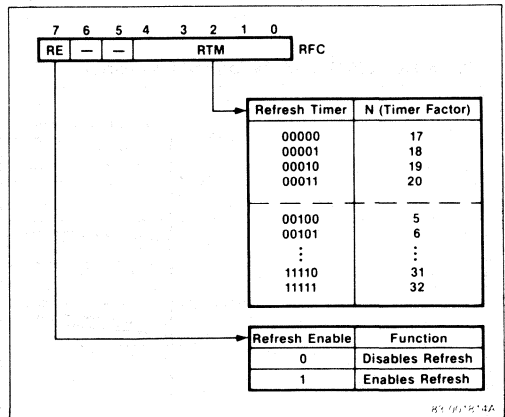
The refresh count interval can be calculated as follows:

$$\text{Refresh interval} = 8 \times N \times t_{CYK}$$

where N is the timer factor selected by the RTM field.

When the μPD70216 is reset, the RE field in the RFC register is unaffacted and the RTM field is set to 01000 (N = 9). No refresh bus cycles occur while  $\overline{RESET}$  is asserted.

Figure 15. Refresh Control Register



The address lines A<sub>9</sub>-A<sub>19</sub> are at level zero if refresh cycle follows a memory or I/O read/write cycle. In refresh cycles after a DMA transfer cycle, the A<sub>9</sub>-A<sub>19</sub> address lines maintain the address of the las DMA transfer.

### Wait Control Unit

The wait control unit (WCU) inserts from zero to three wait states into a bus cycle in order to compensate for the varying access times of memory and I/O devices. The number of wait states for CPU, DMAU, and RCU bus cycles is separately programmable. In addition, the memory address space is divided into three independent partitions to accommodate a wide range of system designs. RESET initializes the WCU to insert three wait states in all bus cycles. This allows operation with slow memory and peripheral devices before the initialization of the WCU registers.

The three system I/O area registers that control the WCU are wait cycle 1 (WCY1), wait cycle 2 (WCY2), and wait state memory boundary (WMB). The WCU always inserts wait states corresponding to the wait count programmed in WCY1 or WCY2 registers into a bus cycle, regardless of the state of the external READY input. After the programmed number of wait states occurs, the WCU will insert Tw states as long as the READY pin remains inactive. When READY is again asserted, the bus cycle continues with T4 as the next cycle. The μPD70216 internal peripherals never require wait states; four clock cycles will terminate an internal peripheral bus cycle.

### CPU Wait States

The WMB register divides the 1M-byte memory address space into three independent partitions: lower, middle, and upper. Figure 16 shows the WMB register format.

Initialization software can then set the number of wait states for each memory partition and the I/O partition via the WCY1 register (figure 17).

### DMA and Refresh Wait States

The WCY2 register (figure 18) specifies the number of wait states to be automatically inserted in DMA and refresh bus cycles.

### DMA and Refresh Wait States

The WCY2 register (figure 18) specifies the number of wait states to be automatically inserted in DMA and refresh bus cycles. DMA wait states must be set to the maximum of the DMA memory and I/O partitions. Refresh wait states should be set to the maximum value of all DRAM memory partitions.

3

Figure 16. Wait State Memory Boundary Register

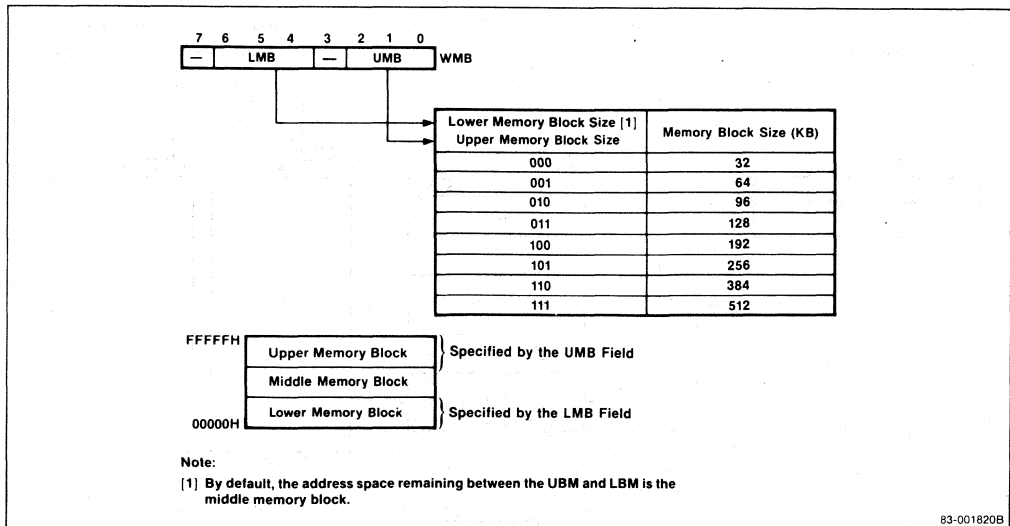


Figure 17. Wait Cycle 1 Register

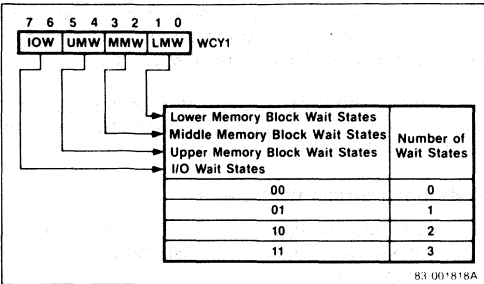
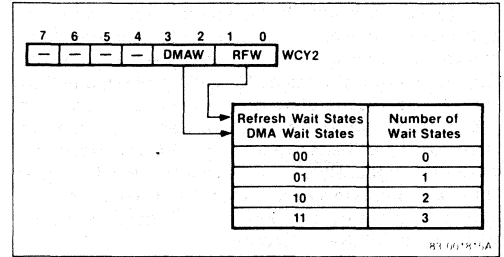


Figure 18. Wait Cycle 2 Register



### Timer/Counter Unit

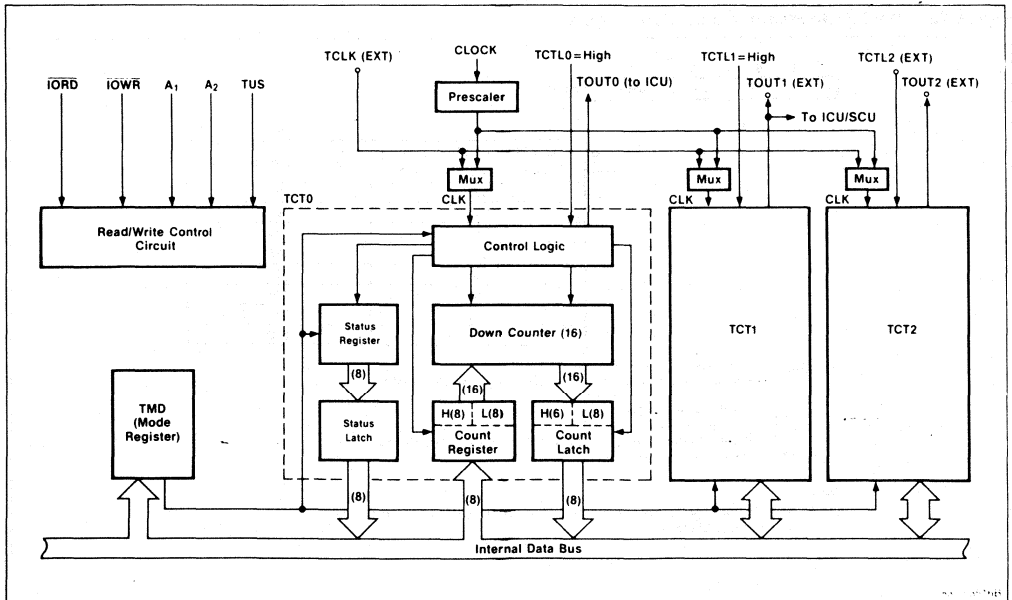
The timer/counter unit (TCU) provides a set of three independent 16-bit timer/counters. The output signal of timer/counter 0 is hardwired internally as an interrupt source. The output of timer/counter 1 is available internally as an interrupt source, used as a baud rate generator, or used as an external output. Due to mode restrictions, the TCU is a subset of the

μPD71054 Programmable Timer/Counter. Figure 19 shows the internal block diagram of the TCU.

The TCU has the following features:

- Three 16-bit timer/counters
- Six programmable count modes
- Binary/BCD counting
- Multiple latch command
- Choice of two clock sources

Figure 19. TCU Block Diagram



Because **RESET** leaves the TCU in an uninitialized state, each timer/counter must be initialized by specifying an operating mode and a count. Once programmed, a timer/counter will continue to operate in that mode until another mode is selected. When the count has been written to the counter and transferred to the down counter, a new count operation starts. Both the current count and the counter status can be read while count operations are in progress.

### TCU Commands

The TCU is programmed by issuing I/O instructions to the I/O port addresses programmed in the OPHA and TULA registers. The individual TCU registers are selected by address bits  $A_2$  and  $A_1$  as follows.

$A_2$	$A_1$	Register	Operation
0	0	TCT0 TST0	Read/Write Read
0	1	TCT1 TST1	Read/Write Read
1	0	TCT2 TST2	Read/Write Read
1	1	TMD	Write

The timer mode (TMD) register selects the operating mode for each timer/counter and issues the latch command for one or more timer/counters. Figure 20 shows the format for the TMD register.

Writes to the timer/counter 2-0 (TCT2-TCT0) registers stores the new count in the appropriate timer/counter. The count latch command is used before reading count data in order to latch the current count and prevent inaccuracies.

The timer status 2-0 (TST2-TST0) registers contain status information for the specified counter (figure 21). The latch command is used to latch the appropriate counter status before reading status information. If both status and counter data are latched for a counter, the first read operation returns the status data and subsequent read operations obtain the count data.

### Count Modes

There are six programmable timer/counter modes. The timing waveforms for these modes are in figure 22.

**Mode 0 [Interrupt on End of Count].** In this mode, TOUT changes from the low to high level when the specified count is reached. This mode is available on all timer/counters. Minimal number of counts is one.

**Mode 1 [Retriggerable One-shot].** In mode 1, a low-level one-shot pulse, triggered by TCTL2 is output from the TOUT2 pin. This mode is available only on timer/counter 2 with one as minimal number of counts.

**Mode 2 [Rate Generator].** In mode 2, TOUT cyclically goes low for one clock period when the counter reaches the 0001H count. A counter in this mode operates as a frequency divider. All timer/counters can operate using mode 2. Minimal number of counts is 2.

**Mode 3 [Square Wave Generator].** Mode 3 is a frequency divider similar to mode 2, but the output has a symmetrical duty cycle. This mode is available on all three timer/counters. Minimal count value is 2.

**Mode 4 [Software Triggerged Strobe].** In mode 4, when the specified count is reached, TOUT goes low for the duration of one clock pulse. Mode 4 is available on all timer/counters. Minimal number of counts is 1.

**Mode 5 [Hardware Triggerged Strobe].** Mode 5 is similar to mode 4 except that operation is triggered by the TCTL2 input and can be retriggered. This mode is available only on timer/counter 2.

### Serial Control Unit

The serial control unit (SCU) is a single asynchronous serial channel that performs serial communication between the μPD70216 and an external serial device. The SCU is similar to the μPD71051 Serial Control Unit except for the lack of synchronous communication protocols. Figure 23 is the block diagram of the SCU.

The SCU has the following features.

- Full-duplex asynchronous serial controller
- Clock rate divisor (x16, x64)
- Baud rates to 250 kb/s supported (with ext. clock)
- 7-, 8-bit character lengths
- 1-, 2-bit stop bit lengths
- Break transmission and detection
- Full-duplex, double-buffered transmitter/receiver
- Even, odd, or no parity
- Parity, overrun, and framing error detection
- Receiver-full/transmitter-empty interrupt

The SCU contains four separately addressable registers for reading/writing data, reading status, and controlling operation of the SCU. The serial receive buffer (SRB) and the serial transmit buffer (STB) store the incoming and outgoing character data. The serial status (SST) register allows software to determine the current state of both the transmitter and receiver. The serial command (SCM) and serial mode (SMD) registers determine the operating mode of the SCU while the serial interrupt mask (SIMK) register allows software control of the SCU receive and transmit interrupts.

Figure 20. Timer Mode Register

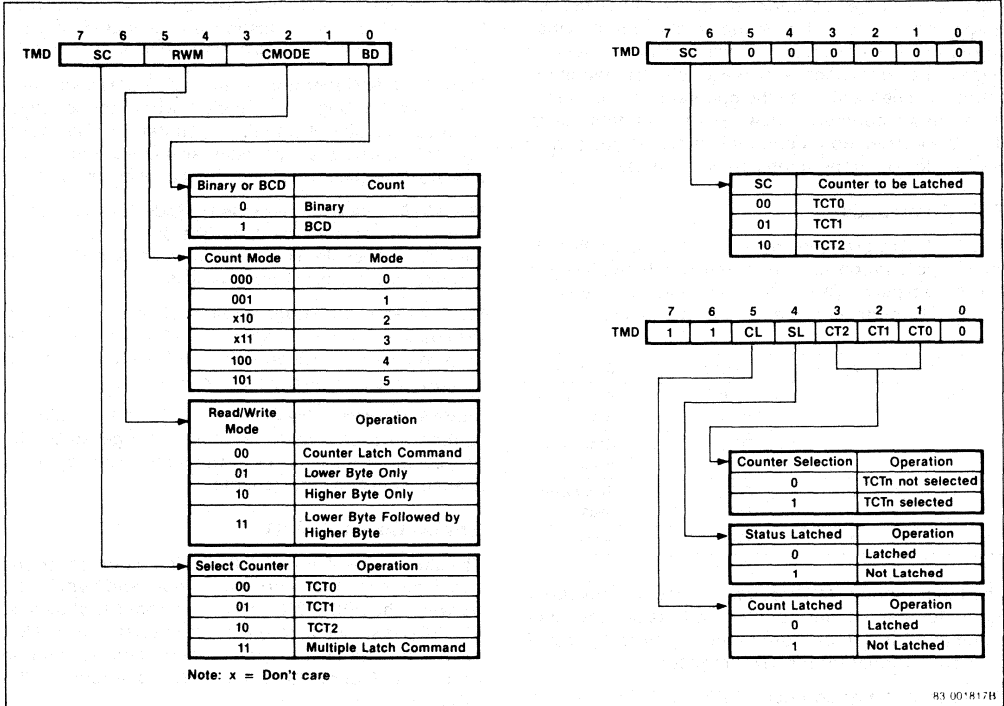


Figure 21. TCU Status Register

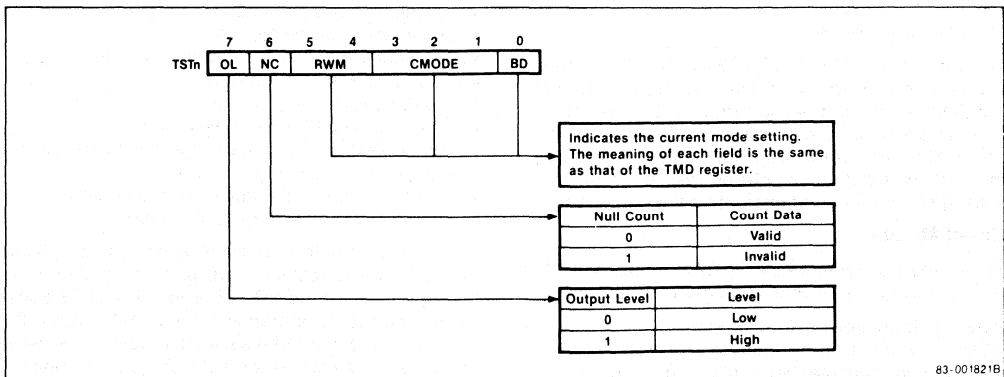


Figure 22. TCU Waveforms (Sheet 1 of 3)

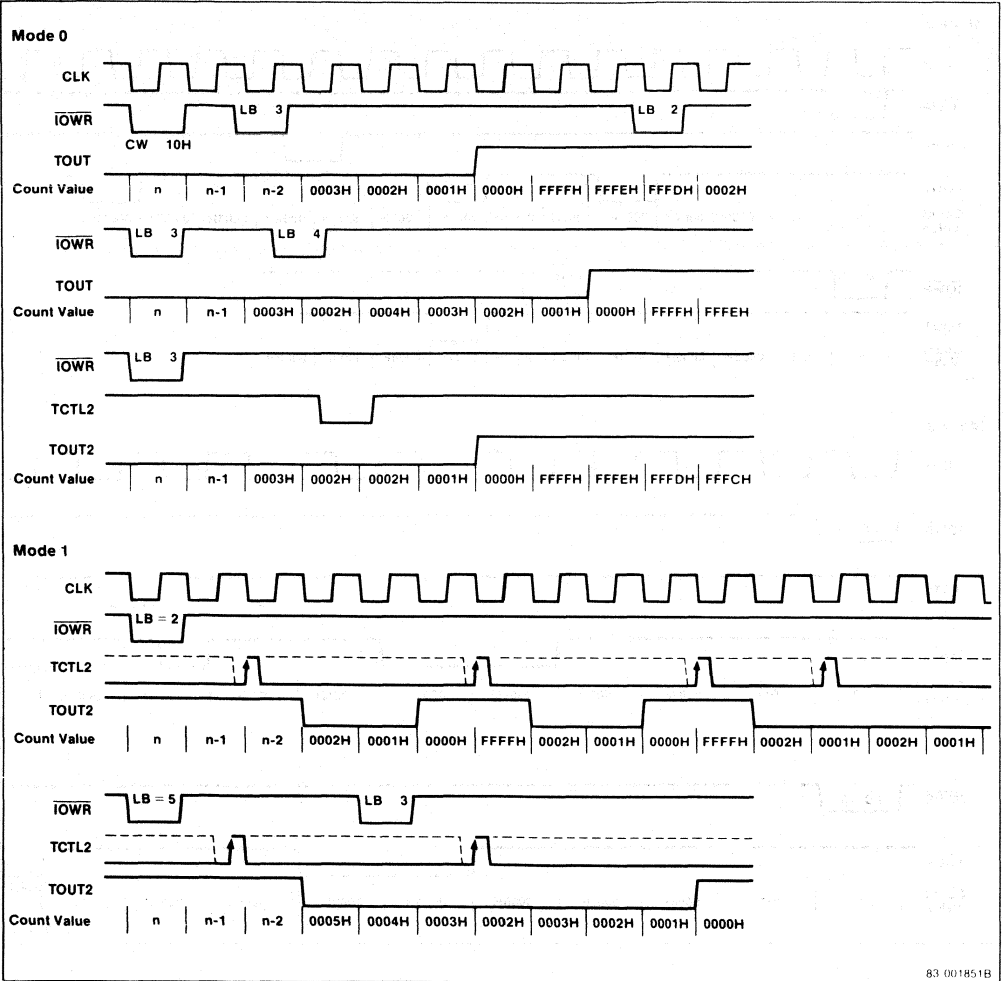




Figure 22. TCU Waveforms (Sheet 2 of 3)

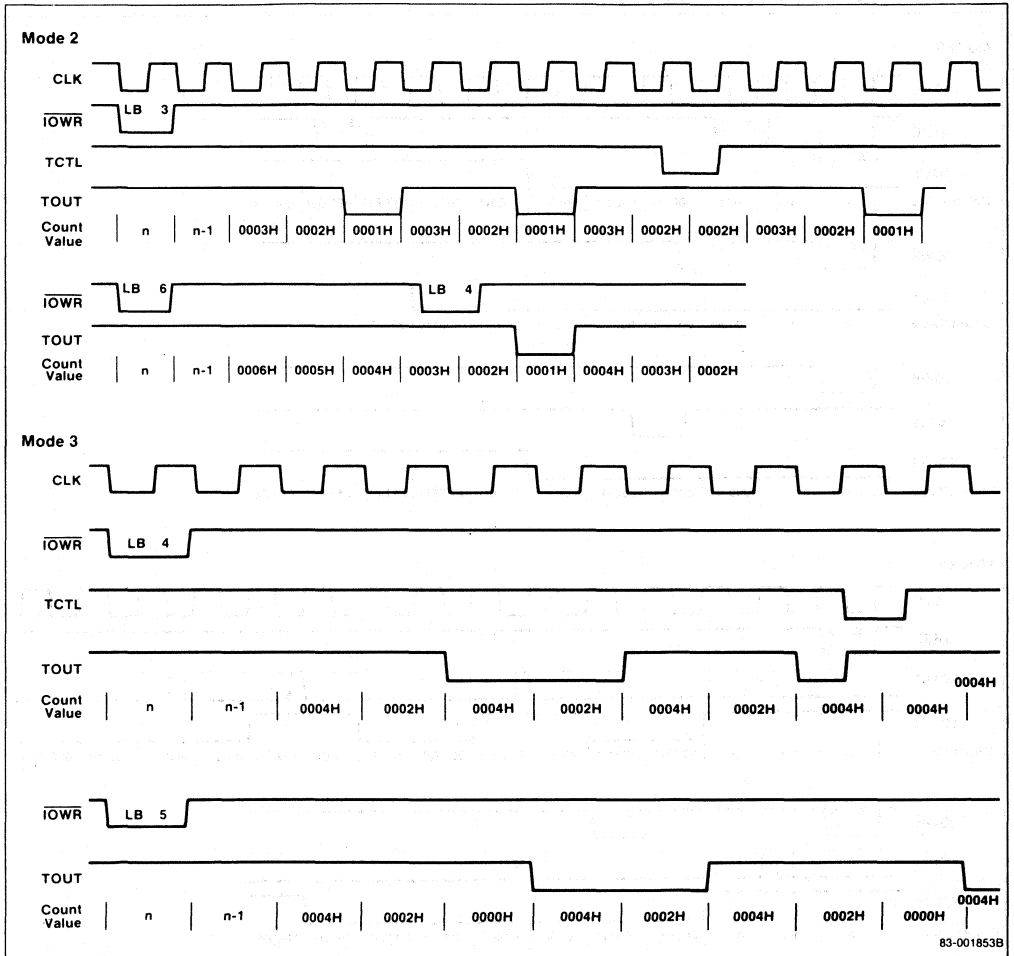


Figure 22. TCU Waveforms (Sheet 3 of 3)

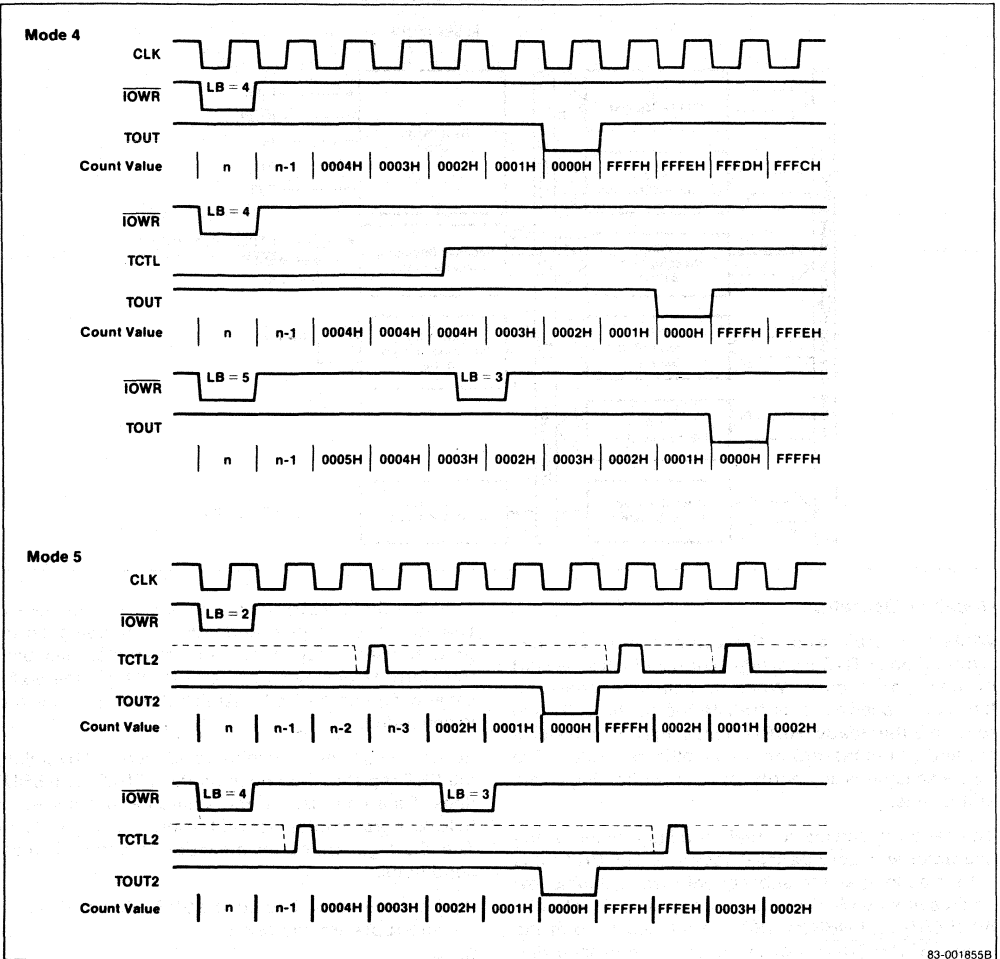
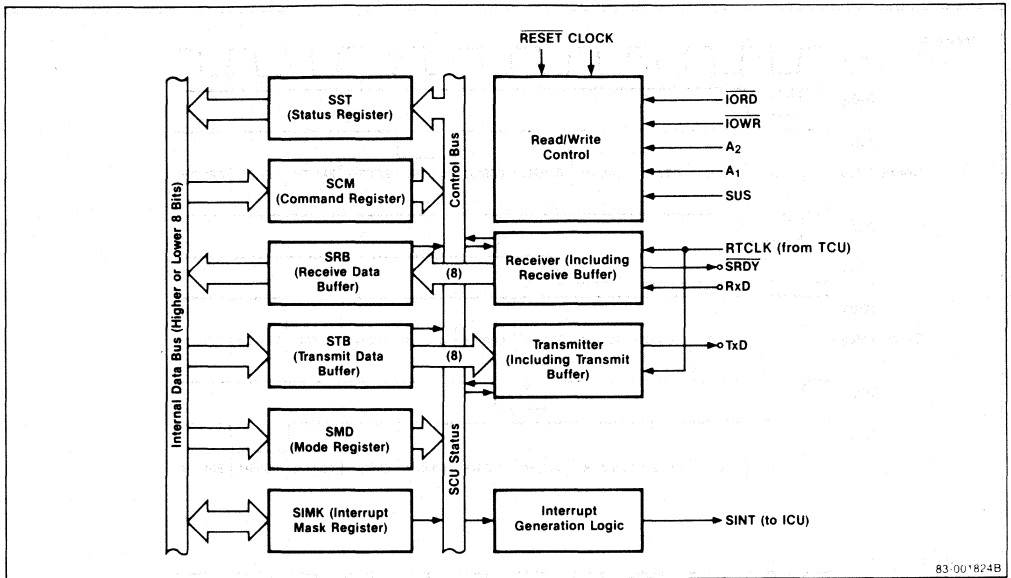


Figure 23. SCU Block Diagram



**Receiver Operation**

While the RxD pin is high, the receiver is in an idle state. A transition on RxD from high to low indicates the start of new serial data. When a complete character has been received, it is transferred to the SRB; the receive buffer ready (RBRDY) bit in the SST register is set and (if unmasked) an interrupt is generated. The SST also latches any parity, overrun, or framing errors at this time.

The receiver detects a break condition when a null character with zero parity is received. The BRK bit is set for as long as the subsequent receive data is low and resets when RxD returns to a high level. The MRDY bit (SCM) and RBRDY (SST) are gated to form the output SRDY. SRDY prevents overruns from occurring when the program is unable to process the input data. Software can control MRDY to prevent data from being sent from the remote transmitter while RBRDY can prevent the immediate overrun of a received character.

**Transmitter Operation**

TxD is kept high while the STB register is empty. When the transmitter is enabled and a character is written to the STB register, the data is converted to serial format and output on the TxD pin. The start bit indicates the start of the transmission and is followed by the character

stream (LSB to MSB) and an optional parity bit. One or two stop bits are then appended, depending on the programmed mode. When the character has been transferred from the STB, the TRBDY bit in the SST is set and if unmasked, a transmit buffer empty interrupt is generated.

Serial data can be transmitted and received by polling the SST register and checking the TBRDY or RBRDY flags. Data can also be transmitted and received by SCU-generated interrupts to the interrupt control unit. The SCU generates an interrupt in either of these conditions:

- (1) The receiver is enabled, the SRB is full, and receive interrupts are unmasked.
- (2) The transmitter is enabled, the STB is empty, and transmit interrupts are unmasked.

### SCU Registers and Commands

I/O instructions to the I/O addresses selected by the OPHA and SULA registers are used to read/write the SCU registers. Address bits A<sub>1</sub> and A<sub>2</sub> and the read/write lines select one of the six internal registers as follows:

A <sub>2</sub>	A <sub>1</sub>	Register	Operation
0	0	SRB	Read
		STB	Write
0	1	SST	Read
		SCM	Write
1	0	SMD	Write
1	1	SIMK	Read/write

The SRB and STB are 8-bit registers. When the character length is 7 bits, the lower 7 bits of the SRB register are valid and bit 7 is cleared to 0. If programmed for 7-bit characters, bit 7 of the STB is ignored.

The SST register (figure 24) contains the status of the transmit and receive data buffers and the error flags. Error flags are persistent. Once an error flag is set, it remains set until a clear error flags command is issued.

Figure 24. SST Register

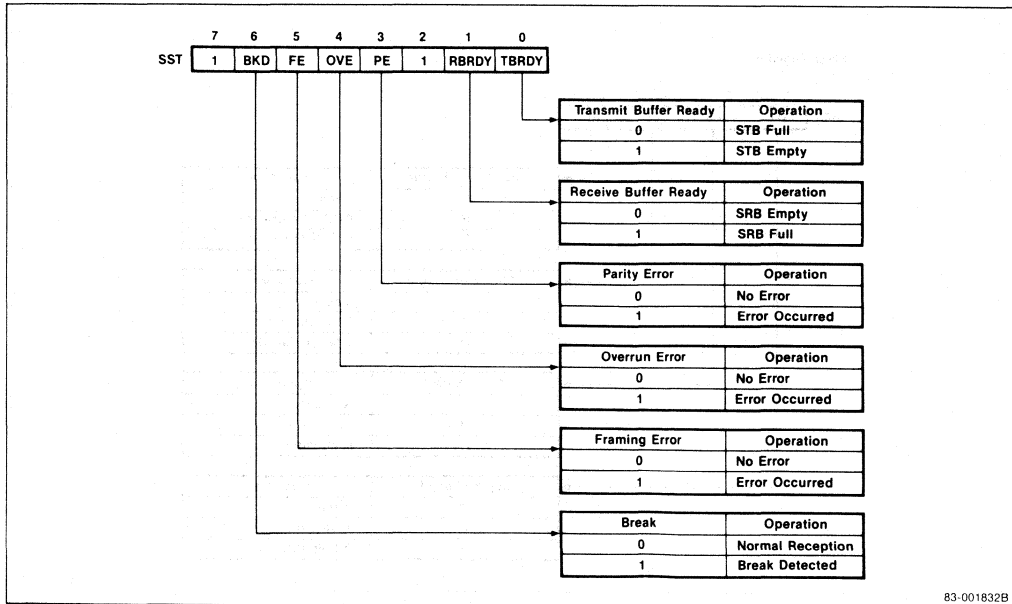


Figure 25 shows the SCM and SMD registers. The SCM register stores the command word that controls transmission, reception, error flag reset, break transmission, and the state of the SRDY pin. The SMD register stores the mode word that determines serial characteristics such as baud rate divisor, parity, character length, and stop bit length.

Initialization software should first program the SMD register followed by the SCM register. Unlike the μPD71051, the SMD register can be modified at any time without resetting the SCU.

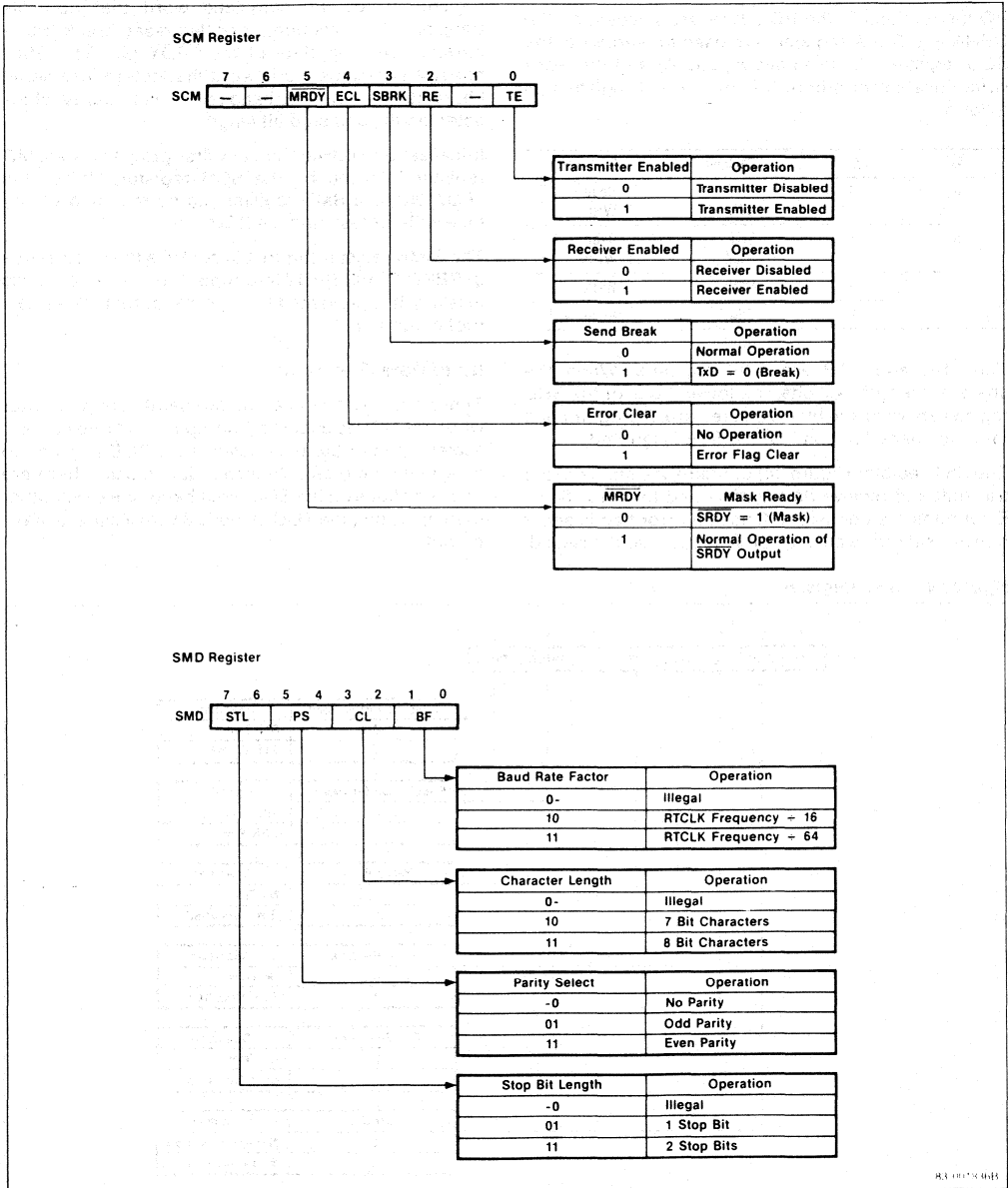
The SIMK register (figure 26) controls the occurrence of RBRDY and TBRDY interrupts. When an interrupt is masked, it is prevented from propagating to the interrupt control unit.

### Baud Rate Generator

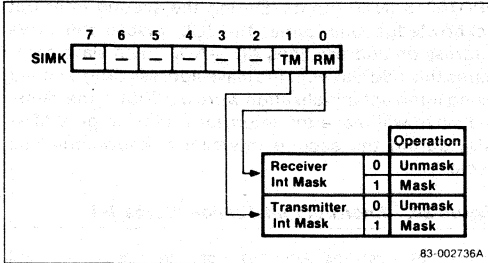
Timer/counter 1 is used as the baud rate generator when the SCU is enabled. The input baud rate clock is scaled by 16 or 64, as selected in the SMD register, to determine the receive/transmit data clock. There are no restrictions on the SCU input baud rate clock other than operating the TCU in mode 3 with a square-wave output.

3

Figure 25. SCM and SMD Registers



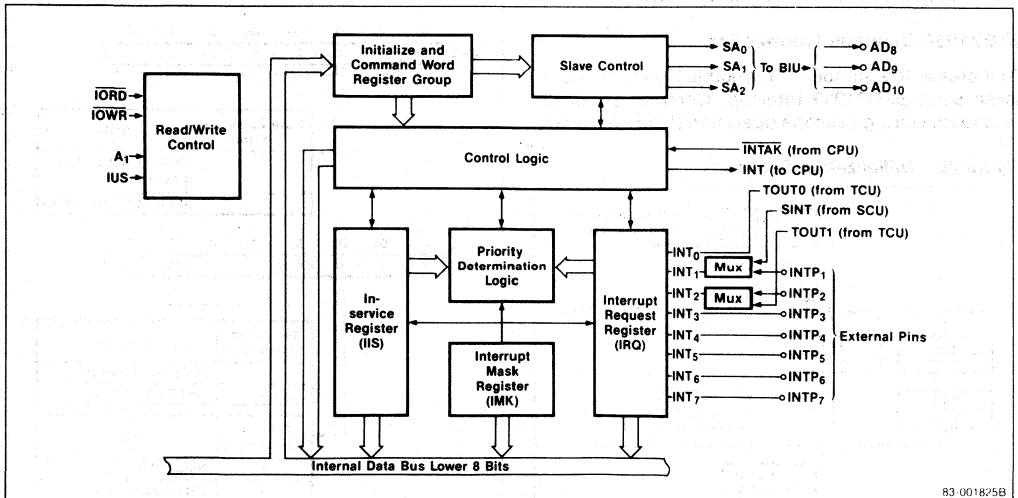
**Figure 26. SIMK Register**



### Interrupt Control Unit

The interrupt control unit (ICU) is a programmable interrupt controller equivalent to the μPD71059. The ICU arbitrates up to eight interrupt inputs, generates a CPU interrupt request, and outputs the interrupt vector number on the internal data bus during an interrupt acknowledge cycle. Cascading up to seven external slave μPD71059s permits the μPD70216 to support up to 56 interrupt sources. Figure 27 is the block diagram for the ICU.

**Figure 27. ICU Block Diagram**



The ICU has the following features.

- Eight interrupt request inputs
- Cascadable with μPD71059 Interrupt Controllers
- Programmable edge- or level-triggered interrupts (TCU, edge-triggered interrupts only)
- Individually maskable interrupt requests
- Programmable interrupt request priority
- Polling mode

### ICU Registers

Use I/O instructions to the I/O addresses selected by the OPHA and IULA registers to read from and write to the ICU registers. Address bit A<sub>1</sub> and the command word selects an ICU internal register.

	A <sub>1</sub>	Other Condition	Operation
Read	0	IMD selects IRQ	CPU ← IRQ data
	0	IMD selects IIS	CPU ← IIS data
	0	Polling phase	CPU ← Polling data
	1	—	CPU ← IMKW
Write	0	D4 = 1	CPU → IIW1
	0	D4 = 0 and D3 = 0	CPU → IPFW
	0	D4 = 0 and D3 = 1	CPU → IMDW
	1	During initialization	CPU → IIW2
	1	—	CPU → IIW3
	1	—	CPU → IIW4
	1	After initialization	CPU → IMKW
	1	—	CPU → IMKW

**Note:**

- (1) In polling phase, polling data has priority over the contents of the IRQ or IIS register when read.

**Initializing the ICU**

The ICU is always used to service maskable interrupts in a μPD70216 system. Prior to accepting maskable interrupts, the ICU must first be initialized (figure 28). Following initialization, command words from the CPU can change the interrupt request priorities, mask/un-mask interrupt requests, and select the polling mode. Figures 29 and 30 list the ICU initialization and command words.

Interrupt initialization words 1-4 (IIW1-IIW4) initialize the ICU, indicate whether external μPD71059s are connected as slaves, select the base interrupt vector, and select edge- or level-triggered inputs for INT1-INT7. Interrupt sources from the TCU are fixed as edge-triggering. INT0 is internally connected to TOUT0, and INT2 may be connected to TOUT1 by the IRSW field in the OPCN.

The interrupt mask word (IMKW) contains programmable mask bits for each of the eight interrupt inputs. The interrupt priority and finish word (IPFW) is used by the interrupt handler to terminate processing of an interrupt or change interrupt priorities. The interrupt mode word (IMDW) selects the polling register, interrupt request (IRQ) or interrupt in service (IIS) register, and the nesting mode.

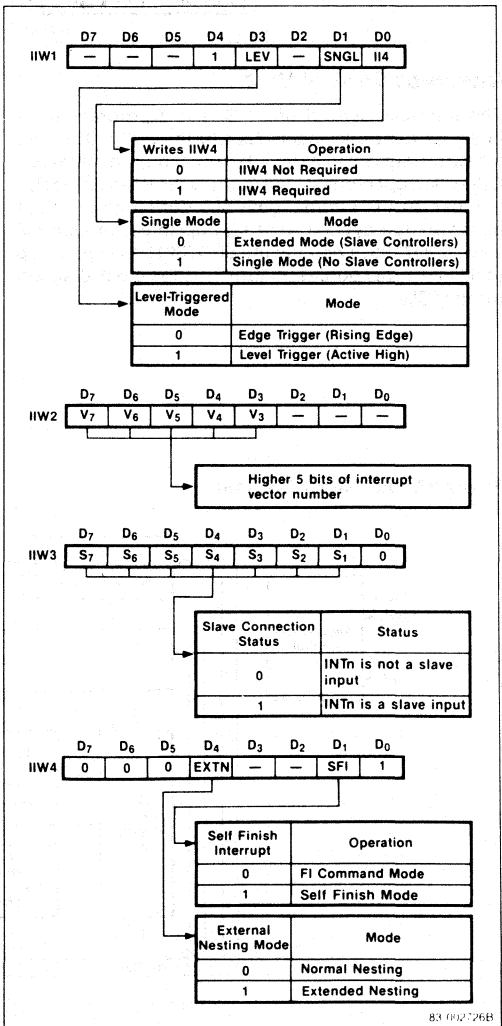
The initialization words are written in consecutive order starting with IIW1. IIW2 sets the interrupt vector. IIW3 specifies which interrupts are connected to slaves. IIW3 is only required in extended systems. The ICU will only expect to receive IIW3 if SNGL = 0 (bit D<sub>1</sub> of IIW1). IIW4 is only written if I14 = 1 (bit D<sub>0</sub> of IIW1).

**μPD71059 Cascade Connection**

To increase the number of maskable interrupts, up to seven slave μPD71059 Interrupt Controllers can be cascaded. During cascade operation (figure 31), each

slave μPD71059 INT output is routed to one of the μPD70216 INTP inputs. During the second interrupt acknowledge bus cycle, the ICU places the slave address on address lines AD<sub>10</sub>-AD<sub>8</sub>. Each slave compares this address with the slave address programmed using interrupt initialization word 3 (IIW3). If the same, the slave will place the interrupt vector on pins AD<sub>7</sub>-AD<sub>0</sub> during the second interrupt acknowledge bus cycle.

**Figure 29. Interrupt Initialization Words 1-4**



**Figure 28. Initialization Sequence**

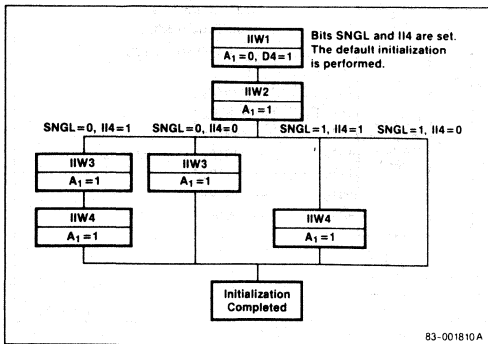


Figure 30. Command Words

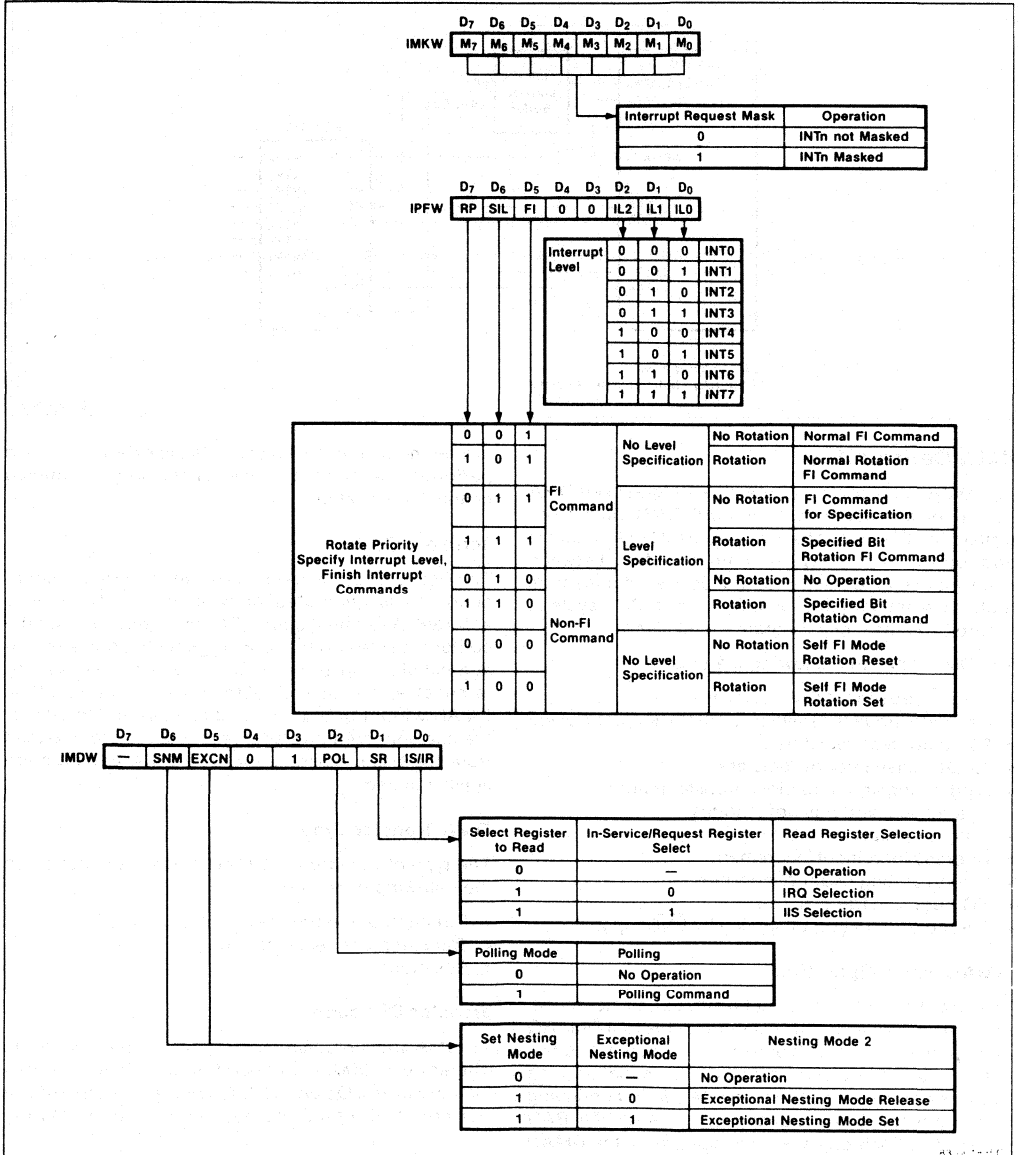
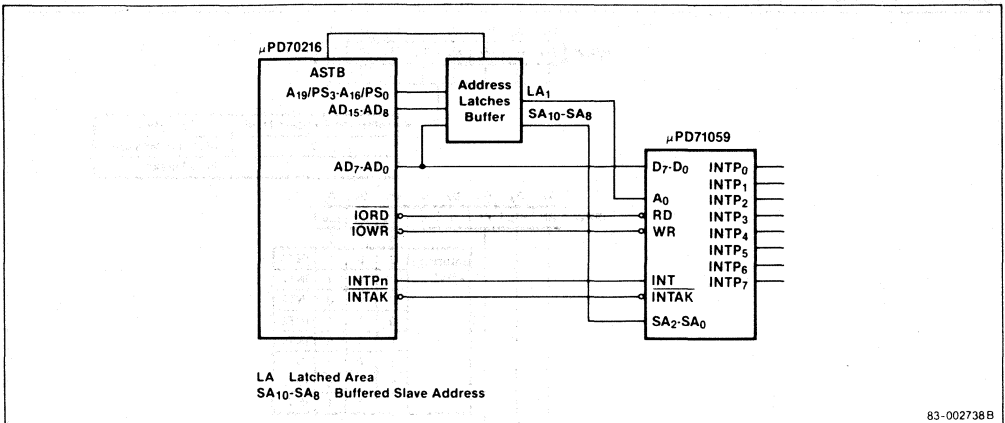




Figure 31. μPD71059 Cascade Connection



### DMA Control Unit

The DMA Control Unit (DMAU) is a high-speed DMA controller compatible with the μPD71071 DMA Controller. The DMAU has four independent DMA channels and performs high-speed data transfers between memory and external peripheral devices at speeds as high as 4 megabytes/second in an 8-MHz system. Figure 32 is the block diagram for the DMAU.

The DMAU has the following features:

- Four independent DMA channels
- Cascade mode for slave μPD71071 DMA controllers
- 20-bit address registers
- 16-bit transfer count registers
- Single, demand, and block transfer modes
- Bus release and bus hold modes
- Autoinitialization
- Address increment/decrement
- Fixed/rotating channel priorities
- TC output at transfer end
- Forced termination of service by  $\overline{\text{END}}$  input

### DMAU Basic Operation

The DMAU operates in either a slave or master mode. In the slave mode, the DMAU samples the four DMARQ input pins every clock. If one or more inputs are active, the corresponding DMA request bits are set and the DMAU sends a bus request to the BAU while continuing to sample the DMA request inputs. After the BAU returns the DMA bus acknowledge signal, the DMAU stops DMA request sampling, selects the DMA channel with the highest priority, and enters the bus master mode to perform the DMA transfer. While in the bus

master mode, the DMAU controls the external bus and performs DMA transfers based on the preprogrammed channel information.

### Terminal Count

The DMAU ends DMA service when the terminal count condition is generated or when the  $\overline{\text{END}}$  input is asserted. A terminal count (TC) is produced when the contents of the current count register becomes zero. If autoinitialization is not enabled when DMA service terminates, the mask bit of the channel is set and the DMARQ input of that channel is masked. Otherwise, the current count and address registers are reloaded from the base registers and new DMA transfers are again enabled.

### DMA Transfer Type

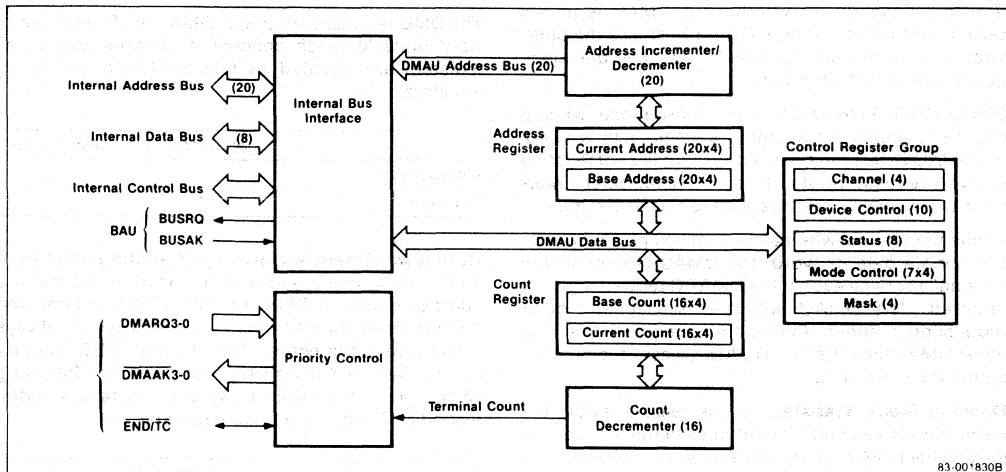
The type of transfer the DMAU performs depends on the following conditions.

- Direction of the transfer (each channel)
- Transfer mode (each channel)
- Bus mode

### Transfer Direction

All DMA transfers use memory as a reference point. Therefore, a DMA read operation transfers data from memory to an I/O port. A DMA write operation reads an I/O port and writes the data into memory. During

**Figure 32. DMAU Block Diagram**



memory-to-I/O transfer, the DMA mode (DMD) register is used to select the transfer directions for each channel and activate the appropriate control signals.

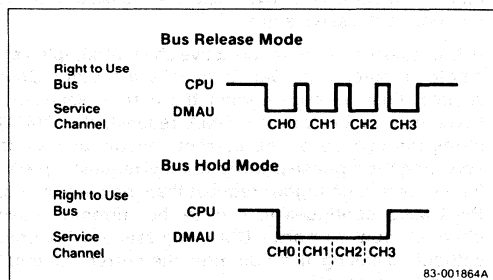
Operation	Transfer Direction	Activated Signals
DMA read	Memory → I/O	IOWR, MRD
DMA write	I/O → Memory	IORD, MWR
DMA verify		Addresses only; no transfer performed

### Bus Mode

The DMA device control (DDC) register selects operation in either the bus release or bus hold mode. The selected bus mode determines the DMAU conditions for return of the bus to the BAU. Figure 33 shows that in bus release mode, only a single channel is serviced after the DMAU obtains the bus. When DMA service ends (termination conditions depend on the transfer mode), the DMAU returns the bus to the BAU regardless of the state of other DMA requests, and the DMAU reenters the slave mode. When the DMAU regains use of the bus, a new DMA operation can begin.

In bus hold mode, several channels can receive contiguous service without releasing the bus. If there is another valid DMA request when a channel's DMA service is finished, the new DMA service can begin immediately after the previous service without returning the bus to the BAU.

**Figure 33. Bus Modes**



### Transfer Modes

The DMD register also selects either single, demand, or block transfer mode for each channel. The conditions for the termination of each transfer characterize each transfer mode. The following table shows the various transfer modes and termination conditions.

Transfer Mode	Termination Conditions
Single	After each byte/word transfer
Demand	END input Terminal count Inactive DMARQ DMARQ of a higher priority channel becomes active (bus hold mode)
Block	END input Terminal count

The operation of single, demand, and block mode transfers depends on whether the DMAU is in bus release or bus hold mode. Figure 34 shows the operation flow for the six possible transfer and bus mode operations in DMA transfer.

**Single Mode Transfer.** In bus release mode, when a channel completes transfer of a single byte or word, the DMAU enters the slave mode regardless of the state of DMA request inputs. In this manner, other lower-priority bus masters will be able to access the bus.

In bus hold mode, when a channel completes transfer of a single byte or word, the DMAU terminates the channel's service even if the DMARQ request signal is asserted. The DMAU will then service any other requesting channel. If there are no requests from any other DMA channels, the DMAU releases the bus and enters the slave state.

**Demand Mode Transfer.** In bus release mode, the currently active channel continues to transfer data as long as the DMA request of that channel is active, even though other DMA channels are issuing higher-priority requests. When the DMA request of the serviced channel becomes inactive, the DMAU releases the bus and enters the slave state.

In bus hold mode, when the active channel completes a single transfer, the DMAU checks the other DMA request lines without ending the current service. If there is a higher-priority DMA request, the DMAU stops the service of the current channel and starts servicing the highest-priority channel requesting service. If there is no higher request than the current one, the DMAU continues to service the currently active channel. Lower-priority DMA requests are honored without releasing the bus after the current channel service is complete.

**Block Mode Transfer.** In bus release mode, the current channel continues DMA transfers until a terminal count or the external  $\overline{END}$  input becomes active. During this time, the DMAU ignores all other DMA requests. After completion of the block transfer, the DMAU releases the bus and enters the slave state, even if DMA requests from other channels are active.

In bus hold mode, the current channel transfers data until an internal or external  $\overline{END}$  signal becomes active. When the service is complete, the DMAU checks all DMA requests without releasing the bus. If there is an active request, the DMAU immediately begins servicing the request. The DMAU releases the bus after it honors all DMA requests or a higher-priority bus master requests the bus.

**Byte/Word Transfer**

The DMD register can specify DMA transfers in byte or word units for each channel. Addresses and count registers are updated as follows during byte/word transfers.

	Byte Transfer	Word Transfer
Address register	$\pm 1$	$\pm 2$
Count register	-1	-1

During word transfers, two bytes starting at an even address are handled as a single word. If the starting address is odd, a DMA transfer is started after first decrementing the address by 1. For this reason, always select even addresses. The  $A_0$  and  $\overline{UBE}$  outputs control byte and word DMA transfers. The following shows the relationship between the data bus width,  $A_0$  and  $\overline{UBE}$  signals, and data bus status.

$A_0$	$\overline{UBE}$	Data Bus Status
0	1	D <sub>7</sub> -D <sub>0</sub> valid
1	0	D <sub>15</sub> -D <sub>8</sub> valid
0	0	D <sub>15</sub> -D <sub>0</sub> valid

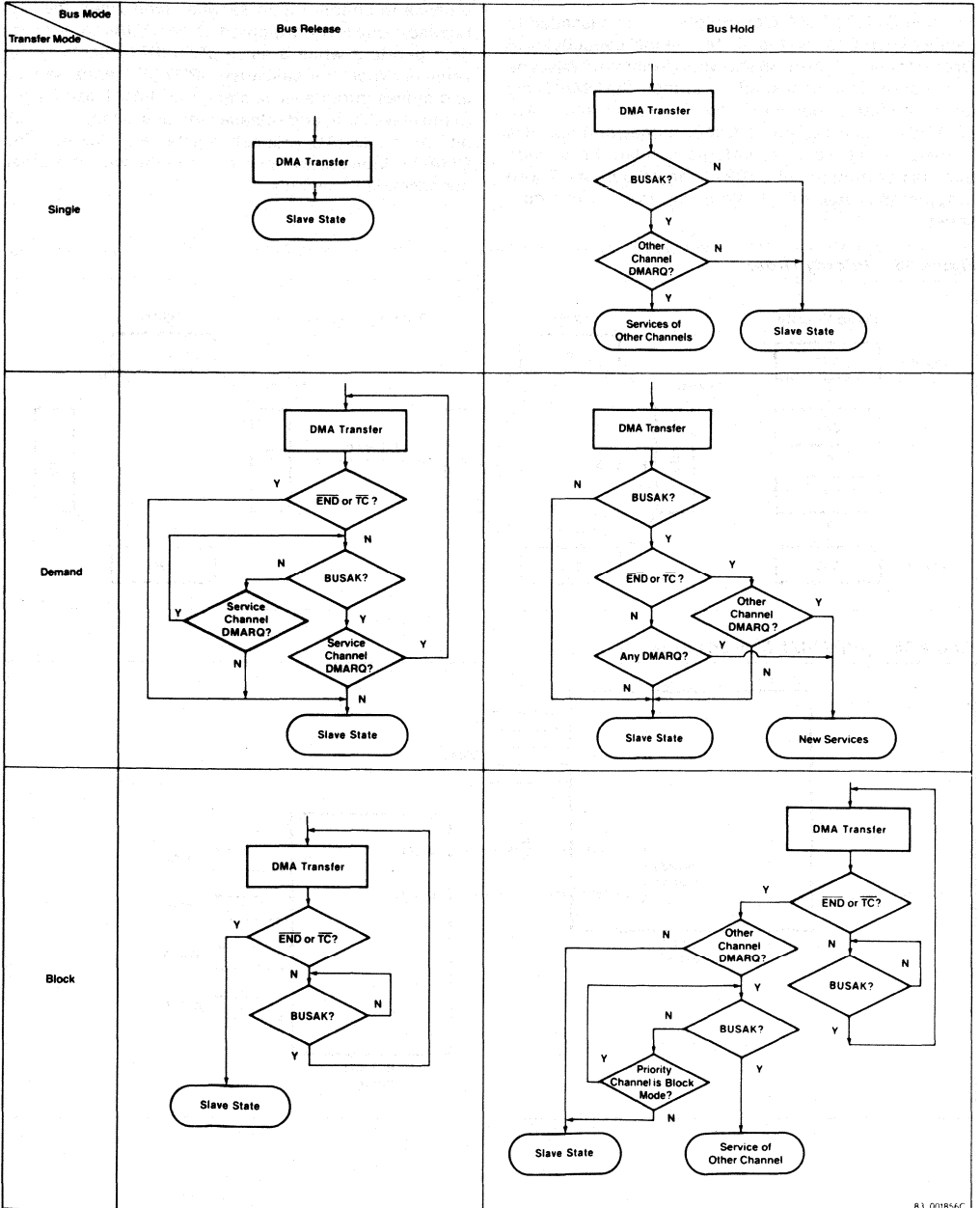
**Autoinitialize**

When the DMD register selects autoinitialize for a channel, the DMAU automatically reinitializes the address and count registers when  $\overline{END}$  is asserted or the terminal count condition is reached. The contents of the base address and base count registers are transferred to the current address and current count registers, and the applicable bit of the mask register remains cleared.

**Channel Priority**

Each of the four DMAU channels is assigned a priority. When multiple DMA requests from several channels occur simultaneously, the channel with the highest priority will be serviced first. The DDC register selects one of two priority schemes: fixed or rotating (figure 35). In fixed priority, channel 0 is assigned the highest priority and channel 3, the lowest. In rotating priority, priority order is rotated after each service so that the channel last serviced receives the lowest priority. This method prevents the exclusive servicing of higher-priority channels and the lockout of lower-priority DMA channels.

**Figure 34. Transfer Modes**





## Bus Waiting Operation

The DMAU will automatically perform a bus waiting operation (figure 37) whenever the RCU refresh request queue fills. When the DMA bus acknowledge goes inactive, the DMAU enters the bus waiting mode and inactivates the DMA bus request signal. Control of the bus is then transferred to the higher-priority RCU by the BAU.

Two clocks later, the DMAU reasserts its internal DMA bus request. The bus waiting mode is continued until the DMA bus acknowledge signal again becomes active and the interrupted DMA service is immediately restarted.

## Programming the DMAU

To prepare a channel for DMA transfer, the following characteristics must be programmed.

- Starting address for the transfer
- Transfer count
- DMA operating mode
- Transfer size (byte/word)

The contents of the OPHA and DULA registers determine the base I/O port address of the DMAU. Addresses A<sub>3</sub>-A<sub>0</sub> are used to select a particular register as follow:

A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Register	Operation
0	0	0	0	DICM	Write
0	0	0	1	DCH	Read/Write
0	0	1	0	DBC/DCC (low)	Read/Write
0	0	1	1	DBC/DCC (high)	Read/Write
0	1	0	0	DBA/DCA (low)	Read/Write
0	1	0	1	DBA/DCA (high)	Read/Write
0	1	1	0	DBA/DCA (upper)	Read/Write
0	1	1	1	Reserved	—
1	0	0	0	DDC (low)	Read/Write
1	0	0	1	DDC (high)	Read/Write
1	0	1	0	DMD	Read/Write
1	0	1	1	DST	Read
1	1	0	0	Reserved	—
1	1	0	1	Reserved	—
1	1	1	0	Reserved	—
1	1	1	1	DMK	Read/Write

Word I/O instructions can be used to read/write the register pairs listed below. All other registers are accessed via byte I/O instructions.

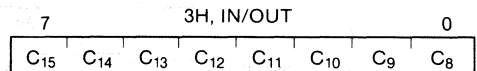
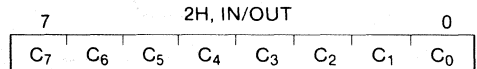
- DBC/DCC
- DBA/DCA (higher/lower only)
- DDC

## DMAU Registers

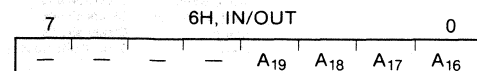
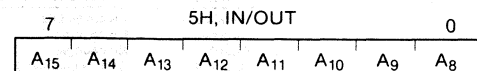
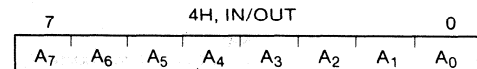
**Initialize.** The DMA initialize command (DICM) register (figure 38) is used to perform a software reset of the DMAU. The DICM is accessed using the byte OUT instruction.

**Channel Register.** Writes to the DMA channel (DCH) register (figure 39) select one of the four DMA channels for programming and also the base/current registers. Reads of the DCH register return the currently-selected channel and the register access mode.

**Count Registers.** When bit 2 of the DCH register is cleared, a write to the DMA count register updates both the DMA base count (DBC) and the DMA current count (DCC) registers with a new count. If bit 2 of the DCH register is set, a write to the DMA count register affects only the DBC register. The DBC register holds the initial count value until a new count is specified. If autoinitialization is enabled, this value is transferred to the DCC register when a terminal count or END condition occurs. For each DMA transfer, the current count register is decremented by one. The format of the DMA count register is shown below. The count value loaded into the DBC/DCC registers is one less than the desired transfer count.



**Address Register.** Use either byte or word I/O instructions with the lower two bytes (4H and 5H) of the DMA address register. However, byte I/O instructions must be used to access the high-order byte (6H) of this register. When bit 2 of the channel register is cleared, a write to the DMA address register updates both the DMA base address (DBA) and the DMA current address (DCA) registers with the new address. If bit 2 of the DCH register is set, a write to the DMA address register affects only the DBA register.



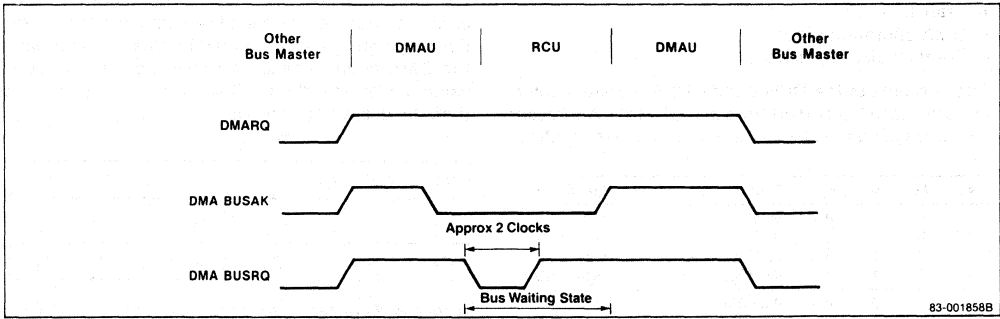
## μPD70216 (V50)

The DBA register holds the starting address value until a new address is specified. This value is transferred to the DCA register automatically if autoinitialization is selected. For each DMA transfer, the current address register is updated by two during word transfers and by one during byte transfers.

**Device Control Register.** The DMA device control (DDC) register (figure 40) is used to program the DMA transfer characteristics common to all DMA channels. It controls the bus mode, write timing, priority logic, and enable/disable of the DMAU.

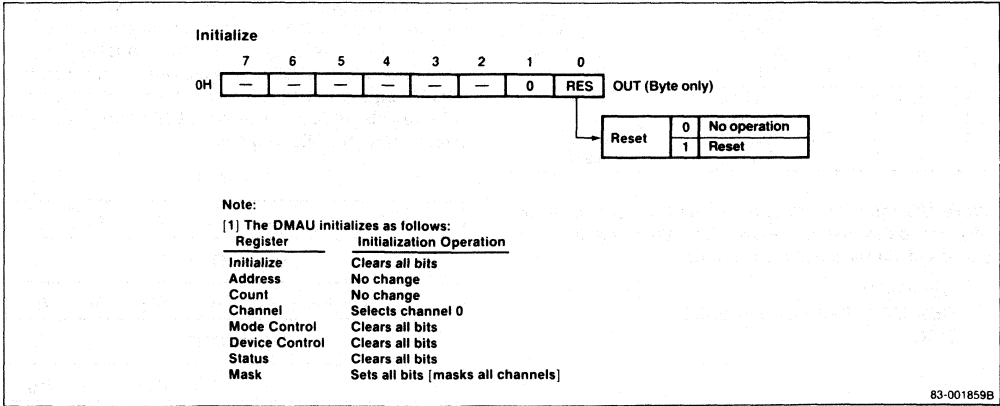
**Status Register.** The DMA status (DST) register (figure 41) contains information about the current state of each DMA channel. Software can determine if a termination condition has been reached (TC<sub>3</sub>-TC<sub>0</sub>) or if a DMA service request is present (RQ<sub>3</sub>-RQ<sub>0</sub>). The byte IN instruction must be used to read this register.

**Figure 37. Bus Waiting Operation**



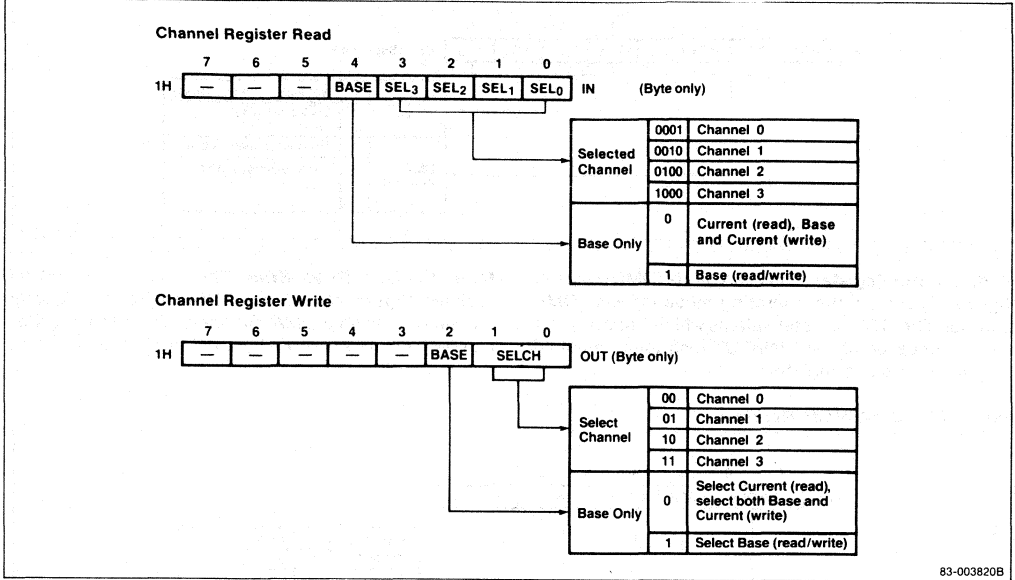
83-001858B

**Figure 38. DMA Initialize Command Register**



83-001859B

Figure 39. DMA Channel Register



3

Figure 40. DMA Device Control Register

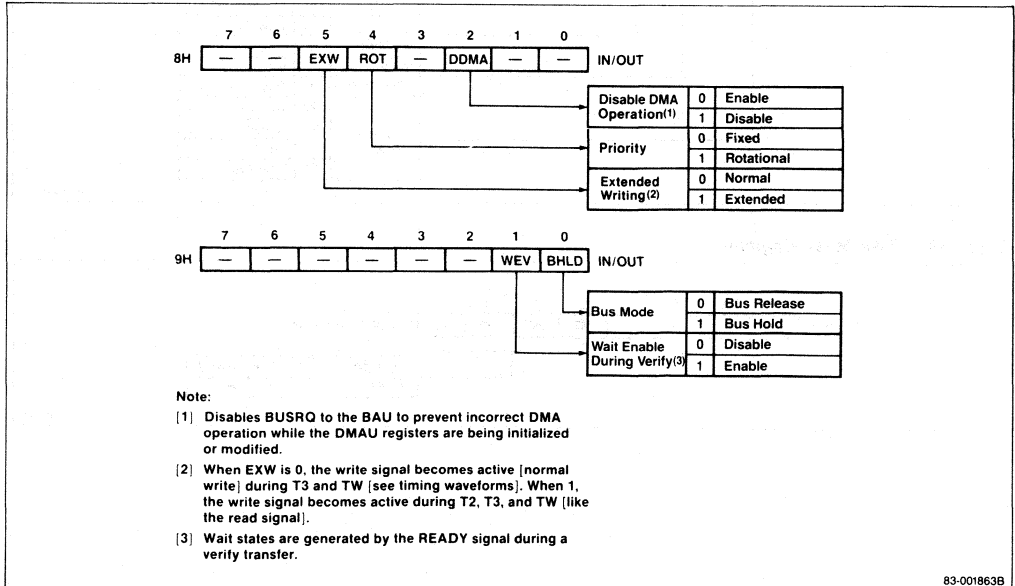
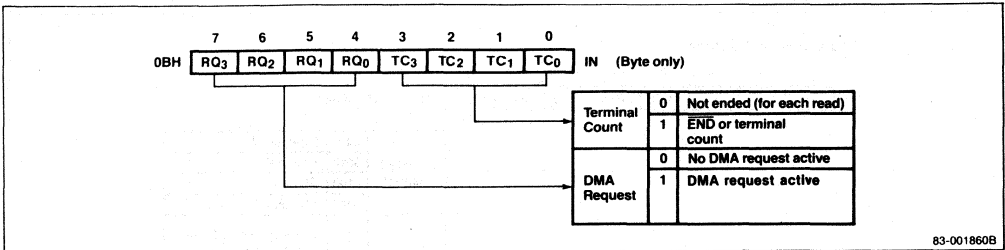




Figure 41. DMA Status Register



**Mode Control Register.** The DMA mode (DMD) register (figure 42) selects the operating mode for each DMA channel. The DCH register selects which DMD register will be accessed. A byte IN/OUT instruction must be used to access this register.

**Mask Register Read/Write.** The DMA mask (DMK) register (figure 43) allows software to individually enable and disable DMA channels. The DMK register can only be accessed via byte I/O instructions.

Figure 42. DMA Mode Register

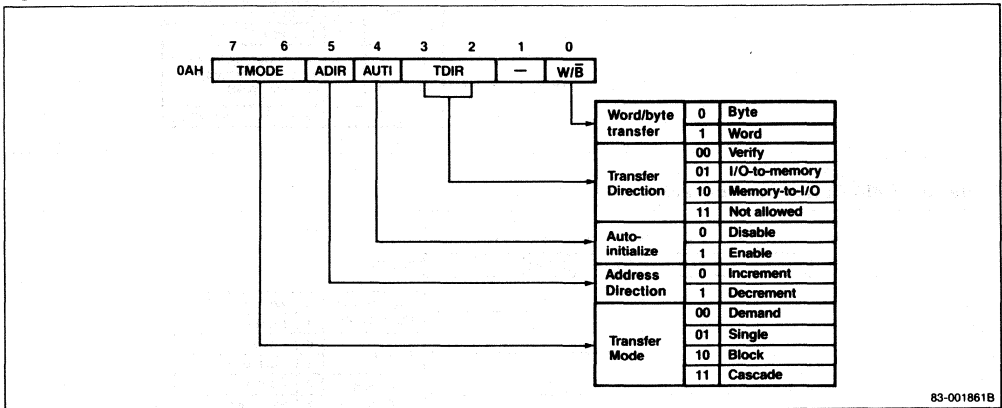
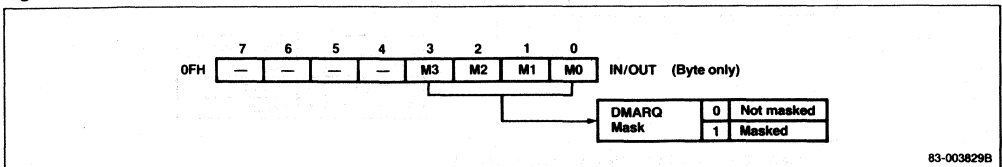


Figure 43. DMA Mask Register



### Reset

The falling edge of the  $\overline{\text{RESET}}$  signal resets the μPD70216. The signal must be held low for at least four clock cycles to be recognized as valid.

CPU Reset State	
Register	Reset Value
PPF	0000H
PC	0000H
PS	FFFFH
SS	0000H
DS0	0000H
DS1	0000H
PSW	F002H
AW, BW, CW, DW, IX, IY, BP, SP	Undefined
Instruction queue	Cleared

When  $\overline{\text{RESET}}$  returns to the high level, the CPU will start fetching instructions from physical address FFFF0H.

### Internal Peripheral Registers

Internal peripheral devices initialized on reset are listed in the following table. I/O devices not listed are not initialized on reset and must be initialized by software.

	Register	Reset Value
System I/O area	OPCN	----0000
	OPSEL	----0000
	WCY1	11111111
	WCY2	----1111
	WMB	-111-111
	TCKS	---00000
SCU	RFC	x--01000
	SMD	01001011
	SCM	--0000-0
	SIMK	-----11
	SST	10000100
	DCH	---00001
DMAU	DMD	000000-0
	DDC (low)	--00-0--
	DDC (high)	-----00
	DST	xxxx0000
	DMK	----1111

Symbols: x = unaffected; 0 = cleared; 1 = set; (-) = unused.

### Output Pin Status

The following table lists output pin status during reset.

Signal	Status
INTAK, BUFEN, BUFR/W, MRD, MWR, END/TC, IOWR, IORD, REFRQ, UBE, BS <sub>2</sub> -BS <sub>0</sub> , BUSLOCK, RESOUT, DMAAR3-DMAAR0	High level
QS <sub>1</sub> -QS <sub>0</sub> , ASTB, HLDK	Low level
A <sub>19</sub> -A <sub>16</sub> /PS <sub>3</sub> -PS <sub>0</sub> , TOUT2	High or low level
AD <sub>15</sub> -AD <sub>0</sub>	High impedance
CLKOUT	Continues to supply clock

**Instruction Set**

**Symbols**

Preceding the instruction set, several tables explain symbols, abbreviations, and codes.

**Clocks**

In the Clocks column of the instruction set, the numbers cover these operations: instruction decoding, effective address calculation, operand fetch, and instruction execution.

Clock timings assume the instruction has been pre-fetched and is present in the six-byte instruction queue. Otherwise, add four clocks for each pair of bytes not present.

Word operands require four additional clocks for each transfer to an unaligned (odd-addressed) memory operand. These times are shown on the right-hand side of the slash (/).

For conditional control transfer or branch instructions, the number on the left side of the slash is applicable if the transfer or branch takes place. The number on the right side is applicable if it does not take place.

If a range of numbers is given, the execution time depends on the operands involved.

**Symbols**

Symbol	Meaning
acc	Accumulator (AW or AL)
disp	Displacement (8 or 16 bits)
dmem	Direct memory address
dst	Destination operand or address
ext-disp8	16-bit displacement (sign-extension byte + 8-bit displacement)
far_label	Label within a different program segment
far_proc	Procedure within a different program segment
fp_op	Floating point instruction operation
imm	8- or 16-bit immediate operand

**Symbols**

Symbol	Meaning
imm3/4	3- or 4-bit immediate bit offset
imm8	8-bit immediate operand
imm16	16-bit immediate operand
mem	Memory field (000 to 111); 8- or 16-bit memory location
mem8	8-bit memory location
mem16	16-bit memory location
mem32	32-bit memory location
memptr16	Word containing the destination address within the current segment
memptr32	Double word containing a destination address in another segment
mod	Mode field (00 to 10)
near_label	Label within the current segment
near_proc	Procedure within the current segment
offset	Immediate offset data (16 bits)
pop_value	Number of bytes to discard from the stack
reg	Register field (000 to 111); 8- or 16-bit general-purpose register
reg8	8-bit general-purpose register
reg16	16-bit general-purpose register
regptr	16-bit register containing a destination address within the current segment
regptr16	Register containing a destination address within the current segment
seg	Immediate segment data (16 bits)
short_label	Label between -128 and +127 bytes from the end of the current instruction
sr	Segment register
src	Source operand or address
temp	Temporary register (8/16/32 bits)
AC	Auxiliary carry flag
AH	Accumulator (high byte)
AL	Accumulator (low byte)
AW	Accumulator (16 bits)
BH	BW register (high byte)
BL	BW register (low byte)
BP	BP register
BRK	Break flag
BW	BW register (16 bits)
CH	CW register (high byte)
CL	CW register (low byte)

### Symbols (cont)

Symbol	Meaning
CW	CW register (16 bits)
CY	Carry flag
DH	DW register (high byte)
DIR	Direction flag
DL	DW register (low byte)
DS0	Data segment 0 register (16 bits)
DS1	Data segment 1 register (16 bits)
DW	DW register (16 bits)
IE	Interrupt enable flag
IX	Index register (source) (16 bits)
IY	Index register (destination) (16 bits)
MD	Mode flag
P	Parity flag
PC	Program counter (16 bits)
PS	Program segment register (16 bits)
PSW	Program status word (16 bits)
R	Register set
S	Sign extend operand field S = 0 No sign extension S = 1 Sign extend immediate byte operand
S	Sign flag
SP	Stack pointer (16 bits)
SS	Stack segment register (16 bits)
V	Overflow flag
W	Word/byte field (0 to 1)
X. XXX. YYY. ZZZ	Data to identify the instruction code of the external floating point arithmetic chip
XXH	Two-digit hexadecimal value
XXXXH	Four-digit hexadecimal value
Z	Zero flag

### Flag Operations

Symbol	Meaning
(blank)	No change
0	Cleared to 0
1	Set to 1
x	Set or cleared according to result
u	Undefined
R	Restored to previous state

### Memory Addressing Modes

mem	mod = 00	mod = 01	mod = 10
000	BW + IX	BW + IX + disp8	BW + IX + disp16
001	BW + IY	BW + IY + disp8	BW + IY + disp16
010	BP + IX	BP + IX + disp8	BP + IX + disp16
011	BP + IY	BP + IY + disp8	BP + IY + disp16
100	IX	IX + disp8	IX + disp16
101	IY	IY + disp8	IY + disp16
110	Direct	BP + disp8	BP + disp16
111	BW	BW + disp8	BW + disp16

### Register Selection (mod = 11)

reg	W = 0	W = 1
000	AL	AW
001	CL	CW
010	DL	DW
011	BL	BW
100	AH	SP
101	CH	BP
110	DH	IX
111	BH	IY

### Segment Register Selection

sr	Segment Register
00	DS1
01	PS
10	SS
11	DS0



### Instruction Set (cont)

Mnemonic	Operand	Opcode																Clocks	Bytes	Flags						
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z	
<b>Block Transfer Instructions (cont)</b>																										
LDM	src	1	0	1	0	1	1	0	W									1								
																	7 (7) + 9n (W = 0)									
																	7 (7) + 9n (W = 1, even addresses)									
																	7 (11) + 13n (W = 1, odd addresses)									
STM	dst	1	0	1	0	1	0	1	W								1									
																	5 (5) + 4n (W = 0)									
																	5 (5) + 4n (W = 1, even addresses)									
																	5 (9) + 8n (W = 1, odd addresses)									

n = number of returns

String instruction execution clocks for a single-instruction execution are in parentheses.

### I/O Instructions

IN	acc, imm8	1	1	1	0	0	1	0	W								9/13	2								
	acc, DW	1	1	1	0	1	1	0	W								8/12	1								
OUT	imm8, acc	1	1	1	0	0	1	1	W								8/12	2								
	DW, acc	1	1	1	0	1	1	1	W								8/12	1								
INM	dst, DW	0	1	1	0	1	1	0	W								1									
																	9 (10) + 8n (W = 0)									
																	9 (10) + 8n (W = 1, even addresses)									
																	9 (18) + 16n (W = 1, odd addresses)									
OUTM	DW, src	0	1	1	0	1	1	1	W								1									
																	9 (10) + 8n (W = 0)									
																	9 (10) + 8n (W = 1, even addresses)									
																	9 (18) + 16n (W = 1, odd addresses)									

n = number of transfers

String instruction execution clocks for a single instruction execution are in parentheses.

Use the right side of the slash (/) for DMA I/O accesses.

### BCD Instructions

ADJBA		0	0	1	1	0	1	1	1								7	1	x	x	u	u	u	u	
ADJ4A		0	0	1	0	0	1	1	1								3	1	x	x	u	x	x	x	
ADJBS		0	0	1	1	1	1	1	1								7	1	x	x	u	u	u	u	
ADJ4S		0	0	1	0	1	1	1	1								3	1	x	x	u	x	x	x	
ADD4S	dst, src	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	7 + 19n	2	u	x	u	u	u	x
SUB4S	dst, src	0	0	0	0	1	1	1	1	0	0	1	0	0	0	1	0	7 + 19n	2	u	x	u	u	u	x
CMP4S	dst, src	0	0	0	0	1	1	1	1	0	0	1	0	0	1	1	0	7 + 19n	2	u	x	u	u	u	x
ROL4	reg8	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	13	3						
	mem8	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	25	3-5						
	mod	0	0	0	0					mem															
ROR4	reg8	0	0	0	0	1	1	1	1	0	0	1	0	1	0	1	0	17	3						
	mem8	0	0	0	0	1	1	1	1	0	0	1	0	1	0	1	0	29	3-5						
	mod	0	0	0	0					mem															

n = number of BCD digits divided by 2

**Instruction Set (cont)**

Mnemonic	Operand	Opcode										Clocks	Bytes	Flags											
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P	S	Z
<b>Data Type Conversion Instructions</b>																									
CVTBD		1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	15	2	u	u	u	x	x	x
CVTDB		1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	7	2	u	u	u	x	x	x
CVTBW		1	0	0	1	1	0	0	0									2	1						
CVTWL		1	0	0	1	1	0	0	1									4/5	1						
<b>Arithmetic Instructions</b>																									
ADD	reg, reg	0	0	0	0	0	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x				
	mem, reg	0	0	0	0	0	0	0	W	mod	reg	mem	13/21	2-4	x	x	x	x	x	x					
	reg, mem	0	0	0	0	0	0	1	W	mod	reg	mem	10/14	2-4	x	x	x	x	x	x					
	reg, imm	1	0	0	0	0	0	S	W	1	1	0	0	0	reg	4	3-4	x	x	x	x	x	x		
	mem, imm	1	0	0	0	0	0	S	W	mod	0	0	0	mem	15/23	3-6	x	x	x	x	x	x			
	acc, imm	0	0	0	0	0	1	0	W							4	2-3	x	x	x	x	x	x		
ADDC	reg, reg	0	0	0	1	0	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x				
	mem, reg	0	0	0	1	0	0	0	W	mod	reg	mem	13/21	2-4	x	x	x	x	x	x					
	reg, mem	0	0	0	1	0	0	1	W	mod	reg	mem	10/14	2-4	x	x	x	x	x	x					
	reg, imm	1	0	0	0	0	0	S	W	1	1	0	1	0	reg	4	3-4	x	x	x	x	x	x		
	mem, imm	1	0	0	0	0	0	S	W	mod	0	1	0	mem	15/23	3-6	x	x	x	x	x	x			
	acc, imm	0	0	0	1	0	1	0	W							4	2-3	x	x	x	x	x	x		
SUB	reg, reg	0	0	1	0	1	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x				
	mem, reg	0	0	1	0	1	0	0	W	mod	reg	mem	13/21	2-4	x	x	x	x	x	x					
	reg, mem	0	0	1	0	1	0	1	W	mod	reg	mem	10/14	2-4	x	x	x	x	x	x					
	reg, imm	1	0	0	0	0	0	S	W	1	1	1	0	1	reg	4	3-4	x	x	x	x	x	x		
	mem, imm	1	0	0	0	0	0	S	W	mod	1	0	1	mem	15/23	3-6	x	x	x	x	x	x			
	acc, imm	0	0	1	0	1	1	0	W							4	2-3	x	x	x	x	x	x		
SUBC	reg, reg	0	0	0	1	1	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x				
	mem, reg	0	0	0	1	1	0	0	W	mod	reg	mem	13/21	2-4	x	x	x	x	x	x					
	reg, mem	0	0	0	1	1	0	1	W	mod	reg	mem	10/14	2-4	x	x	x	x	x	x					
	reg, imm	1	0	0	0	0	0	S	W	1	1	0	1	1	reg	4	3-4	x	x	x	x	x	x		
	mem, imm	1	0	0	0	0	0	S	W	mod	0	1	1	mem	15/23	3-6	x	x	x	x	x	x			
	acc, imm	0	0	0	1	1	1	0	W							4	2-3	x	x	x	x	x	x		
INC	reg8	1	1	1	1	1	1	1	0	1	1	0	0	0	reg	2	2	x	x	x	x	x	x		
	mem	1	1	1	1	1	1	1	W	mod	0	0	0	mem	13/21	2-4	x	x	x	x	x	x			
	reg16	0	1	0	0	0	reg								2	1	x	x	x	x	x	x			
DEC	reg8	1	1	1	1	1	1	1	0	1	1	0	0	1	reg	2	2	x	x	x	x	x	x		
	mem	1	1	1	1	1	1	1	W	mod	0	0	1	mem	13/21	2-4	x	x	x	x	x	x			
	reg16	0	1	0	0	1	reg								2	1	x	x	x	x	x	x			
MULU	reg	1	1	1	1	0	1	1	W	1	1	1	0	0	reg	21-30	2	u	x	x	u	u	u		
	mem	1	1	1	1	0	1	1	W	mod	1	0	0	mem	26-35	2-4	u	x	x	u	u	u			

## Instruction Set (cont)

Mnemonic	Operand	Opcode											Clocks	Bytes	Flags								
		7	6	5	4	3	2	1	0	7	6	5			4	3	2	1	0	AC	CY	V	P
<b>Arithmetic Instructions (cont)</b>																							
MUL	reg	1	1	1	1	0	1	1	W	1	1	1	0	1	reg	33-47	2	u	x	x	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	0	1	mem	38-52	2-4	u	x	x	u	u	u	
	reg16,reg16,imm8	0	1	1	0	1	0	1	1	1	1	reg	reg	28-34	3	u	x	x	u	u	u		
	reg16,mem16,imm8	0	1	1	0	1	0	1	1	mod	reg	mem	33-39	3-5	u	x	x	u	u	u			
	reg16,reg16,imm16	0	1	1	0	1	0	0	1	1	1	reg	reg	36-42	4	u	x	x	u	u	u		
reg16,mem16,imm16	0	1	1	0	1	0	0	1	mod	reg	mem	41-47	4-6	u	x	x	u	u	u				
DIVU	reg	1	1	1	1	0	1	1	W	1	1	1	1	0	reg	19-25	2	u	u	u	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	1	0	mem	24-30	2-4	u	u	u	u	u	u	
DIV	reg	1	1	1	1	0	1	1	W	1	1	1	1	1	reg	29-43	2	u	u	u	u	u	u
	mem	1	1	1	1	0	1	1	W	mod	1	1	1	mem	34-48	2-4	u	u	u	u	u	u	
<b>Comparison Instructions</b>																							
CMP	reg, reg	0	0	1	1	1	0	1	W	1	1	reg	reg	2	2	x	x	x	x	x	x		
	mem, reg	0	0	1	1	1	0	0	W	mod	reg	mem	10/14	2-4	x	x	x	x	x	x			
	reg, mem	0	0	1	1	1	0	1	W	mod	reg	mem	10/14	2-4	x	x	x	x	x	x			
	reg, imm	1	0	0	0	0	0	S	W	1	1	1	1	1	reg	4	3-4	x	x	x	x	x	x
	mem, imm	1	0	0	0	0	0	S	W	mod	1	1	1	mem	12/16	3-6	x	x	x	x	x	x	
acc, imm	0	0	1	1	1	1	0	W					4	2-3	x	x	x	x	x	x			
<b>Logical Instructions</b>																							
NOT	reg	1	1	1	1	0	1	1	W	1	1	0	1	0	reg	2	2						
	mem	1	1	1	1	0	1	1	W	mod	0	1	0	mem	13/21	2-4							
NEG	reg	1	1	1	1	0	1	1	W	1	1	0	1	1	reg	2	2	x	x	x	x	x	x
	mem	1	1	1	1	0	1	1	W	mod	0	1	1	mem	13/21	2-4	x	x	x	x	x	x	
TEST	reg, reg	1	0	0	0	0	1	0	W	1	1	reg	reg	2	2	u	0	0	x	x	x		
	mem, reg	1	0	0	0	0	1	0	W	mod	reg	mem	9/13	2-4	u	0	0	x	x	x			
	reg, imm	1	1	1	1	0	1	1	W	1	1	0	0	0	reg	4	3-4	u	0	0	x	x	x
	mem, imm	1	1	1	1	0	1	1	W	mod	0	0	0	mem	10/14	3-6	u	0	0	x	x	x	
	acc, imm	1	0	1	0	1	0	0	W				4	2-3	u	0	0	x	x	x			
AND	reg, reg	0	0	1	0	0	0	1	W	1	1	reg	reg	2	2	u	0	0	x	x	x		
	mem, reg	0	0	1	0	0	0	0	W	mod	reg	mem	13/21	2-4	u	0	0	x	x	x			
	reg, mem	0	0	1	0	0	0	1	W	mod	reg	mem	10/14	2-4	u	0	0	x	x	x			
	reg, imm	1	0	0	0	0	0	0	W	1	1	1	0	0	reg	4	3-4	u	0	0	x	x	x
	mem, imm	1	0	0	0	0	0	0	W	mod	1	0	0	mem	15/23	3-6	u	0	0	x	x	x	
	acc, imm	0	0	1	0	0	1	0	W				4	2-3	u	0	0	x	x	x			
OR	reg, reg	0	0	0	0	1	0	1	W	1	1	reg	reg	2	2	u	0	0	x	x	x		
	mem, reg	0	0	0	0	1	0	0	W	mod	reg	mem	13/21	2-4	u	0	0	x	x	x			
	reg, mem	0	0	0	0	1	0	1	W	mod	reg	mem	10/14	2-4	u	0	0	x	x	x			
	reg, imm	1	0	0	0	0	0	0	W	1	1	0	0	1	reg	4	3-4	u	0	0	x	x	x
	mem, imm	1	0	0	0	0	0	0	W	mod	0	0	1	mem	15/23	3-6	u	0	0	x	x	x	
	acc, imm	0	0	1	0	0	1	0	W				4	2-3	u	0	0	x	x	x			



**Instruction Set (cont)**

Mnemonic	Operand	Opcode										Clocks	Bytes	Flags											
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P	S	Z
<b>Logical Instructions (cont)</b>																									
XOR	reg, reg	0	0	1	1	0	0	1	W	1	1	reg	reg	2	2	u	0	0	x	x	x				
	mem, reg	0	0	1	1	0	0	0	W	mod	reg	mem	13/21	2-4	u	0	0	x	x	x					
	reg, mem	0	0	1	1	0	0	1	W	mod	reg	mem	10/14	2-4	u	0	0	x	x	x					
	reg, imm	1	0	0	0	0	0	0	W	1	1	1	1	0	reg	4	3-4	u	0	0	x	x	x		
	mem, imm	1	0	0	0	0	0	0	W	mod	1	1	0	mem	15/23	3-6	u	0	0	x	x	x			
	acc, imm	0	0	1	0	0	1	0	W					4	2-3	u	0	0	x	x	x				
<b>Bit Manipulation Instructions</b>																									
INS	reg8, reg8	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	1	31-117/ 35-133	3						
		1	1	reg	reg																				
	reg8, imm8	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	1	31-117/ 35-133	4						
		1	1	0	0	0	reg																		
EXT	reg8, reg8	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1	26-55/ 34-59	3						
		1	1	reg	reg																				
	reg8, imm8	0	0	0	0	1	1	1	1	0	0	1	1	1	0	1	1	26-55/ 34-59	4						
		1	1	0	0	0	reg																		
TEST1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	W	3	3	u	0	0	u	u	x
		1	1	0	0	0	reg																		
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	W	7/11	3-5	u	0	0	u	u	x
		mod	0	0	0	mem																			
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0	W	4	4	u	0	0	u	u	x
		1	1	0	0	0	reg																		
	mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0	W	8/12	4-6	u	0	0	u	u	x
		mod	0	0	0	mem																			
SET1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	0	W	4	3						
		1	1	0	0	0	reg																		
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	0	W	10/18	3-5						
		mod	0	0	0	mem																			
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0	W	5	4						
		1	1	0	0	0	reg																		
	mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0	W	11/19	4-6						
		mod	0	0	0	mem																			
	CY	1	1	1	1	1	0	0	1									2	1			1			
	DIR	1	1	1	1	1	1	0	1									2	1						
CLR1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	W	5	3						
		1	1	0	0	0	reg																		
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	W	11/19	3-5						
		mod	0	0	0	mem																			
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	W	6	4						
		1	1	0	0	0	reg																		
	mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	W	12/20	4-6						
		mod	0	0	0	mem																			
	CY	1	1	1	1	1	0	0	0									2	1			0			
	DIR	1	1	1	1	1	1	0	0									2	1						

### Instruction Set (cont)

Mnemonic	Operands	Opcode										Clocks	Bytes	Flags											
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P	S	Z
<b>Bit Manipulation Instructions (cont)</b>																									
NOT1	reg, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	1	W	4	3						
		1	1	0	0	0		reg																	
	mem, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	1	1	W	10/18	3-5						
		mod	0	0	0	0	mem																		
	reg, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	W	5	4						
	1	1	0	0	0	reg																			
mem, imm3/4	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	W	11/19	4-6							
	mod	0	0	0	0	mem																			
CY	1	1	1	1	0	1	0	1								2	1				x				
<b>Shift/Rotate Instructions</b>																									
SHL	reg, 1	1	1	0	1	0	0	0	W	1	1	1	0	0	reg	2	2	u	x	x	x	x	x		
	mem, 1	1	1	0	1	0	0	0	W	mod	1	0	0	mem	13/21	2-4	u	x	x	x	x	x			
	reg, CL	1	1	0	1	0	0	1	W	1	1	1	0	0	reg	7+n	2	u	x	u	x	x	x		
	mem, CL	1	1	0	1	0	0	1	W	mod	1	0	0	mem	16/24+n	2-4	u	x	u	x	x	x			
	reg, imm8	1	1	0	0	0	0	0	W	1	1	1	0	0	reg	7+n	3	u	x	u	x	x	x		
	mem, imm8	1	1	0	0	0	0	0	W	mod	1	0	0	mem	16/24+n	3-5	u	x	u	x	x	x			
SHR	reg, 1	1	1	0	1	0	0	0	W	1	1	1	0	1	reg	2	2	u	x	x	x	x	x		
	mem, 1	1	1	0	1	0	0	0	W	mod	1	0	1	mem	13/21	2-4	u	x	x	x	x	x			
	reg, CL	1	1	0	1	0	0	1	W	1	1	1	0	1	reg	7+n	2	u	x	u	x	x	x		
	mem, CL	1	1	0	1	0	0	1	W	mod	1	0	1	mem	16/24+n	2-4	u	x	u	x	x	x			
	reg, imm8	1	1	0	0	0	0	0	W	1	1	1	0	1	reg	7+n	3	u	x	u	x	x	x		
	mem, imm8	1	1	0	0	0	0	0	W	mod	1	0	1	mem	16/24+n	3-5	u	x	u	x	x	x			
SHRA	reg, 1	1	1	0	1	0	0	0	W	1	1	1	1	1	reg	2	2	u	x	0	x	x	x		
	mem, 1	1	1	0	1	0	0	0	W	mod	1	1	1	mem	13/21	2-4	u	x	0	x	x	x			
	reg, CL	1	1	0	1	0	0	1	W	1	1	1	1	1	reg	7+n	2	u	x	u	x	x	x		
	mem, CL	1	1	0	1	0	0	1	W	mod	1	1	1	mem	16/24+n	2-4	u	x	u	x	x	x			
	reg, imm8	1	1	0	0	0	0	0	W	1	1	1	1	1	reg	7+n	3	u	x	u	x	x	x		
	mem, imm8	1	1	0	0	0	0	0	W	mod	1	1	1	mem	16/24+n	3-5	u	x	u	x	x	x			
ROL	reg, 1	1	1	0	1	0	0	0	W	1	1	0	0	0	reg	2	2				x	x			
	mem, 1	1	1	0	1	0	0	0	W	mod	0	0	0	mem	13/21	2-4				x	x				
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	0	0	reg	7+n	2				x	u			
	mem, CL	1	1	0	1	0	0	1	W	mod	0	0	0	mem	16/24+n	2-4				x	u				
	reg, imm	1	1	0	0	0	0	0	W	1	1	0	0	0	reg	7+n	3				x	u			
	mem, imm	1	1	0	0	0	0	0	W	mod	0	0	0	mem	16/24+n	3-5				x	u				
ROR	reg, 1	1	1	0	1	0	0	0	W	1	1	0	0	1	reg	2	2				x	u			
	mem, 1	1	1	0	1	0	0	0	W	mod	0	0	1	mem	13/21	2-4				x	x				
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	0	1	reg	7+n	2				x	u			
	mem, CL	1	1	0	1	0	0	1	W	mod	0	0	1	mem	16/24+n	2-4				x	u				
	reg, imm8	1	1	0	0	0	0	0	W	1	1	0	0	1	reg	7+n	3				x	u			
	mem, imm8	1	1	0	0	0	0	0	W	mod	0	0	1	mem	16/24+n	3-5				x	u				

n = number of shifts

**Instruction Set (cont)**

Mnemonic	Operands	Opcode										Clocks	Bytes	Flags										
		7	6	5	4	3	2	1	0	7	6			5	4	3	2	1	0	AC	CY	V	P	S
<b>Shift/Rotate Instructions (cont)</b>																								
ROLC	reg, 1	1	1	0	1	0	0	0	0	W	1	1	0	1	0	reg	2	2	x	x				
	mem, 1	1	1	0	1	0	0	0	0	W	mod	0	1	0	mem	13/21	2-4	x	x					
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	1	0	reg	7 + n	2	x	u					
	mem, CL	1	1	0	1	0	0	1	W	mod	0	1	0	mem	16/24 + n	2-4	x	u						
	reg, imm8	1	1	0	0	0	0	0	W	1	1	0	1	0	reg	7 + n	3	x	u					
	mem, imm8	1	1	0	0	0	0	0	W	mod	0	1	0	mem	16/24 + n	3-5	x	u						
RORC	reg, 1	1	1	0	1	0	0	0	W	1	1	0	1	1	reg	2	2	x	x					
	mem, 1	1	1	0	1	0	0	0	W	mod	0	1	1	mem	13/21	2-4	x	x						
	reg, CL	1	1	0	1	0	0	1	W	1	1	0	1	1	reg	7 + n	2	x	u					
	mem, CL	1	1	0	1	0	0	1	W	mod	0	1	1	mem	16/24 + n	2-4	x	u						
	reg, imm8	1	1	0	0	0	0	0	W	1	1	0	1	1	reg	7 + n	3	x	u					
	mem, imm8	1	1	0	0	0	0	0	W	mod	0	1	1	mem	16/24 + n	3-5	x	u						

n = number of shifts

**Stack Manipulation Instructions**

PUSH	mem16	1	1	1	1	1	1	1	1	mod	1	1	0	mem	15/23	2-4						
	reg16	0	1	0	1	0	reg	6/10	1													
	sr	0	0	0	sr	1	1	0	6/10	1												
	PSW	1	0	0	1	1	1	0	0	6/10	1											
	R	0	1	1	0	0	0	0	0	33/65	1											
	imm	0	1	1	0	1	0	S	0	5-6/9-10	2-3											
POP	mem16	1	0	0	0	1	1	1	1	mod	0	0	0	mem	16/24	2-4						
	reg16	0	1	0	1	1	reg	8/12	1													
	sr	0	0	0	sr	1	1	1	8/12	1												
	PSW	1	0	0	1	1	1	0	1	8/12	1	R	R	R	R	R	R					
	R	0	1	1	0	0	0	0	1	43/75	1											
PREPARE	imm16, imm8	1	1	0	0	1	0	0	0						*	4						

\*imm8 = 0 : 12  
imm8 > 1 : 17 + 8 (imm8 - 1)

DISPOSE		1	1	0	0	1	0	0	1						6/10	1						
---------	--	---	---	---	---	---	---	---	---	--	--	--	--	--	------	---	--	--	--	--	--	--

**Control Transfer Instructions**

CALL	near_proc	1	1	1	0	1	0	0	0						16/20	3							
	regptr	1	1	1	1	1	1	1	1	1	1	0	1	0	reg	14/18	1						
	memptr16	1	1	1	1	1	1	1	1	mod	0	1	0	mem	23/31	2-4							
	far_proc	1	0	0	1	1	0	1	0						21/29	5							
	memptr32	1	1	1	1	1	1	1	1	mod	0	1	1	mem	31/47	2-4							
RET		1	1	0	0	0	0	1	1						15/19	1							
	pop_value	1	1	0	0	0	0	1	0						20/24	3							
		1	1	0	0	1	0	1	1						21/29	1							
	pop_value	1	1	0	0	1	0	1	0						24/32	3							

### Instruction Set (cont)

Mnemonic	Operands	Opcode															Clocks	Bytes	Flags						
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1			0	AC	CY	V	P	S	Z
<b>Control Transfer Instructions (cont)</b>																									
BR	near_label	1	1	1	0	1	0	0	1									13	3						
	short_label	1	1	1	0	1	0	0	1									12	2						
	reg	1	1	1	1	1	1	1	1	1	1	1	0	0	reg			11	2						
	memptr16	1	1	1	1	1	1	1	1	mod	1	0	0	mem			19/23	2-4							
	far_label	1	1	1	0	1	0	1	0									15	5						
	memptr32	1	1	1	1	1	1	1	1	mod	1	0	1	mem			26/34	2-4							
BV	near_label	0	1	1	1	0	0	0	0								14/4	2							
BNV	near_label	0	1	1	1	0	0	0	1								14/4	2							
BC, BL	near_label	0	1	1	1	0	0	1	0								14/4	2							
BNC, BNL	near_label	0	1	1	1	0	0	1	1								14/4	2							
BE, BZ	near_label	0	1	1	1	0	1	0	0								14/4	2							
BNE, BNZ	near_label	0	1	1	1	0	1	0	1								14/4	2							
BNH	near_label	0	1	1	1	0	1	1	0								14/4	2							
BH	near_label	0	1	1	1	0	1	1	1								14/4	2							
BN	near_label	0	1	1	1	1	0	0	0								14/4	2							
BP	near_label	0	1	1	1	1	0	0	1								14/4	2							
BPE	near_label	0	1	1	1	1	0	1	0								14/4	2							
BPO	near_label	0	1	1	1	1	0	1	1								14/4	2							
BLT	near_label	0	1	1	1	1	1	0	0								14/4	2							
BGE	near_label	0	1	1	1	1	1	0	1								14/4	2							
BLE	near_label	0	1	1	1	1	1	1	0								14/4	2							
BGT	near_label	0	1	1	1	1	1	1	1								14/4	2							
DBNZNE	near_label	1	1	1	0	0	0	0	0								14/5	2							
DBNZE	near_label	1	1	1	0	0	0	0	1								14/5	2							
DBNZ	near_label	1	1	1	0	0	0	1	0								13/5	2							
BCWZ	near_label	1	1	1	0	0	0	1	1								13/5	2							
<b>Interrupt Instructions</b>																									
BRK	3	1	1	0	0	1	1	0	0								38/50	1							
	imm8	1	1	0	0	1	1	0	1								38/50	2							
BRKV	imm8	1	1	0	0	1	1	1	0								40/3	1							
RETI		1	1	0	0	1	1	1	1								27/39	1	R	R	R	R	R	R	
CHKIND	reg16, mem32	0	1	1	0	0	0	1	0	mod	reg	mem				17-25/ 52-55	2-4								
BRKEM	imm8	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	38/50	3							
<b>CPU Control Instructions</b>																									
HALT		1	1	1	1	0	1	0	0								2	1							
BUSLOCK		1	1	1	1	0	0	0	0								2	1							
FP01	fp_op	1	1	0	1	1	X	X	X	1	1	Y	Y	Z	Z	Z	2	2							
	fp_op, mem	1	1	0	1	1	X	X	X	mod	Y	Y	Y	mem		10/14	2-4								
FP02	fp_op	0	1	1	0	0	1	1	X	1	1	Y	Y	Z	Z	Z	2	2							
	fp_op, mem	0	1	1	0	0	1	1	X	mod	Y	Y	Y	mem		10/14	2-4								

**Instruction Set (cont)**

Mnemonic	Operand	Opcode															Clocks	Bytes	Flags														
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1			0	AC	CY	V	P	S	Z								
<b>CPU Control Instructions (cont)</b>																																	
POLL		1	0	0	1	1	0	1	1													2 + 5n	1										
		n = number of times POLL pin is sampled.																															
NOP		1	0	0	1	0	0	0	0													3	1										
DI		1	1	1	1	1	0	1	0													2	1										
EI		1	1	1	1	1	0	1	1													2	1										
DS0, DS1, PS, SS, (segment override prefixes)		0	0	1	seg	1	1	0														2	1										
<b>8080 Instruction Set Enhancements</b>																																	
RETEM		1	1	1	0	1	1	0	1	1	1	1	1	1	0	1							27/39	2	R	R	R	R	R	R			
CALLN	imm8	1	1	1	0	1	1	0	1	1	1	1	0	1	1	0	1							38/58	3								

### Description

The  $\mu$ PD70320 and  $\mu$ PD70322 (V25™) are high-performance, 16-bit, single-chip microcomputers with an 8-bit external data bus. They combine the instruction set of the  $\mu$ PD70108 (V20™) with many of the on-chip peripherals in NEC's 78000 series.

The  $\mu$ PD70320/322 processor has software compatibility with the V20 (and subsequently the 8086/8088), faster memory accessing, superior interrupt processing ability, and enhanced control of internal peripherals.

A variety of on-chip components, including 16K bytes of mask programmable ROM ( $\mu$ PD70322 only), 256 bytes of RAM, serial and parallel I/O, comparator port lines, timers, and a DMA controller make the  $\mu$ PD70320/322 a sophisticated microsystem.

Eight banks of registers are mapped into internal RAM below an additional 256-byte special-function register (SFR) area that is used to control on-chip peripherals. Internal RAM and the SFR area are together relocatable to anywhere in the 1M-byte address space. This maintains compatibility with existing system memory maps.

The  $\mu$ PD70322 is the mask ROM version, the  $\mu$ PD70320 is the ROM-less version, and the  $\mu$ PD70P322 is the EPROM version.

### Features

- Complete single-chip microcomputer
  - 16-bit ALU
  - 16K bytes of ROM ( $\mu$ PD70322)
  - 256 bytes of RAM
- Six-byte instruction prefetch queue
- 24 parallel I/O lines
- Eight analog comparator inputs with programmable threshold level
- Two independent DMA channels
- Two 16-bit timers
- Programmable time base counter
- Two full-duplex UARTs
- Programmable interrupt controller
  - Eight priority levels
  - Five external, 12 internal sources
  - Register bank (eight) context switching
  - Eight macro service function channels

V20 and V25 are trademarks of NEC Corporation.

- DRAM refresh pulse output
- Two standby modes
  - HALT
  - STOP
- Internal clock generator
  - 5-MHz maximum CPU clock frequency (0.4- $\mu$ s instruction cycle time)
  - 8-MHz maximum CPU clock frequency (0.25- $\mu$ s instruction cycle time)
- Programmable wait state generation
- Separate address/data bus interface
- CMOS technology

### Ordering Information

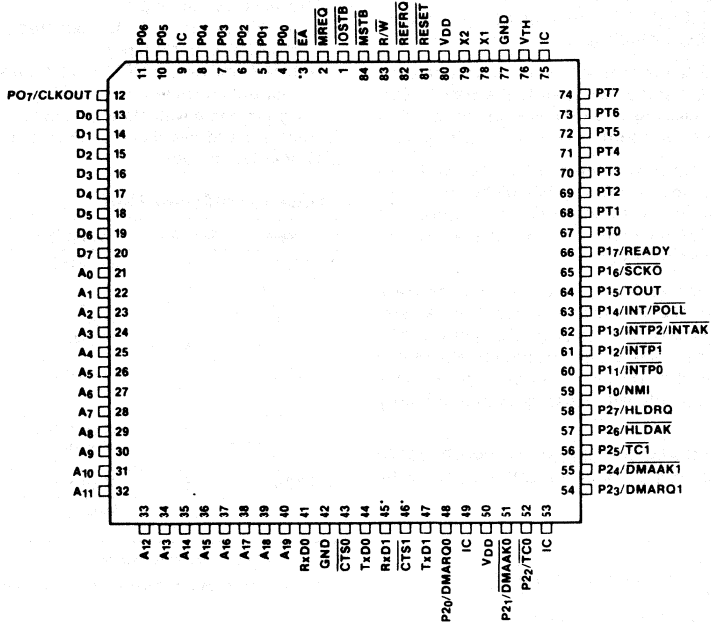
Part Number	Clock (MHz)	Package Type	ROM
$\mu$ PD70320L	5	84-pin PLCC	ROM-less
L-8	8		
GJ	5	94-pin plastic miniflat	
GJ-8	8		
$\mu$ PD70322L	5	84-pin PLCC	Mask ROM
L-8	8		
GJ	5	94-pin plastic miniflat	
GJ-8	8		
$\mu$ PD70P322L	5	84-pin PLCC	PROM (OTP)
K	5	84-pin LCC	EPROM (UV erasable)

PLCC = plastic leaded chip carrier

LCC = ceramic leadless chip carrier (with window)

**Pin Configurations**

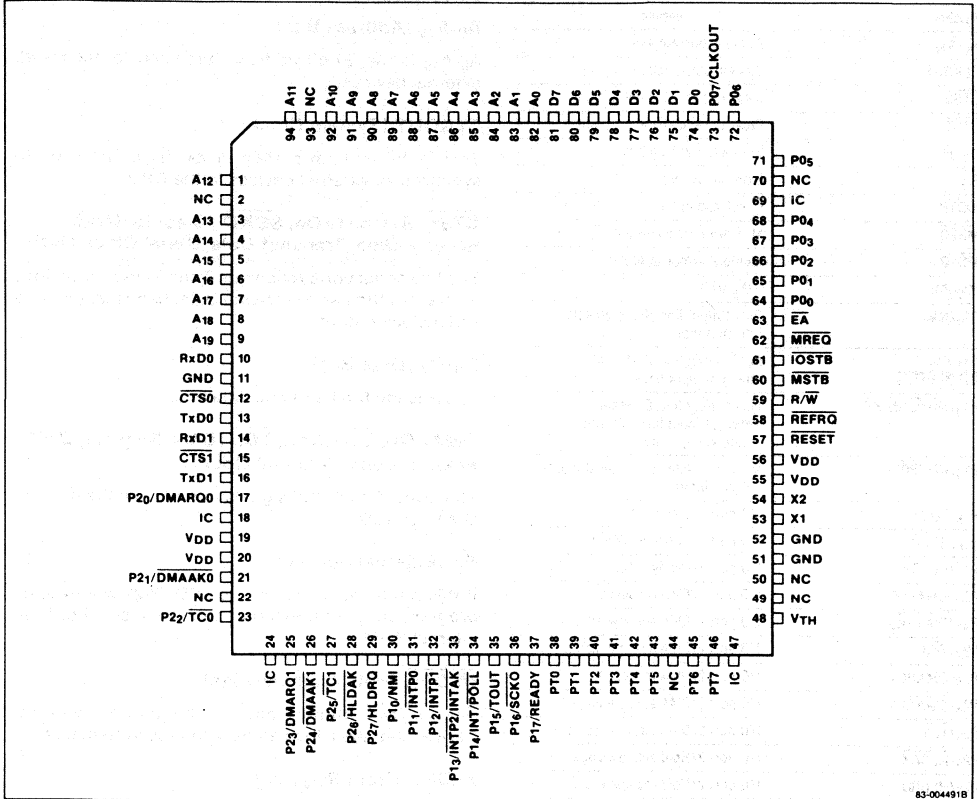
**84-Pin PLCC and LCC**



\*Pin functions for normal operation of the μPD70P322 are changed as follows for programming.

Pin No.	Symbol	Function
3	Vpp	Write power supply input
45	OE	Output enable signal input
46	CE	Chip enable signal input

### 94-Pin Plastic Flat Pack



63-004491B



**Pin Identification**

Symbol	Function
A <sub>0</sub> -A <sub>19</sub>	Address bus outputs
CLKOUT	System clock output
CTS0	Clear to send channel 0 input
CTS1	Clear to send channel 1 input
D <sub>0</sub> -D <sub>7</sub>	Bidirectional data bus
EA	External access
I <sub>OSTB</sub>	I/O strobe output
MREQ	Memory request output
MSTB	Memory strobe output
P <sub>00</sub> -P <sub>07</sub>	I/O port 0
P <sub>10</sub> /NMI	Port 1 input line/Nonmaskable interrupt input
P <sub>11</sub> -P <sub>12</sub> / INTP0-INTP1	Port 1 input lines/External interrupt input lines
P <sub>13</sub> /INTP2/INTAK	Port 1 input line/External interrupt input line/Interrupt acknowledge output
P <sub>14</sub> /INT/POLL	I/O port 1/Interrupt request input/ I/O poll input
P <sub>15</sub> /TOUT	I/O port 1/Timer out
P <sub>16</sub> /SCK0	I/O port 1/Serial clock out
P <sub>17</sub> /READY	I/O port 1/Ready input
P <sub>20</sub> /DMARQ0	I/O port 2/DMA request 0
P <sub>21</sub> /DMAAK0	I/O port 2/DMA acknowledge 0
P <sub>22</sub> /TC0	I/O port 2/DMA terminal count 0
P <sub>23</sub> /DMARQ1	I/O port 2/DMA request 1
P <sub>24</sub> /DMAAK1	I/O port 2/DMA acknowledge 1
P <sub>25</sub> /TC1	I/O port 2/DMA terminal count 1
P <sub>26</sub> /HLDAK	I/O port 2/Hold acknowledge output
P <sub>27</sub> /HLDRQ	I/O port 2/Hold request input
PT0-PT7	Comparator port input lines
REFRQ	Refresh pulse output
RESET	Reset input
RxD0	Serial receive data, channel 0 input
RxD1	Serial receive data, channel 1 input
R/W	Read/Write output
TxD0	Serial transmit data, channel 0 output
TxD1	Serial transmit data, channel 1 output
X1, X2	Crystal connection terminals
V <sub>DD</sub>	Positive power supply voltage
V <sub>TH</sub>	Threshold voltage input
GND	Ground
IC	Internal connection

**Pin Functions**

**A<sub>0</sub>-A<sub>19</sub> [Address Bus]**

A<sub>0</sub>-A<sub>19</sub> is the 20-bit address bus used to access all external devices.

**CLKOUT [System Clock]**

This is the internal system clock. It can be used to synchronize external devices to the CPU.

**CTS<sub>n</sub>, RxD<sub>n</sub>, TxD<sub>n</sub>, SCK0 [Clear to Send, Receive Data, Transmit Data, Serial Clock Out]**

The two serial ports (channels 0 and 1) use these lines for transmitting and receiving data, handshaking, and serial clock output.

**D<sub>0</sub>-D<sub>7</sub> [Data Bus]**

D<sub>0</sub>-D<sub>7</sub> is the 8-bit external data bus.

**DMARQ<sub>n</sub>, DMAAK<sub>n</sub>, TC<sub>n</sub> [DMA Request, DMA Acknowledge, Terminal Count]**

These are the control signals to and from the on-chip DMA controller.

**EA [External Access]**

If this pin is low on reset, the μPD70322 will execute program code from external memory instead of from internal ROM.

**HLDAK [Hold Acknowledge]**

The HLDAK output (active low) informs external devices that the CPU has released the system bus.

**HLDRQ [Hold Request]**

The HLDRQ input (active high) is used by external devices to request the CPU to release the system bus to an external bus master. A pin which has been set for an output port retains the data of the previous mode during the hold or standby mode. A control signal output pin specified for an output port continues the specified operation in the hold or HALT mode, and retains the data of the previous mode in the STOP mode. Pins P20 to P27 become an input port when RESET is input (high impedance output).

### **INT [Interrupt Request]**

INT is a maskable, active-low, vectored interrupt request input. After assertion, external hardware must provide the interrupt vector number.

### **INTAK [Interrupt Acknowledge]**

After INT is asserted, the CPU will respond with  $\overline{\text{INTAK}}$  (active low) to inform external devices that the interrupt request has been granted.

### **INTP0-INTP2 [External Interrupt]**

INTP0-INTP2 allow external devices to generate I/O requests (interrupts). Each can be programmed to be rising or falling edge triggered.

### **IOSTB [I/O Strobe]**

IOSTB is asserted during read and write operations to external I/O.

### **MREQ [Memory Request]**

MREQ (active low) informs external memory that the current bus cycle is a memory access bus cycle.

### **MSTB [Memory Strobe]**

MSTB (active low) is asserted during read and write operations to external memory.

### **NMI [Nonmaskable Interrupt]**

NMI cannot be masked through software and is typically used for emergency processing. Upon execution, the interrupt starting address is obtained from interrupt vector number 2. NMI can release the standby modes and can be programmed to be either rising or falling edge triggered.

### **P0<sub>0</sub>-P0<sub>7</sub> [Port 0]**

P0<sub>0</sub>-P0<sub>7</sub> are the lines of port 0, an 8-bit bidirectional parallel I/O port.

### **P1<sub>0</sub>-P1<sub>7</sub> [Port 1]**

The status of P1<sub>0</sub>-P1<sub>3</sub> can be read but these lines are always control functions. P1<sub>4</sub>-P1<sub>7</sub> are the remaining lines of parallel port 1, each line individually programmable as either an input, an output, or a control function.

### **P2<sub>0</sub>-P2<sub>7</sub> [Port 2]**

P2<sub>0</sub>-P2<sub>7</sub> are the lines of port 2, an 8-bit bidirectional I/O port. The lines can also be used as control signals for the on-chip DMA controller.

### **POLL [Poll]**

Upon execution of the POLL instruction, the CPU checks the status of this pin and, if low, program execution continues. If high, the CPU will check the level of the line every five clock cycles until it is low.  $\overline{\text{POLL}}$  can be used to synchronize program execution to external conditions.

### **PT0-PT7 [Comparator Port]**

PT0-PT7 are inputs to the analog comparator port.

### **READY [Ready]**

After READY is de-asserted low, the CPU will synchronize and insert at least two wait states into a read or write cycle to memory or I/O. This allows the processor to accommodate devices whose access times are longer than normal execution allows.

### **REFRQ [Refresh]**

This active-low output pulse can refresh nonstatic RAM. It can be programmed to meet system specifications and is internally synchronized so that refresh cycles do not interfere with normal CPU operation.

### **RESET [Reset]**

A low on  $\overline{\text{RESET}}$  resets the CPU and all on-chip peripherals.  $\overline{\text{RESET}}$  can also release the standby modes. After  $\overline{\text{RESET}}$  returns high, program execution begins from address FFFF0H.

### **R/W [Read/Write]**

An  $\overline{\text{R/W}}$  output allows external hardware to determine if the current operation is a read or write cycle. It can also control the direction of bidirectional buffers.

### **TOUT [Timer Out]**

TOUT is the square-wave output signal from the internal timer.

### **X1, X2 [Crystal Connections]**

The internal clock generator requires an external crystal across these terminals as shown in figure 35. By programming the PRC register, the system clock frequency can be selected as the oscillator frequency ( $f_{\text{OSC}}$ ) divided by 2, 4, or 8.

### **VDD [Power Supply]**

Two positive power supply pins ( $V_{\text{DD}}$ ) reduce internal noise.

**V<sub>TH</sub> [Threshold Voltage]**

The comparator port uses this pin to determine the analog reference point. The actual threshold to each comparator line is programmable to V<sub>TH</sub> × n/16, where n = 1 to 16.

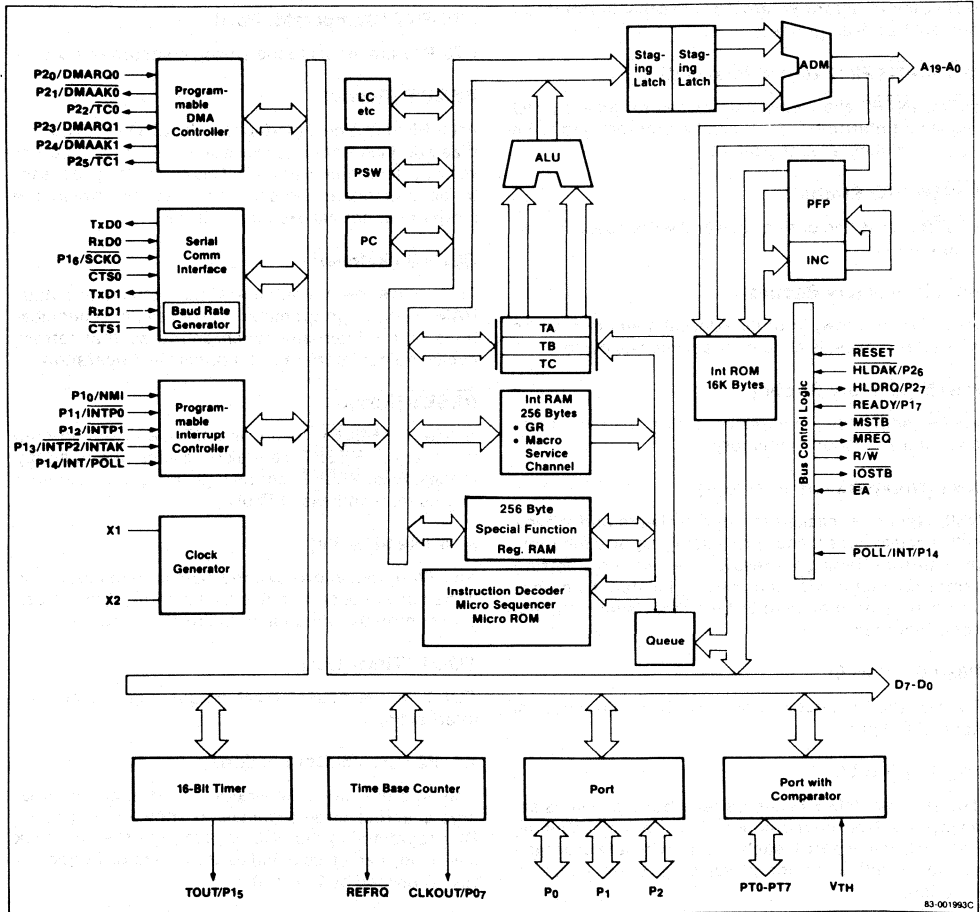
**GND**

Two ground connections reduce internal noise.

**IC [Internal Connection]**

The IC pins should be tied to V<sub>DD</sub>, with a pull-up resistor of 5 to 100 kΩ to each IC pin.

**Block Diagram**



83-001993C

### Functional Description

#### Architectural Enhancements

The following features enable the μPD70320/322 to perform high-speed execution of instructions:

- Dual data bus
- 16-/32-bit temporary registers/shifters (TA, TB, TA + TB)
- 16-bit loop counter (LC)
- Program counter (PC) and prefetch pointer (PPF)
- Internal ROM pass bus (μPD70322 only)

**Dual Data Bus.** The μPD70320/322 has two internal 16-bit data buses: the main data bus and a subdata bus. This reduces the processing time required for addition/subtraction and logical comparison instructions by one-third over single-bus systems. The dual data bus method allows two operands to be fetched simultaneously from the general-purpose registers and transferred to the ALU.

**16-/32-Bit Temporary Registers/Shifters.** The 16-bit temporary registers/shifters (TA, TB) allow high-speed execution of multiplication/division and shift/rotation instructions. By using the temporary registers/shifters, the μPD70320/322 can execute multiplication/division instructions about four times faster than with the microprogramming method.

**Loop Counter [LC].** The dedicated hardware loop counter counts the number of loops for string operations and the number of shifts performed for multiple bit shift/rotation instructions. The loop counter works with internal dedicated shifters to speed the processing of multiplication/division instructions.

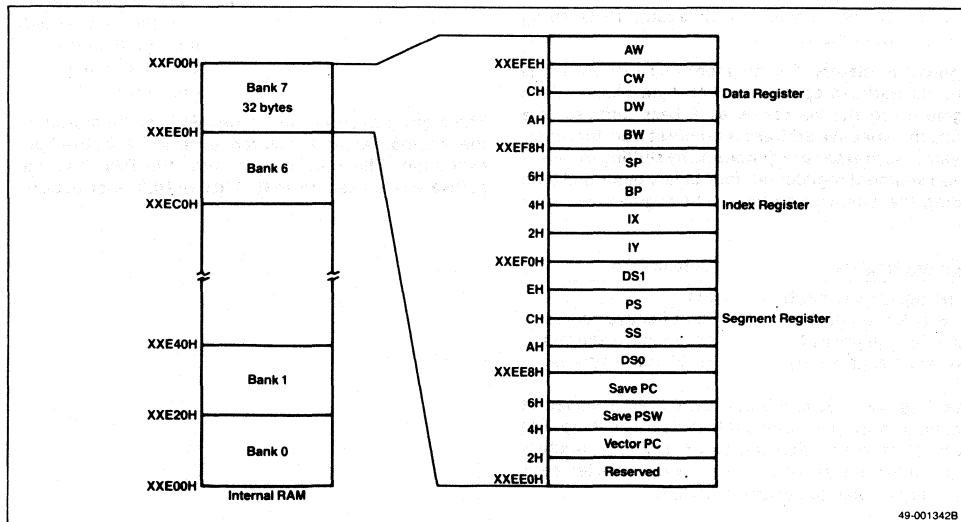
**Program Counter and Prefetch Pointer [PC and PFP].** The hardware PC addresses the memory location of the instruction to be executed next. The hardware PFP addresses the program memory location to be accessed next. Several clocks are saved for branch, call, return, and break instructions compared with processors having only one instruction pointer.

**Internal ROM Pass Bus.** The μPD70322 features a dedicated data bus between the internal ROM and the instruction pre-fetch queue. This allows internal ROM opcode fetches to be performed in a single clock cycle (200 ns at 5 MHz); it also makes it possible for opcode fetches to be performed while the external data bus is busy. This feature gives the V25 a 10-20% performance increase when executing from the internal ROM.

#### Register Set

Figure 1 shows the μPD70320/322 has eight banks of registers functionally mapped into internal RAM. Each bank contains general-purpose registers, pointer and index registers, segment registers, and save areas.

Figure 1. Register Banks in Internal RAM



49-001342B

## μPD70320/322 (V25)

**General-Purpose Registers [AW, BW, CW, DW].** There are four 16-bit general-purpose registers that can each serve as individual 16-bit registers or two independent 8-bit registers (AH, AL, BH, BL, CH, CL, DH, DL). The following instructions use the general-purpose registers for default:

- AW Word multiplication/division, word I/O, data conversion
- AL Byte multiplication/division, byte I/O, BCD rotation, data conversion, translation
- AH Byte multiplication/division
- BW Translation
- CW Loop control branch, repeat prefix
- CL Shift instructions, rotation instructions, BCD operations
- DW Word multiplication/division, indirect addressing I/O

**Pointers [SP, BP] and Index Registers [IX, IY].** These registers are used as 16-bit base pointers or index registers in based addressing, indexed addressing, and based indexed addressing. The registers are used as default registers under the following conditions:

- SP Stack operations
- IX Block transfer (source), BCD string operations
- IY Block transfer (destination), BCD string operations

**Segment Registers.** The segment registers divide the 1M-byte address space into 64K-byte blocks. Each segment register functions as a base address to a block; the effective address is an offset from that base. Physical addresses are generated by shifting the associated segment register left four binary digits and then adding the effective address. The segment registers are:

Segment Register	Default Offset
PS (Program segment)	PC
SS (Stack segment)	SP, Effective address
DS0 (Data segment-0)	IX, Effective address
DS1 (Data segment-1)	IY, Effective address

**Save Registers.** Save PC and Save PSW are used as save areas during register bank context switching. The Vector PC save location contains the effective address of the interrupt service routine when register bank switching is used to service interrupts.

**Program Counter [PC].** The PC is a 16-bit binary counter that contains the offset address from the program segment of the next instruction to be executed. It is incremented every time an instruction is received from the queue. It is loaded with a new location whenever a branch, call, return, break, or interrupt is executed.

**Program Status Word [PSW].** The PSW contains the following status and control flags.

PSW							
15							8
1	RB2	RB1	RB0	V	DIR	IE	BRK
7							0
S	Z	F1	AC	F0	P	BRKI	CY

### Status Flags

- V Overflow bit
- S Sign
- Z Zero
- AC Auxiliary carry
- P Parity
- CY Carry

### Control Flags

- DIR Direction of string processing
- IE Interrupt enable
- BRK Break (after every instruction)
- RBn Current register bank flags
- BRKI I/O trap enable (see software interrupts)
- F0, F1 General-purpose user flags (accessed through the Flag special function register)

The eight low-order bits of the PSW can be stored in the A4 register and restored by a MOV instruction execution. The only way to alter the RBn bits via software is to execute a RETRBI or RETI instruction.

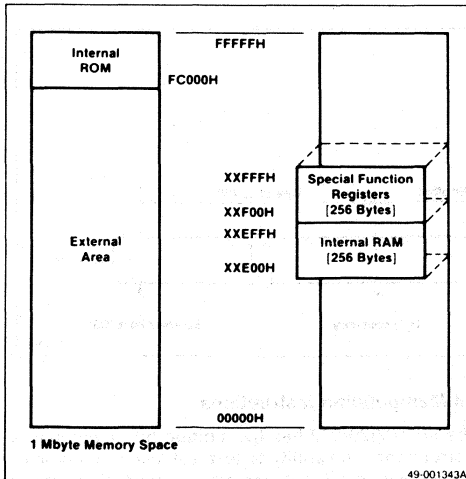
### Memory Map

The μPD70320/322 has a 20-bit address bus that can directly access 1M bytes of memory. Figure 2 shows that the 16K bytes of internal ROM (μPD70322 only) are located at the top of the address space from FC000H to FFFFFH.

Figure 2 shows the internal data area (IDA) is a 256-byte internal RAM area followed consecutively by a 256-byte special function register (SFR) area. All the data and control registers for on-chip peripherals and I/O are mapped into the SFR area and accessed as RAM. The IDA is dynamically relocatable in 4K-byte increments by changing the value in the internal data base (IDB) register. Whatever value is in this register will be assigned as the uppermost eight bits of the IDA address. The IDB register can be accessed from two different memory locations, FFFFFH and XXFFFH, where XX is the value in the IDB register.

On reset, the internal data base register is set to FFH which maps the IDA into the internal ROM space. However, since the μPD70322 has a separate bus to internal ROM, this does not present a problem. When these address spaces overlap, program code cannot be executed from the IDA and internal ROM locations cannot be accessed as data. You can select any of the eight possible register banks, which occupy the entire internal RAM space. Multiple register bank selection allows faster interrupt processing and facilitates multi-tasking.

Figure 2. Memory Map



In larger-scale systems where internal RAM is not required for data memory, the internal RAM can be removed completely from the address space and dedicated entirely to registers and control functions such as macro service and DMA channels. Clearing the RAMEN bit in the processor control register achieves this. When the RAMEN bit is cleared, internal RAM can only be accessed by register addressing or internal control processes. Many instructions are executed faster when the internal RAM is disabled.

### Instruction Types

The μPD70320/322 instruction set is fully compatible with the V20 native mode instruction set. The V20 instruction set is a superset of the μPD8086/8088 instruction set with different execution times and mnemonics.

The μPD70320/322 does not support the V20 8080 emulation mode. All of the instructions pertaining to this have been deleted from the μPD70320/322 instruction set.

### Enhanced Instructions

In addition to the μPD8086/88 instructions, the μPD70320/322 has the following enhanced instructions.

Instruction	Function
PUSH imm	Pushes immediate data onto stack
PUSH R	Pushes eight general registers onto stack
POP R	Pops eight general registers from stack
MUL imm	Executes 16-bit multiply of register or memory contents by immediate data
SHL imm8	Shifts/rotates register or memory by immediate value
SHR imm8	
SHRA imm8	
ROL imm8	
ROR imm8	
ROLC imm8	
RORC imm8	
CHKIND	Checks array index against designated boundaries
INM	Moves a string from an I/O port to memory
OUTM	Moves a string from memory to an I/O port
PREPARE	Allocates an area for a stack frame and copies previous frame pointers
DISPOSE	Frees the current stack frame on a procedure exit

Unique Instructions

The μPD70320/322 has the following unique instructions.

Instruction	Function
INS	Inserts bit field
EXT	Extracts bit field
ADD4S	Performs packed BCD string addition
SUB4S	Performs packed BCD string subtraction
CMP4S	Performs packed BCD string comparison
ROL4	Rotates BCD digit left
ROR4	Rotates BCD digit right
TEST1	Tests bit
SET1	Sets bit
CLR1	Clears bit
NOT1	Complements bit
BTCLR	Tests bit; if true, clear and branch
REPC	Repeat while carry set
REPNC	Repeat while carry cleared

Variable Length Bit Field Operation Instructions

Bit fields are a variable length data structure that can range in length from 1 to 16 bits. The μPD70320/322 supports two separate operations on bit fields: insertion (INS) and extraction (EXT). There are no restrictions

Bit field insertion copies the bit field of specified length from the AW register to the bit field addressed by DS1:IY:reg8 (8-bit general-purpose register). The bit field length can be located in any byte register or supplied as immediate data. Following execution, both the IY and reg8 are updated to point to the start of the next bit field.

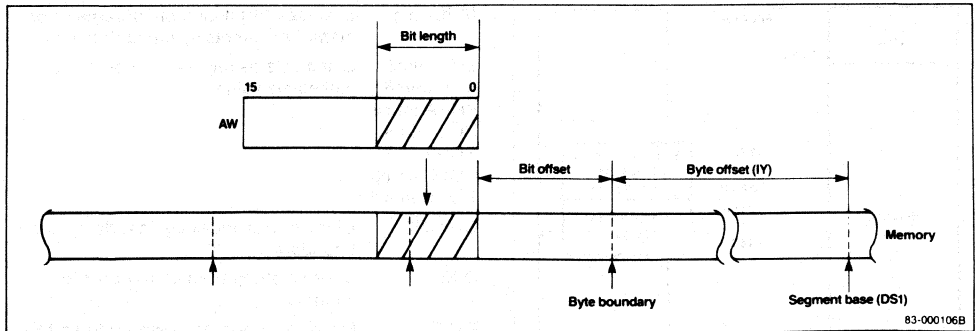
Bit field extraction copies the bit field of specified length from the bit field addressed by DS0:IX:reg8 to the AW register. If the length of the bit field is less than 16 bits, the bit field is right justified with a zero fill. The bit field length can be located in any byte register or supplied as immediate data. Following execution, both IX and reg8 are updated to point to the start of the next bit field.

Figures 3 and 4 show bit field insertion and bit field extraction.

Packed BCD Instructions

Packed BCD instructions process packed BCD data either as strings (ADD4S, SUB4S, CMP4S) or byte format operands (ROR4, ROL4). Packed BCD strings may be 1 to 254 digits in length. The two BCD rotation instructions perform rotation of a single BCD digit in the lower half of the AL register through the register or the memory operand.

Figure 3. Bit Field Insertion

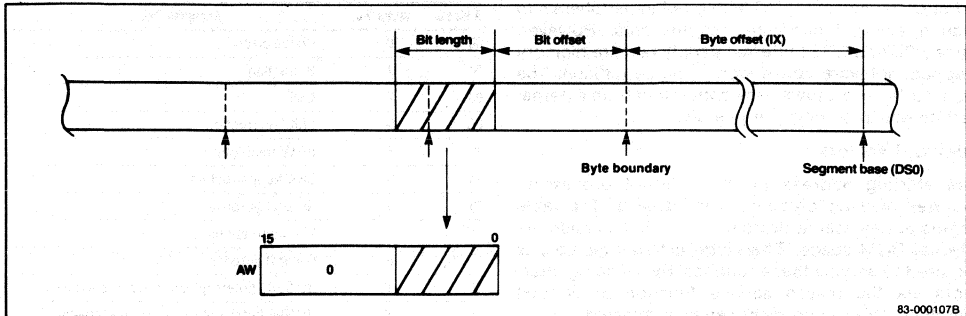


on the position of the bit field in memory. Separate segment, byte offset, and bit offset registers are used for insertion and extraction. Following the execution of these instructions, both the byte offset and bit offset are left pointing to the start of the next bit field, ready for the next operation. Bit field operation instructions are powerful and flexible and are therefore highly effective for graphics, high-level languages, and packing/unpacking applications.

Bit Manipulation Instructions

The μPD70320/322 has five unique bit manipulation instructions. The ability to test, set, clear, or complement a single bit in a register or memory operand increases code readability as well as performance over the logical operations traditionally used to manipulate bit data. This feature further enhances control over on-chip peripherals.

**Figure 4. Bit Field Extraction**



83-000107B

## Additional Instructions

Besides the V20 instruction set, the μPD70320/322 has the four additional instructions described in table 1.

**Table 1. Additional Instructions**

Instruction	Function
BTCLR var, imm8, short label	Bit test and if true, clear and branch; otherwise, no operation
STOP (no operand)	Power down instruction, stops oscillator
RETRBI (no operand)	Return from register bank context switch interrupt
FINT (no operand)	Finished interrupt. After completion of a hardware interrupt request, this instruction must be used to reset the current priority bit in the in-service priority register (ISPR).*
MOVSPA	Copies SS and SP of 'OLD' register bank into 'CURRENT' register bank after register bank switching. Helpful to keep stack continuation before and after register bank switching.
BRKCS reg 16	Switches register bank by the value of 'reg. 16', saves PC and PSW and loads the value of 'VPC' into PC. Useful to call suproutine by reg. bank switching (context switching), etc.
MOVSPB reg 16	Copies SS and SP of 'CURRENT' reg. bank into 'NEW' reg. bank before register bank switching. Helpful to keep stack continuation before and after register bank switching.
TSKSW reg 16	Saves PC and PSW, switches reg. bank by the value of 'reg 16' and loads PS and PSW with new values. Useful to switch tasks by reg. bank switching.

\* Do not use with NMI or INTR interrupt service routines.

## Repeat Prefixes

Two new repeat prefixes (REPC, REPNC) allow conditional block transfer instructions to use the state of the CY flag as the termination condition. This allows inequalities to be used when working on ordered data, thus increasing performance when searching and sorting algorithms.

## Interrupt Structure

The μPD70320/322 can service interrupts generated both by hardware and by software. Software interrupts are serviced through vectored interrupt processing. See table 2 for the various types of software interrupts.

**Table 2. Software Interrupts**

Interrupt	Description
Divide error	The CPU will trap if a divide error occurs as the result of a DIV or DIVU instruction.
Single step	The interrupt is generated after every instruction if the BRK bit in the PSW is set.
Overflow	By using the BRKV instruction, an interrupt can be generated as the result of an overflow.
Interrupt instructions	The BRK 3 and BRK imm8 instructions can generate interrupts.
Array bounds	The CHKIND instruction will generate an interrupt if specified array bounds have been exceeded.
Escape trap	The CPU will trap on an FP01.2 instruction to allow software to emulate the floating point processor.
I/O trap	If the I/O trap bit in the PSW is set, a trap will be generated on every IN or OUT instruction. Software can then provide an updated peripheral address. This feature allows software interchangeability between different systems.



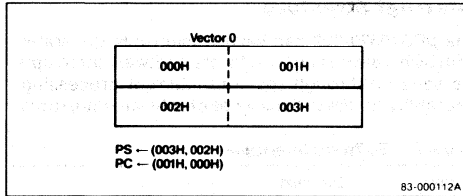
When executing software written for another system, it is better to implement I/O with on-chip peripherals to reduce external hardware requirements. However, since μPD70320/322 internal peripherals are memory mapped, software conversion could be difficult. The I/O trap feature allows easy conversion from external peripherals to on-chip peripherals.

**Interrupt Vectors**

The starting address of the interrupt processing routines may be obtained from table 3. The table begins at physical address 00H, which is outside the internal ROM space. Therefore, external memory is required to service these routines. By servicing interrupts via the macro service function or context switching, this requirement can be eliminated.

Each interrupt vector is four bytes wide. To service a vectored interrupt, the lower addressed word is transferred to the PC and the upper word to the PS. However, the byte order within each word is reversed so that the low-order bytes of the vector address become the most-significant bytes in the PC and PS. See figure 5.

**Figure 5. Interrupt Vector 0**



Execution of a vectored interrupt occurs as follows:

- (SP-1, SP-2) ← PSW
- (SP-3, SP-4) ← PS
- (SP-5, SP-6) ← PC
- SP ← SP-6
- IE ← 0, BRK ← 0
- PS ← vector high bytes
- PC ← vector low bytes

**Hardware Interrupt Configuration**

The V25 features a high-performance on-chip controller capable of controlling multiple processing for interrupts from up to 17 different sources (5 external, 12 internal). The interrupt configuration includes system interrupts that are functionally compatible with those of the V20/V30 and unique high-performance microcontroller interrupts.

**Table 3. Interrupt Vectors**

Address	Vector No.	Assigned Use
00	0	Divide error
04	1	Break flag
08	2	NMI
0C	3	BRK3 instruction
10	4	BRKV instruction
14	5	CHKIND instruction
18	6	General purpose
1C	7	FPO instructions
20-2C	8-11	General purpose
30	12	INTSER0 (Interrupt serial error, channel 0)
34	13	INTSR0 (Interrupt serial receive, channel 0)
38	14	INTST0 (Interrupt serial transmit, channel 0)
3C	15	General purpose
40	16	INTSER1 (Interrupt serial error, channel 1)
44	17	INTSR1 (Interrupt serial receive, channel 1)
48	18	INTST1 (Interrupt serial transmit, channel 1)
4C	19	I/O trap
50	20	INTD0 (Interrupt from DMA, channel 0)
54	21	INTD1 (Interrupt from DMA, channel 1)
58	22	General purpose
5C	23	General purpose
60	24	INTP0 (Interrupt from peripheral 0)
64	25	INTP1 (Interrupt from peripheral 1)
68	26	INTP2 (Interrupt from peripheral 2)
6C	27	General purpose
70	28	INTTU0 (Interrupt from timer unit 0)
74	29	INTTU1 (Interrupt from timer unit 1)
78	30	INTTU2 (Interrupt from timer unit 2)
7C	31	INTTB (Interrupt from time base counter)
080-3FF	32-255	General purpose

### Interrupt Sources

The 17 interrupt sources (table 4) are divided into groups for management by the interrupt controller. Using software, each of the groups can be assigned a priority from 0 (highest) to 7 (lowest). The priority of individual interrupts within a group is fixed in hardware. If interrupts from different groups occur simultaneously and the groups have the same assigned priority level, the priority followed will be as shown in the Simultaneous Priority column of table 4.

The ISPR is an 8-bit register; bits PR<sub>0</sub>-PR<sub>7</sub> correspond to the eight possible interrupt request priorities. The ISPR keeps track of the priority of the interrupt currently being serviced by setting the appropriate bit. The ISPR format is shown below.

PR <sub>7</sub>	PR <sub>6</sub>	PR <sub>5</sub>	PR <sub>4</sub>	PR <sub>3</sub>	PR <sub>2</sub>	PR <sub>1</sub>	PR <sub>0</sub>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

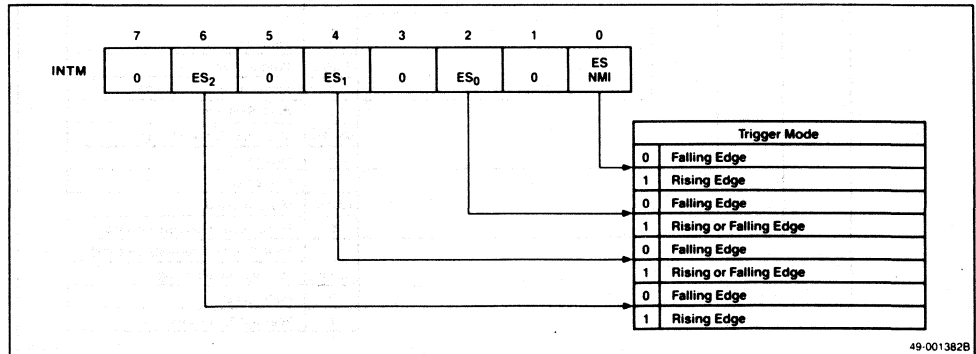
NMI and INTR are system-type external vectored interrupts. NMI is not maskable via software. INTR is maskable (IE bit in PSW) and requires that an external device provide the interrupt vector number. It allows expansion by the addition of an external interrupt controller (μPD71059).

NMI, INTPO, and INTP1 are edge-sensitive interrupt inputs. By selecting the appropriate bits in the interrupt mode register, these inputs can be programmed to be either rising or falling edge triggered. ES<sub>0</sub>-ES<sub>2</sub> correspond to INTPO-INTP2, respectively. See figure 6.

**Table 4. Interrupt Sources**

Group	Interrupt Source (Priority Within Group)			Simultaneous Priority
	1	2	3	
Non-maskable interrupt	NMI	—	—	0
Timer unit	INTTU0	INTTU1	INTTU2	1
DMA controller	INTD0	INTD1	—	2
External peripheral interrupt	INTP0	INTP1	INTP2	3
Serial channel 0	INTSER0	INTSR0	INTST0	4
Serial channel 1	INTSER1	INTSR1	INTST1	5
Time base counter	INTTB	—	—	6
Interrupt request	INTR	—	—	7

**Figure 6. Interrupt Mode Register (INTM)**



### Interrupt Processing Modes

Interrupts, with the exception of NMI, INT, and INTTB, have high-performance capability and can be processed in any of three modes: standard vector interrupt, register bank context switching, or macro service function. The processing mode for a given interrupt can be chosen by enabling the appropriate bits in the corresponding interrupt request control register. The format for all IRC registers is shown in figure 7.

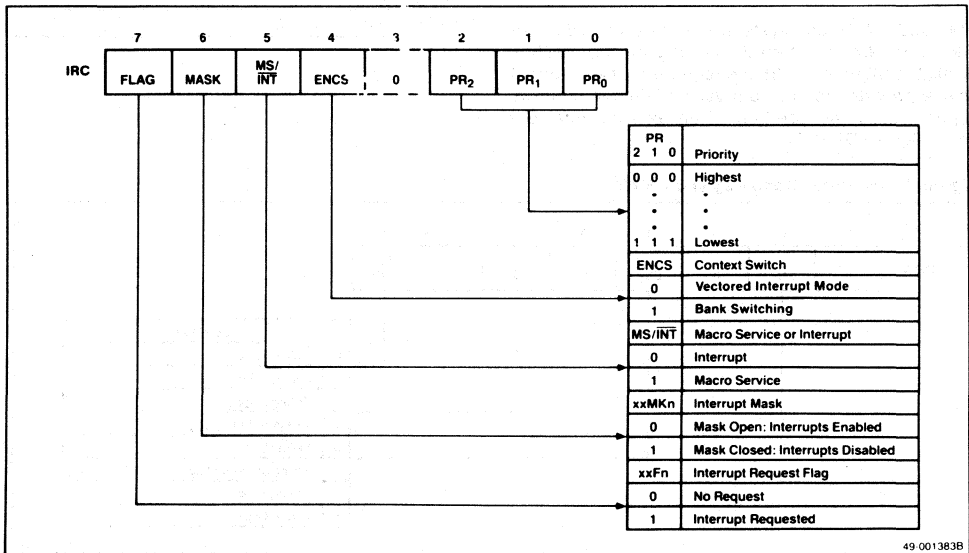
All interrupt processing routines other than those for NMI and INT must end with the execution of an FINT instruction. Otherwise, subsequently, only interrupts of a higher priority will be accepted.

In the vectored interrupt mode, the CPU returns to the vector location shown in table 3.

### Register Bank Switching

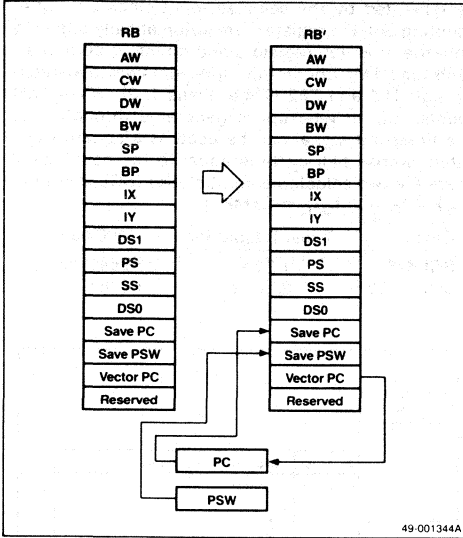
Register bank context switching allows interrupts to be processed rapidly by switching register banks. After an interrupt, the new register bank selected is that which has the same register bank number (0-7) as the priority of the interrupt to be serviced. The PC and PSW are automatically stored in the save areas of the new register bank and the address of the interrupt routine is loaded from the vector PC storage location in the new register bank. After interrupt processing, execution of the RETRBI (return from register bank interrupt) returns control to the former register bank and restores the former PC and PSW. Figures 8 and 9 show register bank context switching and register bank return.

Figure 7. Interrupt Request Control Registers (IRC)



49-001383B

**Figure 8. Register Bank Context Switching**



49-001344A

### Macro Service Function

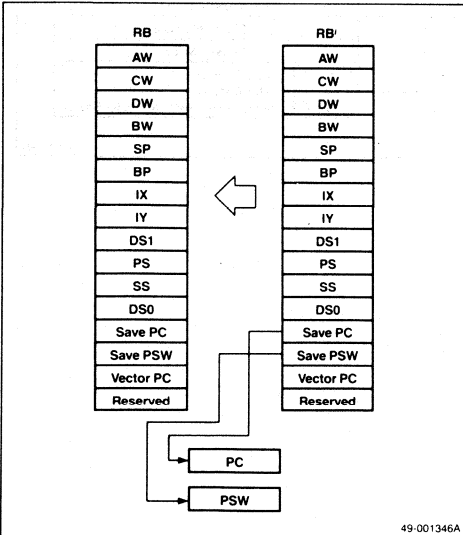
The macro service function (MSF) is a special micro-program that acts as an internal DMA controller between on-chip peripherals (special function registers, SFR) and memory. The MSF greatly reduces the software overhead and CPU time that other processors would require for register save processing, register returns, and other handling associated with interrupt processing.

If the MSF is selected for a particular interrupt, each time the request is received, a byte or word of data will be transferred between the SFR and memory without interrupting the CPU. Each time a request occurs, the macro service counter is decremented. When the counter reaches zero, an interrupt to the CPU is generated. The MSF also has a character search option. When selected, every byte transferred will be compared to an 8-bit search character and an interrupt will be generated if a match occurs or if the macro service counter counts out.

Like the NMI, INT and INTTB, the two DMA controller interrupts (INTD0, INTD1) do not have MSF capability.

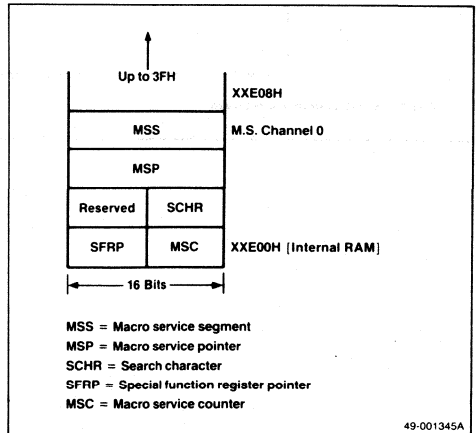
There are eight 8-byte macro service channels mapped into internal RAM from XXE00H to XXE3FH. Figure 10 shows the components of each channel.

**Figure 9. Register Bank Return**



49-001346A

**Figure 10. Macro Service Channels**



49-001345A

## μPD70320/322 (V25)

Setting the macro service mode for a given interrupt requires programming the corresponding macro service control register. Format for all MSC registers is shown in figure 11.

### On-Chip Peripherals

#### Timer Unit

The μPD70320/322 (figure 12) has two programmable 16-bit interval timers (TM0, TM1) on-chip, each with variable input clock frequencies. Each of the two 16-bit timer registers has an associated 16-bit modulus register (MD0, MD1). Timer 0 operates in the interval timer mode or one-shot mode; timer 1 has only the interval timer mode.

**Interval Timer Mode.** In this mode, TM0/TM1 are decremented by the selected input clock and, after counting out, the registers are automatically reloaded from the modulus registers and counting continues. Each time TM1 counts out, interrupts are generated through TF1 and TF2 (Timer Flags 1, 2). When TM0 counts out, an interrupt is generated through TF0. The timer-out signal can be used as a square-wave output whose half-cycle is equal to the count time. There are two selectable input clocks (SCLK: system clock =  $f_{OSC}/2$ ;  $f_{OSC} = 10$  MHz).

Clock	Timer Resolution	Full Count
SCLK/6	1.2 μs	78.643 ms
SCLK/128	25.6 μs	1.678 s

Figure 11. Macro Service Control Registers (MSC)

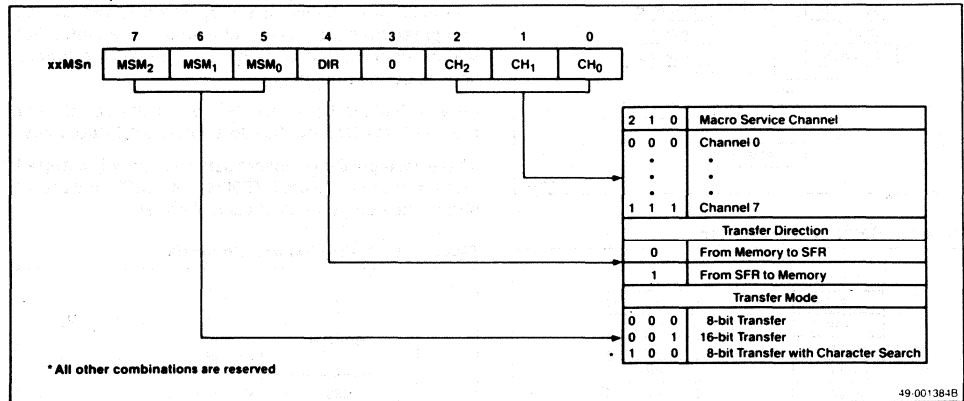
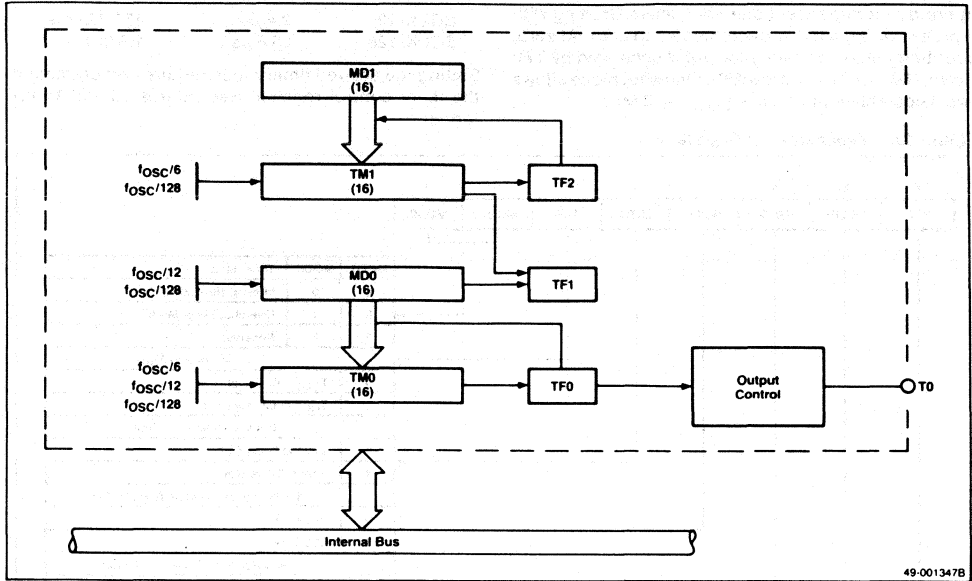


Figure 12. Timer Unit Block Diagram



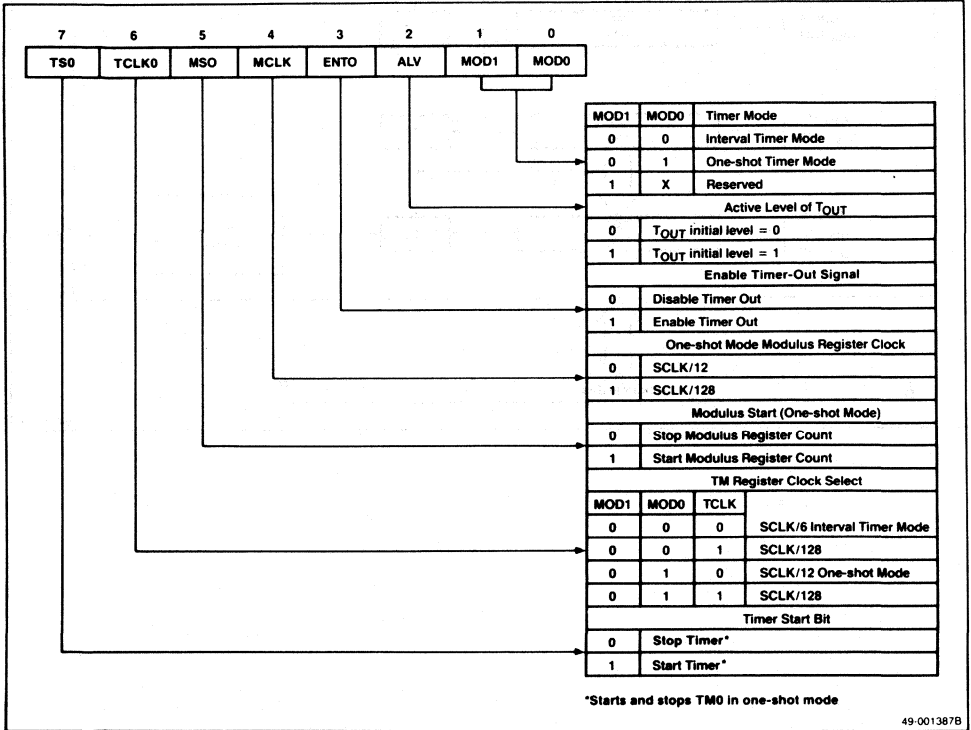
## μPD70320/322 (V25)

**One-Shot Mode.** In the one-shot mode, TM0 and MD0 operate as independent one-shot timers. Starting with a preset value, each is decremented to zero. At zero, counting ceases and an interrupt is generated by TF0 (from TM0) or TF1 (from MD0). One-shot mode allows two selectable input clocks ( $f_{OSC} = 10$  MHz).

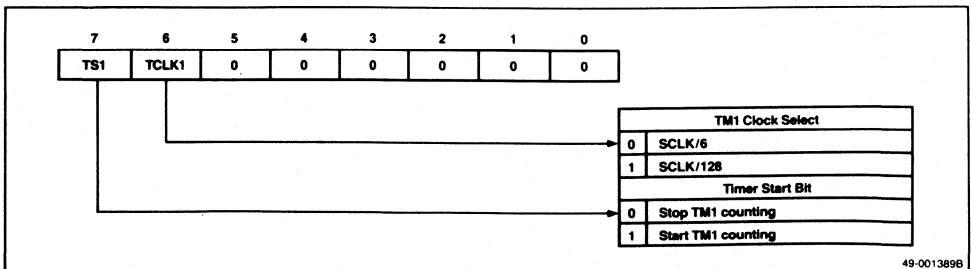
Clock	Timer Resolution	Full Count
SCLK/12	2.4 μs	157.283 ms
SCLK/128	25.6 μs	1.678 s

Setting the desired timer mode requires programming the timer control register. See figures 13 and 14 for format.

**Figure 13. Timer Control Register 0**



**Figure 14. Timer Control Register 1**



### Time Base Counter/Processor Control Register

The 20-bit free-running time base counter controls internal timing sequences and is available to the user as the source of periodic interrupts at lengthy intervals. One of four interrupt periods can be selected by programming the TB0 and TB1 bits in the processor control register (PRC). The TBC interrupt is unlike the others in that it is fixed as a level 7 vectored interrupt. Macro service and register bank switching cannot be used to service this interrupt. See figures 15 and 16.

The RAMEN bit in the PRC register allows the internal RAM to be removed from the memory address space to implement faster instruction execution.

The TBC (figure 17) uses the system clock as the input frequency. The system clock can be changed by programming the PCK0 and PCK1 bits in the processor control register (PRC). Reset initializes the system clock to  $f_{OSC}/8$  ( $f_{OSC}$  = external oscillator frequency).

Figure 17. Time Base Counter (TBC) Block Diagram

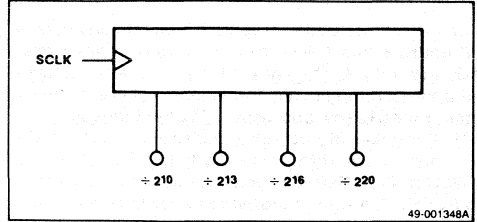


Figure 15. Time Base Interrupt Request Control Register

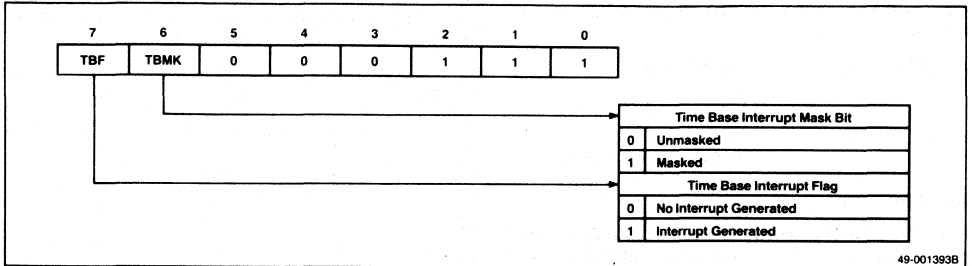
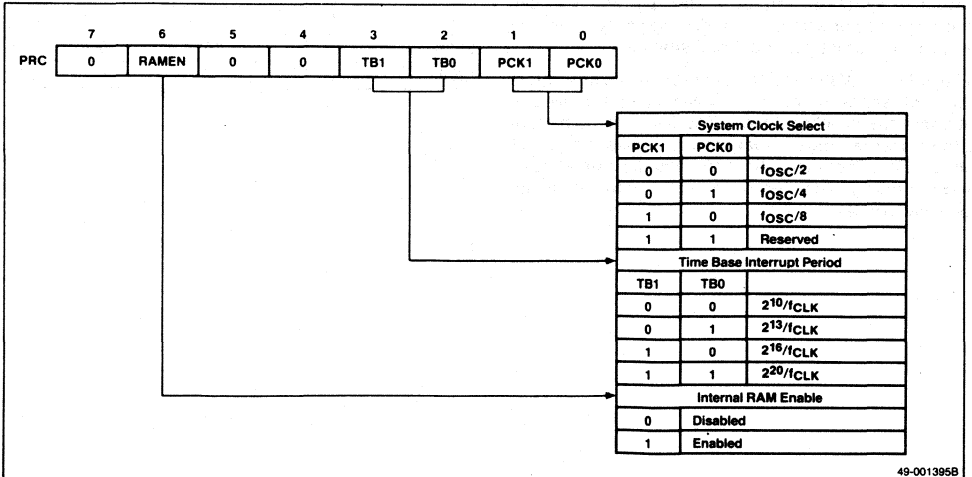


Figure 16. Processor Control Register (PRC)





## μPD70320/322 (V25)

### Refresh Controller

The μPD70320/322 has an on-chip refresh controller for dynamic and pseudostatic RAM mass storage memories. The refresh controller generates refresh addresses and refresh pulses. It inserts refresh cycles between the normal CPU bus cycles according to refresh specifications.

The refresh controller outputs a 9-bit refresh address on address bits A<sub>0</sub>-A<sub>8</sub> during the refresh bus cycle. Address bits A<sub>9</sub>-A<sub>19</sub> are all 1's. The 9-bit refresh address is automatically incremented at every refresh timing for 512 row addresses. The 8-bit refresh mode (RFM) register (figure 18) specifies the refresh operation and allows refresh during both CPU HALT and HOLD modes. Refresh cycles are automatically timed to REFRQ following read/write cycles to minimize the effect on system throughput.

The following shows the REFRQ pin level in relation to bits 4 (RFEN) and 7 (RFLV) of the refresh mode register.

RFEN	RFLV	REFRQ Level
0	0	0
0	1	1
1	0	0
1	1	Refresh pulse output

### Serial Interface

The μPD70320/322 has two full-duplex UARTs, channel 0 and channel 1. Each serial port channel has a transmit line (TxDn), a receive line (RxDn), and a clear to send (CTS<sub>n</sub>) input line for handshaking. Communication is synchronized by a start bit, and you can program the ports for even, odd, or no parity, character lengths of 7 or 8 bits, and 1 or 2 stop bits.

The μPD70320/322 has dedicated baud rate generators for each serial channel. This eliminates the need to obligate the on-chip timers. The baud rate generator allows a wide range of data transfer rates (up to 1.25 Mb/s). This includes all of the standard baud rates without being restricted by the value of the particular external crystal.

Each baud rate generator has an 8-bit baud rate generator (BRG<sub>n</sub>) data register, which functions as a prescaler to a programmable input clock selected by the serial communication control (SCC<sub>n</sub>) register. Together these must be set to generate a frequency equivalent to the desired baud rate.

The baud rate generator can be set to obtain the desired transmission rate according to the following formula:

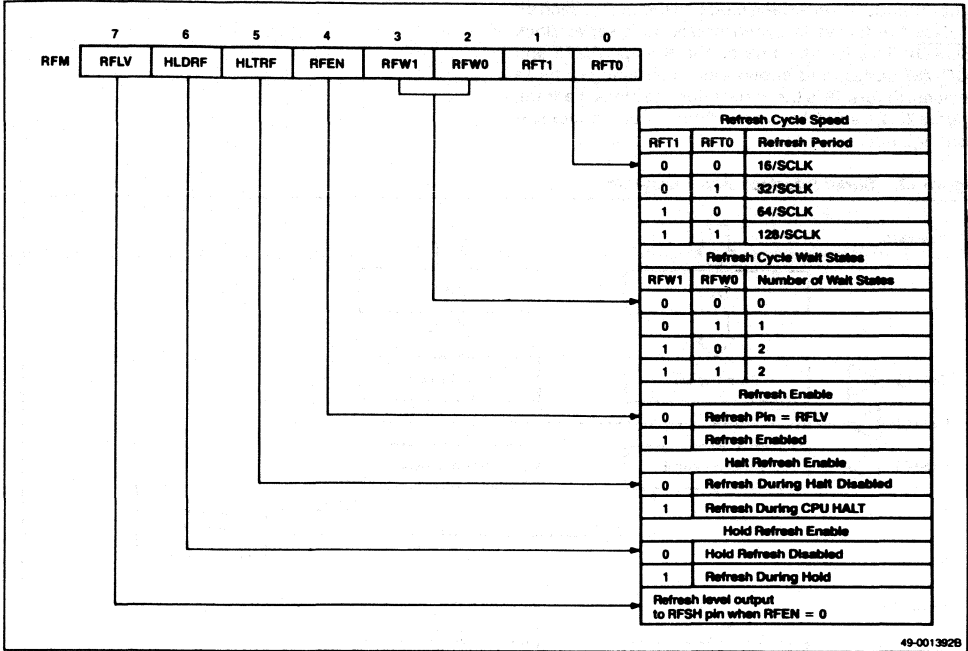
$$B \times G = \frac{SCLK \times 10^6}{2^{n+1}}$$

- where B = baud rate  
 G = baud rate generator register (BRG<sub>n</sub>) value  
 n = input clock specifications (n between 0 and 7)  
 SCLK = system clock frequency (MHz)

Based on the above expression, the following table shows the baud rate generator values used to obtain standard transmission rates when SCLK = 5 MHz.

Baud Rate	n	BRG <sub>n</sub> Value	Error (%)
110	7	178	0.25
150	7	130	0.16
300	6	130	0.16
600	5	130	0.16
1200	4	130	0.16
2400	3	130	0.16
4800	2	130	0.16
9600	1	130	0.16
19,200	0	130	0.16
38,400	0	65	0.16
1.25M	0	2	0

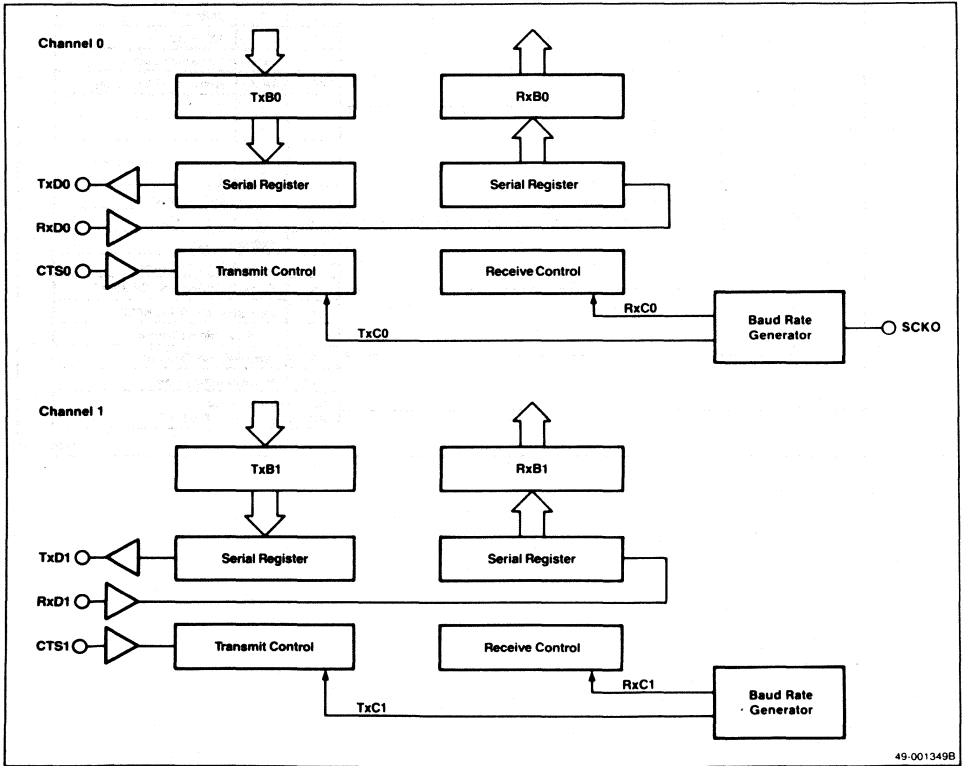
Figure 18. Refresh Mode Register (RFM)



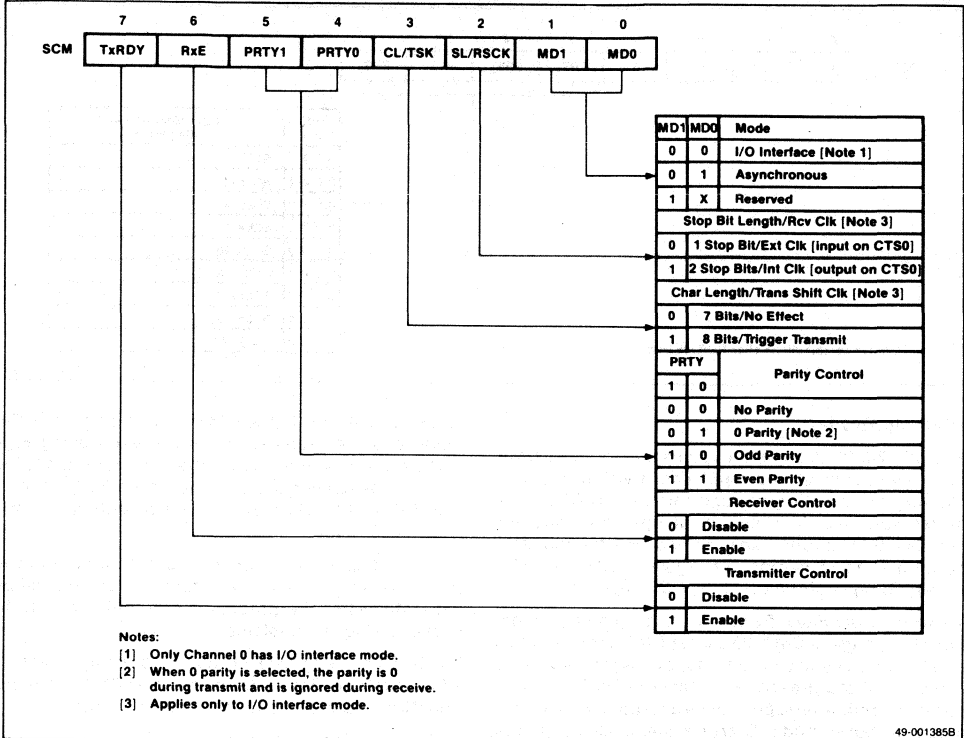
49-001382B

In addition to the asynchronous mode, channel 0 has a synchronous I/O interface mode. In this mode, each bit of data transferred is synchronized to a serial clock (SCK0). This is the same as the NEC μCOM75 and μCOM87 series, and allows easy interfacing to these devices. Figure 19 is the serial interface block diagram; figures 20, 21, and 22 show the three serial communication registers.

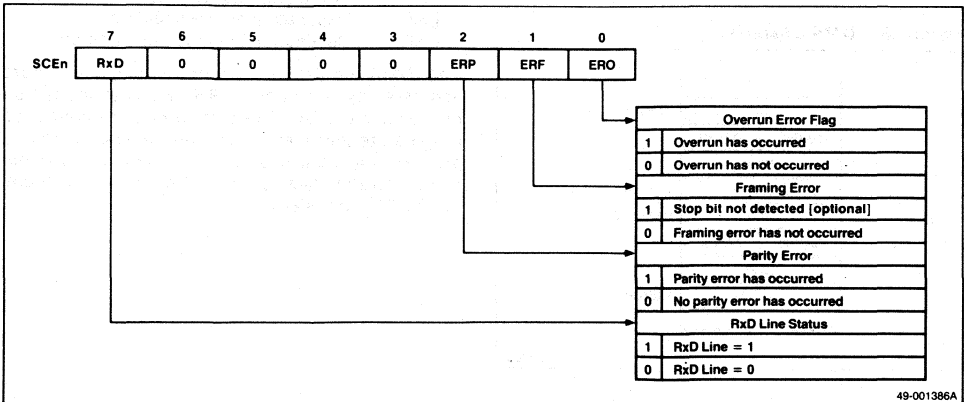
Figure 19. Serial Interface Block Diagram



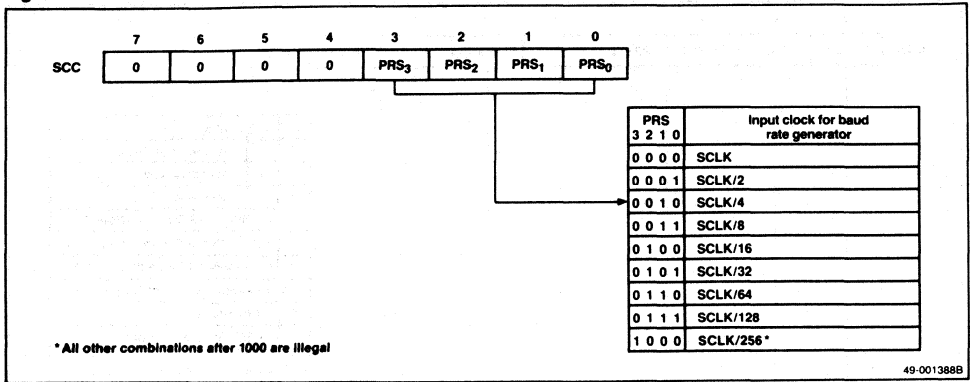
**Figure 20. Serial Communication Mode Register (SCM)**



**Figure 21. Serial Communication Error Registers (SCE)**



**Figure 22. Serial Communication Control Register (SCC)**



### DMA Controller

The μPD70320/322 has a two-channel, on-chip DMA controller. This allows rapid data transfer between memory and auxiliary storage devices. The DMA controller supports four modes of operation, two for memory-to-memory transfers and two for transfers between I/O and memory. See figures 23, 24, and 25 for a graphic representation of the DMA registers.

**Memory-to-Memory Transfers.** In a single-step mode, the falling edge of DMARQ causes DMA transfer cycles and CPU bus cycles to alternate as long as DMARQ is low or until the prescribed number of DMA transfers has occurred. Interrupts can be accepted while in this mode. In burst mode, DMA transfer cycles continue until the DMA terminal counter decrements to zero. Software can also initiate memory-to-memory transfers.

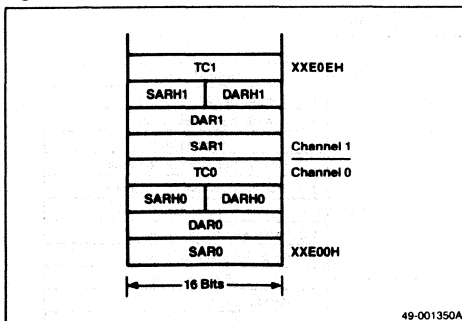
**Transfers Between I/O and Memory.** In single-transfer mode, one DMA transfer occurs after each falling edge of DMARQ. After the transfer, the bus is returned to the CPU. In demand release mode, the falling edge of DMARQ enables DMA cycles, which continue as long as DMARQ is low.

In all modes, the  $\overline{TC}$  (terminal count) output pin will pulse low and a DMA completion I/O request will be generated after the predetermined number of DMA cycles has been completed.

The bottom of internal RAM contains all the necessary address information for the designated DMA channels. The DMA channel mnemonics are as follows:

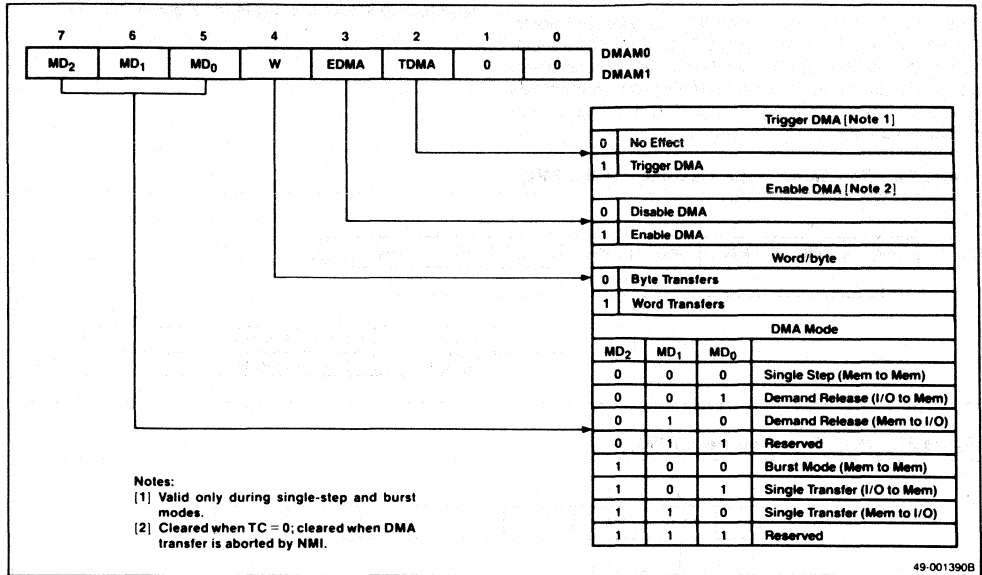
- TC Terminal counter
- SAR Source address register
- SARH Source address register high
- DAR Destination address register
- DARH Destination address register high

**Figure 23. DMA Channels**

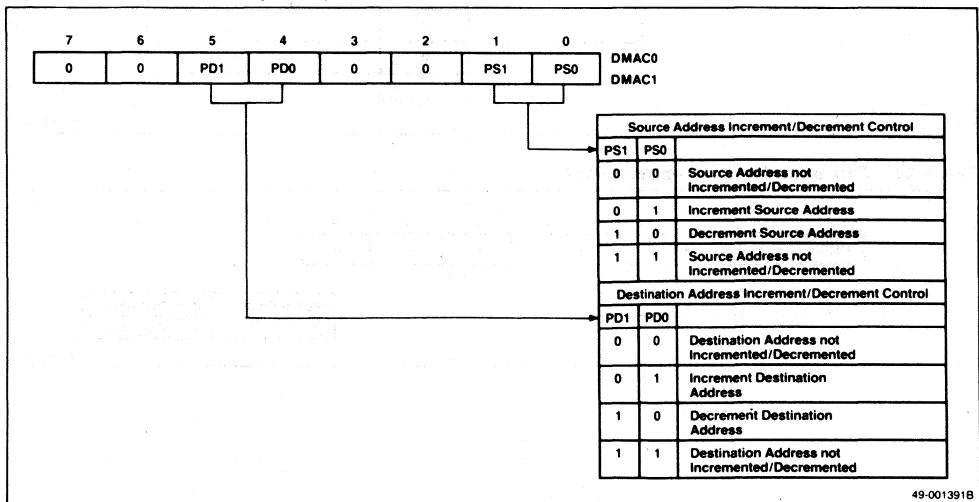


The DMA controller generates physical source addresses by offsetting SARH 12 bits to the left and then adding the SAR. The same procedure is also used to generate physical destination addresses. You can program the controller to increment or decrement source and/or destination addresses independently during DMA transfers.

**Figure 24. DMA Mode Registers (DMAM)**



**Figure 25. DMA Control Registers (DMAC)**

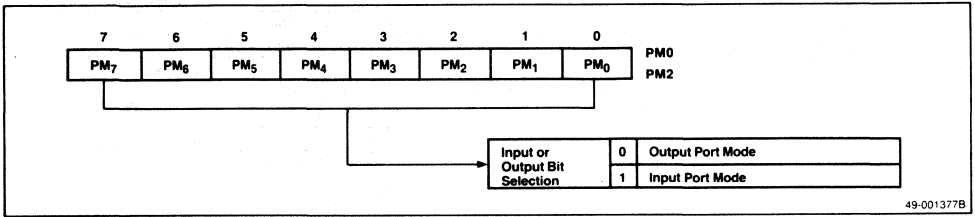


**Parallel Ports**

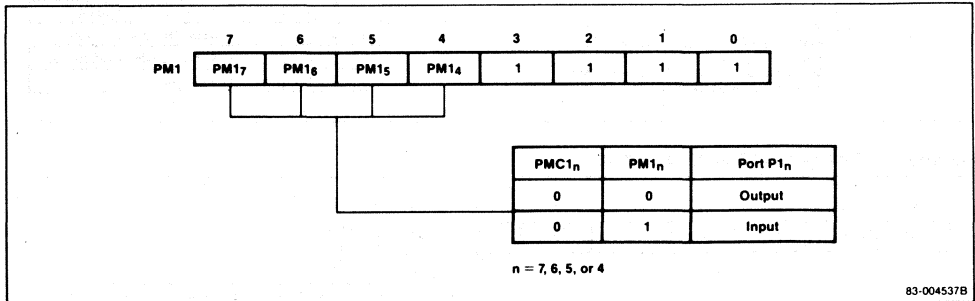
The μPD70320/322 has three 8-bit parallel I/O ports: P0, P1, and P2. Refer to figures 26 through 30. Special function register (SFR) locations can access these ports. The port lines are individually programmable as inputs or outputs. Many of the port lines have dual functions as port or control lines.

Use the associated port mode and port mode control registers to select the mode for a given I/O line.

**Figure 26. Port Mode Registers 0 and 2 (PM0, PM2)**



**Figure 27. Port Mode Register 1 (PM1)**



**Figure 28. Port Mode Control Register 0 (PMC0)**

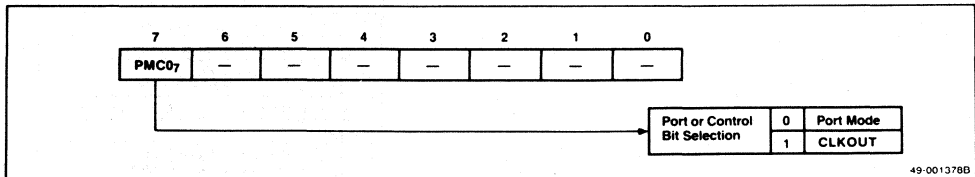
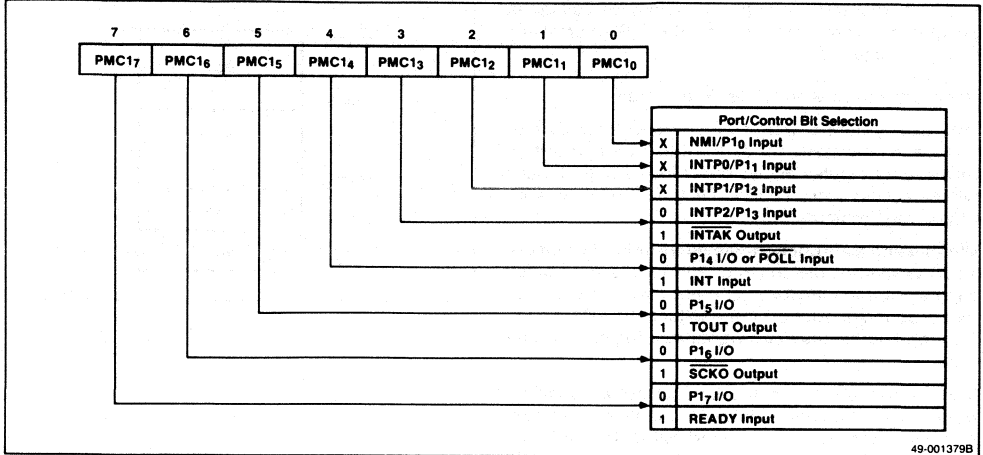
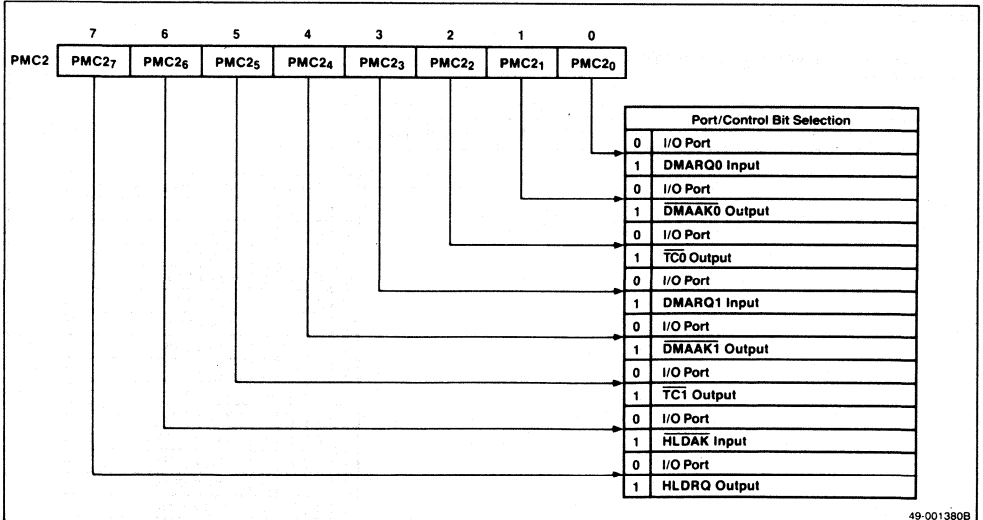


Figure 29. Port Mode Control Register 1 (PMC1)



49-001379B

Figure 30. Port Mode Control Register 2 (PMC2)



49-001380B



## μPD70320/322 (V25)

The analog comparator port (PT) compares each input line to a reference voltage. The reference voltage is programmable to be the  $V_{TH}$  input  $x n/16$ , where  $n = 1$  to 16. See figure 31.

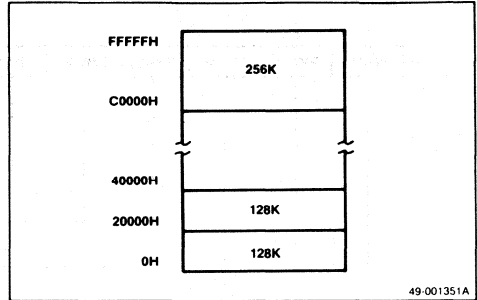
### Programmable Wait State Generation

You can generate wait states internally to further reduce the necessity for external hardware. Insertion of these wait states allows direct interface to devices whose access times cannot meet the CPU read/write timing requirements.

When using this function, the entire 1M-byte memory address space is divided into 128K-blocks. Each block can be programmed for zero, one, or two wait states, or two plus those added by the external READY signal. The top two blocks are programmed together as one unit.

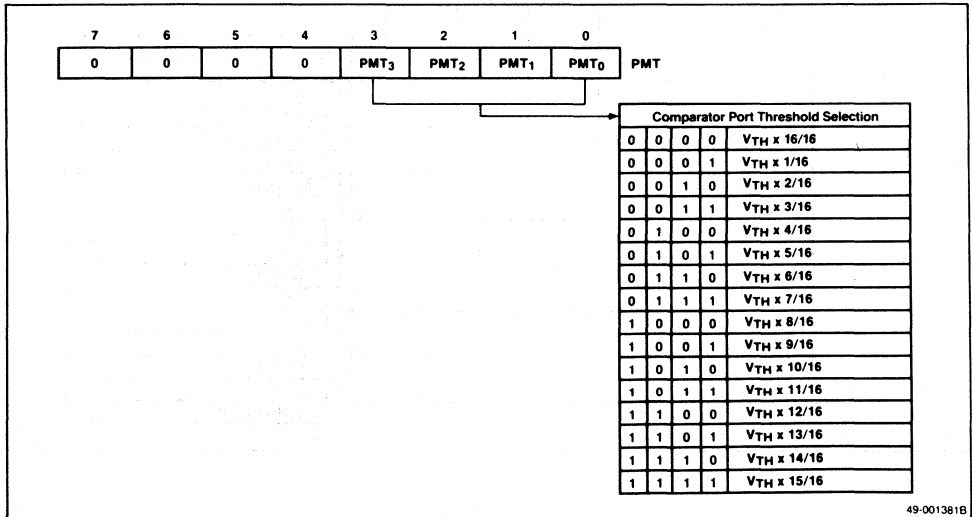
The appropriate bits in the wait control word (WTC) control wait state generation. Programming the upper two bits in the wait control word will set the wait state conditions for the entire I/O address space. Figure 32 shows the memory map for programmable wait state generation; see figure 33 for a graphic representation of the wait control word.

Figure 32. Programmable Wait State Generation



49-001351A

Figure 31. Port Mode Register T (PMT)



49-001381B

### Standby Modes

The two low-power standby modes are HALT and STOP. Software causes the processor to enter either mode.

#### HALT Mode.

In the HALT mode, the processor is inactive and the chip consumes much less power than when operational. The external oscillator remains functional and all peripherals are active. Internal status and output port line conditions are maintained. Any unmasked interrupt can release this mode. In the EI state, interrupts subsequently will be processed in vector mode. In the DI state, program execution is restarted with the instruction following the HALT instruction.

#### STOP Mode.

The STOP mode allows the largest power reduction while maintaining RAM. The oscillator is stopped, halting all internal peripherals. Internal status is maintained. Only a reset or NMI can release this mode.

A standby flag in the SFR area is reset by rises in the supply voltage. The flag is set when its status is read. Its status is maintained during normal operation and standby. The STBC register (figure 34) is not initialized by RESET. Use the standby flag to determine whether program execution is returning from standby or from a cold start.

Figure 33. Wait Control Word

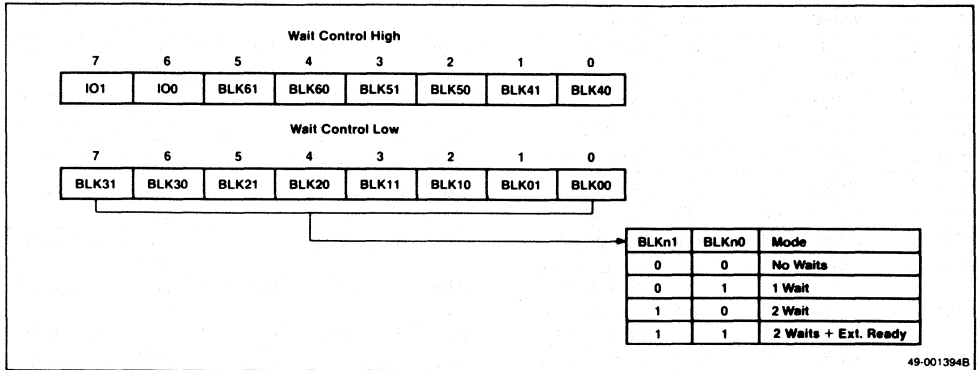
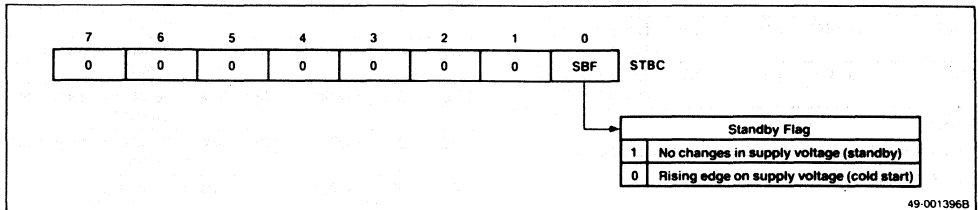


Figure 34. Standby Register



## μPD70320/322 (V25)

### Special Function Registers

Table 5 shows the special function register mnemonic, type, address, reset value, and function. The 8 high-order bits of each address (xx) are specified by the IDB register.

SFR area addresses not listed in table 5 are reserved. If read, the contents of these addresses are undefined, and any write operation will be meaningless.

**Table 5. Special Function Registers**

Name	Byte/ Word	Address	Reset Value (Note 2)	R/W (Note 1)	Function
PD	B	xxF00H		R/W	Port 0
PM0	B	xxF01H	FFH	W	Port mode 0
PMCD	B	xxF02H	00H	W	Port mode control 0
P1	B	xxF08H		R/W	Port 1
PM1	B	xxF09H	FFH	W	Port mode 1
PMCD1	B	xxF0AH	00H	W	Port mode control 1
P2	B	xxF10H		R/W	Port 2
PM2	B	xxF11H	FFH	W	Port mode 2
PMCD2	B	xxF12H	00H	W	Port mode control 2
PT	B	xxF38H		R	Port T
PMT	B	xxF3BH	00H	R/W	Port mode T
INTM	B	xxF40H	00H	R/W	Interrupt mode
EMSO	B	xxF44H		R/W	External interrupt macro service 0
EMS1	B	xxF45H		R/W	External interrupt macro service 1
EMS2	B	xxF46H		R/W	External interrupt macro service 2
EXIC0	B	xxF4CH	47H	R/W	External interrupt control 0
EXIC1	B	xxF4DH	47H	R/W	External interrupt control 1
EXIC2	B	xxF4EH	47H	R/W	External interrupt control 2

**Notes:**

- (1) Indicates if register is available for read/write operations.
- (2) Reset values not specified are undefined.

Name	Byte/ Word	Address	Reset Value (Note 2)	R/W (Note 1)	Function
RXB0	B	xxF60H		R	Receive buffer 0
TXB0	B	xxF62H		W	Transfer buffer 0
SRMS0	B	xxF65H		R/W	Serial receive macro service 0
STMS1	B	xxF66H		R/W	Serial transmit macro service 1
SCM0	B	xxF68H	00H	R/W	Serial communication mode 0
SCC0	B	xxF69H	00H	R/W	Serial communication control 0
BRG0	B	xxF6AH	00H	R/W	Baud rate generator 0
SCE0	B	xxF6BH	00H	R	Serial communication error 0
SEIC0	B	xxF6CH	47H	R/W	Serial error interrupt control 0
SRIC0	B	xxF6DH	47H	R/W	Serial receive interrupt control 0
STIC0	B	xxF6EH	47H	R/W	Serial transmit interrupt control 0
RXB1	B	xxF70H		R	Receive buffer 1
TXB1	B	xxF72H		W	Transmit buffer 1
SRMS1	B	xxF75H		R/W	Serial receive macro service 1
STMS1	B	xxF76H		R/W	Serial transmit macro service 1
SCM1	B	xxF78H	00H	R/W	Serial communication mode 1
SCC1	B	xxF79H	00H	R/W	Serial communication control 1
BRG1	B	xxF7AH	00H	R/W	Baud rate generator register 1
SCE1	B	xxF7BH	00H	R	Serial communication error 1
SEIC1	B	xxF7CH	47H	R/W	Serial error interrupt control 1
SRIC1	B	xxF7DH	47H	R/W	Serial receive interrupt control 1
STIC1	B	xxF7EH	47H	R/W	Serial transmit interrupt control 1
TMO	W	xxF80H		R/W	Timer register 0
TMOL	B	xxF80H		R/W	Timer register 0 low
TMOH	B	xxF81H		R/W	Timer register 0 high
MDO	W	xxF82H		R/W	Modulo register 0

**Table 5. Special Function Registers (cont)**

Name	Byte/ Word	Address	Reset Value (Note 2)	R/W (Note 1)	Function
MD0L	B	xxF82H		R/W	Modulo register 0 low
MD0H	B	xxF83H		R/W	Modulo register 0 high
TM1	W	xxF88H		R/W	Timer register 1
TM1L	B	xxF88H		R/W	Timer register 1 low
TM1H	B	xxF89H		R/W	Timer register 1 high
MD1	W	xxF8AH		R/W	Modulo register 1
MD1L	B	xxF8AH		R/W	Modulo register 1 low
MD1H	B	xxF8BH		R/W	Modulo register 1 high
TMC0	B	xxF90H	00H	R/W	Timer control 0
TMC1	B	xxF91H	00H	R/W	Timer control 1
TMMS0	B	xxF94H		R/W	Timer macro service 0
TMMS1	B	xxF95H		R/W	Timer macro service 1
TMMS2	B	xxF96H		R/W	Timer macro service 2
TMIC0	B	xxF9CH	47H	R/W	Timer interrupt control 0
TMIC1	B	xxF9DH	47H	R/W	Timer interrupt control 1
TMIC2	B	xxF9EH	47H	R/W	Timer interrupt control 2
DMAC0	B	xxFA0H		R/W	DMA control 0
DMAM0	B	xxFA1H	00H	R/W	DMA mode 0
DMAC1	B	xxFA2H		R/W	DMA control 1
DMAM1	B	xxFA3H	00H	R/W	DMA mode 1
DIC0	B	xxFACH	47H	R/W	DMA interrupt control 0
DIC1	B	xxFADH	47H	R/W	DMA interrupt control 1
RFM	B	xxFE1H	10H	R/W	Refresh mode
TBIC	B	xxFECH	47H	R/W	Time base interrupt control
WTC	W	xxFE8H	FFH	R/W	Wait control
WTCL	B	xxFE8H	FFH	R/W	Wait control low
WTCH	B	xxFE9H	FFH	R/W	Wait control high
PSWL	B	xxFEAH	00H	R/W	Flag register
PRC	B	xxFEBH	4EH	R/W	Processor control
SB	B	xxFE0H		R/W	Standby control
IDB	B	FFFFFH	FFH	R/W	Internal data area base

### Absolute Maximum Ratings

T<sub>A</sub> = 25°C

Supply voltage, V <sub>DD</sub>	-0.5 to +7.0 V
Input voltage, V <sub>I</sub>	-0.5 to V <sub>DD</sub> + 0.5 V (≤ +7.0 V)
Output voltage, V <sub>O</sub>	-0.5 to V <sub>DD</sub> + 0.5 V (≤ +7.0 V)
Threshold voltage, V <sub>TH</sub>	-0.5 to V <sub>DD</sub> + 0.5 V (≤ +7.0 V)
Output current, low; I <sub>OL</sub>	
Each output pin	4.0 mA
Total	50 mA
Output current, high; I <sub>OH</sub>	
Each output pin	-2.0 mA
Total	-20 mA
Operating temperature range, T <sub>OP</sub>	-40 to +85°C
Storage temperature range, T <sub>STG</sub>	-65 to +150°C

**Comment:** Exposure to Absolute Maximum Ratings for extended periods may affect device reliability; exceeding the ratings could cause permanent damage.

### DC Characteristics

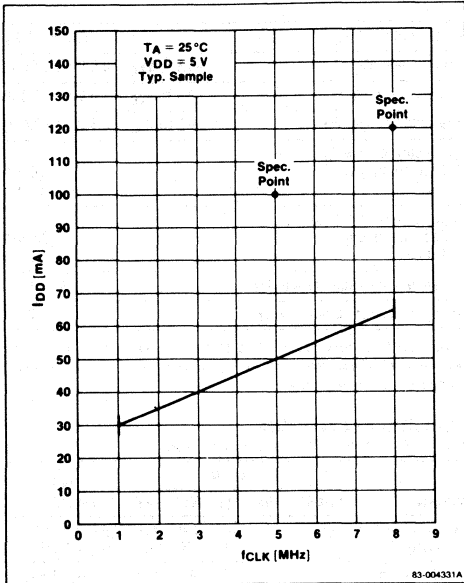
V<sub>DD</sub> = +5 V ±10%; T<sub>A</sub> = -10 to +70°C (Note 1)

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Supply current, operating	I <sub>DD1</sub>	50	100	120	mA	f <sub>CLK</sub> = 5 MHz
		65	120			
Supply current, HALT mode	I <sub>DD2</sub>	20	40	50	mA	f <sub>CLK</sub> = 5 MHz
		25	50			
Supply current, STOP mode	I <sub>DD3</sub>	10	30		μA	
Threshold current	I <sub>TH</sub>	0.5	1.0		mA	V <sub>TH</sub> = 0 to V <sub>DD</sub>
Input voltage, low	V <sub>IL</sub>	0	0.8		V	
Input voltage, high	V <sub>IH1</sub>	2.2		V <sub>DD</sub>	V	All inputs except RESET, P1 <sub>0</sub> /NMI, X1, X2
	V <sub>IH2</sub>	0.8 x V <sub>DD</sub>		V <sub>DD</sub>		
Output voltage, low	V <sub>OL</sub>		0.45		V	I <sub>OL</sub> = 1.6 mA
Output voltage, high	V <sub>OH</sub>	V <sub>DD</sub> - 1.0			V	I <sub>OH</sub> = -0.4 mA
Input current	I <sub>IN</sub>		±20		μA	E <sub>A</sub> , P1 <sub>0</sub> /NMI; V <sub>I</sub> = 0 to V <sub>DD</sub>
Input leakage current	I <sub>LI</sub>		±10		μA	All except E <sub>A</sub> , P1 <sub>0</sub> /NMI; V <sub>I</sub> = 0 to V <sub>DD</sub>
Output leakage current	I <sub>LO</sub>		±10		μA	V <sub>O</sub> = 0 to V <sub>DD</sub>
Data retention voltage	V <sub>DDDR</sub>	2.5	5.5		V	

#### Notes:

- (1) The standard operating temperature range is -10 to +70°C. However, extended temperature range parts (-40 to +85°C) are available.

Supply Current vs Clock Frequency



Comparator Characteristics

V<sub>DD</sub> = +5 V ±10%; T<sub>A</sub> = -10 to +70°C

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Accuracy	V <sub>ACOMP</sub>		±100	mV	
Threshold voltage	V <sub>TH</sub>	0	V <sub>DD</sub> + 0.1	V	
Comparison time	t <sub>COMP</sub>	64	65	t <sub>CYK</sub>	
PT input voltage	V <sub>IPT</sub>	0	V <sub>DD</sub>	V	

Capacitance Characteristics

V<sub>DD</sub> = 0 V; T<sub>A</sub> = 25°C

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input capacitance	C <sub>I</sub>		10	pF	f <sub>c</sub> = 1 MHz;
Output capacitance	C <sub>O</sub>		20	pF	Unmeasured pins returned to 0 V
I/O capacitance	C <sub>IO</sub>		20	pF	

AC Characteristics

V<sub>DD</sub> = +5 V ±10%; T<sub>A</sub> = -10 to +70°C; C<sub>L</sub> = 100 pF (max)

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
V <sub>DD</sub> rise, fall time	t <sub>RDV</sub> , t <sub>FVD</sub>	200		μs	STOP mode
Input rise, fall time	t <sub>IR</sub> , t <sub>IF</sub>		20	ns	Except X1, X2, RESET, NMI
Input rise, fall time	t <sub>IRS</sub> , t <sub>IFS</sub>		30	ns	RESET, NMI (Schmitt)
Output rise, fall time	t <sub>OR</sub> , t <sub>OF</sub>		20	ns	Except CLKOUT
X1 cycle time	t <sub>CYX</sub>	98	250	ns	f <sub>CLK</sub> = 5 MHz
		62	250	ns	f <sub>CLK</sub> = 8 MHz
X1 width, low	t <sub>WXL</sub>	35		ns	f <sub>CLK</sub> = 5 MHz
		20		ns	f <sub>CLK</sub> = 8 MHz
X1 width, high	t <sub>WXH</sub>	35		ns	f <sub>CLK</sub> = 5 MHz
		20		ns	f <sub>CLK</sub> = 8 MHz
X1 rise, fall time	t <sub>XR</sub> , t <sub>XF</sub>		10	ns	
CLKOUT cycle time	t <sub>CYK</sub>	200	2000	ns	f <sub>CLK</sub> = 5 MHz
		125	2000	ns	f <sub>CLK</sub> = 8 MHz
CLKOUT width, low	t <sub>WKL</sub>	0.5T - 15		ns	Note 1
		0.5T - 15		ns	
CLKOUT width, high	t <sub>WKH</sub>	0.5T - 15		ns	
		0.5T - 15		ns	
CLKOUT rise, fall time	t <sub>KR</sub> , t <sub>KF</sub>		15	ns	
Address delay time	t <sub>DKA</sub>		90	ns	
Address hold time	t <sub>HMA</sub>	0.5T - 30		ns	
Address valid to input data valid	t <sub>DADR</sub>		T(n + 1.5) - 120	ns	Note 2
MREQ to data delay	t <sub>DMRD</sub>		T(n + 1) - 90	ns	
MSTB to data delay	t <sub>DMSD</sub>		T(n + 0.5) - 90	ns	
MREQ to MSTB delay	t <sub>DMRMS</sub>	0.5T - 50	0.5T + 50	ns	
MREQ width, low	t <sub>WMRL</sub>		T(n + 1) - 40	ns	
Input data hold time	t <sub>HMDR</sub>		0	ns	
Next control setup time	t <sub>SCC</sub>		T - 25	ns	

Notes:

- (1) T = CPU clock period (t<sub>CYK</sub>).
- (2) n = number of wait states inserted.

### AC Characteristics (cont)

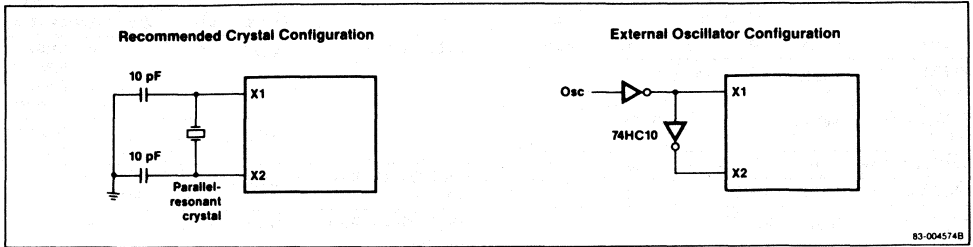
Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
MREQ to TC delay time	tDMRTC		0.5T + 50	ns	
Address data output	tDADW		0.5T + 50	ns	
MREQ delay time	tDAMR	0.5T - 50		ns	
MSTB delay time	tDAMS	T - 50		ns	
MSTB width, low	tWMSL	T(n + 0.5) - 40		ns	
Data output setup time	tSDM	T(n + 1) - 50		ns	
Data output hold time	tHMDW	0.5T - 30		ns	
IOSTB delay time	tDAIS	0.5T - 50		ns	
IOSTB to data input	tDISD		T(n + 1) - 120	ns	
IOSTB width, low	tWISL	T(n + 1) - 40		ns	
Address hold time	tHISA	0.5T - 30		ns	
Input data hold time	tHISDR	0		ns	
Output data setup time	tSDIS	T(n + 1) - 50		ns	
Output data hold time	tHISDW	0.5T - 30		ns	
Next DMARQ setup time	tSDADQ		T	ns	Demand mode
DMARQ hold time	tHDADQ	0		ns	Demand mode
DMAAK read width, low	tWDMRL	T(n + 1.5) - 40		ns	
DMAAK write width, low	tWDMWL	T(n + 1) - 40		ns	
DMAAK to TC delay time	tDDATC		0.5T + 50	ns	
TC width, low	tWTCL	2T - 40		ns	
REFRQ delay time	tDARF	0.5T - 50		ns	
REFRQ width, low	tWRFL	T(n + 1) - 40		ns	
Address hold time	tHRFA	0.5T - 50		ns	

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
RESET width low	tWRSL1	30		ms	STOP/POR (Power-on reset)
	tWRSL2	5		μs	System reset
MREQ, IOSTB to READY setup time	tSCRY		T(n - 1) - 100	ns	n ≥ 2
MREQ, IOSTB to READY hold time	tHCRY	T(n - 1)		ns	n ≥ 2
HLDRO setup time	tSHQK	30		ns	
HLDARQ output delay	tDKHA		80	ns	
Bus control float to HLDARQ ↓	tCFHA	T - 50		ns	
HLDARQ ↑ to control output time	tDHAC	T - 50		ns	
HLDRO to HLDARQ delay	tDHQHA		3T + 160	ns	
HLDRO ↓ to control float	tDHQC	3T + 30		ns	
HLDRO width, low	tWHQL	1.5T		ns	
HLDARQ width, low	tWHAL	T		ns	
INTP, DMARQ setup	tSIQK	30		ns	
INTP, DMARQ width, high	tWIQH	8T		ns	
INTP, DMARQ width, low	tWQIL	8T		ns	
POLL setup time	tSPLK	30		ns	
NMI width, high	tWNIH	5		μs	
NMI width, low	tWNIL	5		μs	
CTS width, low	tWCTL	2T		ns	
INTR setup time	tSIRK	30		ns	
INTAK delay time	tDKIA		80	ns	
INTR hold time	tHIAIQ	0		ns	
INTAK width, low	tWIAL	2T - 40		ns	
INTAK width, high	tWIAH	T - 40		ns	
INTAK to data delay	tDIAD		2T - 130	ns	
INTAK to data hold	tHIAID	0	0.5T	ns	
SCK0 (TSCK) cycle time	tCYTK	1000		ns	
SCK0 (TSCK) width, high	tWSTH	450		ns	
SCK0 (TSCK) width, low	tWSTL	450		ns	
1xD delay time	tPTKD		210	ns	
CTS0 (RSCK) cycle time	tCYRK	1000		ns	
CTS0 (RSCK) width, high	tWSRH	420		ns	
CTS0 (RSCK) width, low	tWSRL	420		ns	
RxD setup time	tSRDK	80		ns	
RxD hold time	tHKRD	80		ns	

3

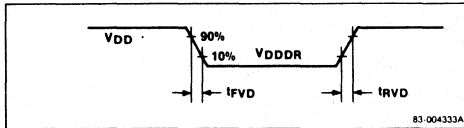
## $\mu$ PD70320/322 (V25)

**Figure 35. External System Clock Control Source**

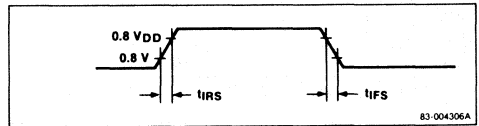


### Timing Waveforms

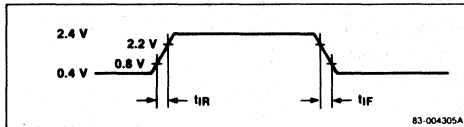
#### Stop Mode Data Retention Timing



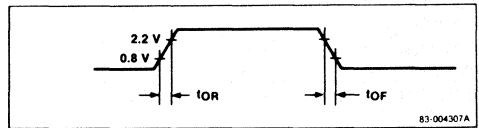
#### AC Input Waveform 2 (RESET, NMI)



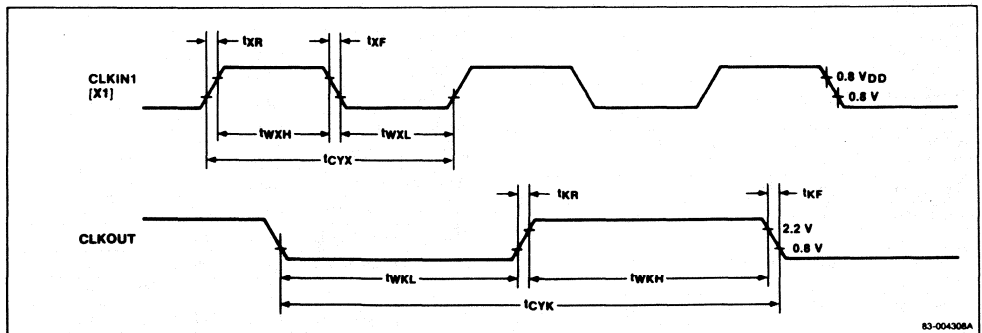
#### AC Input Waveform 1 (Except X1, X2, RESET, NMI)



#### AC Output Test Point (Except CLKOUT)

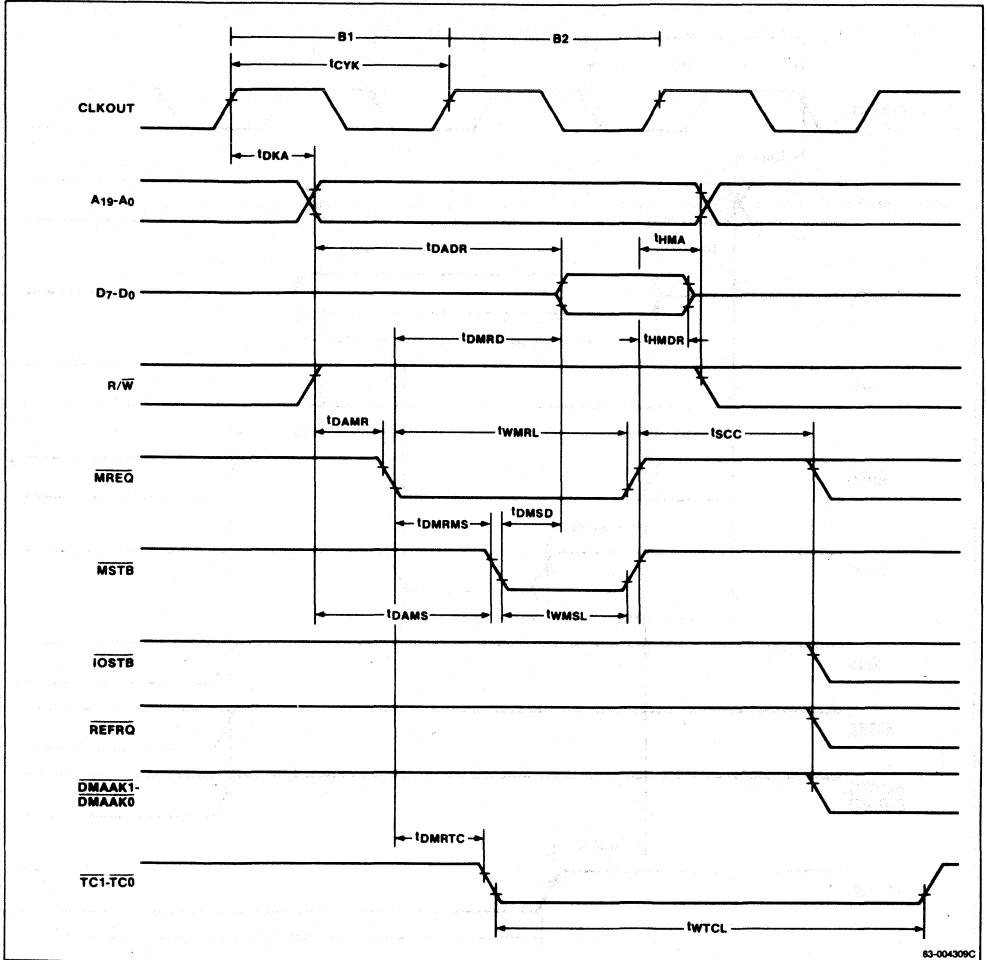


### Clock In and Clock Out



### Timing Waveforms (cont)

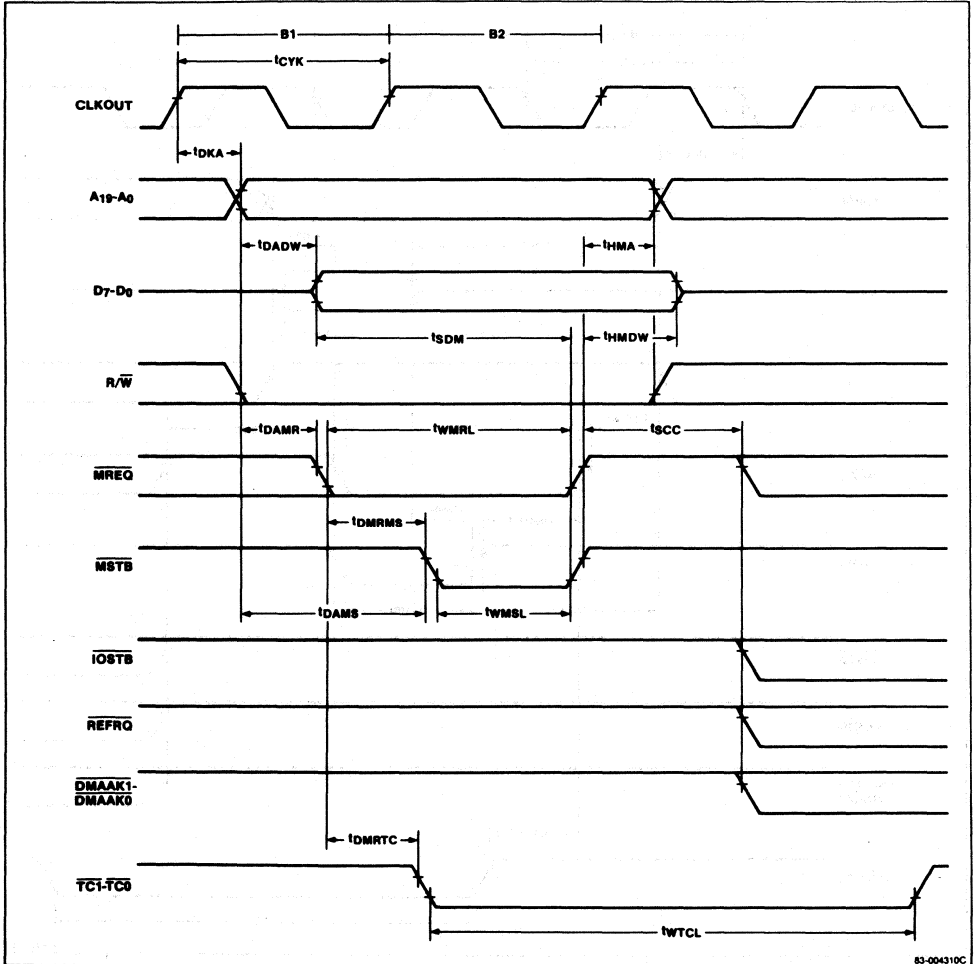
#### Memory Read





Timing Waveforms (cont)

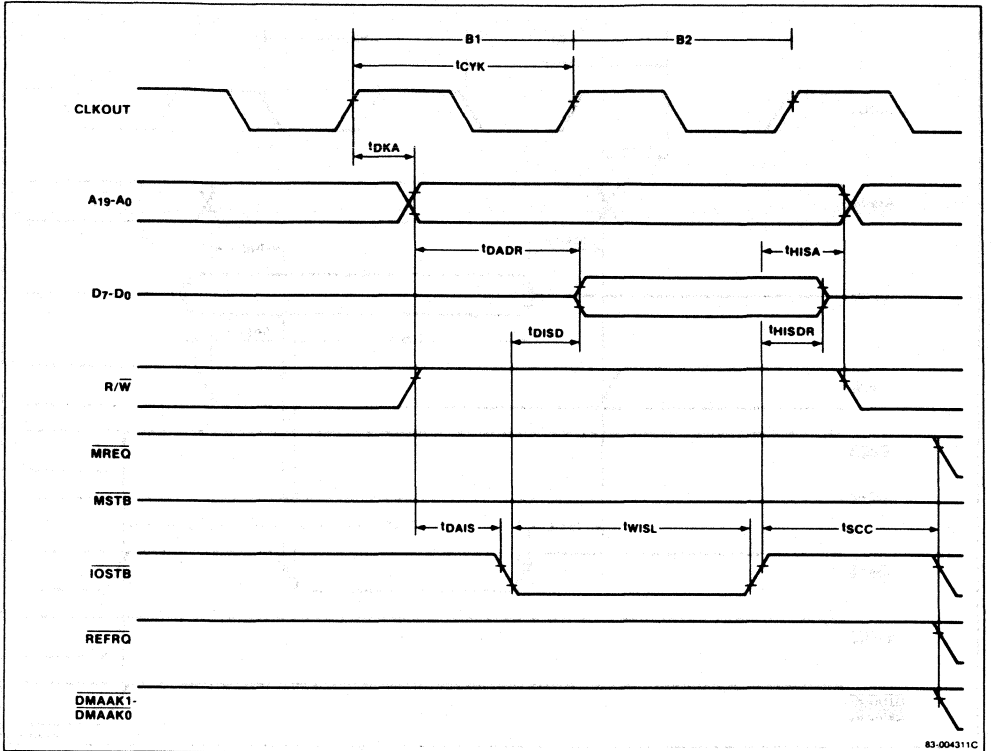
Memory Write



83-004310C

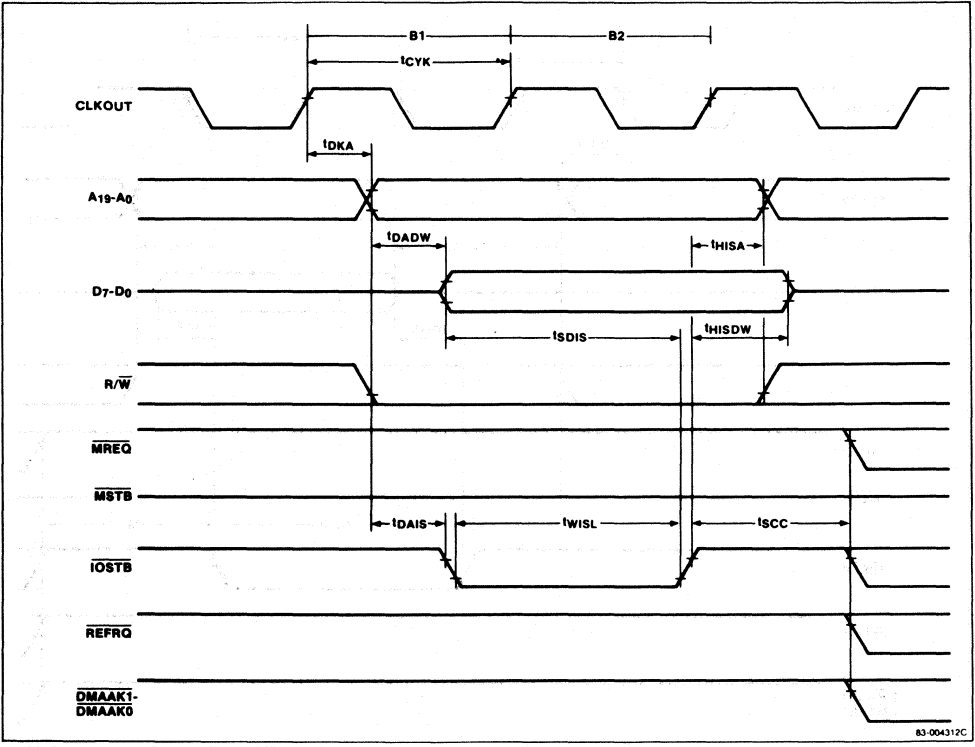
### Timing Waveforms (cont)

#### I/O Read



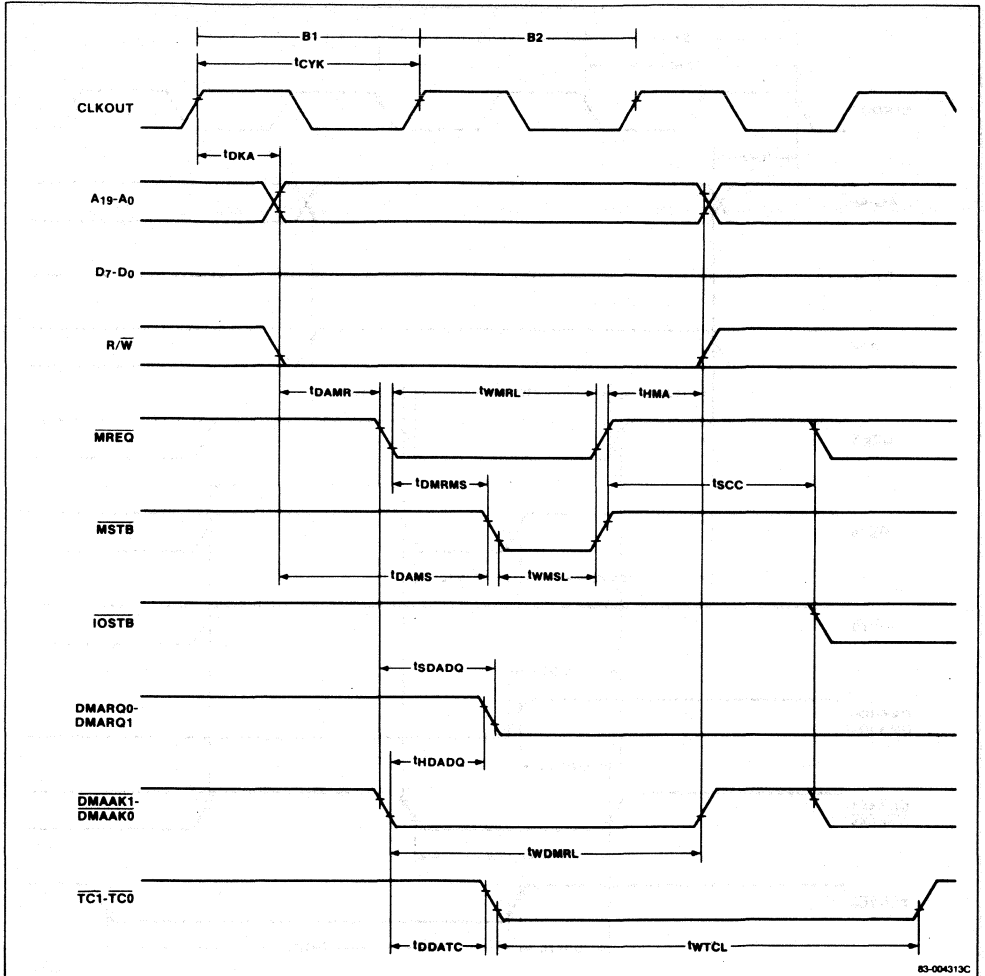
### Timing Waveforms (cont)

I/O Write



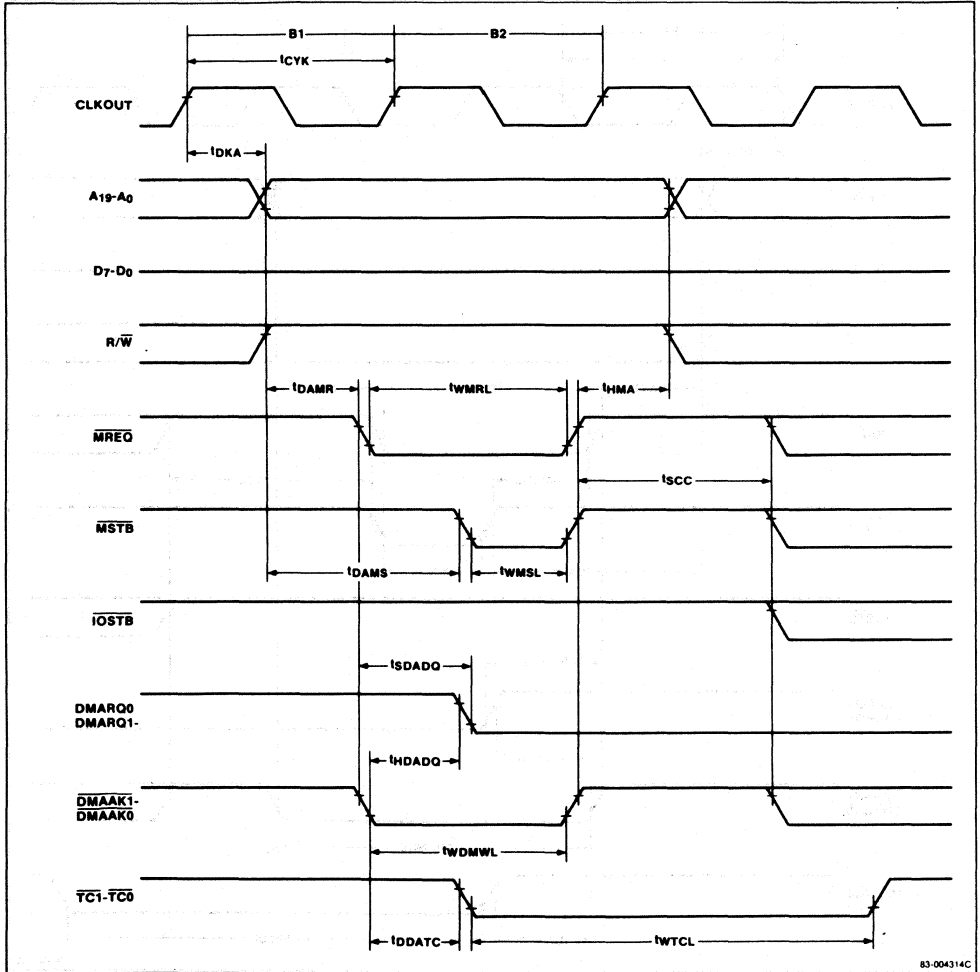
### Timing Waveforms (cont)

#### DMA, I/O to Memory



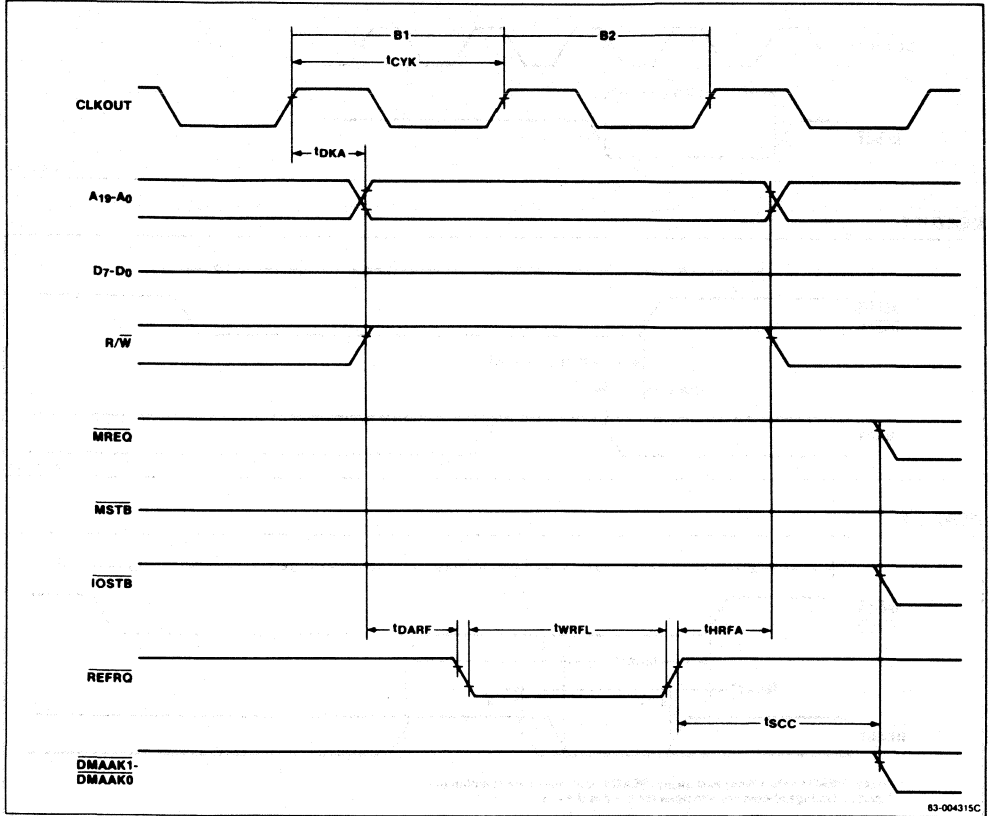
Timing Waveforms (cont)

DMA, Memory to I/O

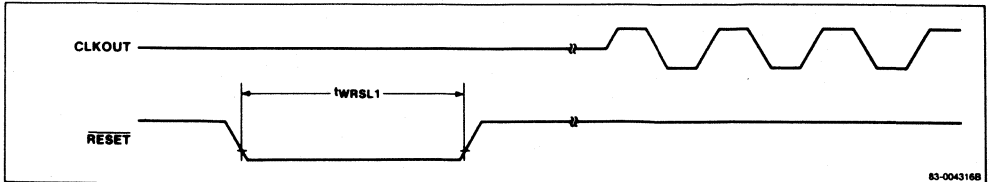


### Timing Waveforms (cont)

#### Refresh

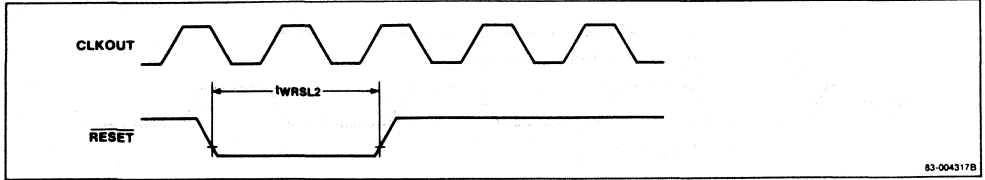


#### RESET 1



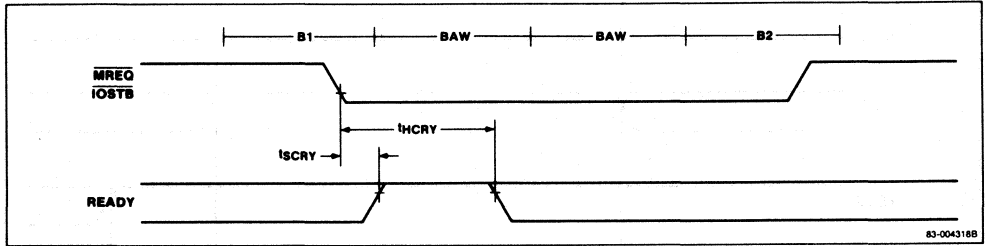
Timing Waveforms (cont)

RESET 2



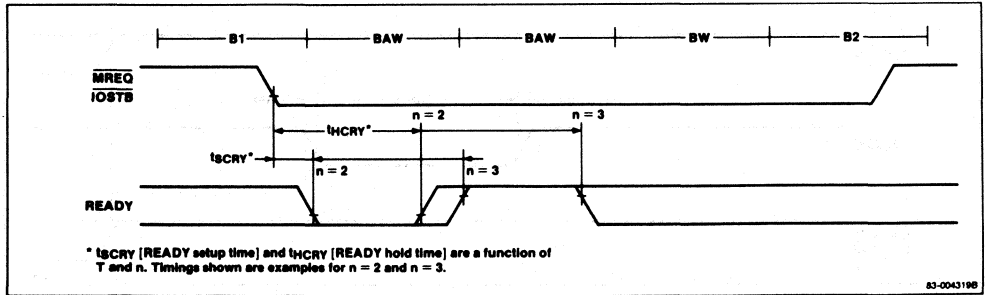
83-004317B

READY 1



83-004318B

READY 2

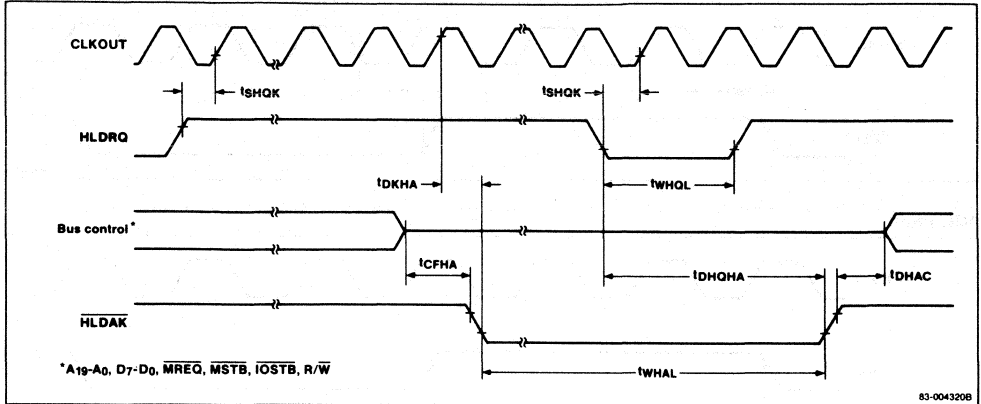


\*  $t_{SCRY}$  (READY setup time) and  $t_{HCRY}$  (READY hold time) are a function of T and n. Timings shown are examples for  $n = 2$  and  $n = 3$ .

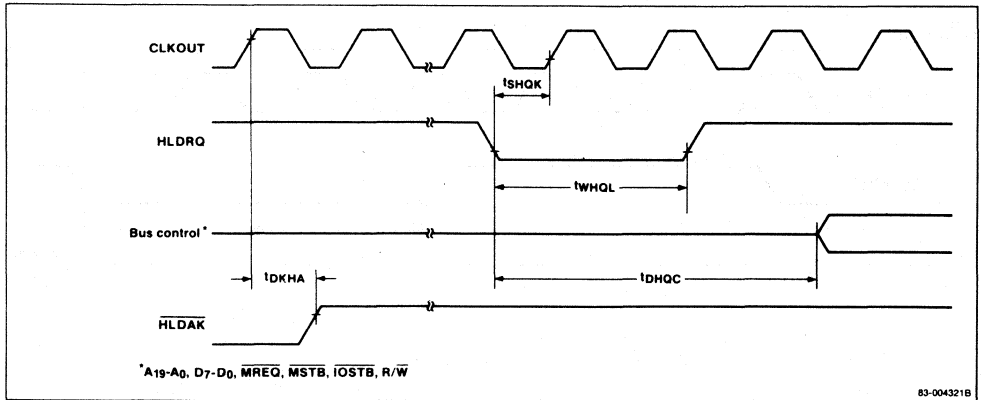
83-004319B

### Timing Waveforms (cont)

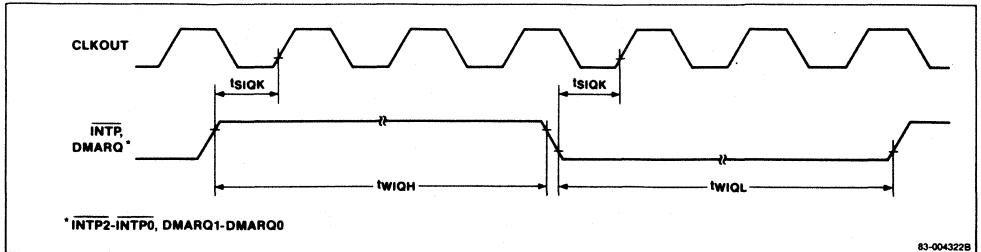
#### HLDRQ/HLDAK 1



#### HLDRQ/HLDAK 2



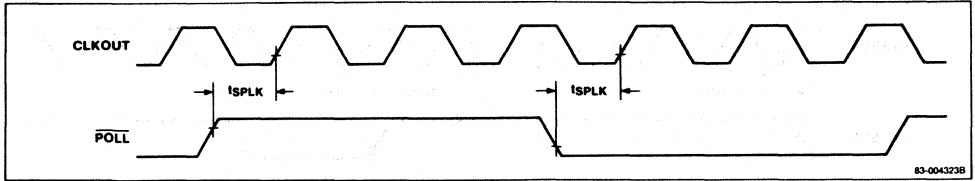
#### INTP, DMARQ Input



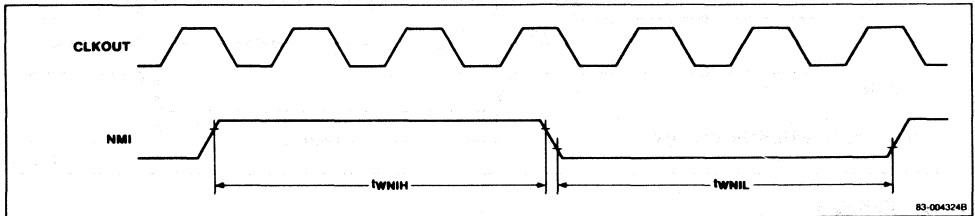


### Timing Waveforms (cont)

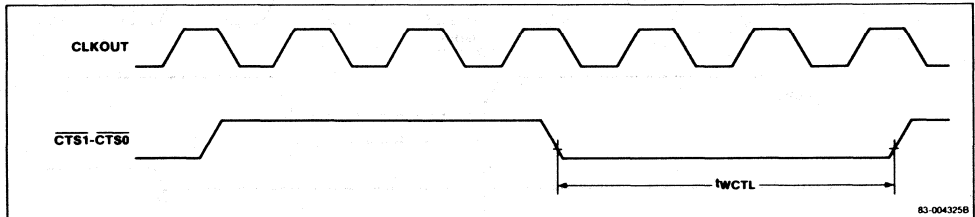
#### $\overline{\text{POLL}}$ Input



#### NMI Input

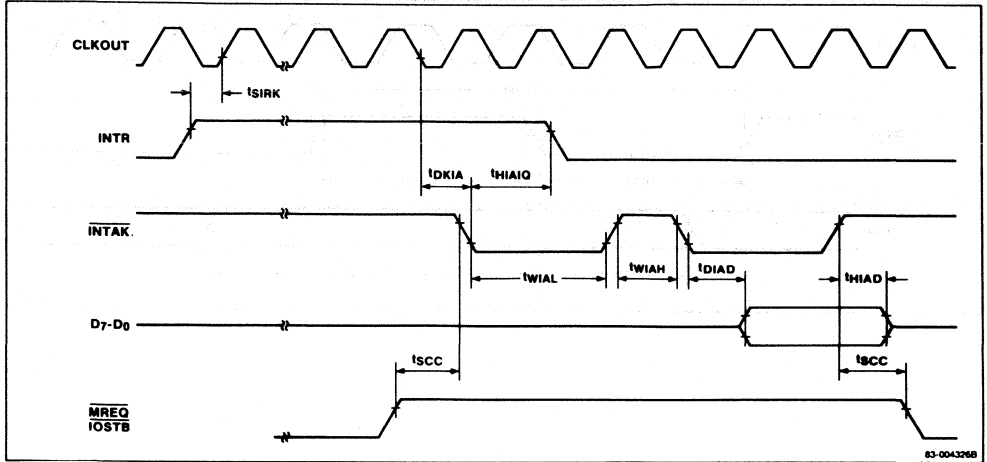


#### $\overline{\text{CTS}}$ Input

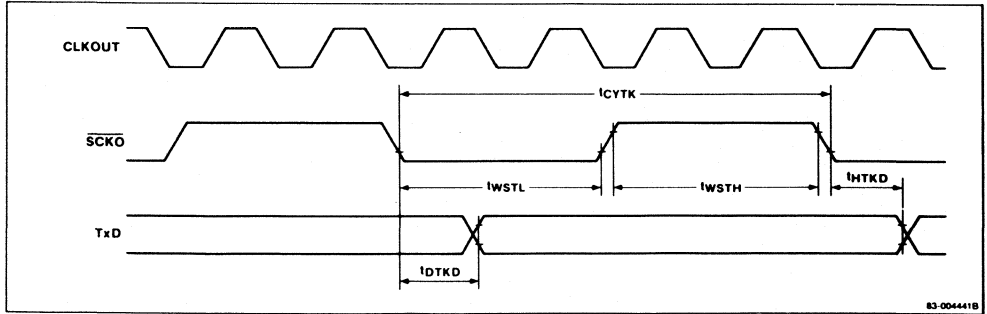


### Timing Waveforms (cont)

#### INTR/INTAK



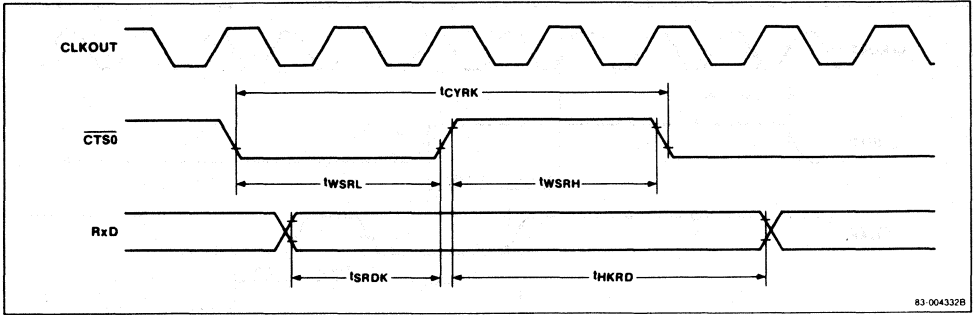
#### Serial Transmit



3

**Timing Waveforms (cont)**

**Serial Receive**



### Instruction Set

Instructions, grouped according to function, are described in a table near the end of this data sheet. Descriptions include source code, operation, opcode, number of bytes, and flag status. Supplementary information applicable to the instruction set is contained in the following tables.

- Symbols and Abbreviations
- Flag Symbols
- 8- and 16-Bit Registers. When mod = 11, the register is specified in the operation code by the byte/word operand (W = 0/1) and reg (000 to 111).
- Segment Registers. The segment register is specified in the operation code by sreg (00, 01, 10, or 11).
- Memory Addressing. The memory addressing mode is specified in the operation code by mod (00, 01, or 10) and mem (000 through 111).
- Instruction Clock Count. This table gives formulas for calculating the number of clock cycles occupied by each type of instruction. The formulas, which depend on byte/word operand and RAM enable/disable, have variables such as EA (effective address), W (wait states), and n (iterations or string instructions).

### Symbols and Abbreviations

Identifier	Description
reg	8- or 16-bit general-purpose register
reg8	8-bit general-purpose register
reg16	16-bit general-purpose register
dmem	8- or 16-bit direct memory location
mem	8- or 16-bit memory location
mem8	8-bit memory location
mem16	16-bit memory location
mem32	32-bit memory location
sfr	8-bit special function register location
imm	Constant (0 to FFFFH)
imm16	Constant (0 to FFFFH)
imm8	Constant (0 to FFH)
imm4	Constant (0 to FH)
imm3	Constant (0 to 7)
acc	AW or AL register
sreg	Segment register
src-table	Name of 256-byte translation table
src-block	Name of block addressed by the IX register

Identifier	Description
dst-block	Name of block addressed by the IY register
near-proc	Procedure within the current program segment
far-proc	Procedure located in another program segment
near-label	Label in the current program segment
short-label	Label between -128 and +127 bytes from the end of instruction
far-label	Label in another program segment
memptr16	Word containing the offset of the memory location within the current program segment to which control is to be transferred
memptr32	Double word containing the offset and segment base address of the memory location to which control is to be transferred
regptr16	16-bit register containing the offset of the memory location within the program segment to which control is to be transferred
pop-value	Number of bytes of the stack to be discarded (0 to 64K bytes, usually even addresses)
fp-op	Immediate data to identify the instruction code of the external floating point operation
R	Register set
W	Word/byte field (0 to 1)
reg	Register field (000 to 111)
mem	Memory field (000 to 111)
mod	Mode field (00 to 10)
S:W	When S:W = 01 or 11, data = 16 bits. At all other times, data = 8 bits.
X, XXX, YYY, ZZZ	Data to identify the instruction code of the external floating point arithmetic chip
AW	Accumulator (16 bits)
AH	Accumulator (high byte)
AL	Accumulator (low byte)
BP	Base pointer register (16 bits)
BW	BW register (16 bits)
BH	BW register (high byte)
BL	BW register (low byte)
CW	CW register (16 bits)
CH	CW register (high byte)
CL	CW register (low byte)
DW	DW register (16 bits)
DH	DW register (high byte)
DL	DW register (low byte)
SP	Stack pointer (16 bits)
PC	Program counter (16 bits)
PSW	Program status word (16 bits)

## μPD70320/322 (V25)

### Symbols and Abbreviations (cont)

Identifier	Description
IX	Index register (source) (16 bits)
IY	Index register (destination) (16 bits)
PS	Program segment register (16 bits)
SS	Stack segment register (16 bits)
DS <sub>0</sub>	Data segment 0 register (16 bits)
DS <sub>1</sub>	Data segment 1 register (16 bits)
AC	Auxiliary carry flag
CY	Carry flag
P	Parity flag
S	Sign flag
Z	Zero flag
DIR	Direction flag
IE	Interrupt enable flag
V	Overflow flag
BRK	Break flag
MD	Mode flag
(...)	Values in parentheses are memory contents
disp	Displacement (8 or 16 bits)
ext-disp8	16-bit displacement (sign-extension byte + 8-bit displacement)
temp	Temporary register (8/16/32 bits)
tmpcy	Temporary carry flag (1-bit)
seg	Immediate segment data (16 bits)
offset	Immediate offset data (16 bits)
←	Transfer direction
+	Addition
-	Subtraction
x	Multiplication
÷	Division
%	Modulo
AND	Logical product
OR	Logical sum
XOR	Exclusive logical sum
XXH	Two-digit hexadecimal value
XXXXH	Four-digit hexadecimal value

### Flag Symbols

Identifier	Description
(blank)	No change
0	Cleared to 0
1	Set to 1
X	Set or cleared according to the result
U	Undefined
R	Value saved earlier is restored

### 8- and 16-Bit Registers (mod = 11)

reg	W = 0	W = 1
000	AL	AW
001	CL	CW
010	DL	DW
011	BL	BW
100	AH	SP
101	CH	BP
110	DH	IX
111	BH	IY

### Segment Registers

sreg	Register
00	DS <sub>1</sub>
01	PS
10	SS
11	DS <sub>0</sub>

### Memory Addressing

mem	mod = 00	mod = 01	mod = 10
000	BW + IX	BW + IX + disp8	BW + IX + disp16
001	BW + IY	BW + IY + disp8	BW + IY + disp16
010	BP + IX	BP + IX + disp8	BP + IX + disp16
011	BP + IY	BP + IY + disp8	BP + IY + disp16
100	IX	IX + disp8	IX + disp16
101	IY	IY + disp8	IY + disp16
110	Direct	BP + disp8	BP + disp16
111	BW	BW + disp8	BW + disp16

### Instruction Clock Count

Mnemonic	Operand	Byte		Word	
		RAM Enable	RAM Disable	RAM Enable	RAM Disable
ADD	reg,reg (Note 2)	2	2	2	2
ADD	mem,reg (Note 3)	EA + 8 + 2W	EA + 6 + 1W	EA + 12 + 4W	EA + 8 + 2W
ADD	reg,mem (Note 2)	EA + 6 + W	EA + 6 + W	EA + 8 + 2W	EA + 8 + 2W
ADD	AL,imm8 (Note 2)	5	5	—	—
ADD	AW,imm16 (Note 2)	—	—	6	6
ADD	reg,imm8 (Note 2)	5	5	—	—
	mem,imm8 (Note 3)	EA + 9 + 2W	EA + 7 + 2W	—	—
ADD	reg,imm16 (Note 2)	—	—	6	6
	mem,imm16 (Note 3)	—	—	EA + 14 + 4W	EA + 10 + 4W
ADD4S		22 + (27+3W)n	22 + (25+3W)n	—	—
ADJ4A		9	9	—	—
ADJ4S		9	9	—	—
ADJBA		17	17	—	—
ADJBS		17	17	—	—
BV		—	—	15/8	15/8
BNV		—	—	15/8	15/8
BC		—	—	15/8	15/8
BL		—	—	15/8	15/8
BNC		—	—	15/8	15/8
BNL		—	—	15/8	15/8
BE		—	—	15/8	15/8
BZ		—	—	15/8	15/8
BNE		—	—	15/8	15/8
BNZ		—	—	15/8	15/8
BNH		—	—	15/8	15/8
BH		—	—	15/8	15/8
BN		—	—	15/8	15/8
BP		—	—	15/8	15/8
BR	near	—	—	12	12
BR	far	—	—	15	15
BR	near short	—	—	12	12

#### Notes:

- (1) EA = effective address (3 or 4 clock cycles)  
= 3 (mod 00 or 01) or 4 (mod 10)  
W = number of wait states selected by WTC register.  
n = number of iterations or string instructions.
- (2) ADD = ADDC, SUBC, CMP, AND, OR, XOR
- (3) ADD = ADDC, SUBC, AND, OR, XOR
- (4) SHL = SHR, SHRA, ROL, ROLC, RORC

## μPD70320/322 (V25)

### Instruction Clock Count (cont)

Mnemonic	Operand	Byte		Word	
		RAM Enable	RAM Disable	RAM Enable	RAM Disable
BR	near reg	—	—	13	13
	mem	—	—	EA + 17 + 2W	EA + 17 + 2W
BR	far mem	—	—	EA + 25 + 4W	EA + 25 + 4W
BPE		—	—	15/8	15/8
BPO		—	—	15/8	15/8
BLT		—	—	15/8	15/8
BGE		—	—	15/8	15/8
BLE		—	—	15/8	15/8
BGT		—	—	15/8	15/8
BRK	3	—	—	55 + 10W	43 + 10W
BRK	n	—	—	56 + 10W	44 + 10W
BRKV		—	—	55 + 10W	43 + 10W
BCWZ		—	—	15/8	15/8
BREAK		—	—	58 + 10W	46 + 10W
BTCLR		29	29	—	—
BUSLOCK		2	2	2	2
CALL	near	—	—	22 + 2W	18 + 2W
CALL	far	—	—	38 + 4W	34 + 4W
CALL	near reg	—	—	22 + 2W	18 + 2W
	mem	—	—	EA + 26 + 4W	EA + 24 + 4W
CALL	far mem	—	—	EA + 36 + 8W	EA + 24 + 8W
CHKIND	reg,mem	—	—	EA + 26 + 4W	EA + 26 + 4W
CLR1	reg, CL	8	8	8	8
	mem,CL	EA + 14 + 2W	EA + 12 + 1W	EA + 18 + 4W	EA + 14 + 2W
CLR1	reg, imm	7	7	7	7
	mem, imm	EA + 11 + 2W	EA + 9 + 1W	EA + 15 + 4W	EA + 10 + 2W
CLR1	CY	2	2	2	2
	DIR	2	2	2	2
CMP	mem,reg	EA + 6 + 1W	EA + 6 + 1W	EA + 8 + 2W	EA + 8 + 2W
CMP	mem,imm8	EA + 7 + W	EA + 7 + W	—	—
CMP	mem,imm16	—	—	EA + 10 + 2W	EA + 10 + 2W
CMP4S		22 + (23+2W)n	22 + (23+2W)n	—	—
CMPM		17 + W	17 + W	19 + 2W	19 + 2W
	(rep)	16 + (15+W)n	16 + (15+W)n	16 + (17+2W)n	16 + (17+2W)n
CMPBK		23 + 2W	19 + 2W	27 + 4W	21 + 4W
	(rep)	16 + (21+2W)n	16 + (21+2W)n	16 + (25+4W)n	16 + (25+4W)n
CVTBW		3	3	—	—
CVTWL		—	—	8	8
CVTDB		19	19	—	—
CVTBD		20	20	—	—

### Instruction Clock Count (cont)

Mnemonic	Operand	Byte		Word	
		RAM Enable	RAM Disable	RAM Enable	RAM Disable
DBNZNE/ DBNZE		—	—	17/8	17/8
DBNZ		—	—	17/8	17/8
DI		4	4	4	4
DIV	reg	46 ~ 56	46 ~ 56	—	—
DIV	mem	EA + 48 + W ~ EA + 58 + W	EA + 48 + W ~ EA + 58 + W	—	—
DIV	reg	—	—	54 ~ 64	54 ~ 64
DIV	mem	—	—	EA + 58 + 2W ~ EA + 68 + 2W	EA + 58 + 2W ~ EA + 68 + 2W
DIVU	reg	31	31	—	—
DIVU	mem	EA + 33 + W	EA + 33 + W	—	—
DIVU	reg	—	—	39	39
DIVU	mem	—	—	EA + 43 + 2W	EA + 43 + 2W
DISPOSE		—	—	12 + 2W	12 + 2W
EI		12	12	12	12
EXT	reg, reg	—	—	41 ~ 121	41 ~ 121
	reg, imm4	—	—	42 ~ 122	42 ~ 122
FINT		2	2	2	2
FP01/FP02		—	—	60 + 10W	48 + 10W
HALT		—	—	—	—
IN	A, imm8	14 + W	14 + W	16 + 2W	16 + 2W
IN	A, DW	13 + W	13 + W	15 + 2W	15 + 2W
INS	reg, reg	—	—	63 ~ 155	63 ~ 155
	reg, imm4	—	—	64 ~ 156	64 ~ 156
INC/DEC	reg	—	—	2	2
INM		19 + 2W	17 + 2W	21 + 4W	17 + 4W
	(rep)	18 + (13+2W)n	18 + (11+2W)n	18 + (15+4W)n	18 + (11+4W)n
INC/DEC	reg	5	5	5	5
	mem	EA + 11 + 2W	EA + 9 + 2W	EA + 15 + 4W	EA + 11 + 4W
LDM		12 + W	12 + W	14 + 2W	14 + 2W
	(rep)	16 + (10+W)n	16 + (10+W)n	16 + (12+2W)n	16 + (12+2W)n
LDEA		—	—	EA + 2	EA + 2
MUL	reg	31 ~ 40	31 ~ 40	—	—
MUL	mem	EA + 33 + W ~ EA + 42 + W	EA + 33 + W ~ EA + 42 + W	—	—



**Instruction Clock Count (cont)**

Mnemonic	Operand	Byte		Word	
		RAM Enable	RAM Disable	RAM Enable	RAM Disable
MUL	reg	—	—	39 ~ 48	39 ~ 48
MUL	mem	—	—	EA + 43 + 2W ~ EA + 52 + 2W	EA + 43 + 2W ~ EA + 52 + 2W
MUL	r1,r2,imm16	—	—	40 ~ 50	40 ~ 50
	r1,mem,imm16	—	—	EA + (44 to 54) + 2W	EA + (44 to 54) + 2W
MUL	r1,r2,imm8	—	—	39 ~ 49	39 ~ 49
	r1,mem,imm8	—	—	EA + (43 to 53) + 2W	EA + (43 to 53) + 2W
MULU	reg	24	24	—	—
MULU	mem	EA + 26 + W	EA + 26 + W	—	—
MULU	reg	—	—	32	32
MULU	mem	—	—	EA + 34 + 2W	EA + 34 + 2W
MOV	reg,reg	2	2	2	2
	mem,reg	EA + 4 + W	EA + 2	EA + 6 + 2W	EA + 2
	reg,mem	EA + 6 + W	EA + 6 + W	EA + 8 + 2W	EA + 8 + 2W
MOV	reg,sreg	—	—	3	3
	mem,sreg	—	—	EA + 7 + 2W	EA + 3
MOV	DS0,reg,mem	—	—	EA + 19 + 4W	EA + 19 + 4W
MOV	DS1,reg,mem	—	—	EA + 19 + 4W	EA + 19 + 4W
MOV	reg,imm8	5	5	—	—
	mem,imm8	EA + 5 + W	EA + 5 + W	—	—
MOV	reg,imm16	—	—	6	6
	mem,imm16	—	—	EA + 5 + 2W	EA + 5 + 2W
MOV	sreg,reg	—	—	4	4
	sreg,mem	—	—	EA + 10 + 2W	EA + 10 + 2W
MOV	PSW,AH	3	3	—	—
MOV	AH,PSW	2	2	—	—
MOV	A,mem	9 + W	9 + W	11 + 2W	11 + 2W
MOV	mem,A	7 + W	5 + W	9 + 2W	5
MOVBK		20 + 2W	16 + 1W	24 + 4W	20 + 2W
	(rep)	16 + (16+2W)n	16 + (12+W)n	16 + (20+4W)n	16 + (12+2W)n
NOP		4	4	4	4
NOT1		2	2	2	2
NOT1	reg, CL	7	7	7	7
	mem CL	EA + 13 + 2W	EA + 11 + 1W	EA + 17 + 4W	EA + 13 + 2W
NOT1	reg, imm	6	6	6	6
	mem, imm	EA + 10 + 2W	EA + 8 + 1W	EA + 14 + 4W	EA + 10 + 2W

### Instruction Clock Count (cont)

Mnemonic	Operand	Byte		Word	
		RAM Enable	RAM Disable	RAM Enable	RAM Disable
NOT/NEG	reg	5	5	5	5
	mem	EA + 11 + 2W	EA + 9 + W	EA + 15 + 4W	EA + 11 + 2W
OUT	DW,A	9 + W	9 + W	9 + 2W	9 + 2W
OUT	imm8,A	10 + W	10 + W	10 + 2W	10 + 2W
OUTM		19 + 2W	17 + 2W	21 + 4W	17 + 4W
	(rep)	18 + (13+2W)n	18 + (11+2W)n	18 + (15+4W)n	18 + (11+4W)n
PUSH	sreg	—	—	11 + 2W	7
POP	sreg	—	—	13 + 2W	13 + 2W
PUSH	reg	—	—	10 + 2W	6
POP	reg	—	—	12 + 2W	12 + 2W
PUSH	R	—	—	82 + 16W	50
POP	R	—	—	82 + 16W	58
PUSH	imm16	—	—	14 + 2W	10
PUSH	imm8	—	—	13 + 2W	9
POP	reg	—	—	12	12
	mem	—	—	EA + 16 + 4W	EA + 12 + 2W
PUSH	PSW	—	—	10 + 2W	6
POP	PSW	—	—	14 + 2W	14 + 2W
PUSH	reg	—	—	14 + 2W	10 + 2W
	mem	—	—	EA + 18 + 4W	EA + 14 + 4W
POLL		—	—	—	—
PREPARE		—	—	—	—
ROL4	reg	17	17	—	—
	mem	EA + 18 + 2W	EA + 16 + 2W	—	—
ROR4	reg	21	21	—	—
	mem	EA + 24 + 2W	EA + 22 + 2W	—	—
RET	near (ADJ)	—	—	20 + 2W	20 + 2W
RET	near	—	—	20 + 2W	20 + 2W
RET	far (ADJ)	—	—	30 + 4W	30 + 4W
RET	far	—	—	29 + 4W	29 + 4W
RETI		—	—	43 + 6W	35 + 2W
REPC		2	2	2	2
REPNC		2	2	2	2
REP/REPE/ REPZ		2	2	2	2
REPNE/ REPNZ		2	2	2	2
RETRBI		—	—	12	12
SET1	reg, CL	7	7	7	7
	mem, CL	EA + 13 + 2W	EA + 11 + 1W	EA + 17 + 4W	EA + 13 + 2W

**Instruction Clock Count (cont)**

Mnemonic	Operand	Byte		Word	
		RAM Enable	RAM Disable	RAM Enable	RAM Disable
SET1	reg, imm	6	6	6	6
	mem, imm	EA + 10 + 2W	EA + 8 + 1W	EA + 14 + 4W	EA + 10 + 2W
SET1	CY	2	2	2	2
	DIR	2	2	2	2
STM		12 + W	10	14 + 2W	10
	(rep)	16 + (8+W)n	16 + (6+W)n	16 + (10+2W)n	16 + (6+2W)n
SHL	reg,imm8 (Note 4)	9 + 2n	9 + 2n	9 + 2n	9 + 2n
SHL	mem,imm8 (Note 4)	EA + 13 + 2W + 2n	EA + 11 + W + 2n	EA + 17 + 4W + 2n	EA + 13 + 2W + 2n
SHL	reg,1 (Note 4)	8	8	8	8
	mem,1 (Note 4)	EA + 14 + 2W	EA + 12 + W	EA + 18 + 4W	EA + 14 + 2W
SHL	reg,CL (Note 4)	11 + 2n	11 + 2n	11 + 2n	11 + 2n
	mem,CL (Note 4)	EA + 17 + 2W + 2n	EA + 15 + W + 2n	EA + 21 + 4W + 2n	EA + 17 + 2W + 2n
SUB4S		22 + (27+3W)n	22 + (25+3W)n	—	—
STOP		—	—	—	—
TEST1	reg, CL	7	7	7	7
	mem, CL	EA + 11 + W	EA + 11 + W	EA + 13 + 2W	EA + 13 + 2W
	reg, imm	6	6	6	6
	mem, imm	EA + 8 + W	EA + 8 + W	EA + 10 + 2W	EA + 10 + 2W
TEST	reg,reg	4	4	4	4
	mem,reg	EA + 8 + W	EA + 8 + W	EA + 10 + 2W	EA + 10 + 2W
	reg, mem	EA + 8 + W	EA + 8 + W	EA + 10 + 2W	EA + 10 + 2W
TEST	AL,imm8	5	5	—	—
TEST	AW,imm16	—	—	6	6
TEST	reg,imm8	7	7	—	—
	mem,imm8	EA + 11 + W	EA + 11 + W	—	—
TEST	reg,imm16	—	—	8	8
	mem,imm16	—	—	EA + 11 + 2W	EA + 11 + 2W
TRANS		10 + W	10 + W	—	—
XCH	reg,reg	3	3	3	3
	mem,reg	EA + 10 + 2W	EA + 8 + 2W	EA + 14 + 4W	EA + 10 + 4W
XCH	AW,reg	—	—	4	4

### Instruction Clock Count (cont)

	Byte		Word	
	RAM Enable	RAM Disable	RAM Enable	RAM Disable
<b>Operations</b>				
Context switch interrupt	—	—	53 + N (max)	53 + N (max)
DMA (Single-step mode)	20 + 2W + N	20 + 2W + N	24 + 4W + N	24 + 4W + N
DMA (Demand release mode)	15 + W	15 + W	17 + 2W + N	17 + 2W + N
DMA (Burst mode)	(12 + 2W)n	(12 + 2W)n	(12 + 4W)n	(12 + 4W)n
DMA (Single-transfer mode)	33 + W + N	33 + W + N	35 + 2W + N	35 + 2W + N
Interrupt	—	—	62 + 10W + N	50 + 10W + N
Macro service, sfr — mem	23 + W	21 + W	27 + 2W	25 + 2W
Macro service, mem — sfr	22 + W	20 + W	26 + 2W	24 + 2W
Macro service (Search char mode), sfr — mem	37 + W	37 + W	—	—
Macro service (Search char mode), mem — sfr	37 + W	37 + W	—	—
Priority interrupt	—	—	58 + 10W + N	46 + 10W + N

N = number of clocks to complete currently executing instruction.



### Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code								Flags													
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	AC	CY	VP	SS	Z	
<b>Repeat Prefixes (cont)</b>																								
REP		While CW ≠ 0, the next byte of the primitive block transfer instruction is executed and CW is decremented (-1). If there is a waiting interrupt, it is processed. If the primitive block transfer instruction is CMPBK or CMPM and Z ≠ 1, exit the loop.	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REPE																								
REPZ																								
REPNE																								
REPNZ																								
<b>Primitive Block Transfer</b>																								
MOVBK	dst-block, src-block	When W = 0 (IV) ← (IX) DIR = 0: IX ← IX + 1, IV ← IV + 1 DIR = 1: IX ← IX - 1, IV ← IV - 1 When W = 1 (IY + 1, IV) ← (IX + 1, IX) DIR = 0: IX ← IX + 2, IV ← IV + 2 DIR = 1: IX ← IX - 2, IV ← IV - 2	1	0	1	0	0	1	0	W														
CMPBK	src-block, dst-block	When W = 0 (IX) ← (IY) DIR = 0: IX ← IX + 1, IV ← IV + 1 DIR = 1: IX ← IX - 1, IV ← IV - 1 When W = 1 (IX + 1, IX) ← (IY + 1, IV) DIR = 0: IX ← IX + 2, IV ← IV + 2 DIR = 1: IX ← IX - 2, IV ← IV - 2	1	0	1	0	0	1	1	W														
CMPM	dst-block	When W = 0 AL ← (IY) DIR = 0: IY ← IY + 1; DIR = 1: IY ← IY - 1 When W = 1 AW ← (IY + 1, IY) DIR = 0: IY ← IY + 2; DIR = 1: IY ← IY - 2	1	0	1	0	1	1	1	W														
LDM	src-block	When W = 0 AL ← (IX) DIR = 0: IX ← IX + 1; DIR = 1: IX ← IX - 1 When W = 1 AW ← (IX + 1, IX) DIR = 0: IX ← IX + 2; DIR = 1: IX ← IX - 2	1	0	1	0	1	1	0	W														
STM	dst-block	When W = 0 (IY) ← AL DIR = 0: IY ← IY + 1; DIR = 1: IY ← IY - 1 When W = 1 (IY + 1, IY) ← AW DIR = 0: IY ← IY + 2; DIR = 1: IY ← IY - 2	1	0	1	0	1	0	1	W														
<b>Bit Field Transfer</b>																								
INS	reg8, reg8	16-Bit field ← AW	0	0	0	1	1	1	0	0	1	1	0	0	0	1								
	reg8, imm4	16-Bit field ← AW	0	0	0	1	1	1	0	0	1	1	0	0	1									

Instruction Set (cont)		Operation Code										No. of			Flags												
Mnemonic	Operand	Operation	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	Bytes	AC	CY	V	P	S	Z		
<b>Bit Field Transfer (cont)</b>																											
EXT	reg8, reg8	AW ← 16-Bit field	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	3
	reg8, imm4	AW ← 16-Bit field	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1	0	1	1	0	1	1	0	1	4
			1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
<b>I/O</b>																											
IN	acc, imm8	When W = 0 AL ← (imm8) When W = 1 AH ← (imm8 + 1), AL ← (imm8)	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	2
	acc, DW	When W = 0 AL ← (DW) When W = 1 AH ← (DW + 1), AL ← (DW)	1	1	1	1	0	1	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1
OUT	imm8, acc	When W = 0 (imm8) ← AL When W = 1 (imm8 + 1) ← AH, (imm8) ← AL	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	2
	DW, acc	When W = 0 (DW) ← AL When W = 1 (DW + 1) ← AH, (DW) ← AL	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1
<b>Primitive Block I/O Transfer</b>																											
INM	dst-block, DW	When W = 0 (Y) ← (DW) DIR = 0: Y ← Y + 1; DIR = 1: Y ← Y - 1 When W = 1 (Y + 1, Y) ← (DW + 1, DW) DIR = 0: Y ← Y + 2; DIR = 1: Y ← Y - 2	0	1	1	0	1	0	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
OUTM	DW, src-block	When W = 0 (DW) ← (X) DIR = 0: X ← X + 1; DIR = 1: X ← X - 1 When W = 1 (DW + 1, DW) ← (X + 1, X) DIR = 0: X ← X + 2; DIR = 1: X ← X - 2	0	1	1	0	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1
<b>Addition/Subtraction</b>																											
ADD	reg, reg	reg ← reg + reg	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	2
	mem, reg	(mem) ← (mem) + reg	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2-4
	reg, mem	reg ← reg + (mem)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2-4
	reg, imm	reg ← reg + imm	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3-4
	mem, imm	(mem) ← (mem) + imm	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3-6
	acc, imm	When W = 0 AL ← AL + imm When W = 1 AW ← AW + imm	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	2-3

**Instruction Set (cont.)**

Mnemonic	Operand	Operation	Operation Code											Flags															
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	No. of Bytes	AC	CY	V	P	S	Z				
<b>Addition/Subtraction (cont.)</b>																													
ADD	reg, reg	reg ← reg + reg + CY	0	0	0	1	0	0	1	W	1	1	reg	reg	2	x	x	x	x	x	x	x	x	x	x	x	x	x	
	mem, reg	(mem) ← (mem) + reg + CY	0	0	0	1	0	0	0	W	mod	reg	mem	2-4	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	reg, mem	reg ← reg + (mem) + CY	0	0	0	1	0	0	1	W	mod	reg	mem	2-4	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	reg, imm	reg ← reg + imm + CY	1	0	0	0	0	S	W	1	1	0	1	0	reg	3-4	x	x	x	x	x	x	x	x	x	x	x	x	x
	mem, imm	(mem) ← (mem) + imm + CY	1	0	0	0	0	S	W	mod	0	1	0	mem	3-6	x	x	x	x	x	x	x	x	x	x	x	x	x	x
acc, imm	When W = 0 AL ← AL + imm + CY When W = 1 AW ← AW + imm + CY	0	0	0	1	0	1	0	W					2-3	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
<b>SUB</b>																													
SUB	reg, reg	reg ← reg - reg	0	0	1	0	1	0	1	W	1	1	reg	reg	2	x	x	x	x	x	x	x	x	x	x	x	x	x	
	mem, reg	(mem) ← (mem) - reg	0	0	1	0	1	0	0	W	mod	reg	mem	2-4	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	reg, mem	reg ← reg - (mem)	0	0	1	0	1	0	1	W	mod	reg	mem	2-4	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	reg, imm	reg ← reg - imm	1	0	0	0	0	S	W	1	1	0	1	reg	3-4	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	mem, imm	(mem) ← (mem) - imm	1	0	0	0	0	S	W	mod	1	0	1	mem	3-6	x	x	x	x	x	x	x	x	x	x	x	x	x	
acc, imm	When W = 0 AL ← AL - imm When W = 1 AW ← AW - imm	0	0	1	0	1	0	W					2-3	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
<b>SUBC</b>																													
SUBC	reg, reg	reg ← reg - reg - CY	0	0	0	1	0	1	0	1	W	1	1	reg	reg	2	x	x	x	x	x	x	x	x	x	x	x	x	
	mem, reg	(mem) ← (mem) - reg - CY	0	0	0	1	0	0	W	mod	reg	mem	2-4	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	reg, mem	reg ← reg - (mem) - CY	0	0	0	1	0	1	0	1	W	mod	reg	mem	2-4	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	reg, imm	reg ← reg - imm - CY	1	0	0	0	0	S	W	1	1	0	1	reg	3-4	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	mem, imm	(mem) ← (mem) - imm - CY	1	0	0	0	0	S	W	mod	0	1	1	mem	3-6	x	x	x	x	x	x	x	x	x	x	x	x	x	x
acc, imm	When W = 0 AL ← AL - imm - CY When W = 1 AW ← AW - imm - CY	0	0	0	1	1	0	W					2-3	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	



**Instruction Set (cont)**

Mnemonic	Operand	Operation	Operation Code 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0	No. of		Flags CY V P S Z
				Bytes	AC	
<b>BCD Operation</b>						
ADDAS		dst BCD string ← dst BCD string + src BCD string	0 0 0 0 1 1 1 1 0 0 1 0 0 0 0 0	2	u	x u u u u x
SUBAS		dst BCD string ← dst BCD string - src BCD string	0 0 0 0 1 1 1 1 0 0 1 0 0 0 1 0	2	u	x u u u u x
CMPS		dst BCD string - src BCD string	0 0 0 0 1 1 1 1 0 0 1 0 0 1 1 0	2	u	x u u u u x
ROL4	reg8	<div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">7</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">AL</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">0</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">ALL</div> <div style="margin-right: 5px;">→</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">Upper 4 bits</div> <div style="margin-right: 5px;">←</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">Lower 4 bits</div> <div style="margin-right: 5px;">←</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">reg</div> </div>	0 0 0 0 1 1 1 1 0 0 1 0 0 0 0 0 1 1 0 0 0 reg	3		
	mem8	<div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">7</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">AL</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">0</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">ALL</div> <div style="margin-right: 5px;">→</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">Upper 4 bits</div> <div style="margin-right: 5px;">←</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">Lower 4 bits</div> <div style="margin-right: 5px;">←</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">mem</div> </div>	0 0 0 0 1 1 1 1 0 0 1 0 1 0 0 0 mem	3-5		
ROR4	reg8	<div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">7</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">AL</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">0</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">ALL</div> <div style="margin-right: 5px;">→</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">Upper 4 bits</div> <div style="margin-right: 5px;">←</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">Lower 4 bits</div> <div style="margin-right: 5px;">←</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">reg</div> </div>	0 0 0 0 1 1 1 1 0 0 1 0 1 0 1 0 1 1 0 0 0 reg	3		
	mem8	<div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">7</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">AL</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">0</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">ALL</div> <div style="margin-right: 5px;">→</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">Upper 4 bits</div> <div style="margin-right: 5px;">←</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">Lower 4 bits</div> <div style="margin-right: 5px;">←</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">mem</div> </div>	0 0 0 0 1 1 1 1 0 0 1 0 1 0 1 0 mem	3-5		
<b>BCD Adjust</b>						
ADJBA		When (AL AND OFH) > 8 or AC = 1, AL ← AL + 6, AH ← AH + 1, AC ← 1, CY ← AC, AL ← AL AND OFH	0 0 1 1 0 1 1 1	1	x	x u u u u u
ADJAA		When (AL AND OFH) > 8 or AC = 1, AL ← AL + 6, CY ← CY OR AC, AC ← 1, When AL > 9FH, or CY = - AL ← AL + 60H, CY ← 1	0 0 1 0 0 1 1 1	1	x	x u x x x x
ADJBS		When (AL AND OFH) > 8 or AC = 1, CY ← AC, AL ← AL AND OFH	0 0 1 1 1 1 1 1	1	x	x u u u u u
ADJAS		When (AL AND OFH) > 8 or AC = 1, AL ← AL - 6, CY ← CY OR AC, AC ← 1, When AL > 9FH, or CY = - AL ← AL + 60H, CY ← 1	0 0 1 0 1 1 1 1	1	x	x u x x x x

### Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code										No. of Bytes			Flags											
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	AC	CY	V	P	S	Z			
<b>Increment/Decrement</b>																											
INC	reg8	reg8 ← reg8 + 1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	2	x	x	x	x	x	x	x	x	x
	mem	(mem) ← (mem) + 1	1	1	1	1	1	1	1	1	W	mod	0	0	0	0	mem	2-4	x	x	x	x	x	x	x	x	x
DEC	reg16	reg16 ← reg16 + 1	0	1	0	0	0	0	0	0	reg	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	reg8	reg8 ← reg8 - 1	1	1	1	1	1	1	1	0	1	0	0	1	reg	2	x	x	x	x	x	x	x	x	x	x	
	mem	(mem) ← (mem) - 1	1	1	1	1	1	1	1	W	mod	0	0	1	mem	2-4	x	x	x	x	x	x	x	x	x	x	
	reg16	reg16 ← reg16 - 1	0	1	0	0	1	reg	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
<b>Multiplication</b>																											
MULLU	reg8	AW ← AL x reg8 AH = 0: CY ← 0, V ← 0 AH ≠ 0: CY ← 1, V ← 1	1	1	1	1	0	1	1	0	1	1	0	1	0	0	reg	2	u	x	x	x	u	u	u	u	
	mem8	AW ← AL x (mem8) AH = 0: CY ← 0, V ← 0 AH ≠ 0: CY ← 1, V ← 1	1	1	1	1	0	1	1	0	mod	1	0	0	mem	2-4	u	x	x	u	u	u	u	u	u	u	
reg16	DW, AW ← AW x reg16 DW = 0: CY ← 0, V ← 0 DW ≠ 0: CY ← 1, V ← 1	1	1	1	1	0	1	1	1	1	1	1	0	0	reg	2	u	x	x	u	u	u	u	u	u	u	
	mem16	DW, AW ← AW x (mem16) DW = 0: CY ← 0, V ← 0 DW ≠ 0: CY ← 1, V ← 1	1	1	1	1	0	1	1	1	mod	1	0	0	mem	2-4	u	x	x	u	u	u	u	u	u	u	
MUL	reg8	AW ← AL x reg8 AH = AL sign expansion: CY ← 0, V ← 0 AH ≠ AL sign expansion: CY ← 1, V ← 1	1	1	1	1	0	1	1	0	1	1	1	0	1	reg	2	u	x	x	u	u	u	u	u	u	
	mem8	AW ← AL x (mem8) AH = AL sign expansion: CY ← 0, V ← 0 AH ≠ AL sign expansion: CY ← 1, V ← 1	1	1	1	1	0	1	1	0	mod	1	0	1	mem	2-4	u	x	x	u	u	u	u	u	u	u	
reg16	DW, AW ← AW x reg16 DW = AW sign expansion: CY ← 0, V ← 0 DW ≠ AW sign expansion: CY ← 1, V ← 1	1	1	1	1	0	1	1	1	1	1	1	0	1	reg	2	u	x	x	u	u	u	u	u	u	u	
	mem16	DW, AW ← AW x (mem16) DW = AW sign expansion: CY ← 0, V ← 0 DW ≠ AW sign expansion: CY ← 1, V ← 1	1	1	1	1	0	1	1	1	mod	1	0	1	mem	2-4	u	x	x	u	u	u	u	u	u	u	
reg16, reg16, imm8	reg16 ← reg16 x imm8 Product ≤ 16 bits: CY ← 0, V ← 0 Product > 16 bits: CY ← 1, V ← 1	0	1	1	0	1	0	1	1	1	1	1	1	1	reg	3	u	x	x	u	u	u	u	u	u	u	
reg16, mem16, imm8	reg16 ← (mem16) x imm8 Product ≤ 16 bits: CY ← 0, V ← 0 Product > 16 bits: CY ← 1, V ← 1	0	1	1	0	1	0	1	1	1	mod	reg	mem	3-5	u	x	x	u	u	u	u	u	u	u	u		

### Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code																Flags								
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	No. of Bytes	AC	CY	V	P	S	Z		
<b>Multiplication (cont)</b>																											
MUL (cont)	reg16, reg16, imm16	reg16 ← reg16 × imm16 Product ≤ 16 bits: CY ← 0, V ← 0 Product > 16 bits: CY ← 1, V ← 1	0	1	1	0	1	0	0	1	1	1	1	1	1	1	0	reg	reg	4	U	X	X	U	U	U	U
	reg16, mem16, imm16	reg16 ← (mem16) × imm16 Product ≤ 16 bits: CY ← 0, V ← 0 Product > 16 bits: CY ← 1, V ← 1	0	1	1	0	1	0	0	1	1	1	1	1	1	0	mod	reg	mem	4	U	X	X	U	U	U	U
<b>Unsigned Division</b>																											
DIVU	reg8	temp ← AW When temp ÷ reg8 > FFH (SP - 1, SP - 2) ← PSW, (SP - 3, SP - 4) ← PS (SP - 5, SP - 6) ← PC, SP ← SP - 6 IE ← 0, BRK ← 0, PS ← (3, 2), PC ← (1, 0) All other times AH ← temp %, reg8, AL ← temp ÷ reg8	1	1	1	1	0	1	1	0	1	1	1	1	1	0	reg	2	U	U	U	U	U	U	U	U	U
	mem8	temp ← AW When temp ÷ (mem8) > FFH (SP - 1, SP - 2) ← PSW, (SP - 3, SP - 4) ← PS (SP - 5, SP - 6) ← PC, SP ← SP - 6 IE ← 0, BRK ← 0, PS ← (3, 2), PC ← (1, 0) All other times AH ← temp %, (mem8), AL ← temp ÷ (mem8)	1	1	1	1	0	1	1	0	1	1	1	1	1	0	mem	2	U	U	U	U	U	U	U	U	U
	reg16	temp ← AW When temp ÷ reg16 > FFFFH (SP - 1, SP - 2) ← PSW, (SP - 3, SP - 4) ← PS (SP - 5, SP - 6) ← PC, SP ← SP - 6 IE ← 0, BRK ← 0, PS ← (3, 2), PC ← (1, 0) All other times AH ← temp %, reg16, AL ← temp ÷ reg16	1	1	1	1	0	1	1	1	1	1	1	1	1	0	reg	2	U	U	U	U	U	U	U	U	U
	mem16	temp ← AW When temp ÷ (mem16) > FFFFH (SP - 1, SP - 2) ← PSW, (SP - 3, SP - 4) ← PS (SP - 5, SP - 6) ← PC, SP ← SP - 6 IE ← 0, BRK ← 0, PS ← (3, 2), PC ← (1, 0) All other times AH ← temp %, (mem16), AL ← temp ÷ (mem16)	1	1	1	1	0	1	1	1	1	1	1	1	1	0	mem	2	U	U	U	U	U	U	U	U	U



Instruction Set (cont)		Operation Code										No. of Bytes			Flags													
Mnemonic	Operand	Operation	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	AC	CY	V	P	S	Z				
<b>Data Conversion</b>																												
CVTBD	AH ← AL - 0AH, AL ← AL % 0AH		1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	2	u	u	u	x	x	x	x
CVTDB	AH ← 0, AL ← AH x 0AH + AL		1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	1	0	2	u	u	u	x	x	x	x
CVTBW	When AL < 80H, AH ← 0, all other times AH ← FFH		1	0	0	1	1	0	0	0										1								
CVTWL	When AL < 8000H, DW ← 0, all other times DW ← FFFFH		1	0	0	1	1	0	0	1										1								
<b>Comparison</b>																												
CMP	reg - reg	reg - reg	0	0	1	1	0	1	0	1	W	1	1	1	1	1	1	1	1	1	reg	reg						
	mem, reg	(mem) - reg	0	0	1	1	0	1	0	0	W	mod	reg	mem	mem						2-4	x	x	x	x	x	x	x
	reg, mem	reg - (mem)	0	0	1	1	0	1	0	1	W	mod	reg	mem	mem						2-4	x	x	x	x	x	x	x
	reg, imm	reg - imm	1	0	0	0	0	0	0	0	S	W	1	1	1	1	1	1	1	1	reg	3-4	x	x	x	x	x	x
	mem, imm	(mem) - imm	1	0	0	0	0	0	0	0	S	W	mod	1	1	1	1	1	1	1	mem	3-6	x	x	x	x	x	x
	acc, imm	When W = 0, AL - imm When W = 1, AW - imm	0	0	1	1	1	0	1	0	W										2-3	x	x	x	x	x	x	x
<b>Complement</b>																												
NOT	reg ← reg	reg ← reg	1	1	1	1	0	1	1	W	1	1	0	1	0	1	0	1	0	1	reg	2						
	mem	(mem) ← (mem)	1	1	1	1	0	1	1	W	mod	0	1	0	1	0	1	0	1	0	mem	2-4						
NEG	reg ← reg + 1	reg ← reg + 1	1	1	1	0	1	1	W	1	1	0	1	1	0	1	1	0	1	1	reg	2	x	x	x	x	x	x
	mem	(mem) ← (mem) + 1	1	1	1	0	1	1	W	mod	0	1	1	0	1	1	0	1	1	mem	2-4	x	x	x	x	x	x	
<b>Logical Operation</b>																												
TEST	reg AND reg	reg AND reg	1	0	0	0	1	0	1	W	1	1	1	1	1	1	1	1	1	1	reg	2	u	0	0	x	x	x
	mem, reg or reg, mem	(mem) AND reg	1	0	0	0	1	0	1	W	mod	reg	mem	mem							2-4	u	0	0	x	x	x	x
	reg, imm	reg AND imm	1	1	1	1	0	1	1	W	1	1	0	0	0	0	0	0	0	0	reg	3-4	u	0	0	x	x	x
	mem, imm	(mem) AND imm	1	1	1	1	0	1	1	W	mod	0	0	0	0	0	0	0	0	0	mem	3-6	u	0	0	x	x	x
	acc, imm	When W = 0, AL AND imm8 When W = 1, AW AND imm8	1	0	1	0	1	0	0	W											2-3	u	0	0	x	x	x	x
AND	reg ← reg AND reg	reg ← reg AND reg	0	0	1	0	0	0	1	W	1	1	1	1	1	1	1	1	1	1	reg	2	u	0	0	x	x	x
	mem, reg	(mem) ← (mem) AND reg	0	0	1	0	0	0	0	W	mod	reg	mem	mem							2-4	u	0	0	x	x	x	x
	reg, mem	reg ← reg AND (mem)	0	0	1	0	0	0	1	W	mod	reg	mem	mem							2-4	u	0	0	x	x	x	x
	reg, imm	reg ← reg AND imm	1	0	0	0	0	0	0	W	1	1	0	0	0	0	0	0	0	0	reg	3-4	u	0	0	x	x	x
	mem, imm	(mem) ← (mem) AND imm	1	0	0	0	0	0	0	W	mod	1	1	0	0	0	0	0	0	0	mem	3-6	u	0	0	x	x	x
	acc, imm	When W = 0, AL AND imm8 When W = 1, AW AND imm8	1	0	1	0	1	0	0	W											2-3	u	0	0	x	x	x	x
	reg, reg	reg ← reg AND reg	0	0	1	0	0	0	1	W	1	1	1	1	1	1	1	1	1	1	reg	2	u	0	0	x	x	x
	mem, reg	(mem) ← (mem) AND reg	0	0	1	0	0	0	0	W	mod	reg	mem	mem							2-4	u	0	0	x	x	x	x
	reg, mem	reg ← reg AND (mem)	0	0	1	0	0	0	1	W	mod	reg	mem	mem							2-4	u	0	0	x	x	x	x
	reg, imm	reg ← reg AND imm	1	0	0	0	0	0	0	W	1	1	0	0	0	0	0	0	0	0	reg	3-4	u	0	0	x	x	x
	mem, imm	(mem) ← (mem) AND imm	1	0	0	0	0	0	0	W	mod	1	0	0	0	0	0	0	0	0	mem	3-6	u	0	0	x	x	x
	acc, imm	When W = 0, AL AND imm8 When W = 1, AW AND imm8	0	0	1	0	1	0	0	W											2-3	u	0	0	x	x	x	x

### Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code										No. of Bytes				Flags									
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	AC	CY	V	P	S	Z		
<b>Logical Operation (cont)</b>																										
OR	reg, reg	reg ← reg OR reg	0	0	0	0	1	0	1	0	1	W	1	1	reg	2	u	0	0	x	x	x	x			
	mem, reg	(mem) ← (mem) OR reg	0	0	0	0	1	0	0	W	mod	reg	mem	2-4	u	0	0	x	x	x	x					
	reg, mem	reg ← reg OR (mem)	0	0	0	0	1	0	1	W	mod	reg	mem	2-4	u	0	0	x	x	x	x					
	reg, imm	reg ← reg OR imm	1	0	0	0	0	0	0	W	1	1	0	0	1	reg	3-4	u	0	0	x	x	x			
	mem, imm	(mem) ← (mem) OR imm	1	0	0	0	0	0	W	mod	0	0	1	mem	3-6	u	0	0	x	x	x	x				
acc, imm	When W = 0, AL ← AL OR imm8 When W = 1, AW ← AW OR imm16	0	0	0	0	1	0	W	2-3	u	0	0	x	x	x	x										
XOR	reg, reg	reg ← reg XOR reg	0	0	1	0	0	1	W	1	1	reg	2	u	0	0	x	x	x	x						
	mem, reg	(mem) ← (mem) XOR reg	0	0	1	0	0	W	mod	reg	mem	2-4	u	0	0	x	x	x	x							
	reg, mem	reg ← reg XOR (mem)	0	0	1	0	0	1	W	mod	reg	mem	2-4	u	0	0	x	x	x	x						
	reg, imm	reg ← reg XOR imm	1	0	0	0	0	0	W	1	1	1	0	reg	3-4	u	0	0	x	x	x	x				
	mem, imm	(mem) ← (mem) XOR imm	1	0	0	0	0	0	W	mod	1	1	0	mem	3-6	u	0	0	x	x	x	x				
acc, imm	When W = 0, AL ← AL XOR imm8 When W = 1, AW ← AW XOR imm16	0	0	1	0	1	0	W	2-3	u	0	0	x	x	x	x										
<b>Bit Operation</b>																										
TEST1	reg8, CL	reg8 bit no. CL = 0: Z ← 1 reg8 bit no. CL = 1: Z ← 0	0	0	0	1	0	0	0	1	1	0	0	0	reg	3	u	0	0	0	u	u	x			
	mem8, CL	(mem8) bit no. CL = 0: Z ← 1 (mem8) bit no. CL = 1: Z ← 0	0	0	0	1	0	0	0	0	mod	0	0	0	mem	3-5	u	0	0	0	u	u	x			
	reg16, CL	reg16 bit no. CL = 0: Z ← 1 reg16 bit no. CL = 1: Z ← 0	0	0	0	1	0	0	0	1	1	1	0	0	reg	3	u	0	0	0	u	u	x			
	mem16, CL	(mem16) bit no. CL = 0: Z ← 1 (mem16) bit no. CL = 1: Z ← 0	0	0	0	1	0	0	0	1	mod	0	0	0	mem	3-5	u	0	0	0	u	u	x			
	reg8, imm3	reg8 bit no. imm3 = 0: Z ← 1 reg8 bit no. imm3 = 1: Z ← 0	0	0	0	1	1	0	0	0	1	1	0	0	reg	4	u	0	0	0	u	u	x			
	mem8, imm3	(mem8) bit no. imm3 = 0: Z ← 1 (mem8) bit no. imm3 = 1: Z ← 0	0	0	0	1	1	0	0	0	mod	0	0	0	mem	4-6	u	0	0	0	u	u	x			
	reg16, imm4	reg16 bit no. imm4 = 0: Z ← 1 reg16 bit no. imm4 = 1: Z ← 0	0	0	0	1	1	0	0	1	1	1	0	0	reg	4	u	0	0	0	u	u	x			
	mem16, imm4	(mem16) bit no. imm4 = 0: Z ← 1 (mem16) bit no. imm4 = 1: Z ← 0	0	0	0	1	1	0	0	1	mod	0	0	0	mem	4-6	u	0	0	0	u	u	x			
	*Note: First byte = 0FH 2nd byte* 3rd byte*																									

**Instruction Set (cont)**

Mnemonic	Operand	Operation	Operation Code										No. of Bytes			Flags						
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	AC	CY	V	P
NOT1	reg8, CL	reg8 bit no. CL ← reg8 bit no. CL	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	reg	3			
	mem8, CL	(mem8) bit no. CL ← (mem8) bit no. CL	0	0	0	1	0	1	1	0	mod	0	0	0	0	mem	3-5					
	reg16, CL	reg16 bit no. CL ← reg16 bit no. CL	0	0	0	1	0	1	1	1	1	0	0	0	0	reg	3					
	mem16, CL	(mem16) bit no. CL ← (mem16) bit no. CL	0	0	0	1	0	1	1	1	mod	0	0	0	0	mem	3-5					
	reg8, imm3	reg8 bit no. imm3 ← reg8 bit no. imm3	0	0	0	1	1	1	0	1	1	0	0	0	0	reg	4					
	mem8, imm3	(mem8) bit no. imm3 ← (mem8) bit no. imm3	0	0	0	1	1	1	0	mod	0	0	0	0	0	mem	4-6					
	reg16, imm4	reg16 bit no. imm4 ← (reg16) bit no. imm4	0	0	0	1	1	1	1	1	1	0	0	0	0	reg	4					
	mem16, imm4	(mem16) bit no. imm4 ← (mem16) bit no. imm4	0	0	0	1	1	1	1	1	mod	0	0	0	0	mem	4-6					
	CY	CY ← CY	1	1	1	1	0	1	0	1	2nd byte*						1	x				
													*Note: First byte = 0FH									
CLR1	reg8, CL	reg8 bit no. CL ← 0	0	0	0	1	0	0	1	0	1	1	0	0	0	reg	3					
	mem8, CL	(mem8) bit no. CL ← 0	0	0	0	1	0	1	0	mod	0	0	0	0	mem	3-5						
	reg16, CL	reg16 bit no. CL ← 0	0	0	0	1	0	0	1	1	1	0	0	0	reg	3						
	mem16, CL	(mem16) bit no. CL ← 0	0	0	0	1	0	0	1	1	mod	0	0	0	mem	3-5						
	reg8, imm3	reg8 bit no. imm3 ← 0	0	0	0	1	1	0	1	0	1	0	0	0	reg	4						
	mem8, imm3	(mem8) bit no. imm3 ← 0	0	0	0	1	1	0	1	0	mod	0	0	0	mem	4-6						
	reg16, imm4	reg16 bit no. imm4 ← 0	0	0	0	1	0	1	1	1	1	0	0	0	reg	4						
	mem16, imm4	(mem16) bit no. imm4 ← 0	0	0	0	1	0	1	1	1	mod	0	0	0	mem	4-6						
	CY	CY ← 0	1	1	1	1	0	0	0	2nd byte*						1	0					
	DIR	DIR ← 0	1	1	1	1	1	0	0	2nd byte*						1						
												*Note: First byte = 0FH										

### Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code													No. of Bytes				Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	AC	CY	V	P	S	Z	
<b>Bit Operation (cont)</b>																									
SETI	reg8, CL	reg8 bit no. CL ← 1	0	0	0	1	0	1	0	0	1	1	0	0	0	0	reg	3							
	mem8, CL	(mem8) bit no. CL ← 1	0	0	0	1	0	1	0	0	mod	0	0	0	0	0	mem	3-5							
	reg16, CL	reg16 bit no. CL ← 1	0	0	0	1	0	1	1	1	0	0	0	0	0	reg	3								
	mem16, CL	(mem16) bit no. CL ← 1	0	0	0	1	0	1	0	1	0	0	0	0	0	mem	3-5								
	reg8, imm3	reg8 bit no. imm3 ← 1	0	0	0	1	1	0	0	1	1	0	0	0	0	reg	4								
	mem8, imm3	(mem8) bit no. imm3 ← 1	0	0	0	1	1	0	0	mod	0	0	0	0	0	mem	4-6								
	reg16, imm4	reg16 bit no. imm4 ← 1	0	0	0	1	1	0	1	1	0	0	0	0	0	reg	4								
	mem16, imm4	(mem16) bit no. imm4 ← 1	0	0	0	1	1	0	1	0	1	0	0	0	0	mem	4-6								
*Note: First byte = 0FH      2nd byte*      3rd byte*																									
CY	CY ← 1		1	1	1	1	0	0	1									1							
DIR	DIR ← 1		1	1	1	1	1	0	1									1							
<b>Shift</b>																									
SHL	reg, 1	CY ← MSB of reg, reg ← reg x 2 When MSB of reg ≠ CY, V ← 1 When MSB of reg = CY, V ← 0	1	1	0	1	0	0	0	W	1	1	0	0	0	reg	2	U	X	X	X	X	X	X	
	mem, 1	CY ← MSB of (mem), (mem) ← (mem) x 2 When MSB of (mem) ≠ CY, V ← 1 When MSB of (mem) = CY, V ← 0	1	1	0	1	0	0	0	W	mod	1	0	0	0	mem	2-4	U	X	X	X	X	X	X	
	reg, CL	temp ← CL, while temp ≠ 0, repeat this operation, CY ← MSB of reg, reg ← reg x 2, temp ← temp - 1	1	1	0	1	0	0	1	W	1	1	1	0	0	reg	2	U	X	X	X	X	X	X	
	mem, CL	temp ← CL, while temp ≠ 0, repeat this operation, CY ← MSB of (mem), (mem) ← (mem) x 2, temp ← temp - 1	1	1	0	1	0	0	1	W	mod	1	0	0	0	mem	2-4	U	X	X	X	X	X		
	reg, imm8	temp ← imm8, while temp ≠ 0, repeat this operation, CY ← MSB of reg, reg ← reg x 2, temp ← temp - 1	1	1	0	0	0	0	0	W	1	1	1	0	0	reg	3	U	X	X	X	X	X		
	mem, imm8	temp ← imm8, while temp ≠ 0, repeat this operation, CY ← MSB of (mem), (mem) ← (mem) x 2, temp ← temp - 1	1	1	0	0	0	0	0	W	mod	1	0	0	0	mem	3-5	U	X	X	X	X	X		
SHR	reg, 1	CY ← LSB of reg, reg ← reg ÷ 2 When MSB of reg ≠ bit following MSB of reg, V ← 1 When MSB of reg = bit following MSB of reg, V ← 0	1	1	0	1	0	0	0	W	1	1	0	1	0	reg	2	U	X	X	X	X	X		



## Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code										No. of Bytes	Flags										
			7	6	5	4	3	2	1	0	7	6		5	4	3	2	1	0	AC	CY	V	P	S
<b>Shift (cont)</b>																								
SHR (cont)	mem, 1	CY ← LSB of (mem), (mem) ← (mem) ÷ 2 When MSB of (mem) ≠ bit following MSB of (mem): V ← 1 When MSB of (mem) = bit following MSB of (mem): V ← 0	1	1	0	1	0	0	0	W	mod	1	0	1	mem	2-4	u	x	x	x	x	x	x	
	reg, CL	temp ← CL, while temp ≠ 0, repeat this operation, CY ← LSB of reg, reg ← reg ÷ 2, temp ← temp - 1	1	1	0	1	0	0	1	W	1	1	1	0	1	reg	2	u	x	x	x	x	x	
	mem, CL	temp ← CL, while temp ≠ 0, repeat this operation, CY ← LSB of (mem), (mem) ← (mem) ÷ 2, temp ← temp - 1	1	1	0	1	0	0	1	W	mod	1	0	1	mem	2-4	u	x	x	x	x	x	x	
	reg, imm8	temp ← imm8, while temp ≠ 0, repeat this operation, CY ← LSB of reg, reg ← reg ÷ 2, temp ← temp - 1	1	1	0	0	0	0	0	W	1	1	1	0	1	reg	3	u	x	x	x	x	x	
	mem, imm8	temp ← imm8, while temp ≠ 0, repeat this operation, CY ← LSB of (mem), (mem) ← (mem) ÷ 2, temp ← temp - 1	1	1	0	0	0	0	0	W	mod	1	0	1	mem	3-5	u	x	x	x	x	x	x	
		n: number of shifts																						
SHRA	reg, 1	CY ← LSB of reg, reg ← reg ÷ 2, V ← 0 MSB of operand does not change	1	1	0	1	0	0	0	W	1	1	1	1	1	1	reg	2	u	x	0	x	x	x
	mem, 1	CY ← LSB of (mem), (mem) ← (mem) ÷ 2, V ← 0, MSB of operand does not change	1	1	0	1	0	0	0	W	mod	1	1	1	1	1	mem	2-4	u	x	0	x	x	x
	reg, CL	temp ← CL, while temp ≠ 0, repeat this operation, CY ← LSB of reg, reg ← reg ÷ 2, temp ← temp - 1 MSB of operand does not change	1	1	0	1	0	0	1	W	1	1	1	1	1	1	reg	2	u	x	x	x	x	x
	mem, CL	temp ← CL, while temp ≠ 0, repeat this operation, CY ← LSB of (mem), (mem) ← (mem) ÷ 2, temp ← temp - 1 MSB of operand does not change	1	1	0	1	0	0	1	W	mod	1	1	1	1	1	mem	2-4	u	x	x	x	x	x
	reg, imm8	temp ← imm8, while temp ≠ 0, repeat this operation, CY ← LSB of reg, reg ← reg ÷ 2, temp ← temp - 1 MSB of operand does not change	1	1	0	0	0	0	0	W	1	1	1	1	1	1	reg	3	u	x	x	x	x	x
	mem, imm8	temp ← imm8, while temp ≠ 0, repeat this operation, CY ← LSB of (mem), (mem) ← (mem) ÷ 2, temp ← temp - 1 MSB of operand does not change	1	1	0	0	0	0	0	W	mod	1	1	1	1	1	mem	3-5	u	x	x	x	x	x
		n: number of shifts																						





### Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code										No. of Bytes			Flags							
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	AC	CY	V	P	S
<b>Rotate (cont)</b>																							
RORC	reg, 1	tmpcy ← CY, CY ← LSB of reg reg ← reg ÷ 2, MSB of reg ← tmpcy MSB of reg ≠ bit following MSB of reg: V ← 1 MSB of reg = bit following MSB of reg: V ← 0	1	1	0	1	0	0	0	W	1	1	0	1	1	reg	2					x	x
	mem, 1	tmpcy ← CY, CY ← LSB of (mem) (mem) ← (mem) ÷ 2, MSB of (mem) ← tmpcy MSB of (mem) ≠ bit following MSB of (mem): V ← 1 MSB of (mem) = bit following MSB of (mem): V ← 0	1	1	0	1	0	0	0	W	mod	0	1	1	mem	2-4					x	x	
	reg, CL	temp ← CL, while temp ≠ 0 repeat this operation, tmpcy ← CY, CY ← LSB of reg, reg ← reg ÷ 2, MSB of reg ← tmpcy, temp ← temp - 1	1	1	0	1	0	0	1	W	1	1	0	1	1	reg	2					x	u
	mem, CL	temp ← CL, while temp ≠ 0, repeat this operation, tmpcy ← CY, CY ← LSB of (mem), (mem) ← (mem) ÷ 2 MSB of (mem) ← tmpcy, temp ← temp - 1	1	1	0	1	0	0	1	W	mod	0	1	1	mem	2-4					x	u	
	reg, imm8	temp ← imm8, while temp ≠ 0 repeat this operation, tmpcy ← CY, CY ← LSB of reg, reg ← reg ÷ 2 MSB of reg ← tmpcy, temp ← temp - 1	1	1	0	0	0	0	0	W	1	1	0	1	1	reg	3					x	u
	mem, imm8	temp ← imm8, while temp ≠ 0, repeat this operation, tmpcy ← CY, CY ← LSB of (mem), (mem) ← (mem) ÷ 2 MSB of (mem) ← tmpcy, temp ← temp - 1	1	1	0	0	0	0	0	W	mod	0	1	1	mem	3-5					x	u	
<b>Subroutine Control Transfer</b>																							
CALL	near-proc	SP ← 1, SP - 2) ← PC, SP ← SP - 2 PC ← PC + disp	1	1	1	0	1	0	0	0						3							
	regptr16	(SP - 1, SP - 2) ← PC, SP ← SP - 2 PC ← regptr16	1	1	1	1	1	1	1	1	1	0	1	0	reg	2							
	memptr16	(SP - 1, SP - 2) ← PC, SP ← SP - 2 PC ← (memptr16)	1	1	1	1	1	1	1	1	1	0	1	0	mem	2-4							
	far-proc	(SP - 1, SP - 2) ← PS, (SP - 3, SP - 4) ← PC SP ← SP - 4, PS ← seg, PC ← offset	1	0	0	1	1	0	1	0	1	0			5								
	memptr32	(SP - 1, SP - 2) ← PS, (SP - 3, SP - 4) ← PC SP ← SP - 4, PS ← (memptr32 + 2), PC ← (memptr32)	1	1	1	1	1	1	1	1	1	1	1	1	mem	2-4							

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code										Flags																	
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	No. of Bytes	AC	CY	V	P	S	Z					
<b>Subroutine Control Transfer (cont)</b>																														
RET		PC ← (SP + 1, SP), SP ← SP + 2	1	1	0	0	0	1	1																					
	pop-value	PC ← (SP + 1, SP) SP ← SP + 2, SP ← SP + pop-value	3	1	1	0	0	0	1	0																				
		PC ← (SP + 1, SP), PS ← (SP + 3, SP + 2) SP ← SP + 4	1	1	1	0	0	1	0	1	1																			
	pop-value	PC ← (SP + 1, SP), PS ← (SP + 3, SP + 2) SP ← SP + 4, SP ← SP + pop-value	3	1	1	0	0	1	0	1	0																			
<b>Stack Manipulation</b>																														
PUSH	mem16	(SP - 1, SP - 2) ← (mem16), SP ← SP - 2	2	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	reg16	(SP - 1, SP - 2) ← reg16, SP ← SP - 2	1	0	1	0	1	0	reg																					
	sreg	(SP - 1, SP - 2) ← sreg, SP ← SP - 2	1	0	0	0	sreg	1	1	0																				
	PSW	(SP - 1, SP - 2) ← PSW, SP ← SP - 2	1	1	0	0	1	1	0	0	0																			
	R	Push registers on the stack	1	0	1	1	0	0	0	0	0																			
	imm	(SP - 1, SP - 2) ← imm SP ← SP - 2, When S = 1, sign extension	2	3	0	1	1	0	1	0	1	0	S	0																
POP	mem16	(mem16) ← (SP + 1, SP), SP ← SP + 2	2	4	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	reg16	reg16 ← (SP + 1, SP), SP ← SP + 2	1	0	1	0	1	1	reg																					
	sreg	sreg ← (SP + 1, SP) sreg : SS, DS0, DS1 SP ← SP + 2	1	0	0	0	sreg	1	1	1																				
	PSW	PSW ← (SP + 1, SP), SP ← SP + 2	1	1	0	0	1	1	0	1	0	1																		
	R	Pop registers from the stack	1	0	1	1	0	0	0	1	1																			
PREPARE	imm16, imm8	Prepare new stack frame	4	1	1	0	0	1	0	0	0	0																		
		* imm8 = 0: 16 imm8 > 1: 25 + 16 (imm8 - 1)																												
DISPOSE		Dispose of stack frame	1	1	0	0	1	0	0	1																				
<b>Branch</b>																														
BR	near-label	PC ← PC + disp	3	1	1	0	1	0	0	1																				
	short-label	PC ← PC + ext-disp8	2	1	1	0	1	0	1	1																				
	reg16	PC ← reg16	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	memptr16	PC ← (memptr16)	2	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	far-label	PS ← seg, PC ← offset	5	1	1	0	1	0	1	0																				
	memptr32	PS ← (memptr32 + 2), PC ← (memptr32)	2	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

### Instruction Set (cont)

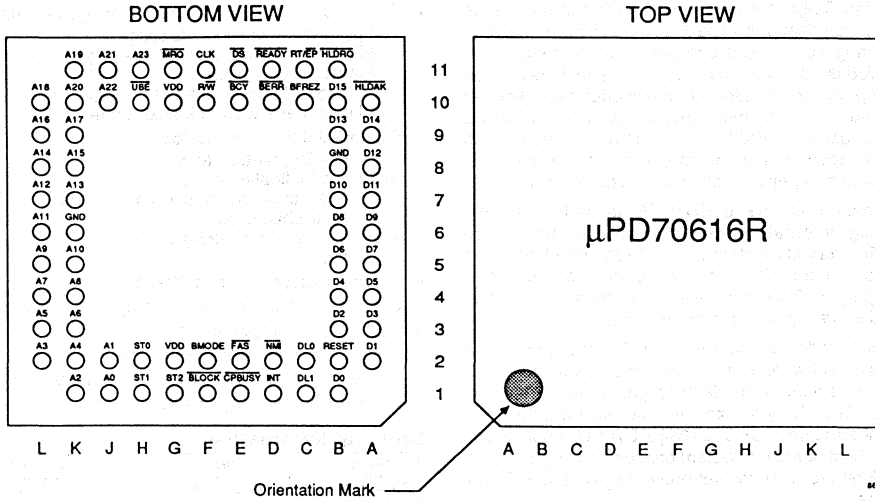
Mnemonic	Operand	Operation	Operation Code										No. of Bytes	Flags											
			7	6	5	4	3	2	1	0	7	6		5	4	3	2	1	0	AC	CV	VP	SS	Z	
<b>Conditional Branch</b>																									
BV	short-label	if V = 1, PC ← PC + ext-disp8	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2			
BNV	short-label	if V = 0, PC ← PC + ext-disp8	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2			
BC, BL	short-label	if CY = 1, PC ← PC + ext-disp8	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2			
BNC, BNL	short-label	if CY = 0, PC ← PC + ext-disp8	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2			
BE, BZ	short-label	if Z = 1, PC ← PC + ext-disp8	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2			
BNE, BNZ	short-label	if Z = 0, PC ← PC + ext-disp8	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2			
BNH	short-label	if CY OR Z = 1, PC ← PC + ext-disp8	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2			
BH	short-label	if CY OR Z = 0, PC ← PC + ext-disp8	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2			
BN	short-label	if S = 1, PC ← PC + ext-disp8	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2			
BP	short-label	if S = 0, PC ← PC + ext-disp8	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2			
BPE	short-label	if P = 1, PC ← PC + ext-disp8	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2			
BPO	short-label	if P = 0, PC ← PC + ext-disp8	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	2			
BLT	short-label	if S XOR V = 1, PC ← PC + ext-disp8	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2			
BGE	short-label	if S XOR V = 0, PC ← PC + ext-disp8	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2			
BLE	short-label	if (S XOR V) OR Z = 1, PC ← PC + ext-disp8	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2			
BGT	short-label	if (S XOR V) OR Z = 0, PC ← PC + ext-disp8	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	2			
DBNZNE	short-label	CW ← CW - 1 if Z = 0 and CW ≠ 0, PC ← PC + ext-disp8	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2			
DBNZE	short-label	CW ← CW - 1 if Z = 1 and CW ≠ 0, PC ← PC + ext-disp8	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2			
DBNZ	short-label	CW ← CW - 1 if CW ≠ 0, PC ← PC + ext-disp8	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2			
BCWZ	short-label	if CW = 0, PC ← PC + ext-disp8	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	2			
BTCLR	sfr, imm3 short-label	if bit no. imm3 of (sfr) = 1, PC ← PC + ext-disp8 bit no. imm3 of (sfr) ← 0	0	0	0	0	1	1	1	0	1	1	0	1	1	0	0	0	0	0	0	5			
<b>Interrupt</b>																									
BRK	3	(SP - 1, SP - 2) ← PSW, (SP - 3, SP - 4) ← PS, (SP - 5, SP - 6) ← PC, SP ← SP - 6 IE ← 0, BRK ← 0 PS ← (15, 14), PC ← (13, 12)	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1			
imm8 (≠ 3)		(SP - 1, SP - 2) ← PSW, (SP - 3, SP - 4) ← PS, (SP - 5, SP - 6) ← PC, SP ← SP - 6 IE ← 0, BRK ← 0 PC ← (n x 4 + 1, n x 4) PS ← (n x 4 + 3, n x 4 + 2) n = imm8	1	1	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	2			







Pin Identification



86-009

Pin	Function	Pin	Function	Pin	Function	Pin	Function
A2	D1	B9	D13	F10	R/W	K4	A8
A3	D3	B10	D15	F11	CLK	K5	A10
A4	D5	B11	HLDRQ	G1	ST2	K6	GND
A5	D7	C1	DL1	G2	V <sub>DD</sub>	K7	A13
A6	D9	C2	DL0	G10	V <sub>DD</sub>	K8	A15
A7	D11	C10	BFREZ	G11	MRQ	K9	A17
A8	D12	C11	RT/EP	H1	ST1	K10	A20
A9	D14	D1	INT	H2	ST0	K11	A19
A10	HLDAK	D2	NMI	H10	UBE	L2	A3
B1	D0	D10	BERR	H11	A23	L3	A5
B2	RESET	D11	READY	J1	A0	L4	A7
B3	D2	E1	CPBUSY	J2	A1	L5	A9
B4	D4	E2	FAS	J10	A22	L6	A11
B5	D6	E10	BCY	J11	A21	L7	A12
B6	D8	E11	DS	K1	A2	L8	A14
B7	D10	F1	BLOCK (MSMAT)	K2	A4	L9	A16
B8	GND	F2	BMODE (FRM)	K3	A6	L10	A18

86-296

### Pin Identification

Symbol		I/O	Function
A23-A0	Address bus	3-state OUT	24-bit address bus
D15-D0	Data bus	3-state I/O	16-bit bidirectional data bus
ST2-ST0	Bus Status	3-state OUT	bus cycle access type
MRQ*	Memory Request	3-state OUT	active low memory access strobe
R/W*	Read/Write	3-state OUT	bus cycle direction output
DS*	Data Strobe	3-state OUT	active low data strobe
BCY*	Bus Cycle	3-state OUT	active low bus cycle signal
DL1-DL0	Data Length	3-state OUT	single mode data length, string mode data direction signals
FAS*	First Data Access Status	OUT	first data access/subsequent data access signal
UBE*	Upper Byte Enable	3-state OUT	valid high order 8-bit data bus (D15-D8) signal
READY*	Ready	IN	bus cycle ready input
BMODE (FRM)	Bus Mode (Functional Redundancy Monitor)	IN	processor mode select input 0 V : master mode, short bus cycle 5 V : master mode, normal bus cycle 10V : check mode, normal bus cycle
BLOCK* (MSMAT*)	Bus Lock (Mismatch)	OUT	active low master mode bus lock output signal check mode mismatch error output signal
BERR*	Bus Error	IN	active low bus error input
BFREZ	Bus Freeze	IN	active high bus freeze command
RT/EP*	Retry/Exception	IN	retry/exception decision input
NMI*	Non-Maskable Interrupt	IN	active low non-maskable interrupt input
INT	Maskable Interrupt	IN	active high maskable interrupt input
HLDRQ*	Hold Request	IN	active low processor address, data and control bus request
HLDK*	Hold Acknowledge	OUT	active low bus external request acknowledgment
CPBUSY*	Coprocessor Busy	IN	coprocessor status polling input
RESET	Reset	IN	active high reset input
CLK	Clock	IN	processor clock input
V <sub>DD</sub>	Power	-	+5 V power supply
GND	Ground	-	ground

**Table of Contents**

<b>Section 1</b>	<b>Design Information</b> .....	<b>3.317</b>
	Pin Function .....	3.317
	AC/DC Specifications .....	3.323
	Timing Waveforms .....	3.324
<b>Section 2</b>	<b>Pipeline Operation</b> .....	<b>3.329</b>
	Bus Control Unit (BCU) .....	3.330
	Prefetch Unit (PFU) .....	3.330
	Instruction Decode Unit (IDU) .....	3.330
	Effective Address Generator (EAG) .....	3.330
	Memory Management Unit (MMU) .....	3.330
	Execution Unit (EXU) .....	3.330
<b>Section 3</b>	<b>μPD70616 Architecture</b> .....	<b>3.331</b>
	Program Register Set .....	3.331
	Privileged Register Set .....	3.333
	Memory, I/O Organization .....	3.339
	Data Types .....	3.339
	Addressing Modes .....	3.341
	Demand Paged Virtual Memory .....	3.347
	Operating System Support .....	3.351
	Interrupts and Exceptions .....	3.353
	μPD70616 Instruction Set .....	3.355
	Software Debug Support .....	3.358
	V20/V30 Emulation Mode .....	3.360
	FRM (Functional Redundancy Monitor) .....	3.362
<b>Section 4</b>	<b>Bus Operation</b> .....	<b>3.364</b>
	Bus Cycle Modes .....	3.364
	Reset Timing .....	3.335
	Memory Read Timing .....	3.367
	Memory Write Timing .....	3.369
	I/O Read Timing .....	3.371
	I/O Write Timing .....	3.373
	Back-to-Back I/O Bus Cycles .....	3.375
	Bus Hold Timing .....	3.376
<b>Section 5</b>	<b>Instruction Set</b> .....	<b>3.377</b>
	Instruction Format .....	3.377
	Instruction Set Description .....	3.379
<b>Appendix</b>	<b>Privileged Registers, PSW, ATE, PTE Formats</b> .....	<b>3.384</b>

# Section 1 Design Information

### Pin Functions

This section describes the operation of the μPD70616 address bus, data bus and control interface. Inputs and outputs are considered at a logic "0" level when a low level signal is present. Likewise, a logic "1" is represented by a high level signal. Bus states are defined and measured from the rising edge of the clock to the rising edge of the next clock.

**ST2-ST0 [Bus Status].....** 3-state outputs

The ST2-ST0 outputs (in conjunction with the MRQ\* output) indicate the type of bus cycle being performed.

Two separate bus access modes are supported by the μPD70616. String mode bus accesses occur during bus cycles for variable length data types. All other bus cycles are single mode.

MRQ	ST2	ST1	ST0	Bus Status	
0	0	0	0	Reserved for Future Use	(Single)
0	0	0	1	String Mode Data Access	(String)
0	0	1	0	Short Path Data Access	(Single)
0	0	1	1	Single Mode Data Access	(Single)
0	1	0	0	System Base Table Access	(Single)
0	1	0	1	Translation Table Access	(Single)
0	1	1	0	Demand Mode Instruction Fetch	
0	1	1	1	Instruction Prefetch	
1	0	0	0	Reserved for Future Use	(Single)
1	0	0	1	String Mode I/O Access	(String)
1	0	1	0	Reserved for Future Use	
1	0	1	1	Single Mode I/O Access	(Single)
1	1	0	0	Machine Fault Acknowledge	
1	1	0	1	Halt Acknowledge	
1	1	1	0	Interrupt Acknowledge	(Single)
1	1	1	1	Reserved for Future Use	

String Mode: variable length data type (character string and bit string) bus cycles  
 Single Mode: all other data access cycles

86-237

**String Mode Memory Access.....** (String Mode)

The processor is performing a character or bit string data access in the memory address space.

**Short Path Memory Access.....** (Single Mode)

A short path memory bus status is substituted for a single mode data access when a read access can be satisfied by the write data for the current bus cycle.

**Single Mode Memory Access.....** (Single Mode)

The processor is performing a single mode access in the memory address space.

**System Base Table Access.....** (Single Mode)

The system base table is being accessed in response to an interrupt or exception condition.

**Translation Table Access.....** (Single Mode)

An area table entry (ATE) or page table entry (PTE) access is taking place for the purpose of address translation.

**Demand Mode Instruction Fetch**

The instruction prefetch queue has been flushed by a control transfer and a demand mode instruction fetch is taking place.

**Instruction Prefetch**

An instruction fetch for the 16-byte instruction prefetch queue is taking place.

**String Mode I/O Access..... (String Mode)**

The processor is performing a character or bit string access in the I/O address space.

**Single Mode I/O Access..... (Single Mode)**

The processor is performing a single mode access in the I/O address space.

**Machine Fault Acknowledge**

The machine fault acknowledge status is generated as a result of a fatal double bus error. This status is presented at the end of the bus cycle and remains until RESET is asserted. DS\* is not asserted during a machine fault acknowledge bus cycle.

**Halt Acknowledge**

A privileged HALT instruction has executed and the processor has halted. The halt status will be output until an interrupt occurs or the processor is reset. DS\* is not asserted during a halt acknowledge bus cycle.

**Interrupt Acknowledge..... (Single Mode)**

The interrupt acknowledge status is generated during the pair of interrupt acknowledge bus cycles.

ST2–ST0 are three-state outputs and enter the high impedance state during hold acknowledge.

**MRQ\* [Memory Request]..... 3-state outputs**

MRQ\* is asserted during memory address space accesses.

MRQ\* is an active-low three-state output and becomes high impedance during hold acknowledge.

**A23-A0 [Address Bus]..... 3-state outputs**

These three-state outputs form the active-high 24-bit physical address bus. Both memory and I/O address spaces are organized into a pair of byte-wide banks and are addressed by A23-A1. The even addressed bank is selected whenever A0 is driven low during an I/O or memory bus cycle. Selection of the odd addressed bank is controlled by the UBE\* signal.

A memory or I/O port address is asserted after the rising edge of a T1 bus cycle. A23-A0 are three-state outputs and enter the high impedance state while HLDAK\* or BFREZ is asserted.

**D15-D0 [Data Bus]..... 3-state I/O**

The sixteen bit bidirectional data bus is organized into a pair of byte-wide data buses, supporting both 8- and 16-bit bus cycles. Eight bit bus cycles and sixteen bit bus cycles are distinguished by A0 and UBE\*. During T4 of a read cycle, the contents of D15–D0 are latched internally in the μPD70616. Data is placed on D15–D0 during T1 of a write bus cycle and remains valid until the start of the next bus cycle.

While the RESET input is asserted, D15–D0 are placed in the high impedance state. D15–D0 are also placed in the high impedance state while HLDAK\* or BFREZ is asserted.

**R/W\* [Read/Write]..... 3-state output**

R/W\* is a three-state output indicating the direction of the current bus cycle. R/W\* is driven low during the T1 state to indicate the current bus cycle will transfer data from the μPD70616 to external memory or a peripheral device. Conversely, R/W\* is driven high to indicate a transfer from an external device to the μPD70616.

**DS\* [Data Strobe]..... 3-state output**

DS\* is a three-state output used to generate memory and I/O read/write strobes for a bus cycle. DS\* is asserted during T1 of a bus cycle and held valid until the T4 bus state. DS\* is negated during the Halt Acknowledge and Machine Fault Acknowledge and reset states.

DS\* is held in the high impedance state during hold acknowledge.

**BCY\*** [Bus Cycle]..... 3-state output

BCY\* is a three-state output and indicates a valid bus cycle is in progress. BCY\* is asserted following the rising edge of T1 and is negated following the rising edge of a T4.

BCY\* is held at a high level during reset.

**DL1–DL0 [Data Length]**..... 3-state outputs

DL1–DL0 are three-state outputs and are used together with FAS\* and the bus status outputs to determine operand size for fixed length data or the position and direction for variable length data types. During instruction fetch accesses, DL1–DL0 are driven low.

- Single Mode Bus Accesses

DL1	DL0	Data Length	
0	0	Byte	(1 byte)
0	1	Halfword	(2 bytes)
1	0	Word	(4 bytes)
1	1	Reserved	

86-228

- String Mode Bus Accesses

DL1	DL0	String Direction	
1	0	Address Increment	
1	1	Address Decrement	

86-229

DL1–DL0 are in an undefined state during reset.

**FAS\*** [First Data Access Status]..... 3-state output

FAS\* is a three-state output indicating the type of data bus cycle. FAS\* is asserted during the first bus cycle of any multiple bus cycle data transfer. FAS\* is negated in any subsequent bus cycle.

FAS\* is undefined during an instruction fetch bus access.

Mode	Data Length/String Direction	DL1	DL0	FAS	MRQ, ST2–ST0	Notes
Single	Byte	0	0	0	Single Mode 0000, 0010 0011, 0100 0101, 1000 1011, 1110	FAS = 0 : First bus cycle FAS = 1 : Subsequent bus cycles
	Halfword	0	1	0/1		
	Word	1	0	0/1		
	Reserved	1	1	0/1		
String	Start Increment	1	0	0	String Mode 0001, 1001	
	Start Decrement	1	1	0		
	Address Increment	1	0	1		
	Address Decrement	1	1	1		

86-230

**UBE\* [Upper Byte Enable]**..... 3-state output

UBE\* is an active low output and is asserted when the upper byte (D15-D8) of the data bus contain valid data. UBE\* is used along with A0 by decoding logic to select the even/odd addressed buses as follows:

UBE	A0	Access Type
0	0	Halfword Access
0	1	Upper Byte Access
1	0	Lower Byte Access
1	1	Reserved

86-301

UBE\* is a three-state output and enters the high impedance state during hold acknowledge.

**READY\* [Ready]**..... input

READY\* is an active-low input used to synchronize external memory and peripheral devices with the μPD70616. Slow memory and I/O devices can lengthen a bus cycle by negating the READY\* input and causing the BCU to insert TW states between states T3 and T4 of a normal bus cycle. When READY\* is once again asserted and recognized by the BCU, the BCU will proceed to the T4 state.

The READY\* input is ignored during any three clock bus cycle or during Halt Acknowledge and Machine Fault Acknowledge states.

**BMODE [Bus Mode]**..... input

**FRM [Functional Redundancy Monitor]**

This three level logic input is used to select the type bus cycle or to enable the FRM (functional redundancy monitor) logic. When at a low level, BMODE selects a three clock bus cycle instead of the normal four clock bus cycle. External logic can select the three clock bus transfer mode when fast memory such as a cache is accessed. The READY\* input is ignored during a short bus cycle.

Bus Mode	Bus Cycle Type	BMODE
Master Mode	Short Cycle (3 clocks, READY pin ignored)	0 V
	Normal Bus Cycle (4 clocks)	5 V
Checker Mode	Normal Bus Cycle (4 clocks)	10 V

86-302

The FRM mode is selected when this pin is attached to a voltage source of +10V. This places the μPD70616 into the checking mode, allowing the μPD70616 to check the operation of a master mode μPD70616 microprocessor.

**BLOCK\* [Bus Lock]**..... output

**MSMAT\* [Mismatch]**

The BLOCK\* output is asserted during a bus cycle to indicate an indivisible read-modify-write bus cycle (TASI, CAXI instructions) is taking place. It is used by external logic to guarantee the integrity of the bus cycle in a multiple bus master system. BLOCK\* is also asserted for the duration of an interrupt acknowledge bus cycle.

MSMAT\* is an FRM check mode output and is asserted if a discrepancy between master and checker processors is detected.

**BERR\* [Bus Error]**..... input

External logic can assert the BERR\* input to indicate the presence of a fault in the current bus cycle and request a retry of the bus cycle or force an exception if the bus fault cannot be corrected. The decision to retry or terminate a bus cycle with an error is determined by the state of the RT/EP\* input pin at the rising edge of the T4 state.

### BFREZ [Bus Freeze]..... input

The BFREZ input causes bus activity to cease after the falling edge of the T4 state and T1 states to be inserted into the bus cycle. Both the address bus and the data bus are placed in the high impedance state and all status outputs are inactivated.

When BFREZ is returned to a low level, the RT/EP\* pin is sampled to determine whether to restart the faulted instruction or force an exception.

### RT/EP\* [Retry/Exception]..... input

The RT/EP\* input pin is used along with the BERR\* and BFREZ inputs to determine the response to a bus error or halt request.

If BERR\* is asserted, RT/EP\* will determine whether to retry the faulty bus cycle (RT/EP\* = 1) or cause an exception (RT/EP\* = 0).

When the BFREZ input is negated, RT/EP\* will determine whether to restart instruction execution (RT/EP\* = 1) or cause an exception (RT/EP\* = 0).

RT/EP	Bus Error Action	BFREZ Release
1	Retry	Interrupted Instruction Restart
0	Exception	Machine Fault Exception

MS-305

3

### NMI\* [Non-Maskable Interrupt]..... input

The NMI\* pin is an active low interrupt input that cannot be masked by software. At the completion of the current instruction, the PC and PSW are pushed on the stack and control is transferred to a non-maskable interrupt handler.

### INT [Interrupt Request]..... input

The INT input is an active high interrupt input that can be masked by the IE bit in the PSW register. If the IE bit is cleared, the INT input is masked and any interrupt requests are held pending until unmasked by software.

If set, the μPD70616 will perform a pair of back-to-back interrupt acknowledge cycles. On the second interrupt acknowledge cycle, an 8-bit interrupt vector is read from the external μPD71059 interrupt controller and used as an offset into the System Base Table. The INT input must be asserted until acknowledged by the μPD70616.

### HLDRQ\* [Hold Request]..... input

This active-low input is asserted by an external bus master desiring to take control of the μPD70616 address, data, and control buses.

### HLDAK\* [Hold Acknowledge]..... output

In response to an asserted HLDRQ\* input, HLDAK\* is asserted indicating that the μPD70616 address, data, and control buses have entered the high impedance state and are available for use by the external bus master.

### CPBUSY\* [Coprocessor Busy]..... input

This active-low input allows an external coprocessor to indicate the status of the previous operation. By sampling the CPBUSY\* input, the μPD70616 can synchronize with asynchronous coprocessors.



**RESET [Reset]**..... input

When this active-high input is asserted, the μPD70616 terminates all operations and initializes the internal registers and logic. While RESET is asserted, the output buffers of the μPD70616 are in the following states:

State	Output
High	R/W, DS, BCY, HLDAR, BLOCK (MSMAT)
High-Z	D15-D0
Undefined	A23-A0, DL1-DL0, FAS, MRQ, UBE, ST2-ST0

86-294

When RESET returns to a low level, the registers assume their initial values and program execution commences in physical address mode from address 0FFFFFF0H.

Register	Reset Value
PSW	1 0 0 0 0 0 0 0 H
PC	F F F F F F F 0 H
SBR	0 0 0 0 0 0 0 0 H
SYCW	0 0 0 0 0 0 7 0 H
TKCW	0 0 0 0 E 0 0 0 H
PSW2	0 0 0 0 F 0 0 2 H
ATBR	Invalid
TLB	Cleared
Others	Undefined

86-295

**CLK [Clock]**..... input

CLK is the μPD70616 system clock input.

**VDD [Power Supply]**

The two VDD pins supply +5 Volt power to the μPD70616.

**GND [Ground]**

The two GND pins are the power supply return.

### Absolute Maximum Ratings

T<sub>A</sub> = +25°C

Power Supply Voltage, V <sub>DD</sub>	-0.5 V to +7.0 V
Input Voltage, V <sub>I</sub>	-0.5 V to V <sub>DD</sub> + 0.3 V
CLK Input Voltage, V <sub>K</sub>	-0.5 V to V <sub>DD</sub> + 1.0 V
Output Voltage, V <sub>O</sub>	-0.5 V to V <sub>DD</sub> + 0.3 V
Operating Temperature, T <sub>OPT</sub>	0°C to +70°C
Storage Temperature, T <sub>STG</sub>	-65°C to +150°C

**Comment:** Exposing the device to stresses above those listed in the Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### Capacitance

T<sub>A</sub> = +25°C, V<sub>DD</sub> = 0 V

Parameter	Symbol	Limits		Unit	Conditions
		Min	Max		
Input Capacitance	C <sub>I</sub>		15	pf	F <sub>c</sub> = 1 MHz
I/O Capacitance	C <sub>IO</sub>		15	pf	returned to 0 V

### DC Characteristics

T<sub>A</sub> = 0°C to +70°C, V<sub>DD</sub> = +5 V ±5%

Parameter	Symbol	Limits		Unit	Conditions
		Min	Max		
Input Voltage High	V <sub>IH</sub>	2.2	V <sub>DD</sub> + 0.3	V	
Input Voltage Low	V <sub>IL</sub>	-0.5	0.8	V	
CLK Input Voltage High	V <sub>KH</sub>	4.0	V <sub>DD</sub> + 0.3	V	
CLK Input Voltage Low	V <sub>KL</sub>	-0.5	0.6	V	
Output Voltage High	V <sub>OH</sub>	2.4		V	I <sub>OH</sub> = -400μA
Output Voltage Low	V <sub>OL</sub>		0.45	V	I <sub>OL</sub> = 3.2mA
Input Leakage Current	I <sub>LI</sub>		±10	μA	
Output Leakage Current	I <sub>LO</sub>		±10	μA	
Supply Current	I <sub>DD</sub>		450	mA	f <sub>IN</sub> = 16 MHz

### AC Characteristics

T<sub>A</sub> = 0°C to +70°C, V<sub>DD</sub> = +5 V ±5%

Output Pin Load Capacitance: 100 pf

Parameter	Symbol	Limits		Unit	Conditions
		Min	Max		
Clock Cycle Time	t <sub>CYK</sub>	62.5	125	ns	
Clock Pulse Width High	t <sub>KKH</sub>	27		ns	V <sub>KH</sub> = 3.0 V
Clock Pulse Width Low	t <sub>KKL</sub>	26		ns	V <sub>KL</sub> = 1.7 V
Clock Rise Time	t <sub>KR</sub>		3	ns	1.7 - 3.0 V
Clock Fall Time	t <sub>KF</sub>		3	ns	3.0 - 1.7 V
RESET Pulse Width from V <sub>DD</sub> Valid	t <sub>HVR</sub>	1000		ns	
RESET Setup to CLK↓	t <sub>SRK</sub>	10		ns	
RESET Pulse Width	t <sub>WRH</sub>	20		t <sub>CYK</sub>	
NMI*, INT, CPBUSY* Setup to CLK↓	t <sub>SIK</sub>	10		ns	
Address Delay from CLK↑	t <sub>DKA</sub>	3	40	ns	
BCY* Delay from CLK↑	t <sub>DKBC</sub>	3	40	ns	
DS* Delay from CLK↓	t <sub>DKDS</sub>	3	40	ns	
BMODE Setup to CLK↓	t <sub>SBMK</sub>	25		ns	
BMODE Hold from CLK↓	t <sub>HKBM</sub>	10		ns	
READY* Setup to CLK↓	t <sub>SRYK</sub>	20		ns	
READY* Hold from CLK↓	t <sub>HKRY</sub>	10		ns	
Data Setup to CLK↓	t <sub>SDK</sub>	10		ns	
Data Hold from CLK↓	t <sub>HKD</sub>	15		ns	
Data Delay from CLK↓	t <sub>DKD</sub>	3	50	ns	
BLOCK* Delay from CLK↑	t <sub>DKBL</sub>	3	40	ns	
BERR* Setup to CLK↓	t <sub>SBEK</sub>	10		ns	
BERR* Hold from CLK↓	t <sub>HKBE</sub>	15		ns	
	t <sub>DBCA</sub>	t <sub>CYK</sub> - 20		ns	
	t <sub>DDSA</sub>	t <sub>KKL</sub> - 15		ns	

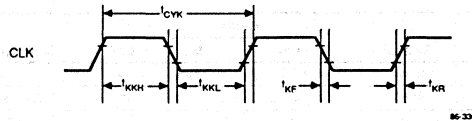
AC Characteristics (cont)  
 $T_A = 0^\circ\text{C to } +70^\circ\text{C}$ ,  $V_{DD} = +5\text{ V } \pm 5\%$

Output Pin Load Capacitance: 100 pf

Parameter	Symbol	Limits		Unit	Conditions
		Min	Max		
RT/EP* Setup to CLK↓	t <sub>SRTK</sub>	25		ns	
RT/EP* Hold from CLK↓	t <sub>HKRT</sub>	10		ns	
HLDROQ* Setup to CLK↑	t <sub>SHQK</sub>	20		ns	
HLDROQ* Hold from CLK↑	t <sub>DKHA</sub>		40	ns	
Output Float Delay from CLK↑	t <sub>FKA</sub>	3	50	ns	
BFREZ Setup to CLK↑	t <sub>SBFK</sub>	10		ns	
RT/EP* Setup to CLK↑	t <sub>SREK</sub>	20		ns	
RT/EP* Hold from CLK↑	t <sub>HKRE</sub>	10		ns	
Address Setup to CLK↓ (FRM Mode)	t <sub>SFRAK</sub>	20		ns	
Address Hold from CLK↓ (FRM Mode)	t <sub>HKFRA</sub>	15		ns	
MSMAT* Delay from CLK↑	t <sub>DKMS</sub>		40	ns	

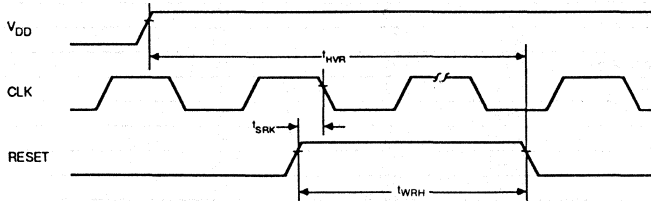
Timing Waveforms

Clock Timing



86-330

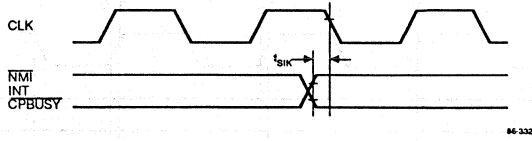
Reset Timing



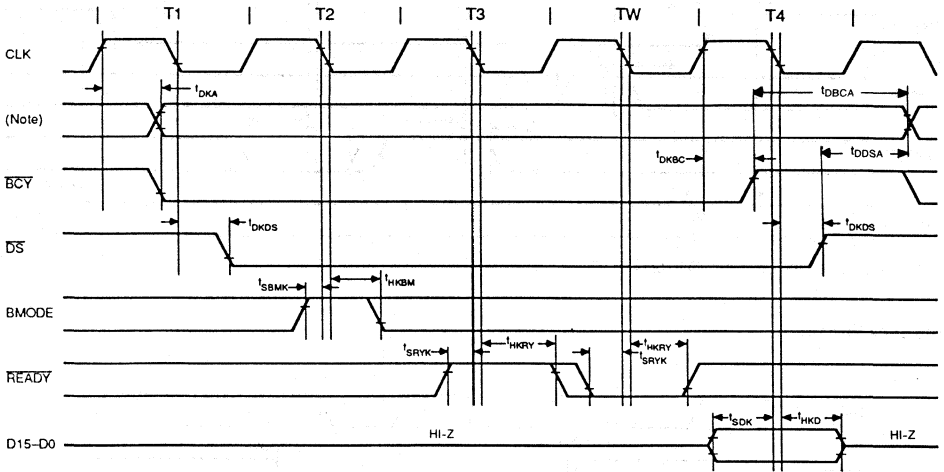
86-331

### Timing Waveforms (cont)

#### Asynchronous Input Timing



#### Read Timing

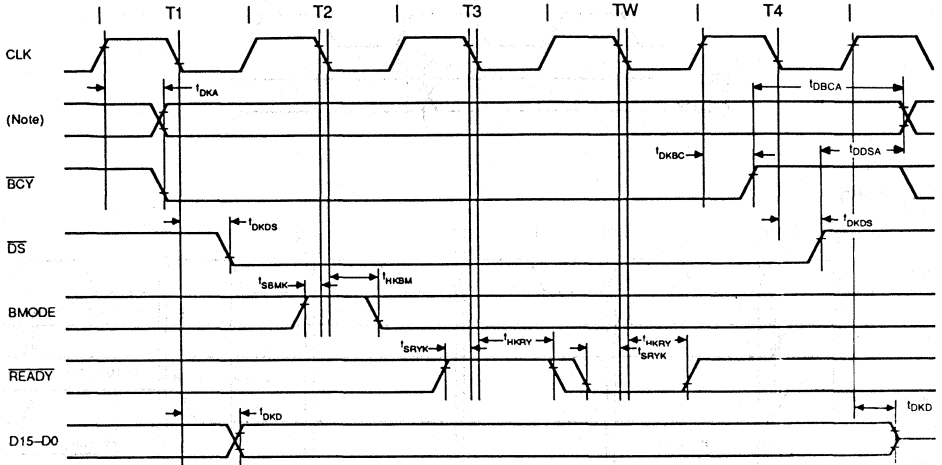


Note - A23-A0,  $\overline{UBE}$ , R $\overline{W}$ , ST2-ST0,  $\overline{MRQ}$ , DL1-DL0, FAS

86-333

Timing Waveforms (cont)

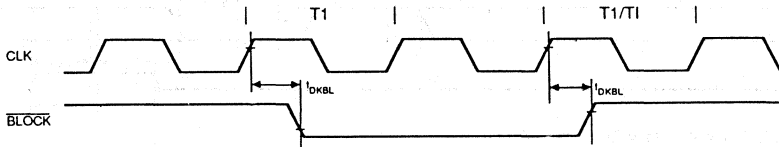
Write Timing



Note - A23-A0,  $\overline{UBE}$ ,  $\overline{R\overline{W}}$ , ST2-ST0,  $\overline{MRQ}$ , DL1-DL0, FAS

86-334

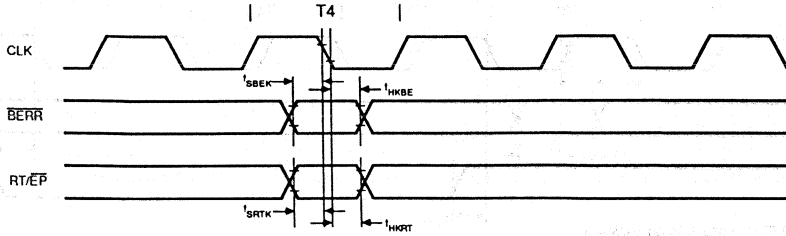
Bus Lock Timing



86-335

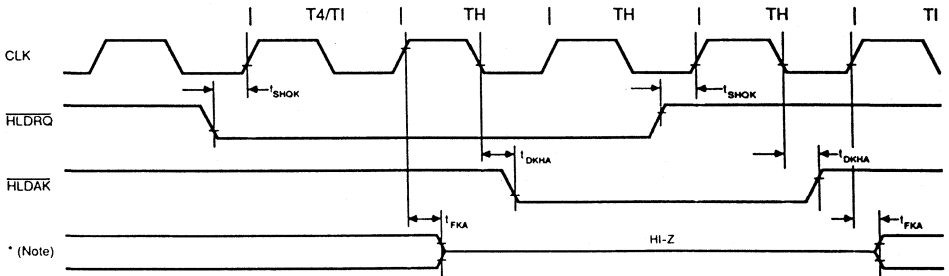
### Timing Waveforms (cont)

#### Bus Error Timing



06-326

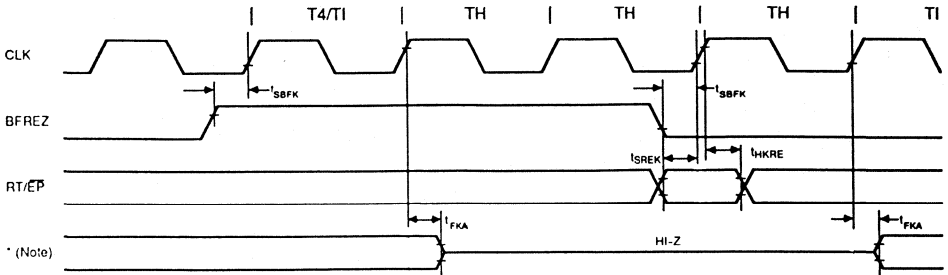
#### Bus Hold Timing



NOTE - A23-A0,  $\overline{UBE}$ ,  $\overline{R\overline{W}}$ ,  $\overline{M\overline{RQ}}$ , ST2-ST0, DL1-DL0,  $\overline{BCY}$ ,  $\overline{DS}$ , D15-D0

06-327

#### Bus Freeze Timing

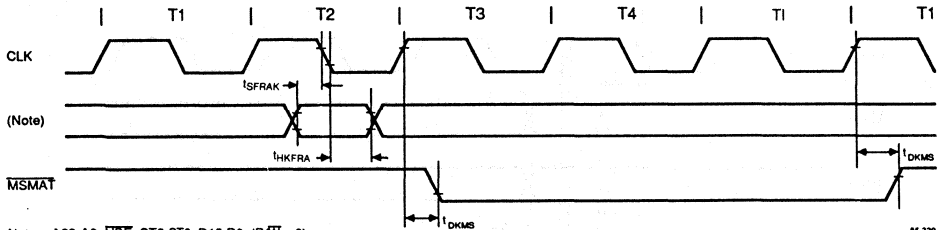


NOTE - A23-A0,  $\overline{UBE}$ ,  $\overline{R\overline{W}}$ ,  $\overline{M\overline{RQ}}$ , ST2-ST0, DL1-DL0,  $\overline{BCY}$ ,  $\overline{DS}$ , D15-D0

06-328

Timing Waveforms (cont)

FRM Timing



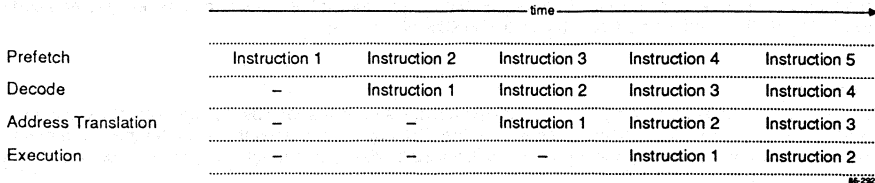
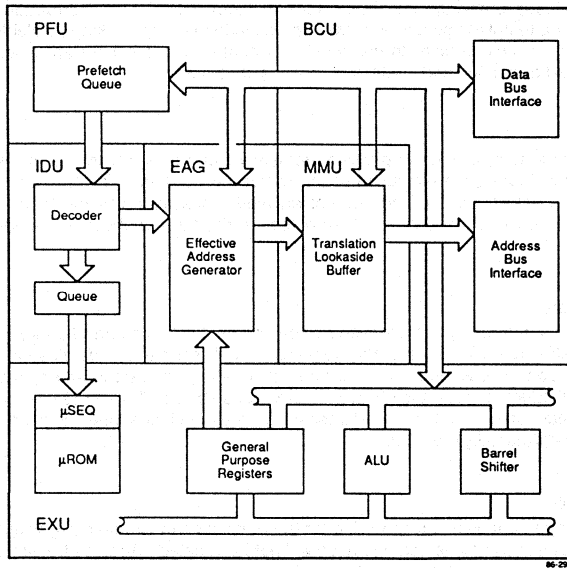
Note - A23-A0,  $\overline{UBE}$ , ST2-ST0, D15-D0 (R/W = 0)

86-339

## Section 2 Pipeline Operation

The μPD70616 contains six functional units which permit concurrent processing of up to four instructions. Pipelining instructions allows the μPD70616 to take advantage of the inherent parallelism in instruction fetch, decode and execution to increase performance. The μPD70616 pipeline organization contains special logic to recognize pipeline hazards and hold up processing until the hazard no longer exists. Exceptions such as instruction prefetch page faults or undefined opcodes are delayed until the faulted instruction is ready to execute, in case a control transfer instruction preceded the non-existent or undefined instruction.

μPD70616 Instruction Pipeline Organization





The μPD70616 functional blocks and the associated operations performed are described below.

**Bus Control Unit**

The BCU (bus control unit) provides the interface between the internal buses of the μPD70616 and external memory and peripheral devices. The BCU accepts requests for bus operations from the other internal units along with a physical address and performs the requested operation. These operations include:

- memory accesses
- I/O accesses
- bus cycle timing (normal/short)
- bus cycle error/retry
- FRM mode comparison

**Prefetch Unit**

The PFU (prefetch unit) performs prefetching of instructions for the instruction decoding logic. As the lowest priority bus requester, the PFU uses otherwise idle bus cycles to keep the 16 byte instruction prefetch queue filled. Prefetching of instructions attempts to minimize instruction execution time by making available without delay the next instruction for the instruction decode unit. In the event of a control transfer, interrupt or exception, the instruction queue contents are flushed and a demand mode instruction fetch is made.

**Instruction Decode Unit**

The IDU (instruction decode unit) is another pipeline element which takes advantage of instruction concurrency to decode the next instruction during the execution of the current instruction. The IDU examines the instruction for operand references and passes the operand addressing mode information to the logic performing the effective address calculation. The decoded instruction is then placed in the three instruction decoded instruction queue for execution.

**Effective Address Generator**

The EAG (effective address generator) accepts both source and destination operand addressing information from the IDU and computes the effective addresses of the operands. Both source and destination operand addresses are computed prior to instruction execution to allow checking of access permissions and availability prior to instruction execution. In the physical address mode, the effective address of an operand is a physical address and is untranslated by the memory management unit. In the virtual address mode, the effective address is a virtual address and requires translation by the MMU before use by the BCU.

**Memory Management Unit**

The MMU (memory management unit) is responsible for the translation of virtual to physical addresses and checking that the access permissions for the requested operation will not be violated. Address translation is performed with the aid of two levels of address translation tables which specify the execution level and access permissions for each virtual address. Because it is desirable to eliminate the overhead of address translation table lookup for each address, the MMU maintains a 16 entry TLB (translation look-aside buffer), containing the sixteen most recently referenced pages, typically eliminating the need for a address translation in 98% of the cases.

**Execution Unit**

The EXU (execution unit) containing the μPD70616 register set, is responsible for the actual execution of each instruction. The EXU takes decoded instructions from the IDU, source operands that were fetched from memory and executes the instruction. Internally, the EXU uses multiple 32-bit data paths to speed operand access and instruction execution.

### Section 3 μPD70616 Architecture

This section contains an overview of the features and implementation of the μPD70616 architecture. For additional information, refer to the μPD70616 Programmer's Reference Manual.

The μPD70616 register set is divided into a program register set and a privileged register set. The program register set consists of the registers and resources visible to application programs. Privileged software such as operating systems also have access to the privileged register set.

#### Program Register Set

The program register set consists of the general purpose registers, program counter and the lower halfword (16-bits) of the program status word.

##### • General Purpose Registers

The general purpose register set consists of thirty-two 32-bit registers named R0–R31. Each of these registers can be used as an accumulator, base register, index register or for temporary storage. General purpose registers can be used to hold byte (8-bit), halfword (16-bit) or word (32-bit) data. Any pair of consecutive registers can also be used to hold doubleword (64-bit) data. For the byte (halfword) data types, the high order 3 (2) bytes are unaffected by operations on the low order part.

The top three of the general purpose registers (R29–R31) are also implicitly selected by certain instructions and can be referred to by alternate names.

R29 : AP (Argument Pointer)

The argument pointer is used as a base register to pass arguments by the procedure call instructions.

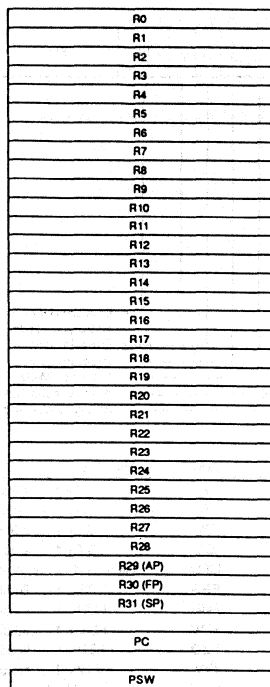
R30 : FP (Frame Pointer)

The frame pointer is used to manage the stack frames generated by high level languages. The FP is referenced by the PREPARE and DISPOSE instructions.

R31 : SP (Stack Pointer)

The stack pointer contains a pointer to the current TOS (top of stack). The SP register is implicitly selected by the push/pop instructions and by the subroutine and procedure call/return instructions.

#### Program Register Set

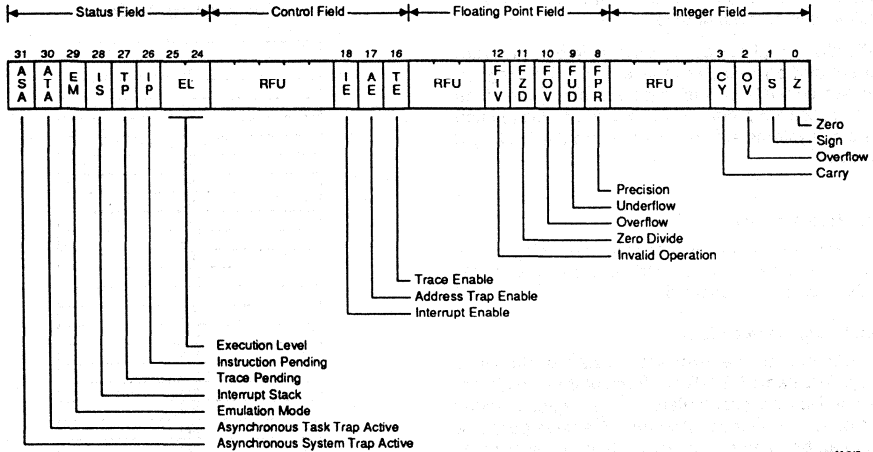


##### • PC (Program Counter)

The program counter contains the address of the first byte of the current instruction. The PC is also used as a base register in any of the five PC relative addressing modes.

• PSW (Program Status Word)

The PSW (program status word) is divided into four 8-bit fields. The lower halfword is accessible to all programs and contains the integer and floating point condition codes. The upper halfword contains the processor control and status fields and can only be modified by programs running at execution level 0.



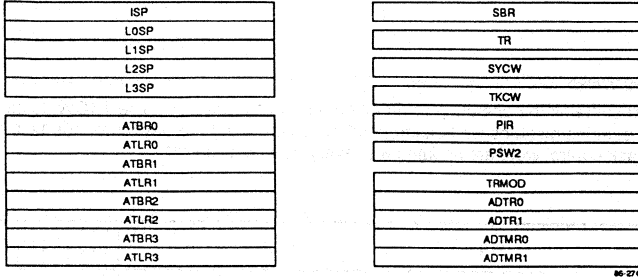
66-217

Bit(s)	Name	Meaning
0	Z	Z = 1 : zero integer result
1	S	S = 1 : negative integer result
2	OV	OV = 1 : integer overflow occurred
3	CY	CY = 1 : a carry (borrow) was generated
4-7	RFU	Reserved for future use (Must be 0)
8	FPR	FPR = 1 : floating point precision fault
9	FUD	FUD = 1 : floating point underflow
10	FOV	FOV = 1 : floating point overflow
11	FZD	FZD = 1 : floating point zero divide
12	FIV	FIV = 1 : floating point invalid operand
13-15	RFU	Reserved for future use (Must be 0)
16	TE	TE = 1 : instruction trace enabled
17	AE	AE = 1 : address traps enabled
18	IE	IE = 1 : maskable interrupts enabled
19-23	RFU	Reserved for future use (Must be 0)
24-25	EL	EL = 00 : execution level 0 (privileged execution level) 01 : execution level 1 10 : execution level 2 11 : execution level 3
26	IP	IP = 1 : instruction pending
27	TP	TP = 1 : instruction trace pending
28	IS	IS = 1 : interrupt stack active
29	EM	EM = 1 : V20/V30 emulation mode enabled
30	ATA	ATA = 1 : asynchronous task trap being serviced
31	ASA	ASA = 1 : asynchronous system trap being serviced

## Privileged Register Set

The privileged register set consists of registers that are accessible only by programs running at execution level 0. The privileged register set includes the stack pointer cache, processor control registers, task control registers, virtual memory management registers and address trap registers. Generally, only operating system software is allowed access to these registers.

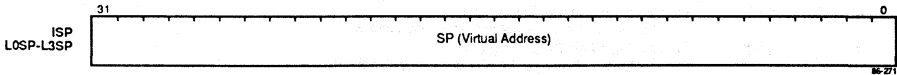
### Privileged Register Set



- **SP (Stack Pointers)**

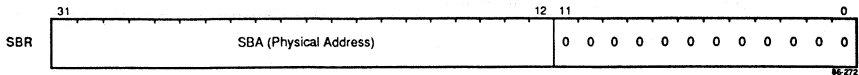
- ISP (Interrupt Stack Pointer)
- L0SP, L1SP, L2SP, L3SP (Level 0, 1, 2, 3 Stack Pointers)

A set of five separate stack pointers are used on the μPD70616 microprocessor. The interrupt stack pointer is used during the processing of interrupts and each execution level has a dedicated stack pointer. The general purpose register R31 contains a pointer to the current TOS and is saved and loaded whenever the execution level changes or the processor switches to the interrupt stack. The active stack pointer can be identified by examining the contents of the PSW.EL and PSW.IS fields.



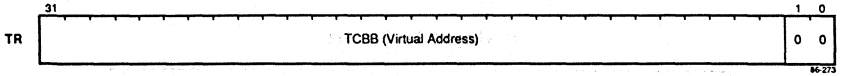
- **SBR (System Base Register)**

The system base register defines the physical base address for the SBT (system base table). The system base table consists of a list of entry points for interrupt and exception handlers. The system base table must be aligned on a page (4KB) boundary, thus the low order 12 bits of the SBR must be zero.



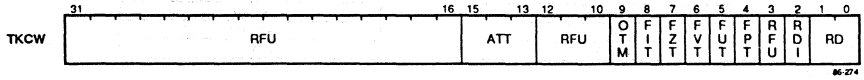
• TR (Task Register)

The task register contains the virtual address of the TCB (task control block) for the current task context. The contents of the task register are modified by the LDTASK instruction when a new task context is installed. The task TCB must be aligned on a word boundary and the low order 2 bits of the task register must be 0.



• TKCW (Task Control Word)

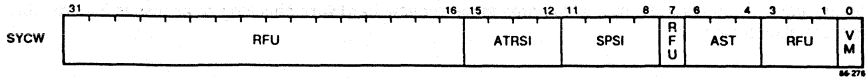
The task control word contains the operating environment for the current task. The TKCW contains the floating point rounding mode, the enables/disables for floating point exceptions and the asynchronous task trap enable/disable. The contents of the task control word are updated with every new task context.



Bit(s)	Name	Meaning
0-1	RD	Floating Point Rounding Control RD = 00 : round toward nearest 01 : round toward $-\infty$ 10 : round toward $+\infty$ 11 : round toward 0
2	RDI	Floating Point Rounding Control for Integer Conversion RDI = 0 : use RD field rounding mode RDI = 1 : round toward 0
3	RFU	Reserved for future use (Must be 0)
4	FPT	Floating Point Precision Trap Enable FPT = 1 : floating point precision traps enabled
5	FUT	Floating Point Underflow Trap Enable FUT = 1 : floating point underflow traps enabled
6	FVT	Floating Point Overflow Trap Enable FVT = 1 : floating point overflow traps enabled
7	FZT	Floating Point Zero Divide Trap Enable FZT = 1 : floating point zero divide traps enabled
8	FIT	Floating Point Invalid Operation Trap Enable FIT = 1 : floating point invalid operation traps enabled
9	OTM	Operand Trap Mask OTM = 0 : floating point invalid operand traps enabled (fixed at 0 on the μPD70616 microprocessor)
10-12	RFU	Reserved for future use (Must be 0)
13-15	ATT	Asynchronous Task Trap Level ATT = 000 : level 0 ATT 001 : level 1 ATT 010 : level 2 ATT 011 : level 3 ATT 1XX : ATT disabled
16-31	RFU	Reserved for future use (Must be 0)

### • SYCW (System Control Word)

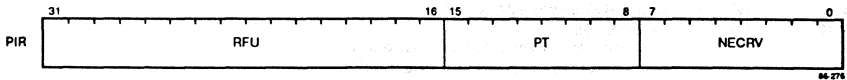
The system control word is used by software to control the system-wide operating environment. The SYCW contains the virtual/physical address mode control, asynchronous system trap enable/disable and the inhibits for area table and stack pointer switching.



Bit(s)	Name	Meaning
0	VM	Virtual Mode VM = 1 : virtual address mode enabled
1-3	RFU	Reserved for future use (Must be 0)
4-5	AST	Asynchronous System Trap Level AST = 000 : level 0 AST 001 : level 1 AST 010 : level 2 AST 011 : level 3 AST 1XX : AST disabled
7	RFU	Reserved for future use (Must be 0)
8-11	SPSI	Stack Pointer Switching Inhibited SPSI = 1 : level 0 (bit 8) / 1 (bit 9) / 2 (bit 10) / 3 (bit 11) stack pointers are saved in the TCB during a context switch
12-15	ATRSI	Area Table Register Switching Inhibited ATRSI = 1 : 00 (bit 12) / 01 (bit 13) / 10 (bit 14) / 11 (bit 15) section area table registers are saved in the TCB during a context switch
16-31	RFU	Reserved for future use (Must be 0)

### • PIR (Processor ID Register)

The processor ID register is used by system software to identify the processor type. When read, the contents of the μPD70616 processor ID register will be of the form 000060XXH.

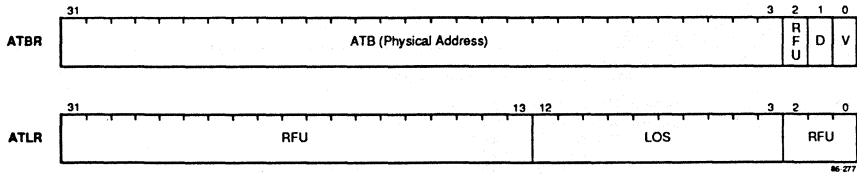


Bit(s)	Name	Meaning
0-7	NECRV	NEC Reserved
8-15	PT	Processor Type
16-31	RFU	Reserved for future use (Must be 0)

• Area Table Register Pairs

- ATBR0, ATBR1, ATBR2, ATBR3 (Area Table Base Registers 0, 1, 2, 3)
- ATLR0, ATLR1, ATLR2, ATLR3 (Area Table Length Registers 0, 1, 2, 3)

The area table register pairs operate in the virtual address mode to identify which of the four area tables is to be used during address translation. Each area table register pair consists of an ATBR (area table base register) and an ATLR (area table length register). The area table base register contains the physical base address of the area table. The area table length register defines the length of the area table to allow hardware checking of accesses into variable length address translation tables.



• ATBR

Bit(s)	Name	Meaning
0	V	Valid V = 1 : this area table register contains valid information
1	D	Growth Direction D = 0 : the section will grow in the positive direction (increasing addresses) D = 1 : the section will grow in the negative direction (decreasing addresses)
2	RFU	Reserved for future use (Must be 0)
3-31	ATB	Area Table Base the physical address of the first area table entry for this section

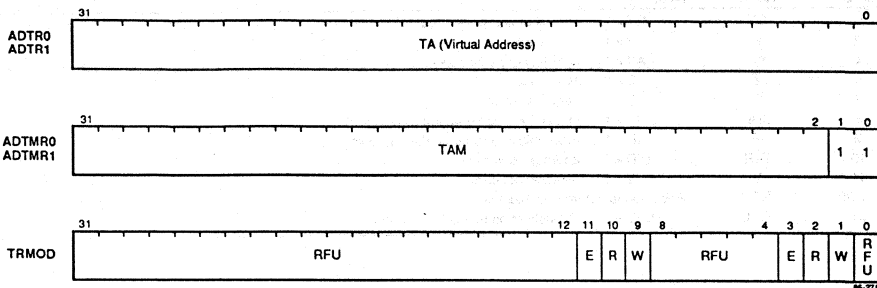
• ATLR

Bit(s)	Name	Meaning
0-2	RFU	Reserved for future use (Must be 0)
3-12	LOS	Limit of Section Determines the size of the section as follows: positive growth direction areas 0 → LOS negative growth direction areas 1023 → LOS
13-31	RFU	Reserved for future use (Must be 0)

### • Address Trap Control Registers

- ADTR0, ADTR1 (Address Trap Registers 0, 1)
- ADTMR0, ADTMR1 (Address Trap Mask Registers 0, 1)
- TRMOD (Trap Mode Register)

The address trap registers define the condition for the occurrence of address traps. Two independent sets of address trap registers are available. The address trap registers and the address trap mask registers contain the virtual base addresses and mask values used to define an address region. The trap mode register is then used for each set of address trap registers to qualify the type access (read/write/execute) necessary to actually cause the trap to occur.



### • TRMOD

Bit(s)	Name	Meaning
0	RFU	Reserved for future use (must be zero)
1	W	Write Access W = 1 : ADTR0, ADTMR0 address range write access trap enable
2	R	Read Access R = 1 : ADTR0, ADTMR0 address range read access trap enable
3	E	Execute Access E = 1 : ADTR0, ADTMR0 address range execute access trap enable
4-7	RFU	Reserved for future use (must be zero)
8	W	Write Access W = 1 : ADTR1, ADTMR1 address range write access trap enable
9	R	Read Access R = 1 : ADTR1, ADTMR1 address range read access trap enable
10	E	Execute Access E = 1 : ADTR1, ADTMR1 address range execute access trap enable
11-31	RFU	Reserved for future use (must be zero)

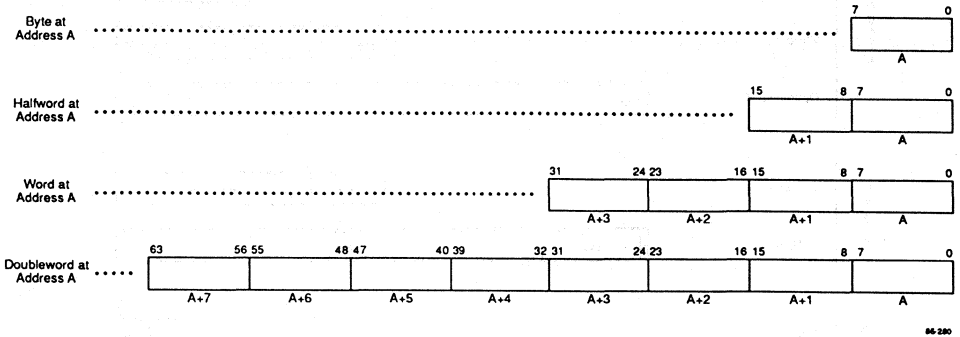




### Memory and I/O Organization

Memory and I/O are organized around a byte addressable, 16 megabyte physical address space. Each byte location in the physical address space has a unique address, determined by the 24-bit address bus. Byte data consists of eight bits with bit 0 designated as the LSB (least significant bit) and bit 7 as the MSB (most significant bit).

For data types larger than a byte, the address of the lowest byte is used as the address of the data. The byte at the lowest address is called the LSB (least significant byte) while the byte at the highest address is the MSB (most significant byte). On the  $\mu$ PD70616, two consecutive bytes of data is called a halfword, four consecutive bytes of data is called a word and eight consecutive bytes of data is called a doubleword. The representation of both memory and I/O data and the corresponding addresses of the component bytes is shown below.



The I/O and memory address spaces are independent and separated by hardware on the  $\mu$ PD70616. However, access to the I/O address space using instructions which normally reference the memory address space can be accomplished by mapping 4KB pages of the virtual address space into the I/O address space.

### Data Types

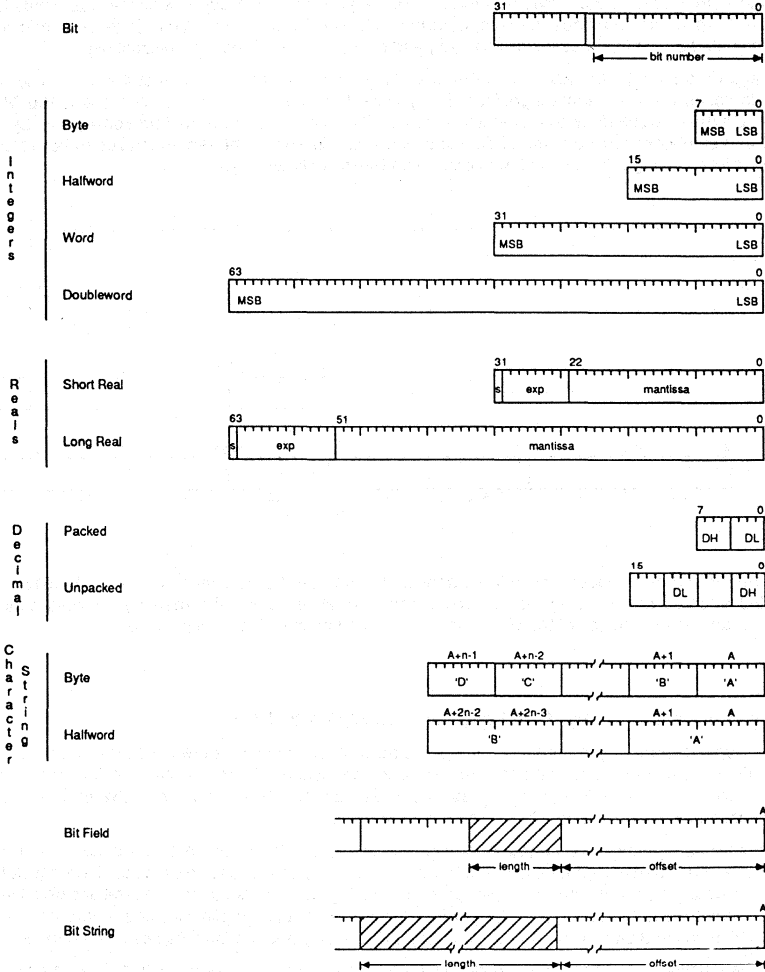
The  $\mu$ PD70616 supports a wide variety of fixed length and variable length data types.

Fixed length data types include signed/unsigned integers and two floating point representations for high performance numeric applications. Business applications can also use the decimal integer data type to speed up decimal calculations. The pointer data type is used to represent addresses for parameter passing between procedures. A bit data type allows fast manipulation of single bits in memory or a register.

The variable length data types include character strings, bit strings and bit fields. A powerful set of character string operations increase performance and minimize the number of bus cycles required to process character data. The bit string data type, unavailable on any other microprocessor, allows logical and data transfer operations on bit strings from 1 to 4 gigabits in length. Bit fields range in length from 0 to 32 bits and simplify packing of integers in high level languages. Both bit fields and bit strings can begin on any bit boundary and are identified by a bit address.

The fixed length data types and bit fields can reside in memory or a register without restriction. The remaining variable length data types can only reside in memory. The format of the data types supported by the  $\mu$ PD70616 are shown below.

Data Type Formats



### Addressing Modes

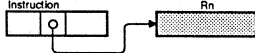
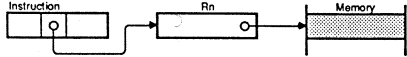
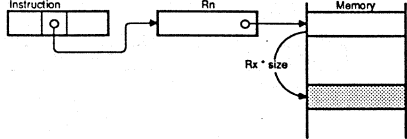
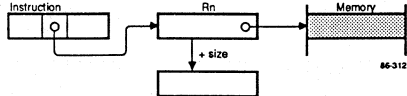
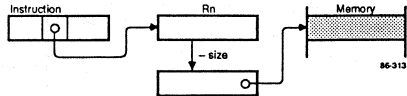
The μPD70616 implements a set of 21 byte and 18 bit addressing modes to allow fast, flexible access to both byte aligned and bit aligned operands.

#### Byte Addressing Modes

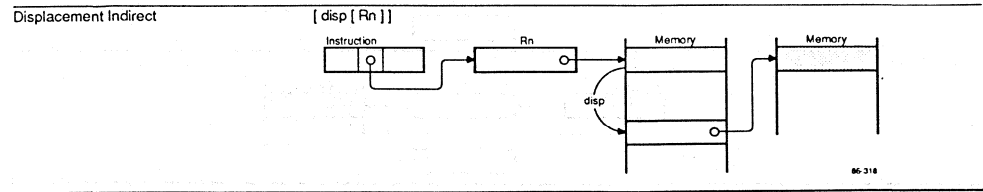
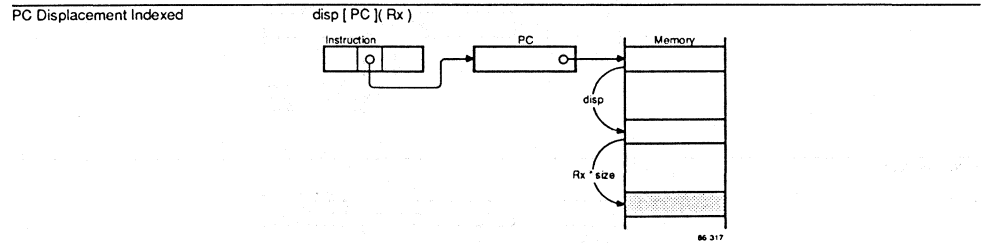
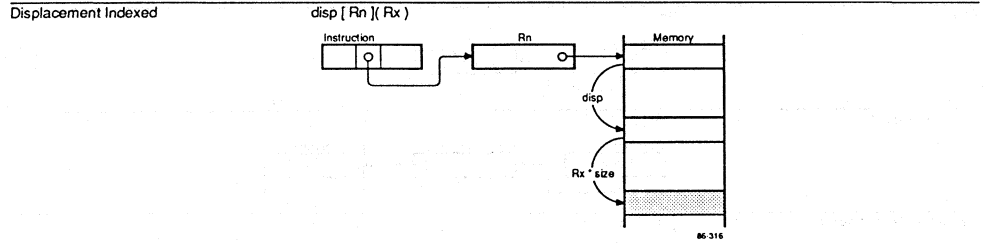
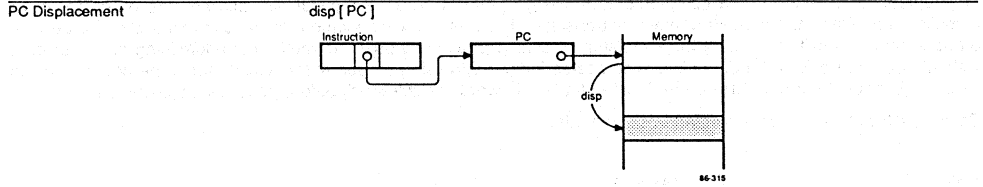
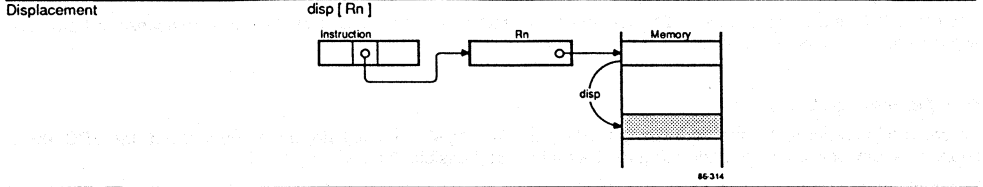
Including all standard operand addressing modes such as register, immediate, base and indexed, the μPD70616 supports a total of twenty-one addressing modes for byte addressible data.

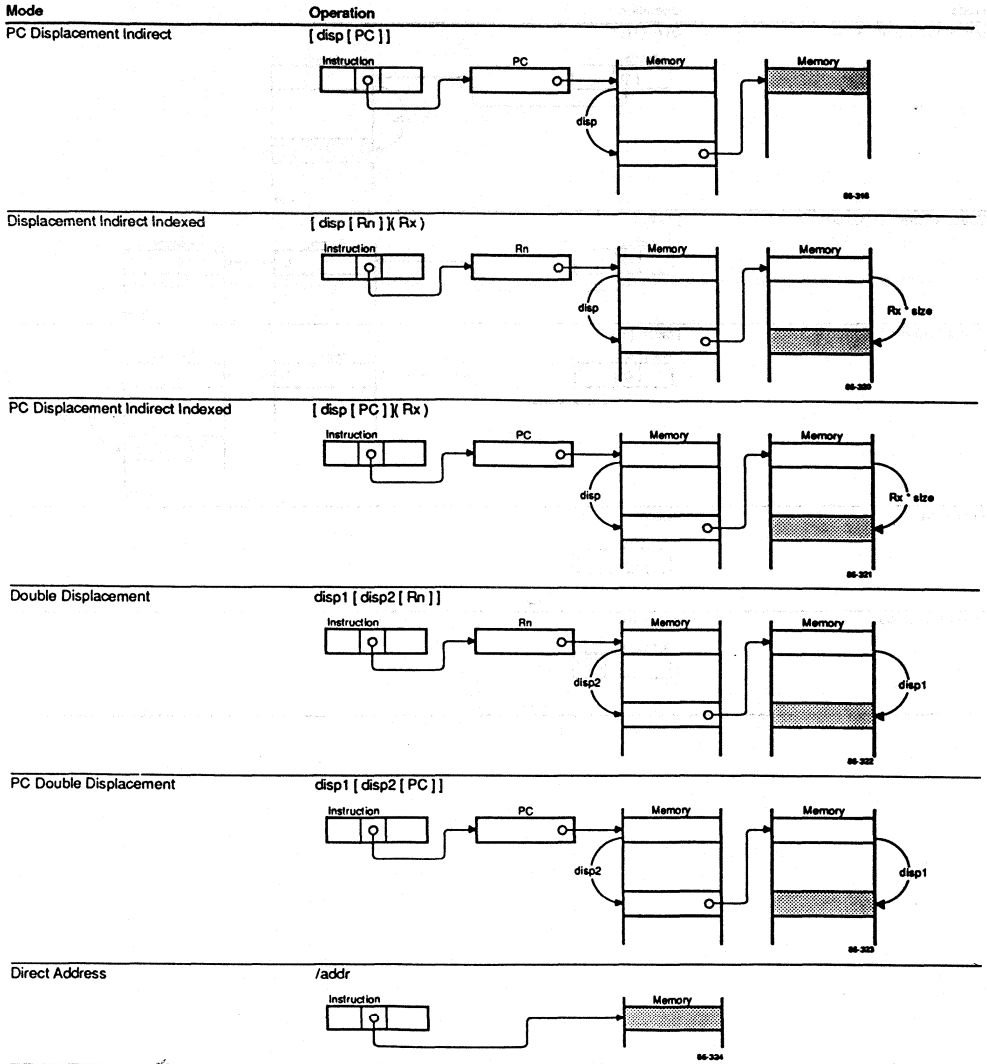
The register addressing mode complements the large general purpose register set, allowing the programmer or compiler to minimize the number of memory transfers. The base, or displacement mode, can use any of the general purpose registers or the PC as a base register. Indexing provides a powerful method of addressing arrays of data without keeping track of the data size. The scaled index addressing modes automatically scale the contents of an index register by the size of the operand (byte / halfword / word / doubleword) before performing the access.

Below is a list of the available byte addressing modes.

Mode	Operation
Register	<p>Rn</p>  <p>86-309</p>
Register Indirect	<p>[ Rn ]</p>  <p>86-310</p>
Register Indirect Indexed	<p>[ Rn ]( Rx )</p>  <p>86-311</p>
Autoincrement	<p>[ +Rn ]</p>  <p>86-312</p>
Autodecrement	<p>[ -Rn ]</p>  <p>86-313</p>

Mode Operation





Mode	Operation
Direct Address Indexed	<p><math>/addr (Rx)</math></p> <p>86-325</p>
Direct Address Deferred	<p><math>[ /addr ]</math></p> <p>86-326</p>
Direct Address Deferred Indexed	<p><math>[ /addr ] (Rx)</math></p> <p>86-327</p>
Immediate	<p><math>\#value</math></p> <p>86-328</p>
Immediate Quick	<p><math>\#value (0-15)</math></p> <p>86-329</p>

The size values used by the scaled index and the autoincrement/autodecrement address modes are listed in the below table.

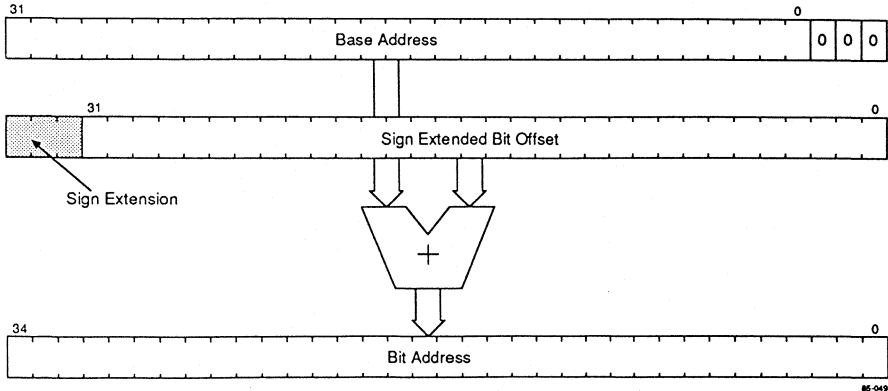
Data Type	Scaling Constant	
	Increment/Decrement	Scaled Index
Byte	1	1
Halfword	2	2
Word	4	3
Doubleword	8	8
Packed Decimal	1	1
Unpacked Decimal	2	2
Byte Character	1	1
Halfword Character	2	2
Bit	4	4
Bit Field	4	-
Bit String	1	-

- Not Available 86-020

### Bit Addressing

In the μPD70616, bit addresses are required to support the bit field and bit string data types which unlike the other data types can be aligned on an arbitrary bit boundary. To address any bit within the 4GB virtual address space, a 35-bit address is required. The μPD70616 generates this 35-bit address using a 32-bit base address and a 32-bit bit offset.

To compute a bit address, the 32-bit base address is zero extended on the right to 35 bit length. Next the 32-bit bit offset is sign extended to 35-bit length and the sum of these two identify the starting address of the bit field or bit string. This process is shown below.





Bit addressing modes are not available for the register or immediate addressing modes.

Bit addressing modes require that a byte base address and an optional non-zero bit offset be specified. In the following table of assembler encodings, the bit offset component of the bit address is shown as underlined. Both address components are combined by the μPD70616 to generate the target bit address.

#### Bit Addressing Modes

Mode	Assembler Encoding	Notes
Register Indirect	@ [ Rn ]	bit offset defaults to 0
Register Indirect Indexed	<u>Rx</u> @ [ Rn ]	
Autoincrement	@ [ Rn+ ]	bit offset defaults to 0
Autodecrement	@ [ -Rn ]	bit offset defaults to 0
Displacement	<u>offset</u> @ [ Rn ]	
PC Displacement	<u>offset</u> @ [ PC ]	
Displacement Indexed	<u>Rx</u> @ disp [ Rn ]	
PC Displacement Indexed	<u>Rx</u> @ disp [ PC ]	
Displacement Indirect	@ [ disp [ Rn ] ]	bit offset defaults to 0
PC Displacement Indirect	@ [ disp [ PC ] ]	bit offset defaults to 0
Double Displacement	<u>offset</u> @ [ disp [ Rn ] ]	
PC Double Displacement	<u>offset</u> @ [ disp [ PC ] ]	
Direct Address	@ /addr	bit offset defaults to 0
Direct Address Indexed	<u>Rx</u> @ /addr	
Direct Address Deferred	@ [ /addr ]	bit offset defaults to 0
Direct Address Deferred Indexed	<u>Rx</u> @ [ /addr ]	

### Demand Paged Virtual Memory

The μPD70616 uses on-chip memory management hardware to implement a high performance demand paged virtual memory system. Memory management allows the design of multi-tasking systems with complete isolation of tasks from the operating system and each other. Demand paging further enhances memory management by allowing programs to use address spaces larger than can be physically addressed in a system. Other advantages include:

- each task can utilize as much of the virtual space as desired
- programs much larger than the physical address space can be accommodated
- protection of tasks is simplified

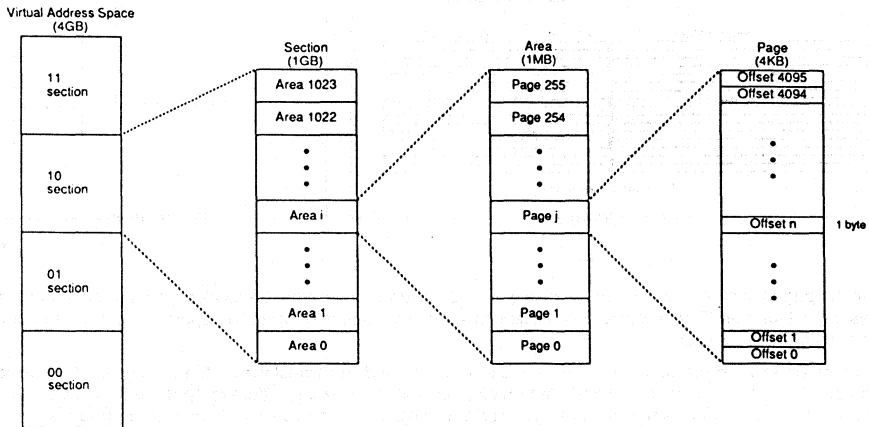
Features of the μPD70616 virtual memory management unit are:

- multiple virtual address spaces
- each virtual address space is 4GB
- three virtual address space partitions (section / area / page)
- 4KB page size
- 16 entry full associative TLB (translation look-aside buffer)
- area / page protection mechanisms
- memory mapped I/O by MMU

### Virtual Address Spaces

A 4 GB virtual space is divided into three components called sections, areas and pages. Sections are a 1GB unit and are the basis for sharing of virtual address spaces in a system employing multiple virtual spaces. Each section contains 1024 areas, each area being 1MB in size. Because of their smaller size, areas are ideally suited for private sharing of virtual space between two tasks. Finally, each area is broken down into 256 4KB pages. Pages are the smallest unit of memory management and are the unit for memory mapping I/O and demand paging.

### Virtual Address Space Organization

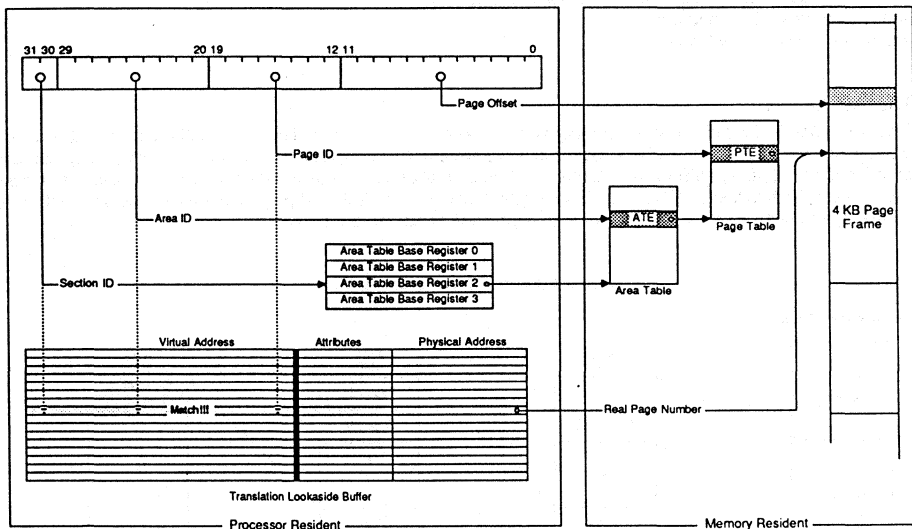


Address Translation

The μPD70616 microprocessor translates virtual addresses into physical addresses using the on-chip memory management unit. Address translation requires the μPD70616 firmware to translate an address using the on-chip memory management registers (area table base and length registers) and the memory resident translation tables (area tables/page tables).

Address translation incurs a great deal of overhead and is unacceptable to perform for each memory access so the results of the last sixteen address translations are cached in a full associative translation look-aside buffer (TLB). Caching of a virtual/physical address pair allows the high speed TLB hardware to access a completed translation the next time the page is referenced, eliminating the address translation overhead. On the μPD70616, the TLB hit ratio is typically over 98%, meaning that for the vast majority of memory references, the address translation tables do not have to be accessed.

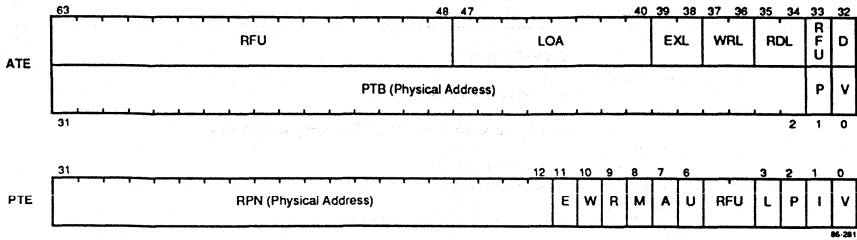
μPD70616 Address Translation



86-203

If a demand paged system is implemented, during an address translation, if the translation tables or page of memory is not present, a fault condition exists and an exception will occur. It is then up to the operating system to correct the fault and restart the instruction.

Two sets of address translation tables are referenced during address translation. Area tables contain up to 1024 entries called ATEs (area table entries). Area tables exist in the memory address space and must be aligned on a doubleword boundary. Each valid area of a virtual address space has an associated page table. Page tables contain up to 256 entries called PTEs (page table entries). Page tables need not be present in memory and can be swapped out to secondary storage if desired. Page tables exist in the memory address space and must be aligned on a word boundary. The formats of the area table and page table entries are shown below.



### • ATE (Area Table Entry)

Bit(s)	Name	Meaning
0	V	Valid V = 1 : this ATE is defined
1	P	Present P = 1 : the page table associated with this ATE is in physical memory
32	D	Growth Direction D = 1 : negative growth direction
33	RFU	Reserved for future use (Must be 0)
34-35	RDL	Read Execution Level RDL = 00 : level 0 01 : level 0, 1 10 : level 0, 1, 2 11 : level 0, 1, 2, 3
36-37	WRL	Write Execution Level WRL = 00 : level 0 01 : level 0, 1 10 : level 0, 1, 2 11 : level 0, 1, 2, 3
38-39	EXL	Execute Execution Level EXL = 00 : level 0 01 : level 0, 1 10 : level 0, 1, 2 11 : level 0, 1, 2, 3
40-47	LOA	Limit of Area Determines the size of the area as follows: positive growth direction pages 0 → LOA negative growth direction pages 255 → LOA
48-63	RFU	Reserved for future use (Must be 0)

• PTE (Page Table Entry)

Bit(s)	Name	Meaning
0	V	Valid V = 1 : this PTE is defined
1	I	I/O Mapped I = 1 : this page is I/O mapped
2	P	Present P = 1 : the page associated with this PTE is in physical memory
3	L	Locked L = 1 : page is locked for I/O
4-5	RFU	Reserved for future use (Must be 0)
6	U	User this field is available for use by the operating system
7	A	Accessed A = 1 : this page has been accessed
8	M	Modified M = 1 : this page has been modified
9	R	Readable R = 1 : this page is readable
10	W	Writable W = 1 : this page is writable
11	E	Executable E = 1 : this page is executable
12-31	RPN	Real Page Number page base physical address

Memory Mapped I/O

The μPD70616 hardware provides a 16MB I/O address space in addition to the 16MB memory address space. Because only privileged I/O instructions can directly access the I/O address space, some other means must be provided to allow any instruction to perform I/O without compromising the protection mechanisms.

In the virtual address mode, this mapping of the virtual address space into the I/O address space is supported by the MMU. Each PTE contains a field (1 bit) which allows the operating system to specify that the page is I/O mapped. When an I/O mapped page is referenced and all protection checks have been satisfied, the bus cycle will be performed but in the I/O address space rather than the memory address space. This allows any instruction to access I/O at any execution level as determined by the operating system. This feature is not available in the physical address mode.

Virtual Address Mode/Physical Address Mode

Following reset, the μPD70616 is in the physical address mode. In the physical address mode, the demand paging and memory management logic are disabled until the initialization software programs the virtual address mode. Virtual address mode is enabled by setting the VM bit within the SYCW.

In the physical address mode, no address translation is performed and there is no need for address translation tables or memory management exception handlers. The four execution levels are present in the physical address mode and privileged instructions can only be executed at execution level 0. On the other hand, physical address mode is not well suited for the design of multi-tasking systems due to the lack of protection mechanisms and the restricted address space.

	Virtual Address Mode	Physical Address Mode
VM Flag	1	0
Address Space	4 Gbytes per task	16 Mbytes
Address Translation Tables	Area Table Page Table	Not used
Protection Mechanisms	2 level (area, page)	None
Software I/O Mapping	I/O mapped pages	Undefined

66-282

3

### Operating System Support

The μPD70616 architecture contains a number of features designed to aid in the design and implementation of modern multi-tasking operating systems.

#### Task Management

In a multi-tasking operating system, multiple tasks must share the system resources in a controlled manner. Certain operations must be prohibited if the system is to guarantee the correct execution of all tasks. It is the responsibility of the operating system to maintain separate task contexts in a data structure called a TCB (task control block). Each TCB consists of information related only this particular instance of the task (such as the general purpose register contents) and other system-wide information (such as shared virtual address space registers) common to all tasks.

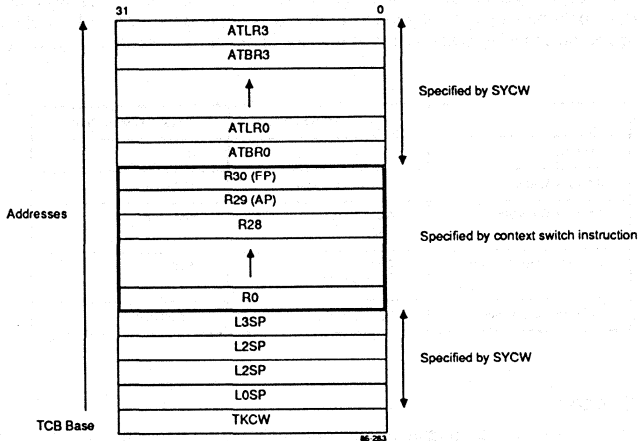
On the μPD70616, the TCB is a variable length data structure consisting of the TKCW (task control word), general purpose registers, stack pointers and memory management register pairs. The TKCW is responsible for the management of local task state which consists of the task's register set. The SYCW (system control word) is responsible for specifying the system-wide context included in the TCB. This information includes the area table register pairs and stack pointers that are to be swapped in and out during context switches. The base address of the TCB for the current task context is contained in the TR (task register). The size of the TCB is determined by the contents of the SYCW and task switch instruction operands.

Instruction set support for the automatic saving and restoring task contexts is provided by the LDTASK and STTASK instructions. The organization of a μPD70616 TCB is shown below.

#### Protection Mechanisms

A multi-tasking system requires that the operating system and other tasks be kept from interfering with one another. Protection and separation of tasks from each other and the operating system is one of the primary advantages of a virtual memory system. Protection mechanisms enhance the operation of the system by detecting and preventing attempts by tasks to access programs and data without first having established the necessary permissions.

TCB (Task Control Block) Organization



The μPD70616 protection mechanisms are implemented using three levels. First an execution level is established which determines the relative trustworthiness of a task. A check of execution level and access type permissions are performed at the area level while an independent access type check is performed again at the page level. Protection mechanisms are enabled only when the processor is operating in the virtual mode. In the physical address mode, all protection mechanisms are disabled and no exceptions can occur.

Area level protection is specified independently for each access method (read, write and execute) by a 2-bit field in each ATE. Each of the three protection fields contain the minimum execution level necessary to allow access to the area. For example, if

```

read level..... 2
write level..... 1
execute level..... 0
    
```

are specified, then access to the area by program execution level is:

```

level 0    read, write and execute
level 1    read and execute
level 2    read
level 3    no access
    
```

Page protections permit or deny access on the basis of contents of the page. For example, if the page protections are

```

read       true
write      true
execute    false
    
```

then the contents of the page are readable and writable but not executable.

An access is possible only when both area and page permissions have been granted. Any attempt at access without both sets of permissions being satisfied will result in an exception.

### Interrupts and Exceptions

An interrupt is an event which occurs asynchronously to the operation of the μPD70616 while an exception is an event which occurs as a direct result of program execution. When an interrupt or exception is recognized, the program is suspended and control is transferred to an interrupt or exception handler. Processing of interrupts and exceptions is similar but differ slightly in the operation of the interrupt enable flag and the stack pointer used during interrupt or exception processing.

Reset is a special type of exception. When reset occurs, all processor activity is stopped and the processor is initialized to the reset state.

### System Base Table

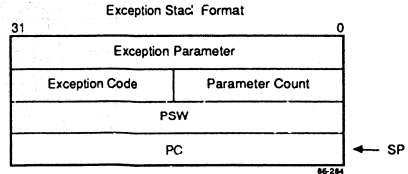
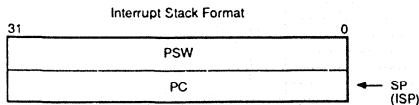
When an interrupt or exception is recognized, the SBT (system base table) is used to locate the appropriate vector. The SBT consists of 256 entries, each containing an entry point to an interrupt or exception handler. An SBT entry consists of a 32-bit virtual address in the virtual address mode or a 24-bit physical address in the physical address mode. In the physical address mode, the high order eight address bits are ignored by the μPD70616 but must be zero for compatibility with future NEC microprocessors.

The SBT is located in the memory address space aligned on a page (4KB) boundary by the SBR (system base register). The first 64 SBT entries (0-63) are reserved for use by μPD70616 interrupts and exceptions. The remaining 192 entries (64-255) are available in user applications as maskable interrupt vectors.

### Interrupt/Exception Recognition

When an interrupt or exception is recognized, the following actions are performed and control is transferred to the specified interrupt/exception handler.

- (i) PSW.EL ← 00  
(If a CHLVL instruction exception or an Asynchronous Task Trap then the specified execution level is set.)
- (ii) PSW.IE flag modification
  - interrupt.....PSW.IE ← 0 (maskable interrupts disabled)
  - exception.....PSW.IE unchanged (bus error and stack invalid exceptions disable interrupts)
- (iii) PSW.TE ← 0  
PSW.TP ← 0  
PSW.AE ← 0
- (iv) PSW.EM ← 0 (native mode)
- (v) PSW.ASA ← 1 (If an ATT occurs then AST are enabled.)
- (vi) temp ← SBT[ vector ]
- (vii) interrupt/exception information is stored on the stack
  - interrupt.....JS (interrupt stack)
  - exception.....LOSP (IS if the previous stack was the interrupt stack or LnSP if a change execution level or ATT exception occurs)



- (viii) PC ← temp



System Base Table

vector		offset	
255	Application Interrupt Vectors (Maskable Interrupts)	+1020	
64			+256
63		Software Trap 15	+252
62		Software Trap 14	+248
61		Software Trap 13	+244
60		Software Trap 12	+240
59		Software Trap 11	+236
58		Software Trap 10	+232
57		Software Trap 9	+228
56		Software Trap 8	+224
55		Software Trap 7	+220
54		Software Trap 6	+216
53		Software Trap 5	+212
52		Software Trap 4	+208
51		Software Trap 3	+204
50		Software Trap 2	+200
49	Software Trap 1	+196	
48	Software Trap 0	+192	
47	RFU	+188	
33			+132
32		Emulation Mode Exception	+128
31		RFU	+124
30		RFU	+120
29		Asynchronous Task Trap	+116
28		Asynchronous System Trap	+112
27		Change to Execution Level 3	+108
26		Change to Execution Level 2	+104
25		Change to Execution Level 1	+100
24		Change to Execution Level 0	+96
23		Decimal Arithmetic Exception	+92
22		Floating Point Arithmetic Exception	+88
21		Integer Arithmetic Exception	+84
20		Illegal Data Field Exception	+80
19		Illegal Addressing Mode Exception	+76
18	Reserved Addressing Mode Exception	+72	
17	Privileged Instruction Exception	+68	
16	Reserved Opcode Exception	+64	
15	RFU	+60	
14	Address Trap	+56	
13	Instruction Breakpoint Exception	+52	
12	Instruction Trace Exception	+48	
11	Address Translation Exception	+44	
10	Memory Protection Exception	+40	
9	Page Not Present Exception	+36	
8	Area Not Present Exception	+32	
7	Stack Invalid Exception	+28	
6	RFU	+24	
5	RFU	+20	
4	System Fault	+16	
3	Serious System Fault	+12	
2	Non-Maskable Interrupt	+8	
1	Bus Freeze	+4	
0	RFU		

← System Base Register (SBR)

86-236

### Interrupts

(i) Maskable Interrupt

Maskable interrupt requests are generated by an external interrupt controller. System software can control the recognition of maskable interrupts by the means of the IE (interrupt enable) bit in the PSW register.

Following the occurrence of a maskable interrupt, further maskable interrupts will be disabled until the PSW.IE is again set.

(ii) Non-Maskable Interrupt (NMI)

Non-maskable interrupts are used to signal the occurrence of catastrophic events such as a loss of power. Non-maskable interrupts cannot be masked by software.

Additional non-maskable interrupts will not be acknowledged until the processing of the first NMI completes and the RETIS instruction is executed.

(iii) Bus Freeze Interrupt

The bus freeze interrupt is used in systems employing functional redundancy monitoring (FRM). If a bus freeze interrupt is detected, the μPD70616 enters the halt state and the address and data buses are placed in the high impedance mode. Actual processing of the bus freeze interrupt is delayed until after the BFREZ input is negated.

During the processing of a bus freeze interrupt, non-maskable interrupts are disabled.

3

### Exceptions

Exceptions are divided into the following categories:

- |   |   |
|---|---|
| <ul style="list-style-type: none"><li>○ Serious System Exceptions</li><li>○ Stack Invalid Exceptions</li><li>○ Software Debug Exceptions</li><li>○ Arithmetic Exceptions</li><li>○ Asynchronous Traps</li><li>○ Emulation Mode Exceptions</li></ul> | <ul style="list-style-type: none"><li>○ System Exceptions</li><li>○ Memory Management Exceptions</li><li>○ Instruction Exceptions</li><li>○ Change Execution Level</li><li>○ Software Traps</li></ul> |
|---|---|

### μPD70616 Instruction Set

The μPD70616 has a large, powerful instruction set characterized by the following features:

- two address instructions
- large number of standard data types
- high level language support
- memory management instructions
- context switch instructions
- multi-processor instructions

A summary of the μPD70616 instruction set can be found in section 5 of this data sheet. For detailed information on the operation of instructions, refer to the μPD70616 Programmer's Reference Manual.

Exception Codes

Code		Code	
<b>Serious System Exceptions</b>		<b>Software Debug Exceptions</b>	
0301	string data write bus error	0C00	instruction trace
0303	fixed length data write bus error	0D00	instruction breakpoint
0305	translation table write bus error	0E01	address trap 0
0309	string I/O write bus error	0E02	address trap 1
030B	fixed length I/O write bus error	0E03	address traps 0 and 1
0311	string data read bus error	<b>Instruction Exceptions</b>	
0313	fixed length data read bus error	1000	reserved instruction
0314	system base table read bus error	1100	privileged instruction
0315	translation table read bus error	1200	reserved address mode
0317	instruction fetch bus error	1300	illegal addressing mode
0319	string I/O read bus error	1301	illegal instruction format
031B	fixed length I/O read bus error	1400	illegal data field
031E	interrupt vector read bus error	<b>Arithmetic Exceptions</b>	
<b>System Exceptions</b>		1500	integer zero divide
0400	illegal interrupt	1501	integer overflow
<b>Stack Invalid Exceptions</b>		1601	floating point precision
0700	area not present	1602	floating point underflow
0701	page not present	1604	floating point overflow
0702	I/O access violation	1608	floating point zero divide
0703	read access violation	1610	invalid floating point operation
0704	write access violation	1680	reserved floating point operand
0705	read/write access violation	1780	decimal format exception
0707	invalid section	<b>Change Execution Level Exceptions</b>	
0708	section length violation	1800	change to execution level 0
0709	invalid area	1900	change to execution level 1
070A	area length violation	1A00	change to execution level 2
070B	invalid page	1B00	change to execution level 3
0780	area not present	<b>Asynchronous Traps</b>	
0781	page not present	1C00	asynchronous system trap
0782	I/O access violation	1D00	asynchronous task trap
0783	read access violation	<b>Emulation Mode Exceptions</b>	
0784	write access violation	2000	emulation mode privileged instruction
0785	read/write access violation	2001	emulation mode reserved instruction
0787	invalid section	2002	emulation mode zero divide
0788	section length violation	2003	emulation mode single step trap
0789	invalid area	2004	emulation mode overflow
078A	area length violation	2005	emulation mode index
078B	invalid page	2006	emulation mode coprocessor not present
<b>Memory Management Exceptions</b>		<b>Software Traps</b>	
0800	area not present	3000	software trap 0
0901	page not present	3100	software trap 1
0A02	I/O access violation	3200	software trap 2
0A03	read access violation	⋮	⋮
0A04	write access violation	3F00	software trap 15
0A05	read/write access violation		
0A06	execute access violation		
0B07	invalid section		
0B08	section length violation		
0B09	invalid area		
0B0A	area length violation		
0B0B	invalid page		

### Two Address Machine

The μPD70616 is a two address machine. μPD70616 instructions are encoded to accept from zero to two operands, generally using any of the available addressing modes without restriction. Two operands allow the specification of a source and destination operand in the same instruction, minimizing the number of instructions that must be executed, especially when instructions frequently reference memory operands.

### Standard Data Types

The μPD70616 includes support for the most common data types encountered in the vast majority of 32-bit applications. The data types include:

- signed and unsigned integers (8-, 16-, 32- and 64-bit sizes)
- IEEE standard binary floating point (32- and 64-bit sizes)
- decimal integers
- bits, bit fields and bit strings
- pointers
- byte (8-bit) and halfword (16-bit) character strings

### High Level Language Support

As the number of programs being written in a high level language increases, the ability of an architecture to support the high speed execution of high level languages becomes even more critical. The μPD70616 provides specific support for:

- subroutine call/return
- procedure call/return
- stack frame management instructions

### MMU Instructions

The inclusion of an on-chip memory management unit mandated instruction set support for the MMU. Instructions to manage the area/page tables and control the contents of the TLB. MMU instructions are privileged instructions and require the processor to be at execution level 0. In addition, MMU instructions can be executed in the physical address mode.

### Context Switching Instructions

A processor must be able to quickly swap out the old context and bring in the new context in a real-time multi-tasking environment. The μPD70616 uses a variable sized TCB to optimize the time and space requirements of the TCB. The instruction set also contains a pair of privileged context switch instructions (LDTASK and STTASK) which simplify the chore of context switching.

### Multiprocessor Support

High performance system designers often turn to multi-processing as a solution to a performance bottleneck. To assist in the design of multi-processor system software primitives for μPD70616 systems, test and set (TAS) and compare and interchange (CAXI) instructions are available.

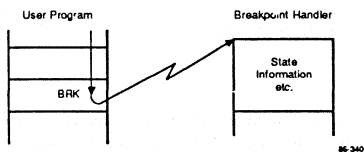
### Software Debug Support

This section describes the software debug facilities of the μPD70616 microprocessor. The size and complexity of 32-bit systems demands that on-chip hardware contribute to the problem of debugging complex application and system software. The on-chip software debug support offered by the μPD70616 aids in quickly identifying the errant sections of a program without spending hours over listings and real-time traces.

The μPD70616 provides three independent hardware aids which address the various requirements of debuggers.

#### Instruction Breakpoints

The instruction breakpoint facility is used to implement program flow debugging. An instruction breakpoint is set by replacing the first byte of an instruction with the one byte BRK instruction. The program then executes at full speed until the breakpoint is reached, causing the breakpoint trap.



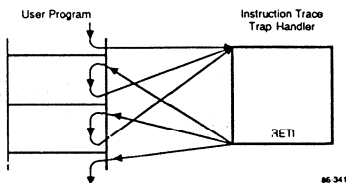
Following the occurrence of an breakpoint trap, the level 0 stack contains the following information:

- PC of the current instruction
- exception code (0D00H)
- PSW image
- parameter count

Note that only the first byte of a multi-byte instruction needs to be replaced with the BRK instruction. Subsequent bytes may be left unmodified without causing an exception since the breakpoint trap guarantees the remaining bytes of the instruction never reach the execution unit.

#### Instruction Trace

Instruction trace (also referred to as single stepping) is a tool used to force an exception following the execution of each instruction. Instruction trace allows a software engineer to observe the execution of a program at the instruction level to locate and correct software errors.



Instruction trace exceptions are controlled by the TE (trace enable) field in the PSW register.

PSW.TE = 0 instruction trace disabled  
 PSW.TE = 1 instruction trace enabled

Because this field is in the upper halfword of the PSW, a privileged instruction is required to enable and disable instruction trace.

When instruction trace exceptions are enabled, following the execution of each instruction an instruction trace exception occurs. Prior to entry into the instruction trace exception handler, the PC and PSW are pushed onto the level 0 stack and the PSW.TE field is cleared to allow the debugger to analyze program execution. To return back to the target program, the RETIS (Return from Interrupt – System) instruction is used.

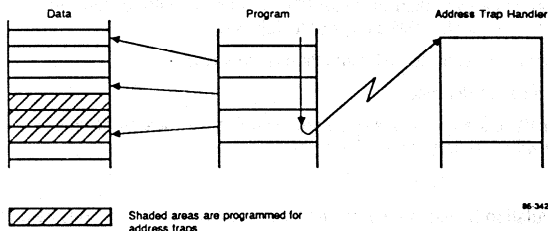
Following the occurrence of an instruction trace exception, the level 0 stack contains the following information:

- PC of the next instruction
- exception code (0C00H)
- PSW image
- parameter count

Refer to the μPD70616 Programmer's Reference Manual for further details.

### Address Traps

Address traps are a powerful debugging facility that combine the occurrence of an address (or range of addresses) with one or more access types (read, write, execute) to generate an exception. Address traps differ from instruction trace and breakpoint traps since they operate on data accesses as well as instruction accesses and can span one or more tasks in a multitasking system.



Address traps are setup and controlled by five privileged registers and a field within the PSW register.

- Address Trap Enable (PSW.AE)

PSW.AE = 0 address traps disabled  
 PSW.AE = 1 address traps enabled

- Address Trap Registers (ADTR0/ADTR1)

The two ADTR registers each contain a 32-bit base trap address. ADTR0 contains the base address for address trap 0 and ADTR1 contains the base address for address trap 1.

- **Address Trap Mask Registers (ADTMR0/ADTMR1)**  
The ADTMR registers each contain a 32-bit mask value that marks the corresponding bits of the ADTR registers as "don't care". ADTMR0 contains the mask value for ADTR0 and ADTMR1 contains the mask value for ADTR1.
- **Trap Mode Register (TRMOD)**  
The TRMOD register contains the access type specifiers (read, write and execute) for each set of ADTR/ADTMR registers.

#### **Address Trap Notes**

- If a memory indirect addressing mode is used and during the effective address calculation the access of the memory resident pointer is within an address trap region with read access enabled, an address trap will occur.
- Address trap logic is disabled during address translation (ATE, PTE accesses) and system base table accesses.

#### **V20/V30 Emulation Mode**

This section describes the operation of the V20/V30 emulation mode. The V20/V30 emulation mode is included in the μPD70616 in order to allow system designers to take advantage of the large installed base of 16-bit software while simultaneously providing an upgrade path to a high performance 32-bit architecture for new applications. V20/V30 emulation mode does not include the μPD8080 emulation mode of the V20/V30.

#### **I/O Emulation Option**

The I/O emulation option permits emulation mode programs to override the privileged status of IN and OUT instructions and permit selective execution of these I/O instructions without the generation of an exception. This option is controlled by the CTL bit in the PSW2 register as follows:

- CTL = 0      I/O emulation enabled (I/O instructions privileged)
- CTL = 1      I/O emulation disabled

Control of the trapping of I/O instructions is particularly useful in the design of virtual machine monitors for device drivers and other software with real-time I/O requirements.

#### **Mode Switching**

μPD70616 native and emulation modes are controlled by the EM (emulation mode) field in the PSW. When the EM field is cleared, the μPD70616 is operating in native mode. When the bit is set, the μPD70616 is operating in emulation mode.

- (i) **Native Mode → Emulation Mode**  
The EM flag is in a privileged field in the upper halfword of the PSW and only software running at execution level 0 can change processor modes. Because both the flag and the instruction streams must change simultaneously, V20/V30 emulation mode is entered by pushing the emulation mode PC on the stack along with a PSW image with the PSW.EM=1 and PSW.EL=11 and executing a RETIS instruction.

**(ii) Emulation Mode → Native Mode**

Return to native mode from emulation mode programs occurs as the result of an interrupt or exception. Emulation mode will be automatically restored upon completion of the interrupt/exception handler.

- **Interrupts**

All interrupts on the μPD70616 microprocessor are processed in native mode. Upon the detection of an interrupt, the emulation mode program will be suspended and native mode will be restored.

- **Exceptions**

Emulation mode exceptions occur as a result of program execution and require processing by a native mode handler to correct a fault or take some other predetermined action.

- **Emulation Mode Privileged Instructions**

The emulation mode system instructions listed below have been made privileged to allow operation in a protected environment:

IN\*, OUT\*, INM, OUTM, HALT, BRK, RETI, POP PSW\*, EI, DI (\*conditionally trapped)

- Emulation Mode Reserved Opcode
  - Emulation Mode Zero Divide
  - Emulation Mode Single Step Trap
  - Emulation Mode Overflow
  - Emulation Mode Index
  - Emulation Mode Floating Point Operation

In either case, the return to emulation mode after processing of the exception or interrupt is by the RETIS instruction.

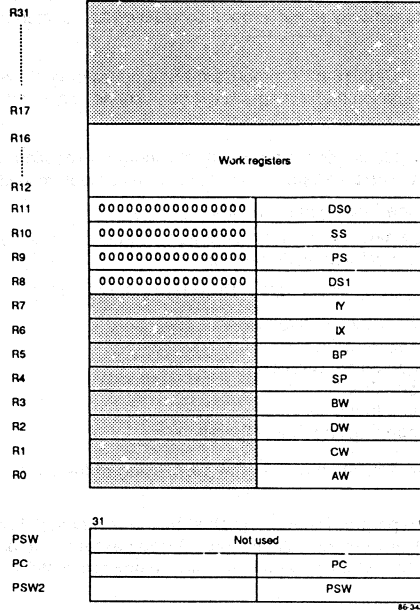
Termination of an emulation program is accomplished by the reserved 63H opcode. Attempted execution of this opcode will cause the Reserved Emulation Mode Opcode exception to be generated along with a specific exception code identifying the exception as an emulation mode terminate operation.

**Emulation Mode Notes**

- (i) Emulation mode operates in both the virtual and physical address modes.
- (ii) Emulation mode programs must run at execution level 3, attempts to enter emulation mode at a level other than 3 will cause an exception.
- (iii) In virtual mode, emulation mode uses virtual addresses in the range 00000H to FFFFFH. The address translation tables are used to provide multiple virtual emulation mode address spaces. Physical address mode is limited to a single emulation address space.



Emulation Mode Register Assignments



Unaffected

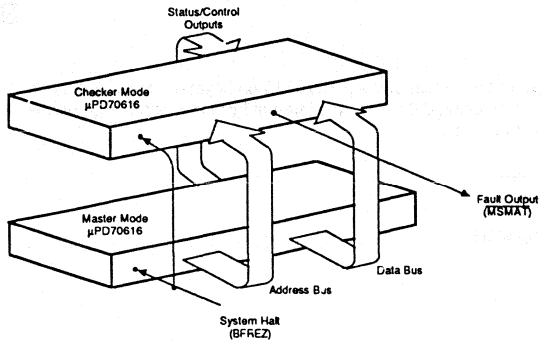
**FRM (Functional Redundancy Monitor)**

The μPD70616 is ideally suited for the design of fault tolerant computing systems. Fault tolerant computer design is driven by requirement to detect and recover from faults without affecting the applications through the use of redundancy. The μPD70616 uses FRM (function redundancy monitor) logic to simplify the hardware design of fault tolerant systems by including on-chip the comparison logic needed to detect and recover from faults.

FRM requires that one processor operate as the master and one or more μPD70616 microprocessors be configured as checkers. This is accomplished by the BMODE/FRM pin which when programmed to operate in the FRM mode, causes checker processors to three-state their output buffers and compare the state of the master with their internal computed state. Should a difference be detected, the MSMAT (mismatch) signal is output to allow external logic to stop the processors, isolate the fault and reconfigure the system.

Following the detection of a fault, the BFREZ (bus freeze) input of each processor must asserted to halt all further processing. The isolation and reconfiguration is then under the control of external hardware which can either quickly reconfigure the system or perform individual testing on each processor to determine the exact cause of the fault. In the case of a transient fault, all processors can be re-initialized to the same state and processing can continue.

### FRM System Example



84-226

Section 4  
Bus Operation

This section describes the operation of each of the μPD70616 bus cycles. Each bus cycle consists of a combination of the seven bus states, each state being defined as the interval from the rising edge of one clock to the rising edge of the next clock. The seven bus states are:

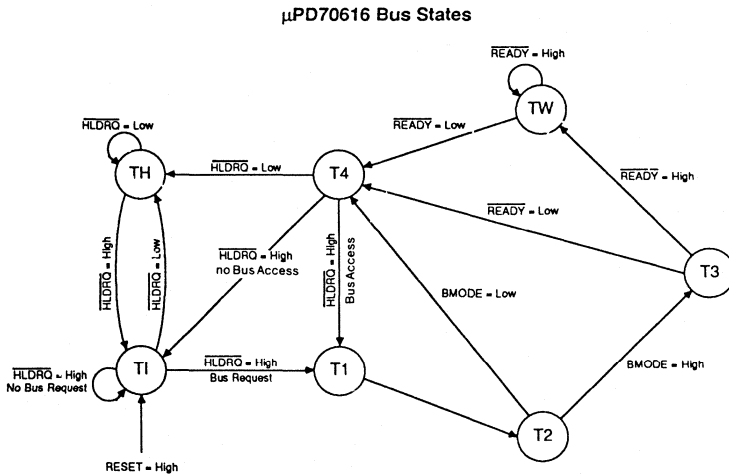
- Tl : bus idle state
- T1 : bus cycle start state
- T2 : bus cycle state 2
- T3 : bus cycle state 3
- T4 : bus cycle ending state
- TW : wait state
- TH : bus hold state

Bus Cycle Modes

The standard μPD70616 bus cycle is a four clock bus cycle consisting of bus states T1, T2, T3 and T4. In addition to these states, external hardware can force the insertion of TW states between the T3 and T4 states of memory and I/O bus cycles by negating the READY\* input.

In addition to the standard four clock bus cycle, a high speed three clock bus cycle is available. The three clock mode eliminates the T3 bus state by passing from bus state T2 directly to T4 without the opportunity of inserting wait states. This mode is provided for accessing high speed memory and I/O peripheral devices. Bus mode selection is made on a cycle by cycle basis by the BMODE input pin. BMODE is sampled at the falling clock edge of the T2 state to determine whether a normal or short bus cycle will be run.

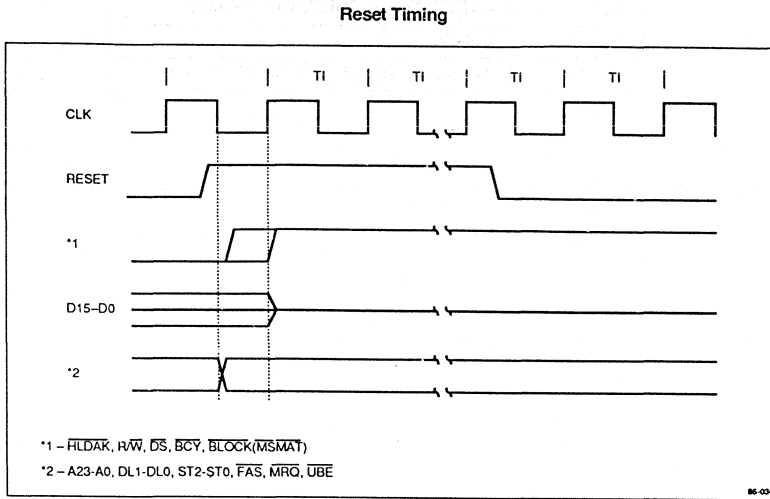
Below is a diagram showing the state transitions of the various bus cycles.



86 011

### RESET Timing

When the active high RESET input is asserted, the μPD70616 stops all processing and enters the reset state.



RESET must be held asserted for a minimum of 20 clock cycles before returning to a low level. Following the release from the reset state, the μPD70616 will exit the idle state and begin execution by performing a memory read (instruction fetch) bus cycle from address 0FFFFFF0H.

Register	Reset Value
PSW	1 0 0 0 0 0 0 0 H
PC	F F F F F F 0 H
SBR	0 0 0 0 0 0 0 0 H
SYCW	0 0 0 0 0 0 7 0 H
TKCW	0 0 0 0 E 0 0 0 H
PSW2	0 0 0 0 F 0 0 2 H
ATBR	Invalid
TLB	Cleared
Others	Undefined

86-295

While in the reset state, the bus interface unit is idle and output pins are in following states:

State	Output
High	R/W, DS, BCY, HLDAR, BLOCK (MSMAT)
High-Z	D15-D0
Undefined	A23-A0, DL1-DL0, FAS, MRQ, UBĒ, ST2-ST0

86-294

### Memory Read Bus Cycles

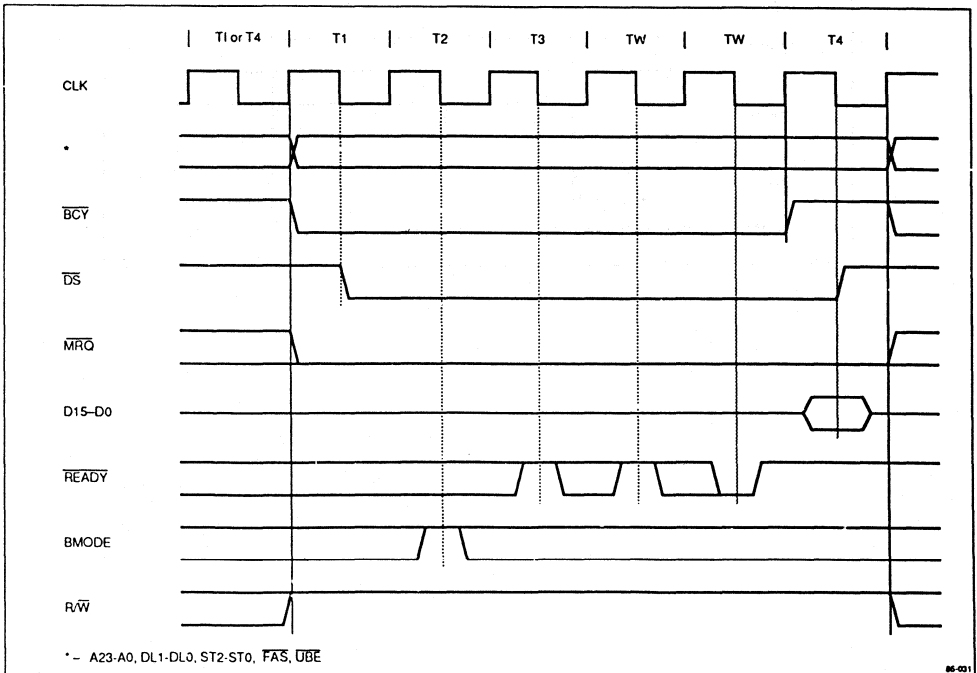
Memory read bus cycles consist of bus states T1, T2, T3 and T4. Memory read bus cycles are distinguished by MRQ\* (memory request) being asserted and R/W\* being at a high level.

At the rising edge of the T1 bus state, the A23-A0, DL1-DL0, ST2-ST0, FAS\*, MRQ\*, R/W\* and UBE\* outputs are asserted and remain valid until the end of the bus cycle. At the same time, the BCY\* pin is asserted indicating the start of a bus cycle. At the falling edge of the T1 clock, the DS\* pin is asserted, indicating that external data bus transceivers can be enabled. At the falling edge of the T2 state, the BMODE input is sampled to determine if the bus cycle is normal (logic high) or a short (logic low) bus cycle. If the normal mode is selected, the T2 state progresses to the T3 state. Otherwise the T3 state is skipped, the READY input ignored and T4 is the next bus state.

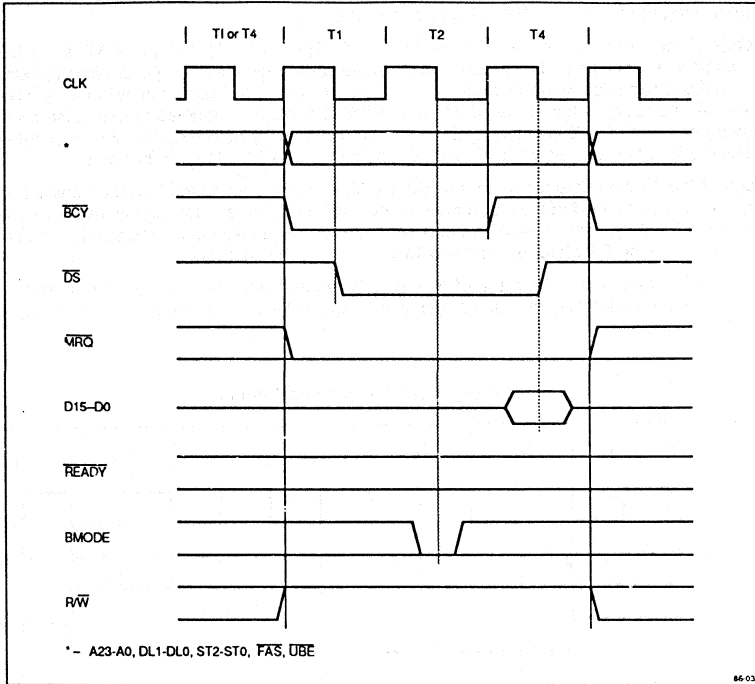
The falling edge of the T3 state is used to sample the READY\* input and if found low, the transition to the T4 state occurs. External hardware can extend the read cycle by negating READY\* and forcing the insertion of TW states after the completion of the T3 state. The READY\* input is continued to be sampled at the falling edge of TW until it returns to a low level. At this time the TW states will end and T4 will occur as the next state.

In both normal and short bus cycle modes, the BCY\* output is negated at the rising edge of the T4 bus state. The read data on the D15-D0 inputs is latched internally at the falling edge of T4 and simultaneously the DS\* output is negated, completing the bus cycle.

Memory Read Timing (Normal Mode)



Memory Read Timing (Short Mode)

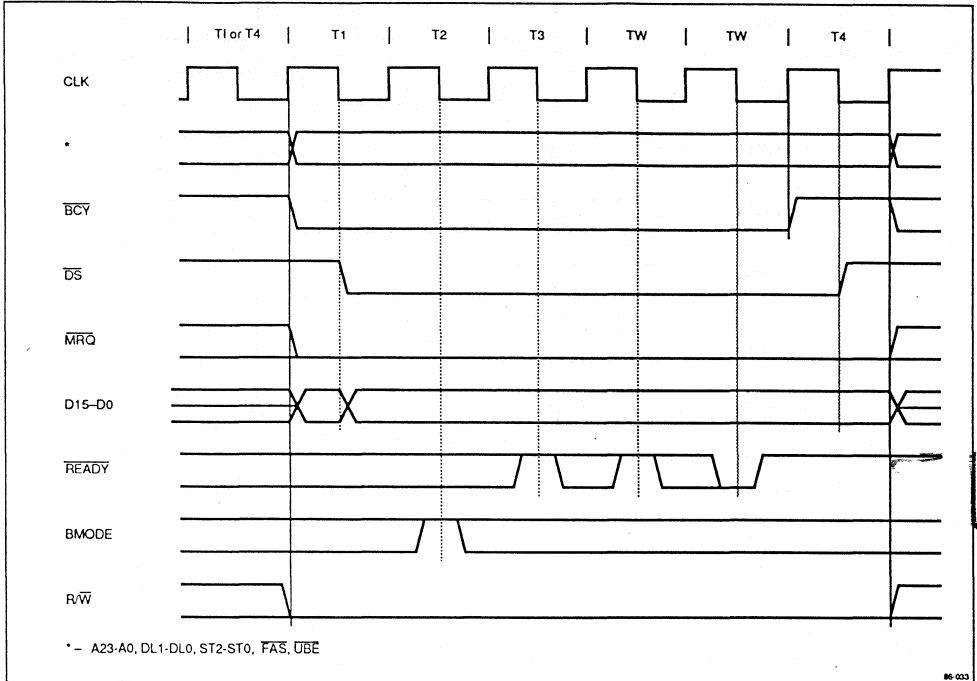


### Memory Write Bus Cycles

Memory write bus cycles consist of bus states T1, T2, T3 and T4. Memory write bus cycles are distinguished by MRQ\* (memory request) being asserted and R/W\* being at a low level. All other outputs (A23-A0, DL1-DL0, ST2-ST0, FAS\*, MRQ\*, UBE\*) and the bus mode selection function are identical.

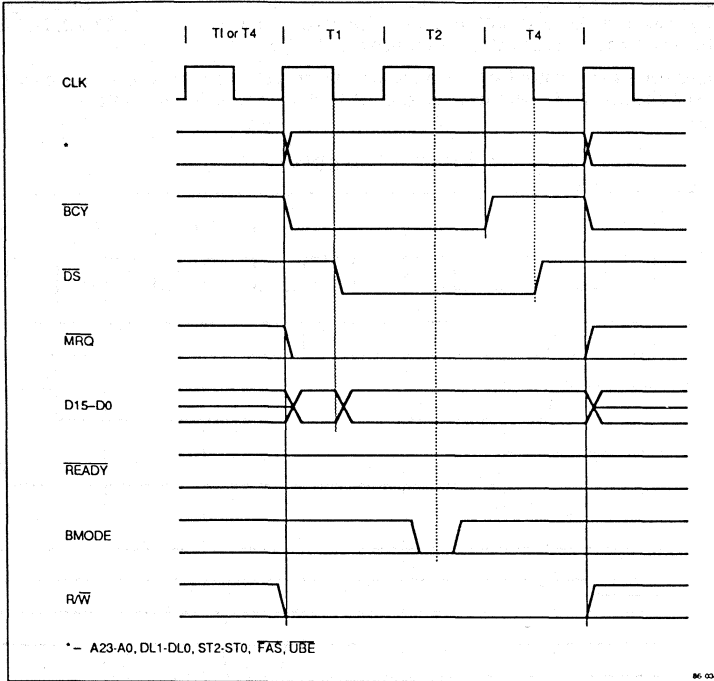
The D15-D0 output buffers are enabled at the start of the T1 state but valid data is not presented until one half clock period later at the falling edge of T1. Write data remains valid until the end of the T4 bus state.

Memory Write Timing (Normal Mode)





Memory Write Timing (Short Mode)



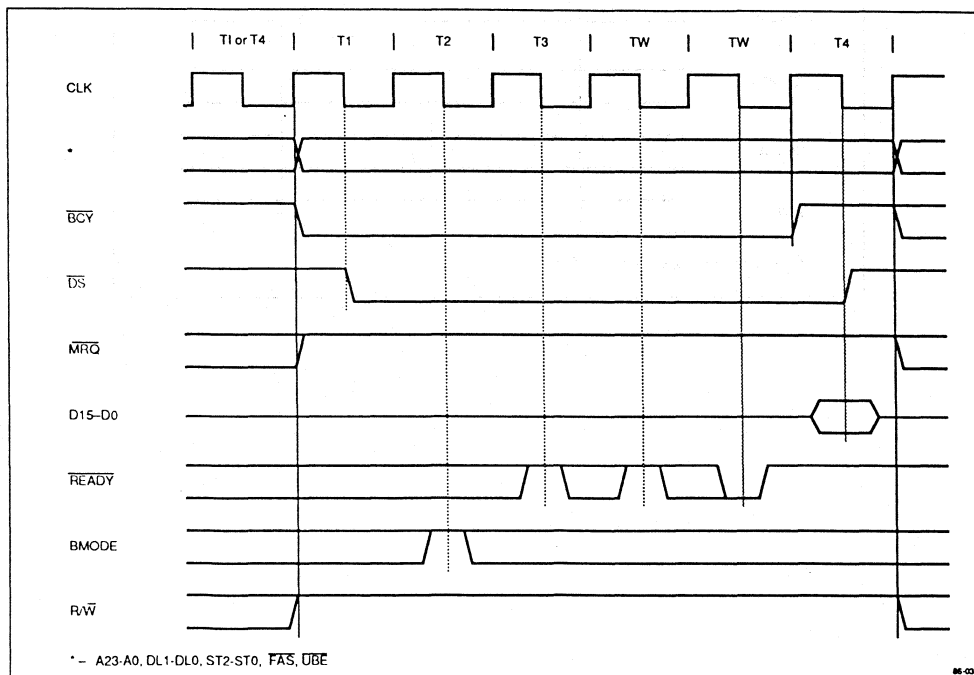
### I/O Read Bus Cycles

I/O read bus cycles consist of bus states T1, T2, T3 and T4. I/O read bus cycles are distinguished by MRQ\* (memory request) being negated and R/W\* being at a high level.

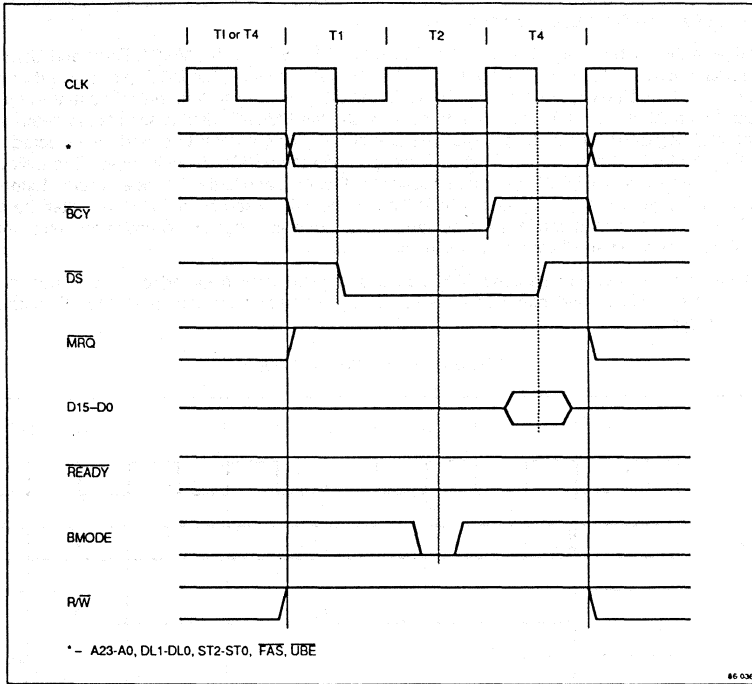
At the rising edge of the T1 bus state, the A23-A0, DL1-DL0, ST2-ST0, FAS\*, MRQ\*, R/W\* and UBE\* outputs are asserted and remain active until the end of the bus cycle. At the same time, the BCY\* pin is asserted indicating the start of a bus cycle. At the falling edge of the T1 clock, the DS\* pin is asserted, indicating that external data bus transceivers can be enabled. At the falling edge of the T2 state, the BMODE input is sampled to determine if the bus cycle will be normal (logic high) or a short (logic low) bus cycle. If the normal bus mode is selected, the T2 state progresses to the T3 state. Otherwise the T3 state is skipped and the READY input ignored. The falling edge of the T3 state is used to sample the READY\* input and if found low, the transition to the T4 state occurs. External hardware can extend an I/O read bus cycle by negating READY\* and forcing the insertion of TW states after the completion of the T3 state. The READY\* input is continued to be sampled at the falling edge of TW until it returns to a low level. At this time the TW states will end and T4 will occur as the next state.

In both normal and short bus cycle modes, the BCY\* output is negated at the rising edge of the T4 bus state. The read data on the D15-D0 inputs is latched internally at the falling edge of T4 and simultaneously the DS\* output is negated, completing the bus cycle.

I/O Read Timing (Normal Mode)



I/O Read Timing (Short Mode)

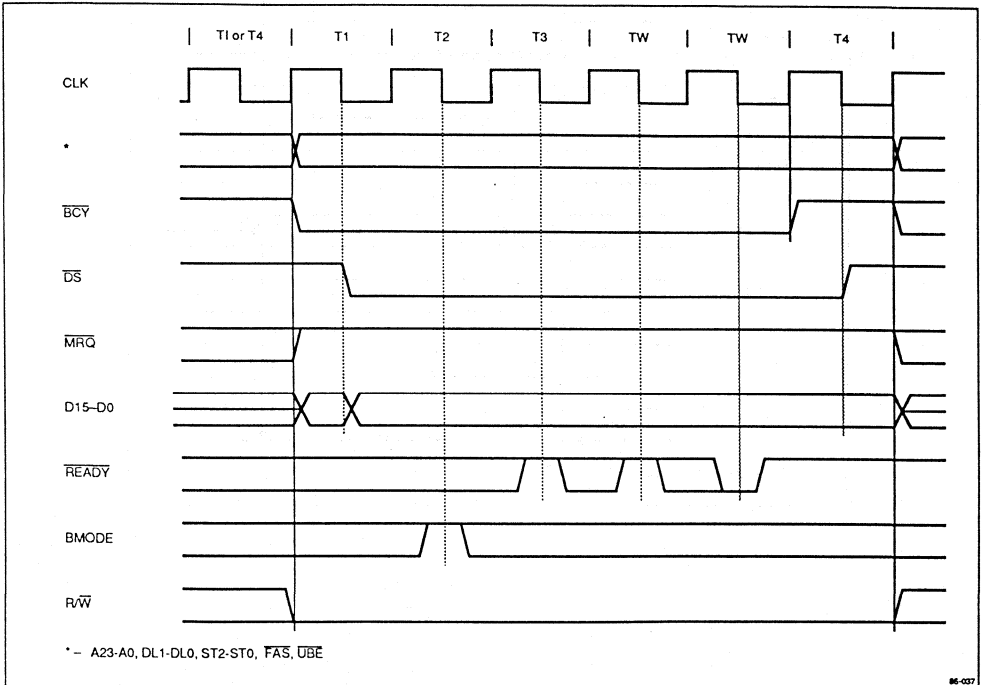


### I/O Write Bus Cycles

I/O write bus cycles consist of bus states T1, T2, T3 and T4. I/O write bus cycles are distinguished by MRQ\* (memory request) being negated and R/W\* being at a low level. All other outputs (A23-A0, DL1-DL0, ST2-ST0, FAS\*, MRQ\*, UBE\*) and the bus mode selection function are identical.

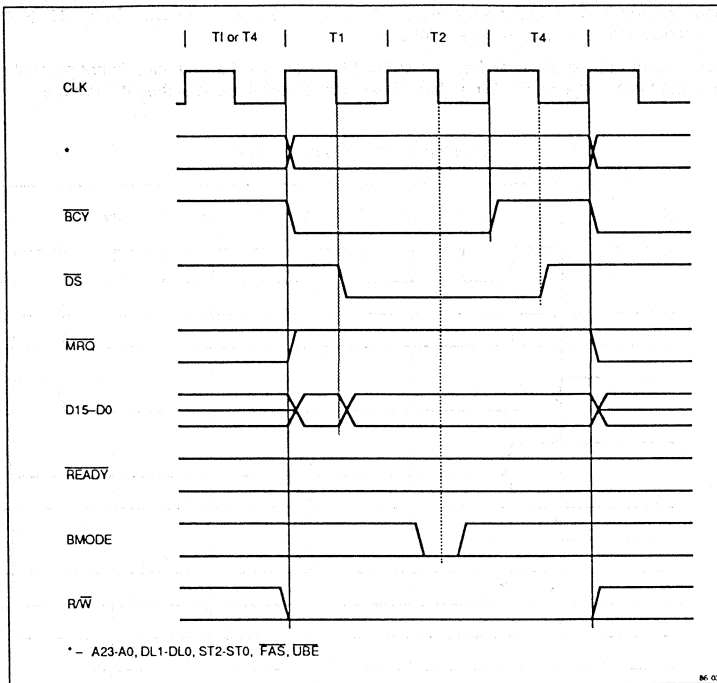
The D15-D0 output buffers are enabled at the start of the T1 state but valid write data is not presented until one half clock period later at the falling edge of T1. Write data remains valid until the end of the T4 bus state.

I/O Write Timing (Normal Mode)



3

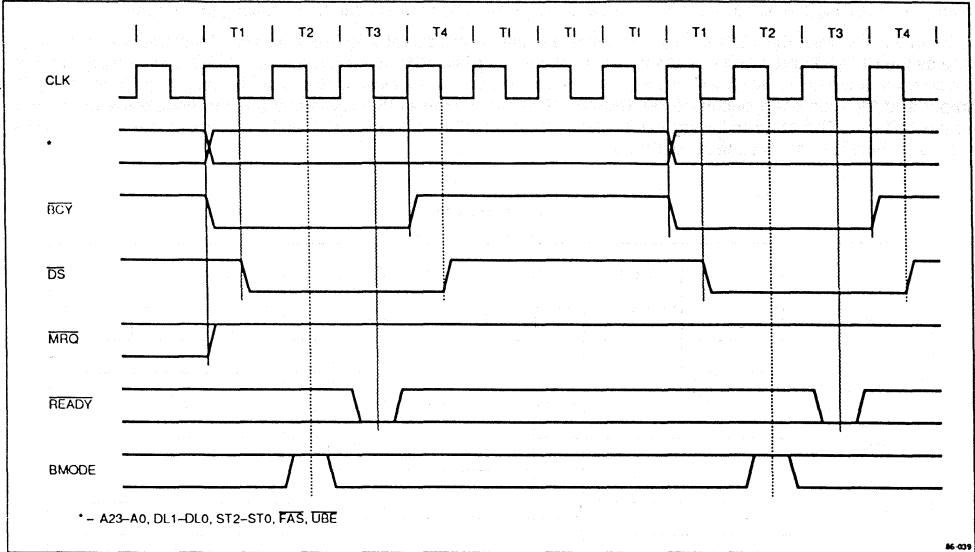
I/O Write Timing (Short Mode)



### Back-to-Back I/O Bus Cycles

I/O bus cycle timing is automatically modified during the execution of back-to-back I/O bus cycles. In order to meet the read and write recovery times of peripheral devices without the need for external logic, three T1 states are inserted between any consecutive pair of I/O bus cycles.

Back-to-Back I/O Bus Cycles



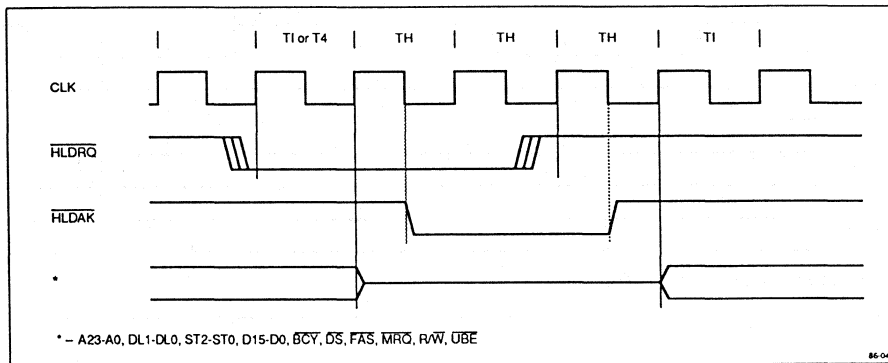
### Bus Hold Timing

External bus masters such as DMA controllers can request the μPD70616 to relinquish the bus to allow external control of the address, data and control buses.

The rising edge of the clock in the T4 or T1 state is used to sample the HLDRO\* input for external bus requests. If the HLDRO\* input is asserted (low level), the next bus state will be the TH or bus hold state. The rising edge of the clock in the TH state will see A23-A0, D15-D0, DL1-DL0, ST2-ST0, BCY\*, DS\*, FAS\*, MRQ\*, R/W\* and UBE\* outputs enter the high impedance state. One half clock cycle later at the falling edge of the initial TH state, the HLDAK\* output will asserted, indicating the bus as been released and the external bus master can begin driving the buses.

While in the TH state, the HLDRO\* input is sampled at the rising edge of each clock and the TH states will continue as long as the HLDRO\* input remains at a low level. When the μPD70616 detects that the HLDRO\* input has returned to a high level, the next TH state will be the last. The falling edge of the clock in the final TH state negates the HLDAK\* output and the next state becomes the T1 state. The rising edge of the T1 state is used to enable the μPD70616 output buffers. The state following the T1 state will be T1 if any internal μPD70616 bus requester has an active request, otherwise, the T1 state is continued.

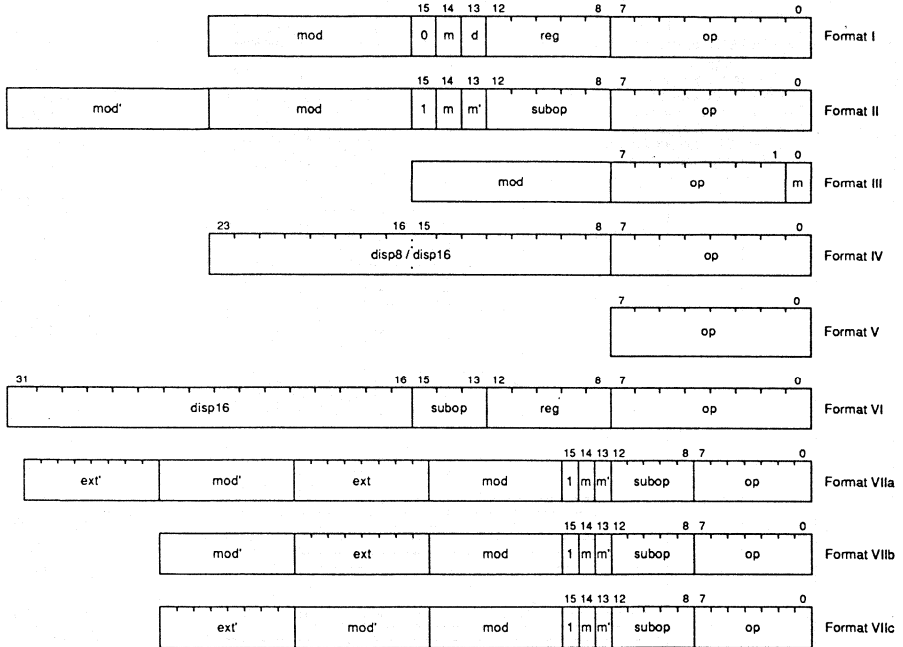
Bus Hold Cycle



### Section 5 Instruction Set

The μPD70616 instruction set operation codes are described by seven instruction formats. This section presents the instruction formats, addressing mode encodings, mnemonics and the effect of instructions on the flags and condition codes.

#### Instruction Formats



op, subop : opcode fields  
 disp : signed displacement  
 d : direction field  
 \* : second operand identifier

reg : register field  
 mod, m : address mode field  
 ext : operand extension field

86-304

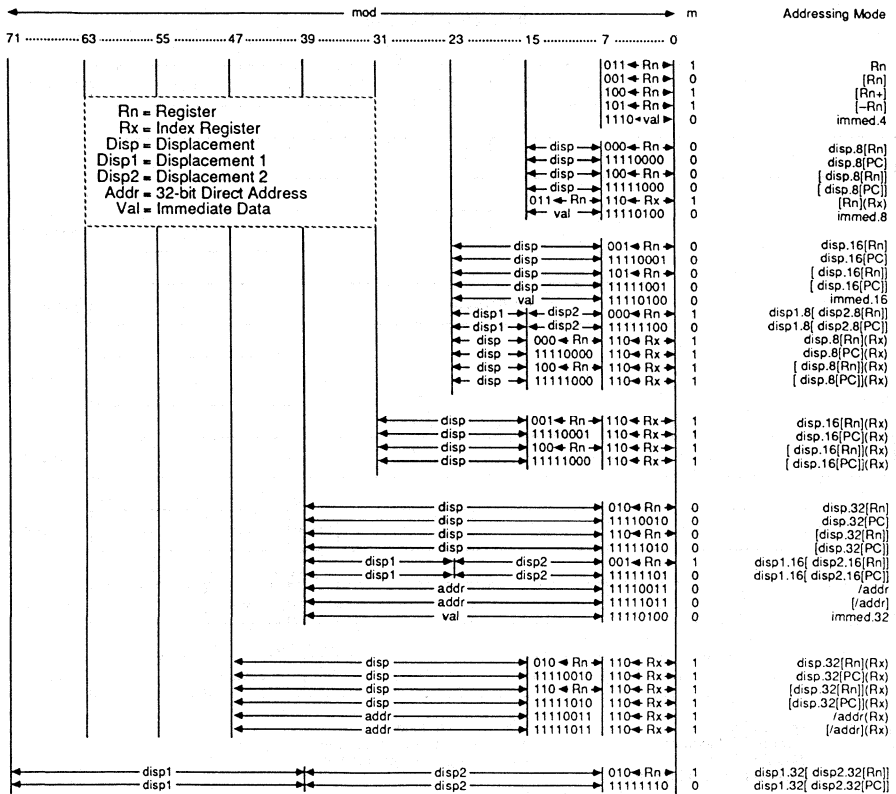
The extension field is used to specify the length of a variable length data type and is encoded as follows:

- bit 7 (ext) = 0 → bits 6:0 (ext) are the operand length
- bit 7 (ext) = 1 → bits 6:0 (ext) contain a pointer (register ID) to the general purpose register containing the operand length



Instructions formats have a 1 to 9 byte mod (modifier) field to specify along with the m field the addressing mode for each operand reference within an instruction. The mod field also contains additional information such as the optional base register, index register and signed displacement fields. The table below lists the byte addressing modes. Bit addressing modes use the same encodings as the equivalent byte addressing modes, only the assembler format differs.

mod Field – Addressing Mode Encoding



### μPD70616 Instruction Set

#### Condition Encodings

c3	c2	c1	c0	Name	Condition
0	0	0	0	Overflow	OV = 1
0	0	0	1	No overflow	OV = 0
0	0	1	0	Carry / Lower	CY = 1
0	0	1	1	No carry / Not lower	CY = 0
0	1	0	0	Zero / Equal	Z = 1
0	1	0	1	Not zero / Not equal	Z = 0
0	1	1	0	Not higher	(CY ∨ Z) = 1
0	1	1	1	Higher	(CY ∨ Z) = 0
1	0	0	0	Sign / Negative	S = 1
1	0	0	1	Not sign / Positive	S = 0
1	0	1	0	True	Always
1	0	1	1	False	Never
1	1	0	0	Less than	(S ⊕ OV) = 1
1	1	0	1	Greater or equal	(S ⊕ OV) = 0
1	1	1	0	Less or equal	((S ⊕ OV) ∨ Z) = 1
1	1	1	1	Greater than	((S ⊕ OV) ∨ Z) = 0

#### Register Selection

reg	Register
0 0 0 0 0	r0
0 0 0 0 1	r1
0 0 0 1 0	r2
...	
1 1 1 0 11	r29
1 1 1 1 00	r30
1 1 1 1 11	r31

#### Integer Data Type Selection

siz	Data Type
00	byte
01	halfword
10	word
11	reserved

#### Floating Point Data Type Selection

s	Data Type
0	short real
1	long real

#### String Direction

d	Direction
0	increment
1	decrement

#### Character Data Type Selection

c	Character Data Type
0	byte
1	halfword

#### Bit Field Extension

ext	Data Type
00	signed
01	unsigned
10	right justified
11	reserved

#### Branch Displacements

b	Displacement
0	byte
1	halfword

Mnemonic	Opcodes										Instruction Format	Clocks	Flags				Exceptions						
	7	6	5	4	3	2	1	0	7	6			5	4	3	2		1	0	CY	OV	S	Z
Data Transfer Instructions																							
MOV.B	0	0	0	0	1	0	0	1									I, II					1, 3	
MOV.H	0	0	0	1	1	0	1	1									I, II					1, 3	
MOV.W	0	0	1	0	1	1	0	1									I, II					1, 3	
MOV.D	0	0	1	1	1	1	1	1									I, II					1, 3	
MOVS.BH	0	0	0	0	1	0	1	0									I, II					1	
MOVS.BW	0	0	0	0	1	1	0	0									I, II					1	
MOVS.HW	0	0	0	1	1	1	0	0									I, II					1	
MOVZ.BH	0	0	0	0	1	0	1	1									I, II					1	
MOVZ.BW	0	0	0	0	1	1	0	1									I, II					1	
MOVZ.HW	0	0	0	1	1	1	0	1									I, II					1	
MOVT.HB	0	0	0	1	1	0	0	1									I, II	-	*	-	-	1	
MOVT.WB	0	0	1	0	1	0	0	1									I, II	-	*	-	-	1	
MOVT.WH	0	0	1	0	1	0	1	1									I, II	-	*	-	-	1	
XCH	0	1	0	0	0	siz	1									I, II					1, 3		
MOVEA	0	1	0	0	0	siz	0									I, II					1		
RVBYT	0	0	1	0	1	1	0	0									I, II					1	
RVBIT	0	0	0	0	1	0	0	0									I, II					1	
Integer Arithmetic Instructions																							
ADD	1	0	0	0	0	siz	0									I, II		*	*	*	*	1	
ADDC	1	0	0	1	0	siz	0									I, II		*	*	*	*	1	
SUB	1	0	1	0	1	siz	0									I, II		*	*	*	*	1	
SUBC	1	0	0	1	1	siz	0									I, II		*	*	*	*	1	
MUL	1	0	0	0	0	siz	1									I, II	-	*	*	*	*	1	
MULU	1	0	0	1	0	siz	1									I, II	-	*	*	*	*	1	
MULX	1	0	0	0	1	1	0									I, II	-	*	*	*	*	1	
MULUX	1	0	0	1	0	1	1	0									I, II	-	*	*	*	*	1
DIV	1	0	1	0	0	siz	1									I, II	-	*	*	*	*	1, 4	
DIVU	1	0	1	1	0	siz	1									I, II	-	0	*	*	*	1, 4	
DIVX	1	0	1	0	0	1	1	0									I, II	-	*	*	*	*	1, 4
DIVUX	1	0	1	1	0	1	1	0									I, II	-	*	*	*	*	1, 4
REM	0	1	0	1	0	siz	0									I, II	-	0	*	*	*	1, 4	
REMU	0	1	0	1	0	siz	1									I, II	-	0	*	*	*	1, 4	
INC	1	1	0	1	1	siz	-									III		*	*	*	*	1	
DEC	1	1	0	1	0	siz	-									III		*	*	*	*	1	
NEG	0	0	1	1	1	siz	1									I, II		*	*	*	*	1	
CMP	1	0	1	1	1	siz	0									I, II		*	*	*	*		
TEST	1	1	1	1	0	siz	-									III	0	0	*	*			
Logical Instructions																							
AND	1	0	1	0	0	siz	0									I, II	-	0	*	*	*	1	
OR	1	0	0	0	1	siz	0									I, II	-	0	*	*	*	1	
XOR	1	0	1	1	0	siz	0									I, II	-	0	*	*	*	1	
NOT	0	0	1	1	1	siz	0									I, II	-	0	*	*	*	1	

Mnemonic	Opcode										Instruction Format	Clocks	Flags			Exceptions							
	7	6	5	4	3	2	1	0	7	6			5	4	3		2	1	0	CY	OV	S	Z
<b>Shift/Rotate Instructions</b>																							
SHA	1	0	1	1	0	siz	1												*	*	*	*	1
SHL	1	0	1	0	0	siz	1												*	0	*	*	1
ROT	1	0	0	0	1	siz	1												*	0	*	*	1
ROTC	1	0	0	1	1	siz	1												*	0	*	*	1
<b>Floating Point Instructions</b>																							
MOVF	0	1	0	1	1	1	s	0	0	0	0	0	1	0	0	0			*	0	*	*	1, 3, 7, 9
ADDF	0	1	0	1	1	1	s	0	0	0	0	1	1	0	0	0			*	0	*	*	1, 3, 6, 7, 8, 9
SUBF	0	1	0	1	1	1	s	0	0	0	0	1	1	0	0	1			*	0	*	*	1, 3, 6, 7, 8, 9
MULF	0	1	0	1	1	1	s	0	0	0	0	1	1	0	1	0			*	0	*	*	1, 3, 6, 7, 8, 9
DIVF	0	1	0	1	1	1	s	0	0	0	0	1	1	0	1	1			*	0	*	*	1, 3, 6, 7, 8, 9, 10, 11
CMPF	0	1	0	1	1	1	s	0	0	0	0	0	0	0	0	0			*	*	*	*	1, 3, 6, 7, 8, 9
NEGF	0	1	0	1	1	1	s	0	0	0	0	0	1	0	0	1			*	0	*	*	1, 3, 7, 9
ABSF	0	1	0	1	1	1	s	0	0	0	0	0	1	0	1	0			0	0	0	*	1, 3, 7, 9
SCLF	0	1	0	1	1	1	s	0	0	0	0	1	0	0	0	0			*	0	*	*	1, 3, 6, 7, 8, 9
CVTF	0	1	0	1	1	1	1	1	0	0	0	0	1	0	0	0			*	0	*	*	1, 3, 6, 7, 8, 9
CVT.WS	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0			*	0	*	*	1, 3, 6, 7, 8, 9
CVT.WL	0	1	0	1	1	1	1	1	0	0	0	1	0	0	0	1			*	0	*	*	1, 3, 6, 7, 8, 9
CVT.SW	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	1			-	*	*	*	1, 3, 6, 7, 8, 9
CVT.LW	0	1	0	1	1	1	1	1	0	0	0	0	1	0	0	1			-	*	*	*	1, 3, 6, 7, 8, 9
TRAPFL	1	1	0	0	1	0	1	1															1, 3, 6, 7, 9
<b>Decimal Arithmetic Instructions</b>																							
ADDDC	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0			V	i	c	1, 5	
SUBDC	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	1			V	i	c	1, 5	
SUBRDC	0	1	0	1	1	0	0	1	0	0	0	0	0	0	1	0			V	i	c	1, 5	
CVTD.PZ	0	1	0	1	1	0	0	1	0	0	0	1	0	0	0	0			V	i	c	1, 5	
CVTD.ZP	0	1	0	1	1	0	0	1	0	0	0	1	1	0	0	0			V	i	c	1, 5	
<b>Bit Manipulation Instructions</b>																							
TEST1	1	0	0	0	0	1	1	1											I	, II	1, 2		
SET1	1	0	0	1	0	1	1	1											I	, II	1, 2		
CLR1	1	0	1	0	0	1	1	1											I	, II	1, 2		
NOT1	1	0	1	1	0	1	1	1											I	, II	1, 2		
<b>Bit Field Instructions</b>																							
EXTBF	0	1	0	1	1	1	0	1	0	0	0	0	1	0	ext				V	i	b	1, 2	
INSBF	0	1	0	1	1	1	0	1	0	0	0	1	1	0	ext				V	i	c	1, 2	
CMPBF	0	1	0	1	1	1	0	1	0	0	0	0	0	0	ext				V	i	b	1, 2	
<b>Bit String Instructions</b>																							
MOVBS	0	1	0	1	1	0	1	1	0	0	0	0	1	0	0	d				V	i	b	1
NOTBS	0	1	0	1	1	0	1	1	0	0	0	0	1	0	1	d				V	i	b	1
ANDBS	0	1	0	1	1	0	1	1	0	0	0	1	0	0	0	d				V	i	b	1
ANDNBS	0	1	0	1	1	0	1	1	0	0	0	1	0	0	1	d				V	i	b	1
ORBS	0	1	0	1	1	0	1	1	0	0	0	1	0	1	0	d				V	i	b	1

Mnemonic	Opcode										Instruction Format	Clocks	Flags				Exceptions						
	7	6	5	4	3	2	1	0	7	6			5	4	3	2		1	0	CY	OV	S	Z
Bit String Instructions (cont)																							
ORNBS	0	1	0	1	1	0	1	1	1	0	0	0	1	0	0	1	d	VIIb					1
XORBS	0	1	0	1	1	0	1	1	1	0	0	0	1	1	0	0	d	VIIb					1
XORNBS	0	1	0	1	1	0	1	1	1	0	0	0	1	0	0	1	d	VIIb					1
SCH0BS	0	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	d	VIIb					1
SCH1BS	0	1	0	1	1	0	1	1	1	0	0	0	0	1	0	1	d	VIIb					1
Character Manipulation Instructions																							
MOVc	0	1	0	1	1	0	c	0		0	0	0	0	1	0	0	d	VIIa					1,3
MOVcF	0	1	0	1	1	0	c	0		0	0	0	0	1	0	1	d	VIIa					1,3
MOVCS	0	1	0	1	1	0	0	0		0	0	0	0	1	1	0	0	VIIa					1,3
CMPC	0	1	0	1	1	0	c	0		0	0	0	0	0	0	0	0	VIIa	-	-	*	*	1,3
CMPCF	0	1	0	1	1	0	c	0		0	0	0	0	0	0	0	1	VIIa	-	-	*	*	1,3
CMPCS	0	1	0	1	1	0	c	0		0	0	0	0	0	0	1	0	VIIa	-	-	*	*	1,3
SCHc	0	1	0	1	1	0	c	0		0	0	0	1	1	0	0	d	VIIb	-	-	-	*	1,3
SKPC	0	1	0	1	1	0	c	0		0	0	0	1	1	0	1	d	VIIb	-	-	-	*	1,3
Stack Manipulation Instructions																							
PUSH	1	1	1	0	1	1	1	1	-									II					1,3
POP	1	1	1	0	0	1	1	1	-									II					1,3
PUSHM	1	1	1	0	1	1	0	1	-									II					1,3
POPM	1	1	1	0	0	1	0	1	-									II	R	R	R	R	1,3 (Note 1)
PREPARE	1	1	0	1	1	1	1	1	-									II					1,3
DISPOSE	1	1	0	0	1	1	0	0										V					
Control Transfer Instructions																							
Bcc	0	1	1	b	c <sub>3</sub>	c <sub>2</sub>	c <sub>1</sub>	c <sub>0</sub>										IV					
DBcc	1	1	0	0	0	1	1	c <sub>0</sub>		reg		c <sub>3</sub>	c <sub>2</sub>	c <sub>1</sub>				VI					
TB	1	1	0	0	0	1	1	1		reg		1	0	1				VI					
JMP	1	1	0	1	0	1	1	1	-									III					1
BSR	0	1	0	0	1	0	0	0										IV					
JSR	1	1	1	0	1	0	0	-									III					1	
RSR	1	1	0	0	1	0	1	0										V					
CALL	0	1	0	0	1	0	0	1										II					1
RET	1	1	1	0	0	0	1	-									III						
BRK	1	1	0	0	1	0	0	0										V					
BRKV	1	1	0	0	1	0	0	1										V					
TRAP	1	1	1	0	1	0	0	-									III						
RETIU	1	1	1	0	1	0	1	-									III						
Miscellaneous Instructions																							
NOP	1	1	0	0	0	1	1	0	1									V					
GETPSW	1	1	1	1	0	1	1	1	-									III					1
UPDPSW.H	0	1	0	0	1	0	1	0										I, II	*	*	*	*	
CHLVL	0	1	0	0	1	0	1	1										I, II					1
CHKAR	0	1	0	0	1	1	0	1										I, II	*	-	*	*	1

Mnemonic	Opcode										Instruction Format	Cycles	Flags			Exceptions							
	7	6	5	4	3	2	1	0	7	6			5	4	3		2	1	0	CY	OV	S	Z
<b>Miscellaneous Instructions (cont)</b>																							
CHKAW	0	1	0	0	1	1	1	0									I, II		*	-	*	*	1
CHKAE	0	1	0	0	1	1	1	1									I, II	*	-	*	*	*	1
TASI	1	1	1	0	0	0	0	-									III	*	*	*	*	*	1
CAXI	0	1	0	0	1	1	0	0									I	*	*	*	*	*	1
SETF	0	1	0	0	0	1	1	1									I, II						1
<b>Privileged Instructions</b>																							
LDPR	0	0	0	1	0	0	1	0									I, II						2, 12
STPR	0	0	0	0	0	0	1	0									I, II						1, 2, 12
CLRTL	1	1	1	1	1	1	1	-									III						12
CLRTLBA	0	0	0	1	0	0	0	0									V						12
GETATE	0	0	0	0	0	1	0	1									I, II	-	-	-	*		1, 12
UPDATE	0	0	0	1	0	1	0	1									I, II	-	-	-	*		1, 12
GETPTE	0	0	0	0	0	1	0	0									I, II	*	-	-	*		1, 12
UPDPTE	0	0	0	1	0	1	0	0									I, II	*	-	-	*		12
GETRA	0	0	0	0	0	0	1	1									I, II	*	-	-	*		1, 12
IN	0	0	1	0	0	#iz	0									I, II						1, 12	
OUT	0	0	1	0	0	#iz	1									I, II						1, 12	
LDTASK	0	0	0	0	0	0	0	1									I, II						12
STTASK	1	1	1	1	1	1	0	-									III						12
RETIS	1	1	1	1	1	0	1	-									III	*	*	*	*	*	2, 12
UPDPSW.W	0	0	0	1	0	0	1	1									I, II	*	*	*	*	*	12
HALT	0	0	0	0	0	0	0	0									V						12

### Exceptions

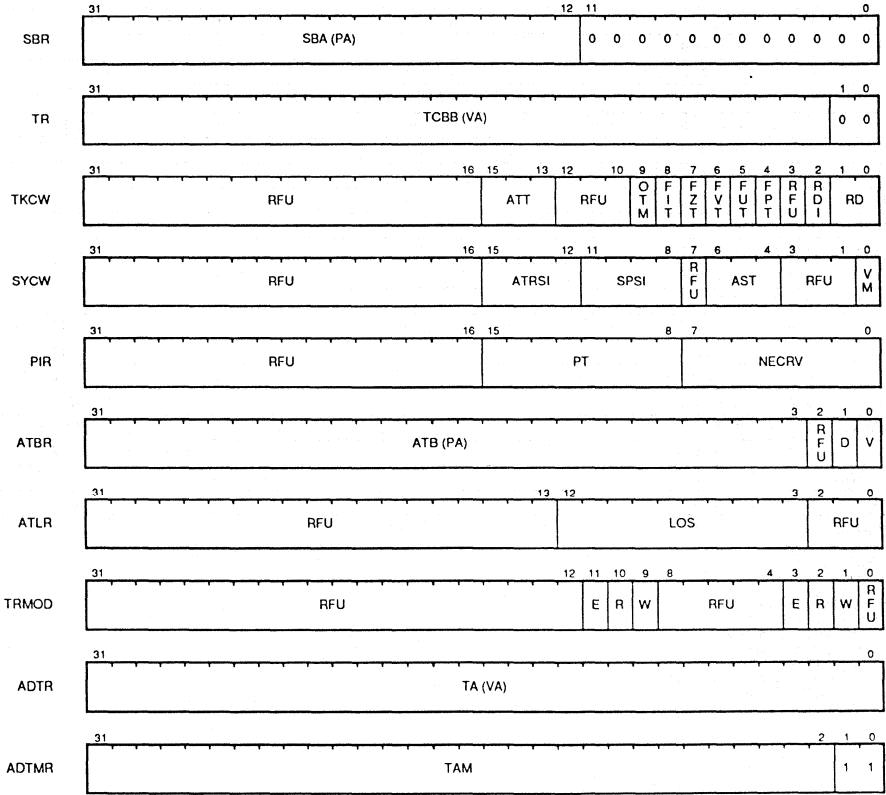
1. Illegal Addressing Mode
2. Illegal Data Type
3. Reserved Addressing Mode
4. Integer Zero Divide
5. Illegal Decimal Format
6. Floating Point Overflow
7. Floating Point Underflow
8. Floating Point Precision
9. Reserved Floating Point Operand
10. Invalid Floating Point Operation
11. Floating Point Zero Divide
12. Privileged Instruction

### Notes

1. Flags updated if PSW is specified in the register list

Appendix

Privileged Register Set



Note 1 (V) → virtual address  
(R) → real (physical) address

Note 2 Fields marked as RFU (Reserved for Future Use) must be 0







## NMOS MICROPROCESSORS

4

**Section 4 – NMOS Microprocessors**

$\mu$ PD780	High-Performance CP/M® - Compatible 8-Bit Microprocessor .....	4.3
$\mu$ PD8085A/AH	8-Bit, Single-Chip Microprocessors .....	4.27
$\mu$ PD8086	16-Bit Microprocessor .....	4.47
$\mu$ PD8088	High-Performance 8-Bit Microprocessor .....	4.59

CP/M is a registered trademark of Digital Research Inc.

### Description

The  $\mu$ PD780 is a microprocessor that utilizes a highly consistent architectural organization, a comprehensive instruction set that is a superset of the industry-standard 8080A instruction set, and third-generation technology, to provide a flexible, high-performance, efficient CPU easily adaptable to a very broad range of industrial and commercial applications.

All software developed on 8080A-based systems may be run on 780-based systems as a subset of the full 780 instruction set. In addition, the NEC  $\mu$ PD780 is fully pin-compatible and software-compatible with the Z80® microprocessor and is therefore perfectly suited for CP/M® designs. The NEC  $\mu$ PD780 provides system designers with powerful, wide-range logic capability that requires minimal additional circuitry to complete a microcomputer system.

The output signals of the  $\mu$ PD780 are fully decoded and signal timing is fully compatible with industry-standard memory and peripheral devices. Two faster versions of the basic  $\mu$ PD780 (2.5 MHz master clock rate) are offered by the  $\mu$ PD780-1 (4 MHz master clock rate) and the  $\mu$ PD780-2 (6 MHz master clock rate). Other than clock rates, all three versions are identical.

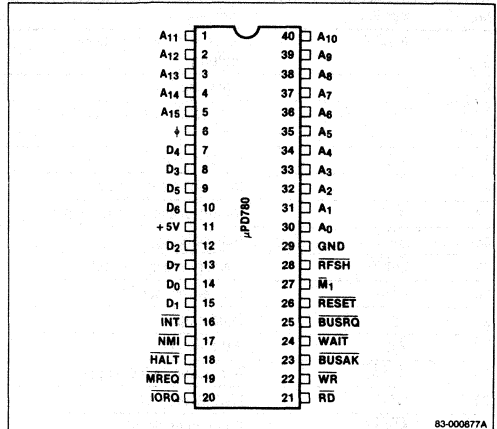
### Features

- Powerful, wide-range logic capability requiring minimal support circuitry
- Fully Z80®-compatible
- Industry-standard 8080A software compatibility
- CP/M®-compatible
- Comprehensive, powerful instruction set featuring 158 instruction types
- Vectored, multilevel interrupt structure
- Highly consistent architectural structure featuring dual register set
- Foreground/background programming
- Automatic refreshing of external dynamic memory
- Signal timing compatible with industry-standard memory and peripheral devices
- TTL-compatible signals
- Single-phase +5 V clock and +5 V DC power supply

\* Z80 is a registered trademark of Zilog, Inc.

\* CP/M is a registered trademark of Digital Research Corporation.

### Pin Configuration



83-000877A

### Ordering Information

Part Number	Package Type	Max Frequency of Operation
$\mu$ PD780C	40-pin plastic DIP	2.5 MHz
$\mu$ PD780C-1	40-pin plastic DIP	4 MHz
$\mu$ PD780C-2	40-pin plastic DIP	6 MHz

**Pin Identification**

No.	Symbol	Function
1-5, 30-40	A <sub>0</sub> -A <sub>15</sub>	Three-state address bus (output)
6	φ	Clock input
7-10, 12-15	D <sub>0</sub> -D <sub>7</sub>	Three-state, I/O data bus
11	+5 V	Power supply
16	$\overline{\text{INT}}$	Interrupt request input
17	$\overline{\text{NMI}}$	Non-maskable interrupt input
18	$\overline{\text{HALT}}$	Halt state input
19	$\overline{\text{MREQ}}$	Memory request output
20	$\overline{\text{IORQ}}$	I/O request output
21	$\overline{\text{RD}}$	Read output
22	$\overline{\text{WR}}$	Write output
23	$\overline{\text{BUSAK}}$	Bus acknowledge output
24	$\overline{\text{WAIT}}$	Wait state input
25	$\overline{\text{BUSRQ}}$	Bus request input
26	$\overline{\text{RESET}}$	Reset input
27	$\overline{\text{M}}_1$	Machine cycle 1
28	$\overline{\text{RFSH}}$	Refresh output
29	GND	Ground

**Pin Functions****A<sub>0</sub>-A<sub>15</sub> (Address Bus)**

16-bit, three-state output address bus. During refresh operations, lines A<sub>0</sub>-A<sub>6</sub> output the external memory address.

**D<sub>0</sub>-D<sub>7</sub> (Data Bus)**

8-bit, three-state I/O data bus.

 **$\overline{\text{NMI}}$  (Non-Maskable Interrupt)**

This active low input line is used for non-maskable interrupts. A non-maskable interrupt is always acknowledged at the end of the current instruction, regardless of whether the interrupt enable flip flop has been turned on, except when the  $\overline{\text{BUSRQ}}$  signal is asserted. Because of the higher priority of the  $\overline{\text{BUSRQ}}$  signal, it is acknowledged before the  $\overline{\text{NMI}}$  signal. When  $\overline{\text{NMI}}$  is acknowledged, program execution automatically restarts from location 0066H.

 **$\overline{\text{INT}}$  (Interrupt Request)**

This active low input line is used for interrupt requests by external I/O devices. Interrupts are serviced upon completion of the current instruction if the interrupt enable flip flop has been turned on by the software. There are three interrupt response modes: the mode 0 response is equivalent to an 8080 interrupt response; mode 1 uses location 0038H as a restart address; and mode 2 is a simple vectoring to an interrupt service routine that can be located anywhere in memory.

 **$\overline{\text{BUSRQ}}$  (Bus Request)**

This active low input signal is used to place the data bus, address bus, and all three-state bus control signals ( $\overline{\text{WR}}$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{IORQ}}$ , and  $\overline{\text{MREQ}}$ ) in a high-impedance state to allow a requesting device to assume bus control. The  $\overline{\text{BUSRQ}}$  signal has a higher priority than the  $\overline{\text{NMI}}$  signal and is always honored at the end of the current machine cycle.

Excessive DMA operations resulting in long periods in which  $\overline{\text{BUSRQ}}$  is asserted can impair the CPU's ability to adequately refresh the dynamic RAMs. Also,  $\overline{\text{BUSRQ}}$  does not have an internal pull-up resistor. For input signals to this pin in a wire-OR'ed configuration, an external pull-up resistor should be used.

 **$\overline{\text{BUSAK}}$  (Bus Acknowledge)**

This active low output line is used to inform the device requesting bus control that the data bus, address bus, and all three-state bus controls ( $\overline{\text{WR}}$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{IORQ}}$ , and  $\overline{\text{MREQ}}$ ) are in a high-impedance state and the requesting device can now assume control.

 **$\overline{\text{WR}}$  (Write)**

This three-state active low output is used to strobe data from the data bus to external memory or I/O devices.  $\overline{\text{WR}}$  is asserted to indicate the data bus holds valid data. This line is three-stated during halt or reset conditions.

 **$\overline{\text{IORQ}}$  (I/O Request)**

This three-state active low output is used to indicate the lower half of the address bus holds a valid address for an I/O read or write. During interrupt acknowledge cycles,  $\overline{\text{IORQ}}$  and  $\overline{\text{M}}_1$  are asserted together to indicate that a vector address can be sent to the data bus.

## **$\overline{RD}$ (Read)**

This three-state active low output is used to strobe data from external memory or I/O devices onto the data bus.  $\overline{RD}$  is asserted to indicate the CPU is requesting data from external memory or I/O devices. This line is three-stated during halt or reset conditions.

## **$\overline{MREQ}$ (Memory Request)**

This three-state active low output is used to indicate that the address specified for the memory read or write is valid.

## **$\overline{M_1}$ (Machine Cycle 1)**

This active low output is used to indicate that the current machine cycle is the opcode fetch phase of an instruction execution.

## **$\overline{HALT}$ (Halt State)**

This active low input is used with the  $\overline{HALT}$  instruction to initiate a halt state. When  $\overline{HALT}$  is asserted, program execution stops and does not resume until an interrupt is generated. During the halt state, NOPs are executed in order to continue memory refresh operations.

## **$\overline{WAIT}$ (Wait State)**

This active low input is used to indicate that the external memory or I/O devices addressed by the CPU are not ready to transfer data. When  $\overline{WAIT}$  is asserted, the CPU is placed in a wait condition.

## **$\overline{RESET}$**

This active low input signal is used to initialize the CPU. When  $\overline{RESET}$  is asserted, the interrupt enable flip flop is reset, the program counter and the I and R registers are cleared, and interrupt response mode 0 is enabled. In a reset condition, the address and data busses are three-stated and all output control signals are inactive, after which program execution begins from address 0000.

The pulse width of  $\overline{RESET}$  must be a minimum of 3 clock cycles in length to reinitialize the CPU and stabilize operation.

## **$\overline{RFSH}$ (Refresh)**

This active low output is used in conjunction with the  $\overline{MREQ}$  signal to initiate a refresh read of all external dynamic memory.  $\overline{RFSH}$  and  $\overline{MREQ}$  are both asserted when the least significant 7 bits of the address on the address bus hold a valid external dynamic memory address.

## **$\phi$ (Clock)**

This line is an input for external clock sources.

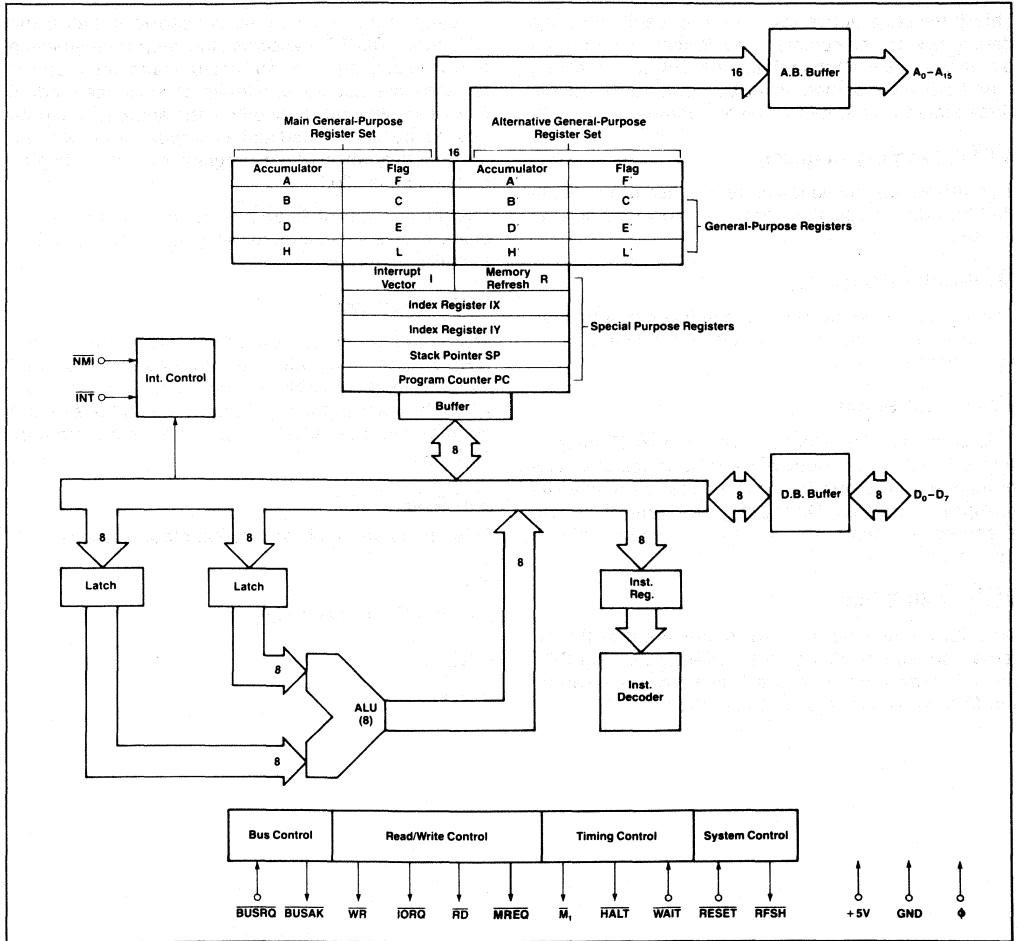
## **+5 V**

Single +5 V power supply.

## **GND**

Ground.

Block Diagram



## Architecture

The architecture includes a dual set of six 8-bit general-purpose registers and two 8-bit accumulators and flag registers. A flexible vectored interrupt structure is supported by an 8-bit interrupt vector register that provides the most-significant 8 bits of a pointer to a table of vector addresses, while the requesting device generates the least-significant 8 bits of the pointer. Two 16-bit index registers enable the manipulation of tabular data as well as facilitating code relocation.

Multilevel interrupts as well as virtually unlimited sub-routine nesting are supported by a 16-bit stack pointer and complimentary 16-bit program counter, enhancing the speed and efficiency of a wide variety of data-handling operations. Processing efficiency is additionally supported by a special memory refresh register that enables automatic refreshing of all external dynamic memory with minimal processor overhead.

The dual set of general-purpose registers may be used as individual 8-bit registers or paired as 16-bit registers. The dual register set (including a dual accumulator and flag register) not only allows more powerful addressing and data transfer operations, but also permits programming in foreground/background mode for vastly improved throughput.

## Standard Test Conditions

The standard test conditions reference all voltages to ground (0 V) and follow the convention that positive current flows into the referenced pin. The listing of AC parameters is based on a load capacitance of 50 pF unless explicitly stated otherwise. For every 50 pF increase in load capacitance there is a 10 ns delay, up to a maximum increase of 200 pF for the data bus and 100 pF for the address bus and the bus control lines.

The operating temperature range is: 0°C to +70°C; +4.75 V ≤ V<sub>CC</sub> ≤ +5.25 V.

## Absolute Maximum Ratings

T<sub>A</sub> = 25°C

Operating temperature	0°C to +70°C
Storage temperature	-65°C to +150°C
Voltage on any pin	-0.3 to +7 V (1)
Power dissipation	1.5 W

### Note:

(1) With respect to ground.

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## Capacitance

T<sub>A</sub> = 25°C

Parameter	Symbol	Limits			Test Conditions
		Min	Max	Unit	
Clock capacitance	C <sub>φ</sub>		35	pF	f <sub>c</sub> = 1 MHz
Input capacitance	C <sub>IN</sub>		5	pF	Unmeasured pins returned to ground.
Output capacitance	C <sub>OUT</sub>		10	pF	

## DC Characteristics

T<sub>A</sub> = 0°C to +70°C; V<sub>CC</sub> = +5 V ± 5% unless otherwise specified

Parameter	Symbol	Limits			Test Conditions
		Min	Typ	Max	
Clock input low voltage	V <sub>ILC</sub>	-0.3		0.45	V
Clock input high voltage	V <sub>IHC</sub>	V <sub>CC</sub> - 0.6		V <sub>CC</sub> + 0.3	V
Input low voltage	V <sub>IL</sub>	-0.3		0.8	V
Input high voltage	V <sub>IH</sub>	2.0		V <sub>CC</sub>	V
Output low voltage	V <sub>OL</sub>		0.4		V I <sub>OL</sub> = 1.8 mA
Output high voltage	V <sub>OH</sub>	2.4			V I <sub>OH</sub> = -250 μA
Power supply current μPD780	I <sub>CC</sub>			150	mA t <sub>C</sub> = 400 ns
Current μPD780-1	I <sub>CC</sub>		90	200	mA t <sub>C</sub> = 250 ns
Input leakage current	I <sub>LI</sub>			10	μA V <sub>IN</sub> = 0 to V <sub>CC</sub>
Three-state output leakage current in float	I <sub>LOH</sub>			10	μA V <sub>OUT</sub> = 2.4 to V <sub>CC</sub>
Three-state output leakage current in float	I <sub>LOL</sub>			-10	μA V <sub>OUT</sub> = 0.4 V
Data bus leakage current in input mode	I <sub>LD</sub>			±10	μA 0 ≤ V <sub>IN</sub> ≤ V <sub>CC</sub>



**AC Characteristics**

T<sub>A</sub> = 0°C to +70°C; V<sub>CC</sub> = +5 V ± 5%; unless otherwise specified

Parameter	Symbol	Limits						Unit	Test Conditions
		μPD780 (2.5 MHz)		μPD780-1(4 MHz)		μPD780-2 (6 MHz)			
		Min	Max	Min	Max	Min	Max		
Clock period	t <sub>C</sub>	0.4	(1)	0.25	(1)	0.165	(1)	μs	
Clock pulse width, clock high	t <sub>W(φH)</sub>	180	(2)	110	(2)	65	(2)	ns	
Clock pulse width, clock low	t <sub>W(φL)</sub>	180	2000	110	2000	72	2000	ns	
Clock rise and fall time	t <sub>rf</sub>		30		30		20	ns	
Address output delay	t <sub>D(AD)</sub>		145		110		90	ns	
Delay to float	t <sub>F(AD)</sub>		110		90		80	ns	
Address stable prior to $\overline{MREQ}$ (Memory cycle)	t <sub>ACM</sub>	(3)		(3)		(3)		ns	C <sub>L</sub> = 50 pF
Address stable prior to $\overline{IORQ}$ , $\overline{RD}$ or $\overline{WR}$ (I/O cycle)	t <sub>ACI</sub>	(4)		(4)		(4)		ns	
Address stable from $\overline{RD}$ or $\overline{WR}$	t <sub>CA</sub>	(5)		(5)		(5)		ns	
Address stable from $\overline{RD}$ or $\overline{WR}$ during Float	t <sub>CAF</sub>	(6)		(6)		(6)		ns	
Data output delay	t <sub>D(D)</sub>		230		150		130	ns	
Delay to float during write cycle	t <sub>F(D)</sub>		90		90		80	ns	
Data setup time to rising edge of clock during M <sub>1</sub> cycle	t <sub>Sφ(D)</sub>	50		35		30		ns	
Data setup time to falling edge of clock during M <sub>2</sub> to M <sub>5</sub> cycles	t <sub>Sφ(D)</sub>	60		50		40		ns	C <sub>L</sub> = 200 pF
Data stable prior to $\overline{WR}$ (Memory cycle)	t <sub>DCM</sub>	(7)		(7)		(7)		ns	
Data stable prior to $\overline{WR}$ (I/O cycle)	t <sub>DCI</sub>	(8)		(8)		(8)		ns	
Data stable from $\overline{WR}$	t <sub>CDF</sub>	(9)		(9)		(9)		ns	
$\overline{BUSRQ}$ setup time to rising edge of clock	t <sub>S(BQ)</sub>	80		50		50		ns	
$\overline{BUSAK}$ delay from rising edge of clock to $\overline{BUSAK}$ low	t <sub>DL(BA)</sub>		120		100		90	ns	
$\overline{BUSAK}$ delay from falling edge of clock to $\overline{BUSAK}$ high	t <sub>DH(BA)</sub>		110		100		90	ns	C <sub>L</sub> = 50 pF
Delay to float ( $\overline{MREQ}$ , $\overline{IORQ}$ , $\overline{RD}$ and $\overline{WR}$ )	t <sub>F(C)</sub>		100		80		70	ns	
$\overline{M}_1$ stable prior to $\overline{IORQ}$ (Interrupt ack.)	t <sub>MR</sub>	(10)		(10)		(10)		ns	
Any hold time for setup time	t <sub>H</sub>	0		0		0		ns	
$\overline{HALT}$ delay time from falling edge of clock	t <sub>D(HT)</sub>		300		300		260	ns	C <sub>L</sub> = 50 pF
$\overline{INT}$ setup time to rising edge of clock	t <sub>S(IT)</sub>	80		80		70		ns	
$\overline{IORQ}$ delay from rising edge of clock to $\overline{IORQ}$ low	t <sub>DLφ(IR)</sub>		90		75		65	ns	
$\overline{IORQ}$ delay from falling edge of clock to $\overline{IORQ}$ low	t <sub>DLφ(IR)</sub>		110		85		70	ns	
$\overline{IORQ}$ delay from rising edge of clock to $\overline{IORQ}$ high	t <sub>DHφ(IR)</sub>		100		85		70	ns	
$\overline{IORQ}$ delay from falling edge of clock to $\overline{IORQ}$ high	t <sub>DHφ(IR)</sub>		110		85		70	ns	C <sub>L</sub> = 50 pF
$\overline{M}_1$ delay from rising edge of clock to $\overline{M}_1$ low	t <sub>DL(M<sub>1</sub>)</sub>		130		100		80	ns	
$\overline{M}_1$ delay from rising edge of clock to $\overline{M}_1$ high	t <sub>DH(M<sub>1</sub>)</sub>		130		100		80	ns	
$\overline{MREQ}$ delay from falling edge of clock to $\overline{MREQ}$ low	t <sub>DLφ(MR)</sub>		100		85		70	ns	
$\overline{MREQ}$ delay from rising edge of clock to $\overline{MREQ}$ high	t <sub>DHφ(MR)</sub>		100		85		70	ns	
$\overline{MREQ}$ delay from falling edge of clock to $\overline{MREQ}$ high	t <sub>DHφ(MR)</sub>		100		85		70	ns	
Pulse width, $\overline{MREQ}$ low	t <sub>W(MRL)</sub>	(11)		(11)		(11)		ns	
Pulse width, $\overline{MREQ}$ high	t <sub>W(MRH)</sub>	(12)		(12)		(12)		ns	
Pulse width, $\overline{NMI}$ low	t <sub>W(NML)</sub>	80		80		70		ns	
$\overline{RESET}$ setup time to rising edge of clock	t <sub>S(RS)</sub>	90		60		60		ns	
$\overline{RD}$ delay from rising edge of clock to $\overline{RD}$ low	t <sub>DLφ(RD)</sub>		100		85		70	ns	
$\overline{RD}$ delay from falling edge of clock to $\overline{RD}$ low	t <sub>DLφ(RD)</sub>		130		95		80	ns	
$\overline{RD}$ delay from rising edge of clock to $\overline{RD}$ high	t <sub>DHφ(RD)</sub>		100		85		70	ns	
$\overline{RD}$ delay from falling edge of clock to $\overline{RD}$ high	t <sub>DHφ(RD)</sub>		110		85		70	ns	C <sub>L</sub> = 30 pF

## AC Characteristics (cont)

$T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ;  $V_{CC} = +5\text{ V} \pm 5\%$ ; unless otherwise specified

Parameter	Symbol	Limits						Unit	Test Conditions
		μPD780 (2.5 MHz)		μPD780-1(4 MHz)		μPD780-2 (6 MHz)			
		Min	Max	Min	Max	Min	Max		
RFSH delay from rising edge of clock to RFSH low	$t_{DL}(\text{RF})$		180		130		110	ns	$C_L = 30\text{ pF}$
RFSH delay from rising edge of clock to RFSH high	$t_{DH}(\text{RF})$		150		120		100	ns	
WAIT setup time to falling edge of clock	$t_S(\text{WT})$	70		70		60		ns	
WR delay from rising edge of clock to WR low	$t_{DL}(\text{WR})$		80		65		60	ns	
WR delay from falling edge of clock WR low	$t_{DL}(\text{WR})$		90		80		70	ns	
WR delay from falling edge of clock to WR high	$t_{DH}(\text{WR})$		100		80		70	ns	
Pulse width to WR low	$t_W(\text{WRL})$	(13)		(13)		(13)		ns	

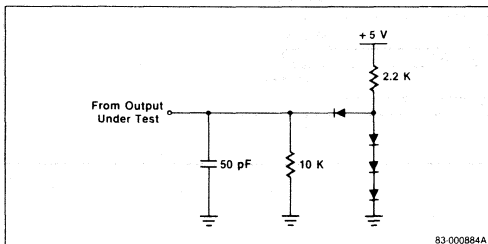
### Notes:

- (1)  $t_C = t_W(\uparrow\text{H}) + t_W(\uparrow\text{L}) + t_R + t_F$
- (2) Though the structure of the 780 is static, 200μs is guaranteed maximum.
- (3)  $t_{ACM} = t_W(\uparrow\text{H}) + t_F - 65\text{ (75)* (50)**}$
- (4)  $t_{ACI} = t_C - 70\text{ (80)* (55)**}$
- (5)  $t_{CA} = t_W(\uparrow\text{L}) + t_R - 50\text{ (40)* (50)**}$
- (6)  $t_{CAF} = t_W(\uparrow\text{L}) + t_R - 45\text{ (60)* (40)**}$
- (7)  $t_{DCM} = t_C - 170\text{ (210)* (140)**}$
- (8)  $t_{DCI} = t_W(\uparrow\text{L}) + t_R - 170\text{ (210)* (140)**}$
- (9)  $t_{CDF} = t_W(\uparrow\text{L}) + t_R - 70\text{ (80)* (55)**}$
- (10)  $t_{MR} = 2t_C + t_W(\uparrow\text{H}) + t_F - 65\text{ (80)* (50)**}$
- (11)  $t_W(\text{MRL}) = t_C - 30\text{ (40)* (30)**}$
- (12)  $t_W(\text{MRH}) = t_W(\uparrow\text{H}) + t_F - 20\text{ (30)* (20)**}$
- (13)  $t_W(\text{WR}) = t_C - 30\text{ (40)* (30)**}$

\* These values apply to the μPD780.

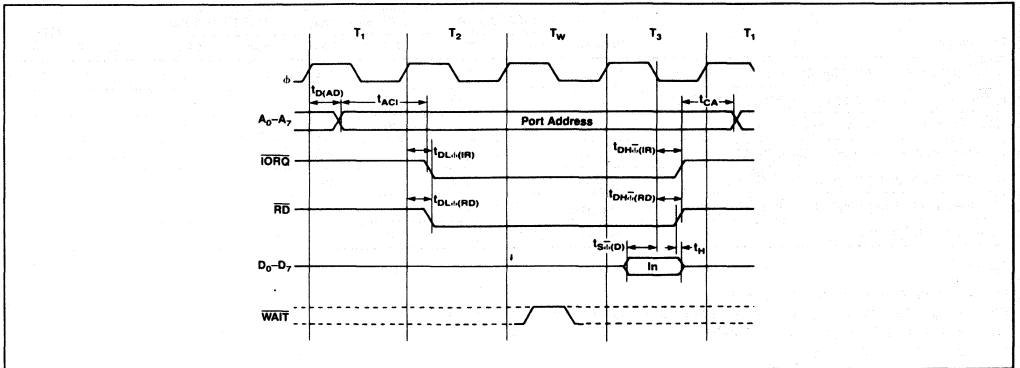
\*\* These values apply to the μPD780-2.

### Load Circuit for Output

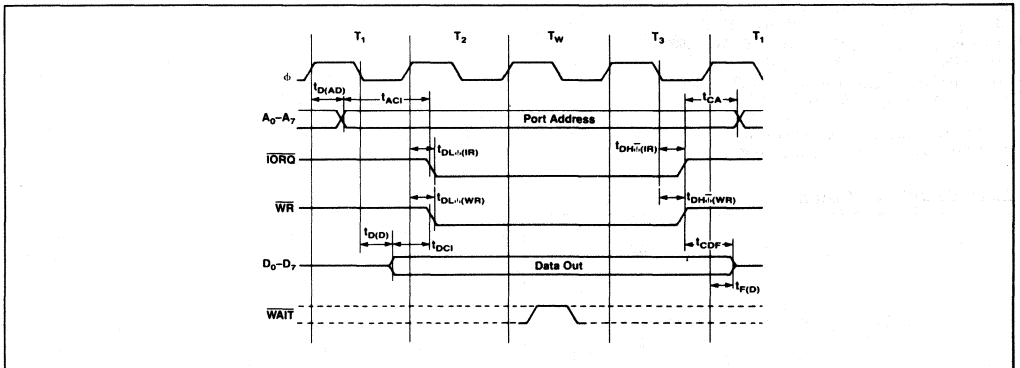


Timing Waveforms

Input Cycle

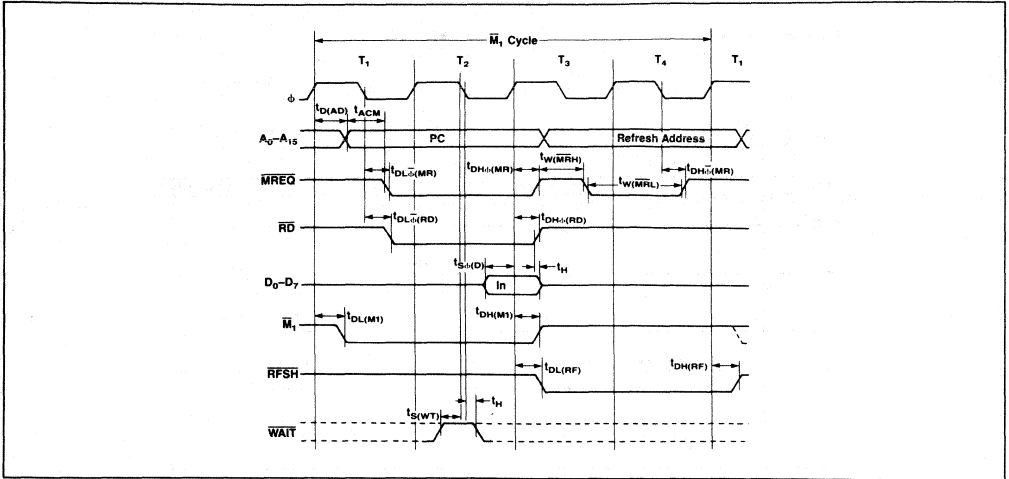


Output Cycle



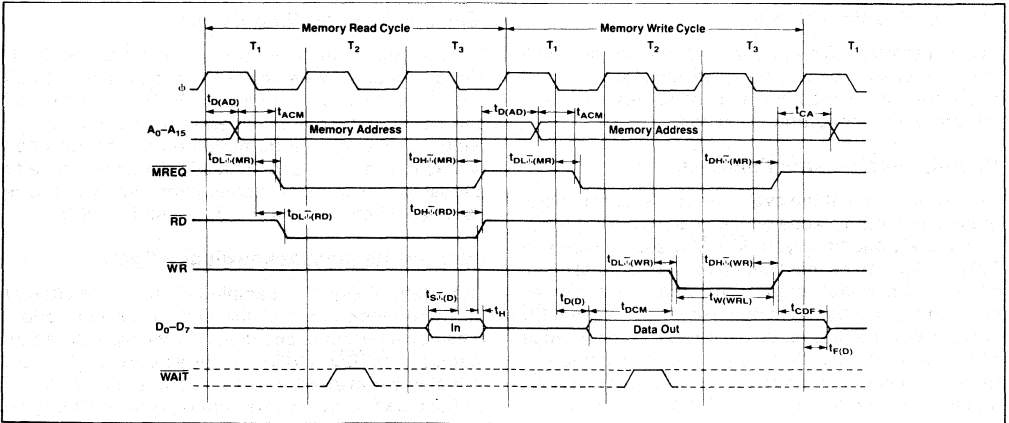
## Timing Waveforms (cont)

### M<sub>1</sub> Cycle



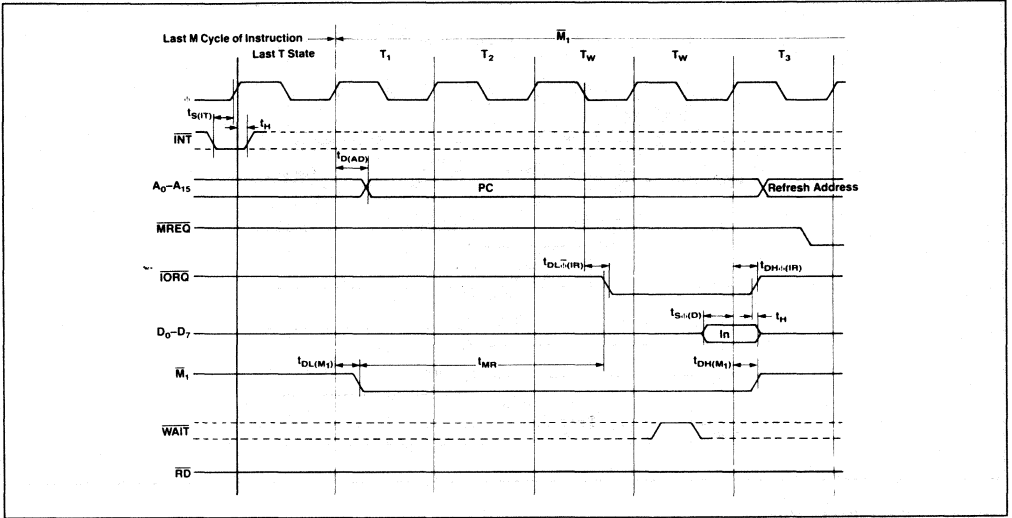
4

### Memory Read/Write Cycles



Timing Waveforms (cont)

Interrupt Request/Acknowledge Cycle



Input and Output Cycles

In I/O operations, a single wait state ( $T_W$ ) is automatically included to provide adequate time for an I/O port to decode the address from the port address lines and initiate a wait condition if needed.

Opcode Fetch Instruction Cycle

At the beginning of the cycle, the contents of the program counter are placed on the address bus. After approximately one-half cycle,  $\overline{MREQ}$  is asserted and its falling edge can be used directly by the external memory as a chip enable signal. The data from the external memory can be gated onto the data bus when  $\overline{RD}$  is asserted. The CPU reads the data at the rising edge of  $T_3$ . During  $T_3$  and  $T_4$ , external dynamic memory is refreshed while the instruction is decoded and executed. The assertion of  $\overline{RFSH}$  indicates that the external dynamic memory requires a refresh read.

Memory Read or Write Cycles

In read and write operations, the  $\overline{MREQ}$  and  $\overline{RD}$  signals function the same as they do in opcode fetch operations. In a write operation  $\overline{MREQ}$  is asserted and can be used directly by external memory as a chip enable signal when information on the address bus is stable. The  $\overline{WR}$  signal is used as a write strobe to almost any type of semiconductor memory, and is asserted when data on the data bus is stable.

Interrupt Request/Acknowledge Cycle

The interrupt signal is sampled at the rising edge of the final clock pulse at the end of an instruction. When an interrupt is accepted, an  $M_1$  cycle is begun. Instead of  $\overline{MREQ}$ ,  $\overline{IORQ}$  is asserted during this cycle to indicate that an 8-bit vector address can be placed on the data bus by the interrupting device. This cycle includes the automatic addition of two wait states to facilitate the implementation of a daisy-chain priority interrupt protocol.

## Instruction Set

The instruction set of the μPD780 consists of 158 types of instructions divided into 16 categories as follows:

8-bit load operations	8-bit arithmetic and logic operations
register exchanges	bit set, reset, and test operations
memory block searches	I/O operations
16-bit arithmetic operations	call operations
rotate and shift operations	return operations
jump operations	general-purpose accumulator and flag operations
restart operations	
miscellaneous operations	
16-bit load operations	
memory block transfers	

This comprehensive instruction set is made more powerful by the array of addressing modes implemented by the architecture, as follows:

bit addressing	relative addressing
register-indirect addressing	immediate-extended addressing
immediate addressing	indexed addressing
extended addressing	modified page zero addressing
implied addressing	
register addressing	

## Instruction Set Symbol Definitions

Symbol	Description
•	Flag not affected
0	Flag set
X	Flag
‡	Flag affected according to result of operation
V	Overflow set
P	Parity set
IFF	Interrupt flip-flop set
C	Carry/Link
Z	Zero
P/V	Parity/Overflow
S	Sign
N	Add/Subtract
H	Half Carry

### Instruction Set

Mnemonic	Operation	Description	Operation Code							No. of Clocks	No. of Bytes	Flags								
			7	6	5	4	3	2	1			0	C	Z	P/V	S	N	H		
ADC HL, ss	HL ← HL + ss + CY	Add with carry reg. pair ss to HL	1	1	1	0	1	0	1	(A)	15	1	†	†	†	V	†	0	X	
ADC A, r	A ← A + r + CY	Add with carry Reg. r to ACC	1	0	0	0	1	r	r	(B)	4	1	†	†	†	V	†	0	†	
ADC A, n	A ← A + n + CY	Add with carry value n to ACC	1	1	0	0	1	1	0	n	7	2	†	†	†	V	†	0	†	
ADC A, (HL)	A ← A + (HL) + CY	Add with carry loc. (HL) to ACC	1	0	0	0	1	1	0	HL	7	1	†	†	†	V	†	0	†	
ADC A, (IX + d)	A ← A + (IX + d) + CY	Add with carry loc. (IX + d) to ACC	1	1	0	1	1	0	1	IX + d	19	3	†	†	†	V	†	0	†	
ADC A, (IY + d)	A ← A + (IY + d) + CY	Add with carry loc. (IY + d) to ACC	1	0	0	1	1	0	IY + d	19	3	†	†	†	V	†	0	†		
ADD A, n	A ← A + n	Add value n to ACC	1	1	0	0	0	1	1	0	n	7	2	†	†	†	V	†	0	†
ADD A, r	A ← A + r	Add Reg. r to ACC	1	0	0	0	0	r	r	(B)	4	1	†	†	†	V	†	0	†	
ADD A, (HL)	A ← A + (HL)	Add location (HL) to ACC	1	0	0	0	0	1	1	0	HL	7	1	†	†	†	V	†	0	†
ADD A, (IX + d)	A ← A + (IX + d)	Add location (IX + d) to ACC	1	1	0	1	1	0	1	IX + d	19	3	†	†	†	V	†	0	†	
ADD A, (IY + d)	A ← A + (IY + d)	Add location (IY + d) to ACC	1	1	1	1	1	0	1	IY + d	19	3	†	†	†	V	†	0	†	
ADD HL, ss	HL ← HL + ss	Add Reg. pair ss to HL	0	0	s	1	0	0	1	(A)	11	1	†	•	•	•	•	0	X	
ADD IX, pp	IX ← IX + pp	Add Reg. pair pp to IX	1	1	0	1	1	0	1	(C)	15	2	†	•	•	•	•	0	X	
ADD IY, rr	IY ← IY + rr	Add Reg. pair rr to IY	1	1	1	1	0	1	0	(D)	15	2	†	•	•	•	•	0	X	
AND r	A ← A ∧ r	Logical 'AND' of Reg. r ∧ ACC	1	0	1	0	0	r	r	(B)	4	1	0	†	P	†	0	†	†	
AND n	A ← A ∧ n	Logical 'AND' of value n ∧ ACC	1	1	0	0	1	1	0	n	7	2	0	†	P	†	0	†	†	
AND (HL)	A ← A ∧ (HL)	Logical 'AND' of loc. (HL) ∧ ACC	1	0	1	0	0	1	1	0	HL	7	1	0	†	P	†	0	†	†
AND (IX + d)	A ← A ∧ (IX + d)	Logical 'AND' of loc. (IX + d) ∧ ACC	1	1	0	1	1	0	1	IX + d	19	3	0	†	P	†	0	†	†	
AND (IY + d)	A ← A ∧ (IY + d)	Logical 'AND' of loc. (IY + d) ∧ ACC	1	1	1	1	1	0	1	IY + d	19	3	0	†	P	†	0	†	†	





**Instruction Set (cont)**

Mnemonic	Operation	Description	Operation Code	No. of ClocKs	No. of Bytes	Flags
			7 6 5 4 3 2 1 0			C Z P/V S N H
BIT b, (HL)	$Z \leftarrow (\overline{HL})_b$	Test BIT b of location (HL)	1 1 0 0 1 0 1 1 (E) 0 1 b b b 1 1 0	12	2	• • • • X X 0 1
BIT b, (IX + d)	$Z \leftarrow (\overline{IX + d})_b$	Test BIT b at location (IX + d)	1 1 0 0 1 1 0 1 (E) 1 1 0 0 1 0 1 1 d d d d d d d d 0 1 b b b 1 1 0	20	4	• • • • X X 0 1
BIT b, (IY + d)	$Z \leftarrow (\overline{IY + d})_b$	Test BIT b at location (IY + d)	1 1 1 1 1 1 0 1 (E) 1 1 0 0 1 0 1 1 d d d d d d d d 0 1 b b b 1 1 0	20	4	• • • • X X 0 1
BIT b, r	$Z \leftarrow \overline{r}_b$	Test BIT of Reg. r	1 1 0 0 1 0 1 1 0 1 b b b r r (B) (E)	8	2	• • • • X X 0 1
CALL cc, nn	If condition cc false continues, else same as CALL nn	Call subroutine at location nn if condition cc is true	1 1 ← cc → 1 0 0 (H) n n n n n n n n n n n n n n n n	10	3	• • • • • • • •
CALL nn	$(SP - 1) \leftarrow PCH$ $(SP - 2) \leftarrow PCL$ $PC \leftarrow nn$	Unconditional call subroutine at location nn	1 1 0 0 1 1 0 1 n n n n n n n n n n n n n n n n	17	3	• • • • • • • •
CCF	$CY \leftarrow \overline{CY}$	Complement carry flag	0 0 1 1 1 1 1 1	4	1	• • • • • • 0 X
CP r	$A - r$	Compare Reg. r with ACC	1 0 1 1 1 r r r (B)	4	1	• • • • V • • •
CP n	$A - n$	Compare value n with ACC	1 1 1 1 1 1 1 0 n n n n n n n n	7	2	• • • • V • • •
CP (HL)	$A - (HL)$	Compare loc. (HL) with ACC	1 0 1 1 1 1 1 0	7	1	• • • • V • • •
CP (IX + d)	$A - (IX + d)$	Compare loc. (IX + d) with ACC	1 1 0 1 1 1 0 1 1 0 1 1 1 1 1 0 d d d d d d d d 1 1 1 1 1 1 1 0	19	4	• • • • V • • •
CP (IY + d)	$A - (IY + d)$	Compare loc. (IY + d) with ACC	1 0 1 1 1 1 1 0 d d d d d d d d	19	2	• • • • V • • •
CPD	$A - (HL)$ $HL \leftarrow HL - 1$ $BC \leftarrow BC - 1$	Compare location (HL) and ACC, decrement HL and BC	1 1 1 0 1 1 0 1 1 0 1 0 1 0 0 1	16	2	• • • • (2) • (1) • • •
CPDR	$A - (HL)$ $HL \leftarrow HL - 1$ $BC \leftarrow BC - 1$ until A = (HL) or BC = 0	Compare location (HL) and ACC, decrement HL and BC, repeat until BC = 0	1 1 1 0 1 1 0 1 1 0 1 1 0 1 0 1	21 if BC = 0 and A ≠ (HL) 16 if BC = 0 or A = (HL)	2	• • • • (2) • (1) • • •
CPI	$A - (HL)$ $HL \leftarrow HL + 1, BC \leftarrow BC - 1$	Compare location (HL) and ACC, increment HL and decrement BC	1 1 1 0 1 1 0 1 1 0 1 0 0 0 0 1	16	2	• • • • (2) • (1) • • •

## Instruction Set (cont)

Mnemonic	Operation	Description	Operation Code							No. of Clocks	No. of Bytes	Flags						
			7	6	5	4	3	2	1			0	C	Z	P/V	S	M	H
IM 0		Set interrupt mode 0	1	1	1	0	1	1	0	1	8	2	•	•	•	•	•	•
			0	1	0	0	0	1	1	0								
IM 1		Set interrupt mode 1	1	1	1	0	1	1	0	1	8	2	•	•	•	•	•	•
			0	1	0	1	0	1	1	0								
IM 2		Set interrupt mode 2	1	1	1	0	1	1	0	1	8	2	•	•	•	•	•	•
			0	1	0	1	1	1	1	0								
IN A, (n)	A ← (n)	Load ACC with input from device n	1	1	0	1	0	1	1	1	11	2	•	•	•	•	•	•
			n	n	n	n	n	n	n	n								
IN r, (C)	r ← (C)	Load Reg. r with input from device (C)	1	1	1	0	1	0	1	(1)	12	2	•	•	•	•	•	•
			0	1	r	r	0	0	0	0								
INC (HL)	(HL) ← (HL) + 1	Increment location (HL)	0	0	1	1	0	1	0	0	11	1	•	•	•	•	V	↑
			0	0	1	0	0	0	0	1								
INC IX	IX ← IX + 1	Increment IX	1	1	0	1	1	0	1	1	10	2	•	•	•	•	•	•
			0	0	1	0	0	0	1	1								
INC (IX + d)	(IX + d) ← (IX + d) + 1	Increment location (IX + d)	1	1	0	1	1	0	1	1	23	3	•	•	•	•	V	↑
			0	0	1	1	0	1	0	0								
			d	d	d	d	d	d	d	d								
INC IY	IY ← IY + 1	Increment IY	1	1	1	1	0	1	1	1	10	2	•	•	•	•	•	•
			0	0	1	0	0	0	1	1								
INC (IY + d)	(IY + d) ← (IY + d) + 1	Increment location (IY + d)	1	1	1	1	0	1	1	1	23	3	•	•	•	•	V	↑
			0	0	1	1	0	1	0	0								
			d	d	d	d	d	d	d	d								
INC r	r ← r + 1	Increment Reg. r	0	0	r	r	1	0	0	(B)	4	1	•	•	•	•	V	↑
			0	0	1	0	0	1	1	(A)								
INC ss	ss ← ss + 1	Increment Reg. pair ss	0	0	s	s	0	0	1	1	6	1	•	•	•	•	•	•
			0	0	1	0	1	0	1	0								
IND	(HL) ← (C) B ← B' - 1, HL ← HL - 1	Load location (HL) with input from port (C), decrement HL and B	1	1	1	0	1	0	1	1	16	2	•	•	†(3)	X	X	1
			1	0	1	0	1	0	1	0								
INDR	(HL) ← (C) B ← B - 1 HL ← HL - 1 until B = 0	Load location (HL) with input from port (C), decrement HL and decrement B, repeat until B = 0	1	1	1	0	1	0	1	0	21	2	•	•	•	•	1	X
			1	0	1	1	0	1	0	0								
INI	(HL) ← (C) B ← B - 1, HL ← HL + 1	Load location (HL) with input from port (C), and increment HL and decrement B	1	1	0	1	0	1	0	1	16	2	•	•	†(3)	X	X	1
			1	0	1	0	0	1	0	0								
INIR	(HL) ← (C) B ← B - 1 HL ← HL + 1 until B = 0	Load location (HL) with input from port (C), increment HL and decrement B, repeat until B = 0	1	1	1	0	1	0	1	0	21	2	•	•	•	•	1	X
			1	0	1	0	0	1	0	0								
JP (HL)	PC ← HL	Unconditional jump to (HL)	1	1	1	0	1	0	0	1	4	1	•	•	•	•	•	•
			1	1	0	1	1	0	1	0								
JP (IX)	PC ← IX	Unconditional jump to (IX)	1	1	0	1	1	0	1	0	8	2	•	•	•	•	•	•
			1	1	0	1	0	0	1	0								
JP (IY)	PC ← IY	Unconditional jump to (IY)	1	1	1	1	0	1	0	1	8	2	•	•	•	•	•	•
			1	1	1	0	1	0	0	1								



## Instruction Set (cont)

Mnemonic	Operation	Description	Operation Code										No. of Clocks	No. of Bytes	Flags							
			7	6	5	4	3	2	1	0	C	Z			P/V	S	N	H				
LD IX, nn	IX ← nn	Load IX with value nn	1	1	0	1	1	1	0	0	1	1	19	4	.	.	.	.	.	.	.	.
LD IX, (nn)	IX <sub>H</sub> ← (nn + 1) IX <sub>L</sub> ← (nn)	Load IX with location (nn)	1	1	0	1	1	1	0	1	0	1	20	4	.	.	.	.	.	.	.	.
LD (IX + d), n	(IX + d) ← n	Load location (IX + d) with value n	1	1	0	1	1	1	0	1	1	19	4	.	.	.	.	.	.	.	.	.
LD (IX + d), r	(IX + d) ← r	Load location (IX + d) with Reg. r	0	0	1	1	0	1	1	0	0	19	3	.	.	.	.	.	.	.	.	.
LD IY, nn	IY ← nn	Load IY with value nn	1	1	1	1	1	0	1	0	1	14	4	.	.	.	.	.	.	.	.	.
LD IY, (nn)	IY <sub>H</sub> ← (nn + 1) IY <sub>L</sub> ← (nn)	Load IY with location (nn)	1	1	1	1	1	0	1	0	1	20	4	.	.	.	.	.	.	.	.	.
LD ss, (nn)	SS <sub>H</sub> ← (nn + 1) SS <sub>L</sub> ← (nn)	Load Reg. pair dd with location (nn)	1	1	1	0	1	1	0	1	0	20	4	.	.	.	.	.	.	.	.	.
LD (IY + d), n	(IY + d) ← n	Load (IY + d) with value n	1	1	1	1	1	0	1	0	1	19	4	.	.	.	.	.	.	.	.	.
LD (IY + d), r	(IY + d) ← r	Load location (IY + d) with Reg. r	0	1	1	1	0	1	0	1	0	19	3	.	.	.	.	.	.	.	.	.
LD (nn), A	(nn) ← A	Load location (nn) with ACC	0	0	1	1	0	0	1	0	0	13	3	.	.	.	.	.	.	.	.	.
LD (nn), ss	(nn + 1) ← SS <sub>H</sub> (nn) ← SS <sub>L</sub>	Load location (nn) with Reg. pair dd	1	1	1	0	1	0	1	0	1	20	4	.	.	.	.	.	.	.	.	.

**Instruction Set (cont)**

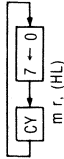
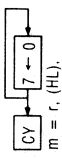
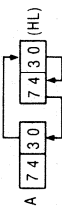
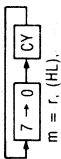
Mnemonic	Operation	Description	Operation Code										No. of Clocks	No. of Bytes	Flags						
			7	6	5	4	3	2	1	0	Z	P/V			S	N	H				
LD (nn), HL	(nn + 1) ← H (nn) ← L	Load location (nn) with HL	0	0	1	0	0	0	1	0	0	16	3	.	.	.	.	.	.	.	.
LD (nn), IX	(nn + 1) ← IX <sub>H</sub> (nn) ← IX <sub>L</sub>	Load location (nn) with IX	1	1	0	1	1	1	0	1	0	20	4	.	.	.	.	.	.	.	.
LD(nn), IY	(nn + 1) ← IY <sub>H</sub> (nn) ← IY <sub>L</sub>	Load location (nn) with IY	1	1	1	1	1	1	0	1	0	20	4	.	.	.	.	.	.	.	.
LD R, A	R ← A	Load R with ACC	1	1	1	0	1	1	0	1	1	9	2	.	.	.	.	.	.	.	.
LD r, (HL)	r ← (HL)	Load Reg. r with location (HL)	0	1	r	r	r	r	1	0	(B)	7	1	.	.	.	.	.	.	.	.
LD r, (IX + d)	r ← (IX + d)	Load Reg. r with location (IX + d)	1	1	0	1	1	1	0	1	(B)	19	3	.	.	.	.	.	.	.	.
LD r, (IY + d)	r ← (IY + d)	Load Reg. r with location (IY + d)	0	1	r	r	r	1	1	0	d	19	3	.	.	.	.	.	.	.	.
LD r, n	r ← n	Load Reg. r with value n	0	0	r	r	r	1	1	0	(B)	7	2	.	.	.	.	.	.	.	.
LD, r'	r ← r'	Load Reg. r with Reg. r'	0	1	r	r	r	r'	r'	r'	(F)	4	1	.	.	.	.	.	.	.	.
LD SP, HL	SP ← HL	Load SP with HL	1	1	1	1	1	0	0	1		6	1	.	.	.	.	.	.	.	.
LD SP, IX	SP ← IX	Load SP with IX	1	1	0	1	1	0	1	1		10	2	.	.	.	.	.	.	.	.
LD SP, IY	SP ← IY	Load SP with IY	1	1	1	1	1	0	1	1		10	2	.	.	.	.	.	.	.	.
LDD	(DE) ← (HL) DE ← DE - 1 HL ← HL - 1, BC ← BC - 1	Load location (DE) with location (HL), decrement DE, HL and BC	1	1	1	0	1	1	0	1	1	16	2	.	.	.	.	.	.	.	.
LDDR	(DE) ← (HL) DE ← DE - 1 HL ← HL - 1, BC ← BC - 1 BC = 0	Load location (DE) with location (HL)	1	1	1	0	1	1	0	1	1	21	2	.	.	.	.	.	.	.	.



**Instruction Set (cont)**

Mnemonic	Operation	Description	Operation Code							No. of Clocks	No. of Bytes	Flags						
			7	6	5	4	3	2	1			0	C	Z	P/V	S	N	H
POP IX	IX <sub>H</sub> ← (SP + 1) IX <sub>L</sub> ← (SP)	Load IX with top of stack	1	1	0	1	1	0	1	14	2	•	•	•	•	•	•	
POP IY	IY <sub>H</sub> ← (SP + 1) IY <sub>L</sub> ← (SP)	Load IY with top of stack	1	1	1	1	1	0	1	14	2	•	•	•	•	•	•	
POP qq	qq <sub>H</sub> ← (SP + 1) qq <sub>L</sub> ← (SP)	Load Reg. pair qq with top of stack	1	1	q	0	0	0	1 (G)	10	1	•	•	•	•	•	•	
PUSH IX	(SP - 2) ← IX <sub>L</sub> (SP - 1) ← IX <sub>H</sub>	Load IX onto stack	1	1	0	1	1	0	1	15	2	•	•	•	•	•	•	
PUSH IY	(SP - 2) ← IY <sub>L</sub> (SP - 1) ← IY <sub>H</sub>	Load IY onto stack	1	1	1	1	0	1	1	15	2	•	•	•	•	•	•	
PUSH qq	(SP - 2) ← qq <sub>L</sub> (SP - 1) ← qq <sub>H</sub>	Load Reg. pair qq onto stack	1	1	q	0	0	1 (G)	1	11	1	•	•	•	•	•	•	
RES b,r	S <sub>b</sub> ← 0	Reset Bit b of Reg. r	1	1	0	0	1	0	1 (B)	8	2	•	•	•	•	•	•	
RES b, (HL)	S <sub>b</sub> ← 0, (HL)	Reset Bit b of loc. (HL)	1	1	0	0	1	0	1 (E)	15	2	•	•	•	•	•	•	
RES b, (IX + d)	S <sub>b</sub> ← 0 (IX + d)	Reset Bit b of loc. (IX + d)	1	1	0	1	1	0	1	23	4	•	•	•	•	•	•	
RES b, (IY + d)	S <sub>b</sub> ← 0, (IY + d)	Reset Bit b of loc. (IY + d)	1	1	1	1	1	0	1	23	4	•	•	•	•	•	•	
RET	PC <sub>L</sub> ← (SP) PC <sub>H</sub> ← (SP + 1)	Return from subroutine	1	1	0	0	1	0	0	10	1	•	•	•	•	•	•	
RET cc	If condition cc is false cont. else (PC <sub>L</sub> ← (SP) PC <sub>H</sub> ← (SP + 1)	Return from subroutine if condition cc is true	1	1	←	cc	→	0	0 (H)	5, if CC false 11 if CC true	1	•	•	•	•	•	•	
RETI		Return from interrupt	1	1	0	1	1	0	1	14	2	•	•	•	•	•	•	
RETN		Return from non-maskable interrupt	1	1	0	1	1	0	1	14	2	•	•	•	•	•	•	
RL r,		Rotate left through carry Reg. r	1	1	0	0	1	0	1 (B)	2	2	↑	↑	↑	↑	P	↑	0
RL (HL)		Rotate left through carry loc. (HL)	1	1	0	0	1	0	1	4	2	↑	↑	↑	↑	P	↑	0

## Instruction Set (cont)


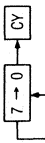
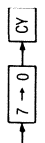
Mnemonic	Operation	Description	Operation Code										No. of Clocks	No. of Bytes	Flags									
			7	6	5	4	3	2	1	0	C	Z			P	S	N	H						
RL (IX + d)		Rotate left through carry loc. (IX + d)	1	1	0	1	1	0	1	0	1	0	1	6	4	↑	↑	↑	P	↑	0	0		
RL (IY + d)	(IX + d), (IY + d), A m = r, (HL)	Rotate left through carry loc. (IY + d)	1	1	1	1	1	0	1	0	1	0	1	6	4	↑	↑	↑	P	↑	0	0		
RLA		Rotate left ACC through carry	0	0	0	1	0	1	1	1	1	0	4	1	↑	•	•	•	•	•	•	0	0	
RLC (HL)		Rotate location (HL) left circular	1	1	0	1	0	1	1	1	0	1	15	2	↑	↑	↑	P	↑	0	0	0		
RLC (IX + d)		Rotate location (IX + d) left circular	1	1	0	1	1	0	1	0	1	1	23	4	↑	↑	↑	P	↑	0	0	0		
RLC (IY + d)		Rotate location (IY + d) left circular	1	1	1	1	1	0	1	0	1	1	23	4	↑	↑	↑	P	↑	0	0	0		
RLC r	m = r, (HL), (IX + d), (IY + d), A	Rotate Reg. r left circular	1	1	0	0	1	0	1	1	(B)	8	2	↑	↑	↑	P	↑	0	0	0	0		
RLCA		Rotate left circular ACC	0	0	0	0	0	1	1	1	1	4	1	↑	•	•	•	•	•	•	•	0	0	
RLD		Rotate digit left and right between ACC and location (HL)	1	1	1	0	1	0	1	0	1	18	2	•	↑	↑	P	↑	0	0	0	0		
RR r		Rotate right through carry Reg. r	1	1	0	0	1	0	1	1	(B)	2	2	↑	↑	↑	P	↑	0	0	0	0		
RR (HL)		Rotate right through carry loc. (HL)	1	1	0	0	1	0	1	1	1	4	2	↑	↑	↑	P	↑	0	0	0	0		
RR (IX + d)		Rotate right through carry loc. (IX + d)	1	1	0	1	1	0	1	0	1	6	4	↑	↑	↑	P	↑	0	0	0	0		
RR (IY + d)		Rotate right through carry loc. (IY + d)	1	1	1	1	1	0	1	0	1	6	4	↑	↑	↑	P	↑	0	0	0	0		
RRR	m = r, (HL), (IX + d), (IY + d), A	Rotate right ACC through carry	0	0	0	1	1	1	1	1	1	4	1	↑	•	•	•	•	•	•	•	•	0	0
RRC r		Rotate Reg. r right circular	1	1	0	0	1	0	1	1	(B)	2	2	↑	↑	↑	P	↑	0	0	0	0		



**Instruction Set (cont)**

Mnemonic	Operation	Description	Operation Code				No. of Clocks	No. of Bytes	Flags															
			7	6	5	4			3	2	1	0	C	Z	P/V	S	N	H						
RRC (HL)		Rotate loc. (HL) right circular	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0			
RRC (IX + d)		Rotate loc. (IX + d) right circular	1	1	0	1	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0		
RRC (IY + d)	$m = r, (HL), (IX + d), (IY + d), A$	Rotate loc. (IY + d) right circular	1	1	1	1	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	
RRCA		Rotate right circular ACC	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
RRD		Rotate digit right and then left between ACC and location (HL)	1	1	1	0	1	1	0	1	0	1	0	1	1	0	1	0	0	1	1	0	1	
RST <sub>T</sub>	$(SP - 1) \leftarrow PC_H$ $(SP - 2) \leftarrow PC_L$ $PC_H \leftarrow 0, PC_L \leftarrow T$	Restart to location T	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
SBC A, r	$A \leftarrow A - r$ CY	Subtract Reg. r from ACC w/carry	1	0	0	1	1	r	r	r	r	0	0	0	0	0	0	0	0	0	0	0	0	
SBC A, n	$A \leftarrow A - n - CY$	Subtract value n from ACC with carry	1	1	0	1	1	1	1	0	n	n	n	n	n	n	n	n	n	n	n	n	n	n
SBC A, (HL)	$A \leftarrow A - (HL) - CY$	Sub. loc. (HL) from ACC w/carry	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SBC A, (IX + d)	$A \leftarrow A - (IX + d) - CY$	Subtract loc. (IX + d) from ACC with carry	1	1	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
SBC A, (IY + d)	$A \leftarrow A - (IY + d) - CY$	Subtract loc. (IY + d) from ACC with carry	1	1	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
SBC HL, ss	$HL \leftarrow HL - ss - CY$	Subtract Reg. pair ss from HL with carry	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
SCF	$CY \leftarrow 1$	Set carry flag (C = 1)	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
SET b, (HL)	$(HL)_b \leftarrow 1$	Set Bit b of location (HL)	1	1	0	0	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	
SET b, (IX + d)	$(IX + d)_b \leftarrow 1$	Set Bit b of location (IX + d)	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	

## Instruction Set (cont)

Mnemonic	Operation	Description	Operation Code							No. of Clocks	No. of Bytes	Flags						
			7	6	5	4	3	2	1			0	C	Z	P	S	N	H
SET b, (IY + d)	(IY + d) <sub>b</sub> ← 1	Set Bit b of location (IY + d)	1	1	1	1	1	0	1	(E)	23	4	.	.	.	.	.	.
SET b, r	r <sub>b</sub> ← 1	Set Bit b of Reg. r	1	1	0	0	1	0	1	(B)	8	2	.	.	.	.	.	.
SLA r		Shift Reg. r left arithmetic	1	1	0	0	1	0	1	(B)	8	2	.	.	.	.	.	.
SLA (HL)		Shift loc. (HL) left arithmetic	0	0	1	0	0	1	1	0	15	2	.	.	.	.	.	.
SLA (IX + d)	m = r, (HL), (IX + d), (IY + d)	Shift loc. (IX + d) left arithmetic	1	1	0	1	1	0	1		23	4	.	.	.	.	.	.
SLA (IY + d)		Shift loc. (IY + d) left arithmetic	1	1	1	1	1	0	1		23	4	.	.	.	.	.	.
SRA r		Shift Reg. r right arithmetic	0	0	1	0	1	1	(B)		8	2	.	.	.	.	.	.
SRA (HL)		Shift loc. (HL) right arithmetic	1	1	0	0	1	0	1	1	15	2	.	.	.	.	.	.
SRA (IX + d)	m = r, (HL), (IX + d), (IY + d)	Shift loc. (IX + d) right arithmetic	1	1	0	1	1	0	1		23	4	.	.	.	.	.	.
SRA (IY + d)		Shift loc. (IY + d) right arithmetic	1	1	1	1	0	1	1		23	4	.	.	.	.	.	.
SRL r		Shift Reg. r right logical	1	1	0	0	1	0	1	(B)	8	2	.	.	.	.	.	.
SRL (HL)		Shift loc. (HL) right logical	1	1	0	0	1	0	1	1	15	2	.	.	.	.	.	.
SRL (IX + d)	m = r, (HL), (IX + d), (IY + d)	Shift loc. (IX + d) right logical	1	1	0	1	1	0	1		23	4	.	.	.	.	.	.

**Instruction Set (cont)**

Mnemonic	Operation	Description	Operation Code							No. of Cycles	No. of Bytes	Flags						
			7	6	5	4	3	2	1			0	Z	P/V	S	N	H	
SRL (Y + d)		Shift loc. (Y + d) right logical	1	1	1	1	1	0	1	23	4	↑	↑	↑	P	↑	0	0
SUB r	A ← A - r	Subtract Reg. r from ACC	1	0	0	1	0	r	r	(B)	4	1	↑	↑	V	↑	1	↑
SUB n	A ← A - n	Subtract value n from ACC	1	1	0	1	0	1	0	7	2	↑	↑	V	↑	1	↑	
SUB (HL)	A ← A - (HL)	Subtract loc. (HL) from ACC	n	n	n	n	n	n	n	7	1	↑	↑	V	↑	1	↑	
SUB (IX + d)	A ← A - (IX + d)	Subtract loc. (IX + d) from ACC	1	0	1	1	0	1	1	19	3	↑	↑	V	↑	1	↑	
SUB (IY + d)	A ← A - (IY + d)	Subtract loc. (IY + d) from ACC	1	0	1	1	0	1	1	19	3	↑	↑	V	↑	1	↑	
XOR r	A ← A ∨ r	Exclusive 'OR' Reg. r and ACC	1	1	0	1	1	0	1	4	1	↑	↑	P	↑	1	↑	
XOR n	A ← A ∨ n	Exclusive 'OR' value n and ACC	1	1	0	1	1	0	1	7	2	↑	↑	P	↑	1	↑	
XOR (HL)	A ← A ∨ (HL)	Exclusive 'OR' loc. (HL) and ACC	n	n	n	n	n	n	n	7	1	↑	↑	P	↑	1	↑	
XOR (IX + d)	A ← A ∨ (IX + d)	Exclusive 'OR' loc. (IX + d) and ACC	1	1	0	1	1	0	1	19	3	↑	↑	P	↑	1	↑	
XOR (IY + d)	A ← A ∨ (IY + d)	Exclusive 'OR' loc. (IY + d) and ACC	1	1	0	1	1	0	1	19	3	↑	↑	P	↑	1	↑	

**Note:**

- (1) P/V flag is 0 if B = 0, else P/V = 1
- (2) Z = 1 if A = (HL), else Z = 0
- (3) If B = 0, Z flag set, else reset

Reg ss	Reg r	Reg pp	Reg rr	Bit b	Reg r, r'	Reg qq	CC	Condition	Relevant Flag	Reg r		
BC 00	A 111	BC 00	BC 00	0	000	A 111	BC 00	000	NZ	Non zero	Z	B 000
DE 01	B 000	DE 01	DE 01	1	001	B 000	DE 01	001	Z	Zero	Z	C 001
HL 10	C 001	IX 10	IY 10	2	010	C 001	HL 10	010	NC	Non carry	C	D 010
SP 11	D 010	SP 11	SP 11	3	011	D 010	AF 11	011	C	Carry	C	E 011
	E 011			4	100	E 011	100	PO	Parity odd	P/V	H 100	
	H 100			5	101	H 100	101	PE	Parity even	P/V	L 101	
	L 101			6	110	L 101	110	P	Sign positive	S	F 110	
				7	111		111	M	Sign negative	S	A 111	

## Description

The μPD8085A-2, μPD8085AH, and μPD8085AH-2 8-bit, single-chip microprocessors are 100 percent software compatible with the industry standard 8080A. They have the ability of increasing system performance of the 8080A by operating at a higher speed. Using the μPD8085A in conjunction with its family of ICs allows the designer complete flexibility with minimum chip count. The H (HMOS) versions have lower power consumptions than the non-H versions.

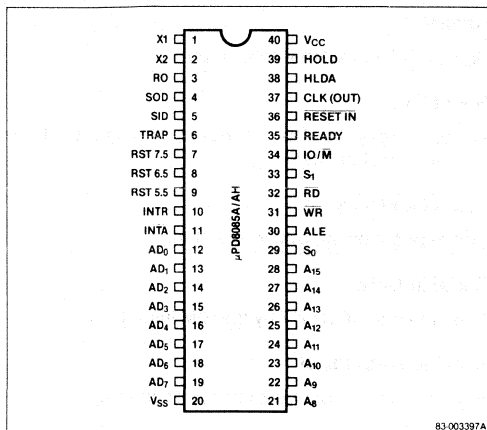
## Features

- Single power supply, +5V, ±10%
- Internal clock generation and system control
- Internal serial in/out port
- Fully TTL-compatible
- Internal four-level interrupt structure
- Multiplexed address/data bus for increased system performance
- Complete family of components for design flexibility
- Software compatible with industry standard 8080A
- Higher throughput
  - μPD8085A-2 — 5 MHz
  - μPD8085AH — 3 MHz
  - μPD8085AH-2 — 5 MHz

## Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD8085AC-2	40-pin plastic DIP	5 MHz
μPD8085AHC	40-pin plastic DIP	3 MHz
μPD8085AHC-2	40-pin plastic DIP	5 MHz

## Pin Configuration



## Pin Identification

No.	Symbol	Function
1, 2	X1, X2	Crystal in
3	RO	Reset out
4	SOD	Serial out data
5	SID	Serial in data
6	TRAP	Trap interrupt input
7	RST 7.5	Restart interrupts
8	RST 6.5	Restart interrupts
9	RST 5.5	Restart interrupts
10	INTR	Interrupt request in
11	INTA	Interrupt acknowledge
12-19	AD <sub>0</sub> -AD <sub>7</sub>	Low address / data bus
20	V <sub>SS</sub>	Ground
21-28	A <sub>8</sub> -A <sub>15</sub>	High address bus
29, 33	S <sub>0</sub> , S <sub>1</sub>	Status outputs
30	ALE	Address latch enable out
31, 32	WR, RD	Write / read strobes out
34	IO / M	I / O or memory indicator
35	READY	Ready input
36	RESET IN	Reset input
37	CLK	Clock out
38, 39	HLDA, HOLD	Hold acknowledge out and hold input request
40	V <sub>CC</sub>	+5 V supply

**Pin Functions****Crystal In**

Crystal, RC, or external clock input.

**Reset Out**

Acknowledges that the processor is being reset to be used as a system reset.

**Serial Out Data**

1-bit data out by the SIM instruction.

**Serial In Data**

1-bit data into ACC bit 7 by the RIM instruction.

**Trap Interrupt Input**

Highest priority nonmaskable restart interrupt.

**Restart Interrupts**

Priority restart interrupt inputs, of which 7.5 is the highest and 5.5 the lowest priority.

**Interrupt Request In**

A general interrupt input which stops the PC from incrementing, generates  $\overline{INTA}$ , and samples the data bus for a restart or call instruction.

**Interrupt Acknowledge**

An output which indicates that the processor has responded to INTR.

**Low Address/Data Bus**

Multiplexed low address and data bus.

**Ground**

Ground Reference.

**High Address Bus**

Nonmultiplexed high 8 bits of the address bus.

**Status Outputs**

Outputs which indicate data bus status: Halt, Write, Read, Fetch.

**Address Latch Enable Out**

A signal which indicates that the lower 8 bits of address are valid on the AD lines.

**Write/Read Strobes Out**

Signals out which are used as write and read strobes for memory and I/O devices.

**I/O or Memory Indicator**

A signal out which indicates whether  $\overline{RD}$  or  $\overline{WR}$  strobes are for I/O or memory devices.

**Ready Input**

An input which is used to increase the data and address bus access times (can be used for slow memory).

**Reset Input**

An input which is used to start the processor activity at address 0, resetting IE and HLDA flip-flops.

**Clock Out**

System clock output.

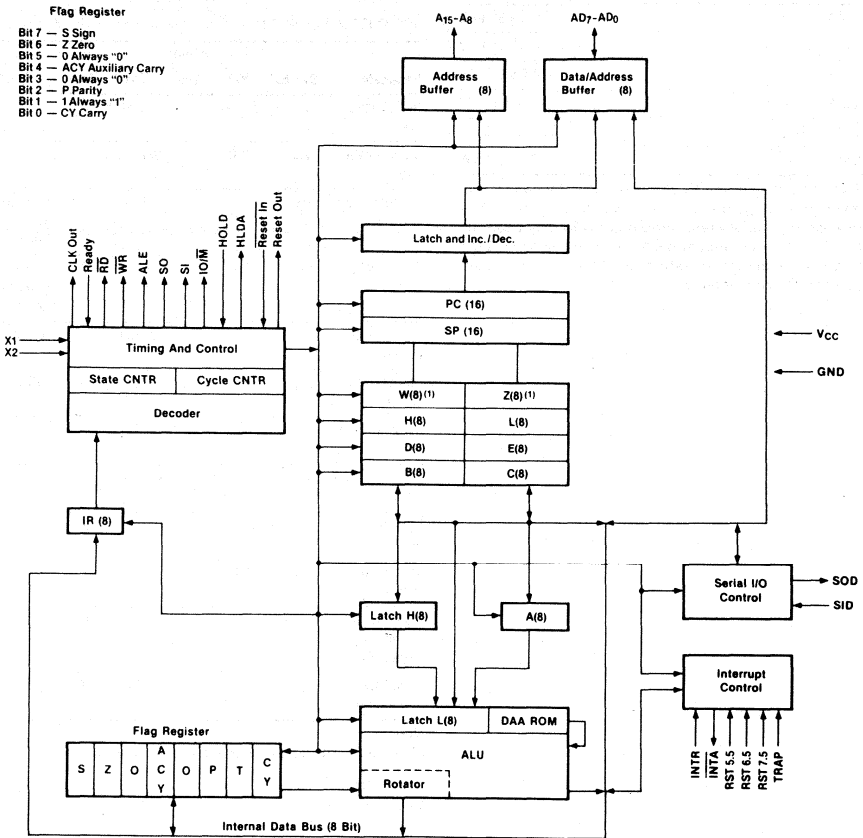
**Hold Acknowledge Out and Hold Input Request**

Used to request and indicate that the processor should relinquish the bus for DMA activity. When hold is acknowledged,  $\overline{RD}$ ,  $\overline{WR}$ , IO/M, address and data buses are all three-stated.

**+5 V Supply**

Power supply input.

## Block Diagram



Note:  
(1) Temporary Register

**Absolute Maximum Ratings**

μPD8085A-2: T<sub>A</sub> = 25°C; V<sub>CC</sub> = +5 V ± 5%

Power supply voltage, V <sub>DD</sub>	-0.5 V to +7 V
Input voltage, V <sub>I</sub>	-0.5 V to +7 V
Output voltage, V <sub>O</sub>	-0.5 V to +7 V
Operating temperature, T <sub>OP</sub> T	0°C to +70°C
Storage temperature, T <sub>STG</sub>	-65°C to +150°C
Power dissipation, P <sub>D</sub>	1.5 W

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**DC Characteristics**

μPD8085AH, μPD8085AH-2: T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = +5 V ± 10%, V<sub>SS</sub> = GND

μPD8085A-2: T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = +5 V ± 5%, V<sub>SS</sub> = GND

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Typ		
Input voltage low	V <sub>IL</sub>	V <sub>SS</sub> -0.5		V <sub>SS</sub> +0.8	V
Input voltage high	V <sub>IH</sub>	2.0		V <sub>CC</sub> +0.5	V
Output voltage low	V <sub>OL</sub>			+0.45	V I <sub>OL</sub> = 2.0 mA, I <sub>OH</sub> = -400 μA. (Notes 1 & 2)
Output voltage high	V <sub>OH</sub>	2.4		V	I <sub>OH</sub> = -400 μA, I <sub>OL</sub> = 2 mA. (Notes 1 & 2)
Input leakage current	I <sub>LI</sub>			±10(1)	μA 0 V ≤ V <sub>IN</sub> ≤ V <sub>CC</sub>
Output leakage current	I <sub>LO</sub>			±10(1)	μA 0.45 V ≤ V <sub>OUT</sub> ≤ V <sub>CC</sub>
Input level low. reset	V <sub>ILR</sub>	-0.5		+0.8	V
Input level high. reset	V <sub>IHR</sub>	2.4		V <sub>CC</sub> +0.5	V
Hysteresis. reset	V <sub>HY</sub>	0.25			V
X1, X2 input voltage high	V <sub>IHX</sub>	4.0		V <sub>CC</sub> +0.5	V
Power supply current (V <sub>CC</sub> ) μPD8085A-2	I <sub>CC(AV)</sub>			170	mA t <sub>CY</sub> min
μPD8085AH, μPD8085AH-2				135	mA t <sub>CY</sub> min. (Note 3)

**Note:**

- (1) Minus (-) designates current flow out of the device.
- (2) On all outputs.
- (3) Maximum unit test.

## AC Characteristics

μPD8085A-2: T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5 V ± 5%

μPD8085AH, μPD8085AH-2: T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5 V ± 10%

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8085AH		μPD8085AH-2, μPD8085A-2			
		Min	Max	Min	Max		
CLK cycle period	t <sub>CYC</sub>	320	2000	200	2000	ns	
CLK low time	t <sub>1</sub>	80		40		ns	
CLK high time	t <sub>2</sub>	120		70		ns	
CLK rise time	t <sub>r</sub>		30		30	ns	
CLK fall time	t <sub>f</sub>		30		30	ns	
X1 rising to CLK rising	t <sub>XKR</sub>	30	120	30	100	ns	
X1 rising to CLK falling	t <sub>XKF</sub>	30	150	30	110	ns	
A <sub>8</sub> -A <sub>15</sub> valid to leading edge of CONTROL	t <sub>AC</sub>	270		115		ns	(Note 1)
A <sub>0</sub> -A <sub>7</sub> valid to leading edge of CONTROL	t <sub>ACL</sub>	240		115		ns	
A <sub>0</sub> -A <sub>15</sub> valid to data input	t <sub>AD</sub>		575		350	ns	
Address float after leading edge of RD (INTA)	t <sub>AFR</sub>		0		0	ns	
A <sub>8</sub> -A <sub>15</sub> valid before trailing edge of ALE	t <sub>AL</sub>	115		50		ns	(Note 1)
A <sub>0</sub> -A <sub>7</sub> valid before trailing edge of ALE	t <sub>ALL</sub>	90		50		ns	
READY valid from address valid	t <sub>ARY</sub>		220		100	ns	
A <sub>8</sub> -A <sub>15</sub> valid after CONTROL	t <sub>CA</sub>	120		60		ns	
Width of control low (RD, WR, INTA)	t <sub>CC</sub>	400		230		ns	
Trailing edge of CONTROL to leading edge of ALE	t <sub>CL</sub>	50		25		ns	
Data valid to trailing edge of WR	t <sub>DW</sub>	420		230		ns	
HLDA to bus enable	t <sub>HABE</sub>		210		150	ns	
Bus float after HLDA	t <sub>HABF</sub>		210		150	ns	
HLDA valid to trailing edge of CLK	t <sub>HACK</sub>	110		40		ns	
HOLD hold time	t <sub>HDH</sub>	0		0		ns	
HOLD setup time to trailing edge of CLK	t <sub>HDS</sub>	170		120		ns	
INTR hold time	t <sub>INH</sub>	0		0		ns	
INTR, RST, TRAP setup time to trailing edge of CLK	t <sub>INS</sub>	160		150		ns	
Address hold time after ALE	t <sub>LA</sub>	100		50		ns	
Trailing edge of ALE to leading edge of CONTROL	t <sub>LC</sub>	130		60		ns	
ALE low time during CLK high	t <sub>LCK</sub>	100		50		ns	
ALE to valid data input during read	t <sub>LDR</sub>		460		270	ns	
ALE to valid data during write	t <sub>LDW</sub>		200		120	ns	
ALE pulse width	t <sub>LL</sub>	140		80		ns	
ALE to READY stable	t <sub>LRV</sub>		110		30	ns	



**AC Characteristics (cont)**

μPD8085A-2: T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5 V ± 5%

μPD8085AH, μPD8085AH-2: T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5 V ± 10%

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8085AH		μPD8085AH-2, μPD8085A-2			
		Min	Max	Min	Max		
Trailing edge of $\overline{RD}$ to re-enabling of address	t <sub>RAE</sub>	150		90		ns	
$\overline{RD}$ (or $\overline{INTA}$ ) to valid data	t <sub>RD</sub>		300		150	ns	
Trailing edge of CONTROL to leading edge of next CONTROL	t <sub>RV</sub>	400		220		ns	
Data hold time after $\overline{RD}$ ( $\overline{INTA}$ )	t <sub>RDH</sub>	0		0		ns	(Note 7)
READY hold time	t <sub>RYH</sub>	0		0		ns	
READY setup time to leading edge of CLK	t <sub>RYs</sub>	110		100		ns	
Leading edge data valid after trailing edge of $\overline{WR}$	t <sub>WD</sub>	100		60		ns	
Leading edge of $\overline{WR}$ to data valid	t <sub>WDL</sub>		40		20	ns	

- Note:**
- A<sub>8</sub>-A<sub>15</sub> address specs apply to IO/ $\overline{M}$ . S<sub>0</sub> and S<sub>1</sub> except A<sub>8</sub>-A<sub>15</sub> are undefined during T<sub>4</sub>-T<sub>6</sub> of OF cycle whereas IO/ $\overline{M}$ , S<sub>0</sub> and S<sub>1</sub> are stable.
  - Test conditions: t<sub>CYC</sub> = 320 ns (8085AH)/200 ns (8085A-2) C<sub>L</sub> = 150 pF
  - For all output timing except where C<sub>L</sub> = 150 pF use the following correction factors:  
 25 pF, C<sub>L</sub> = 150 pF: - 0.10 ns/pF  
 150 pF, C<sub>L</sub> = 300 pF: + 0.3 ns/pF
  - Output timings are measured with purely capacitive load.
  - All timings are measured as the following:  
 Output voltage: V<sub>L</sub> = 0.8 V, V<sub>H</sub> = 2.0 V  
 Input voltage: 1.5 V; t<sub>n</sub>, t<sub>f</sub> = 20 ns
  - To calculate timing specifications at other values of t<sub>CYC</sub> use Bus Timing Specifications.
  - Data hold time is guaranteed under all loading conditions.

**Bus Timing Specifications**

**t<sub>CYC</sub> as a Dependent**

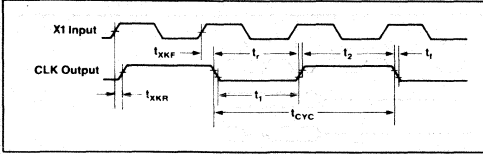
Symbol	Timing Formula		Min/Max
	μPD8085AH	μPD8085A-2, μPD8085AH-2	
t <sub>AL</sub>	(1/2) t <sub>CY</sub> - 45	(1/2) t <sub>CY</sub> - 50	Min
t <sub>LA</sub>	(1/2) t <sub>CY</sub> - 60	(1/2) t <sub>CY</sub> - 50	Min
t <sub>LL</sub>	(1/2) t <sub>CY</sub> - 20	(1/2) t <sub>CY</sub> - 20	Min
t <sub>LCK</sub>	(1/2) t <sub>CY</sub> - 60	(1/2) t <sub>CY</sub> - 50	Min
t <sub>LC</sub>	(1/2) t <sub>CY</sub> - 30	(1/2) t <sub>CY</sub> - 40	Min
t <sub>AD</sub>	(5/2 + N) t <sub>CY</sub> - 225	(5/2 + N) t <sub>CY</sub> - 150	Max
t <sub>RD</sub>	(3/2 + N) t <sub>CY</sub> - 180	(3/2 + N) t <sub>CY</sub> - 150	Max
t <sub>RAE</sub>	(1/2) t <sub>CY</sub> - 10	(1/2) t <sub>CY</sub> - 10	Min
t <sub>CA</sub>	(1/2) t <sub>CY</sub> - 40	(1/2) t <sub>CY</sub> - 40	Min
t <sub>DW</sub>	(3/2 + N) t <sub>CY</sub> - 60	(3/2 + N) t <sub>CY</sub> - 70	Min
t <sub>WD</sub>	(1/2) t <sub>CY</sub> - 60	(1/2) t <sub>CY</sub> - 40	Min

Symbol	Timing Formula		Min/Max
	μPD8085AH	μPD8085A-2, μPD8085AH-2	
t <sub>CC</sub>	(3/2 + N) t <sub>CY</sub> - 80	(3/2 + N) t <sub>CY</sub> - 70	Min
t <sub>CL</sub>	(1/2) t <sub>CY</sub> - 110	(1/2) t <sub>CY</sub> - 75	Min
t <sub>ARY</sub>	(3/2) t <sub>CY</sub> - 260	(3/2) t <sub>CY</sub> - 200	Max
t <sub>HACK</sub>	(1/2) t <sub>CY</sub> - 50	(1/2) t <sub>CY</sub> - 60	Min
t <sub>HABF</sub>	(1/2) t <sub>CY</sub> + 50	(1/2) t <sub>CY</sub> - 50	Max
t <sub>HABE</sub>	(1/2) t <sub>CY</sub> + 50	(1/2) t <sub>CY</sub> - 50	Max
t <sub>AC</sub>	(2/2) t <sub>CY</sub> - 50	(2/2) t <sub>CY</sub> - 85	Min
t <sub>1</sub>	(1/2) t <sub>CY</sub> - 80	(1/2) t <sub>CY</sub> - 60	Min
t <sub>2</sub>	(1/2) t <sub>CY</sub> - 40	(1/2) t <sub>CY</sub> - 30	Min
t <sub>RV</sub>	(3/2) t <sub>CY</sub> - 80	(3/2) t <sub>CY</sub> - 80	Min
t <sub>LDR</sub>	(4/2 + N) t <sub>CY</sub> - 180	(4/2 + N) t <sub>CY</sub> - 130	Max

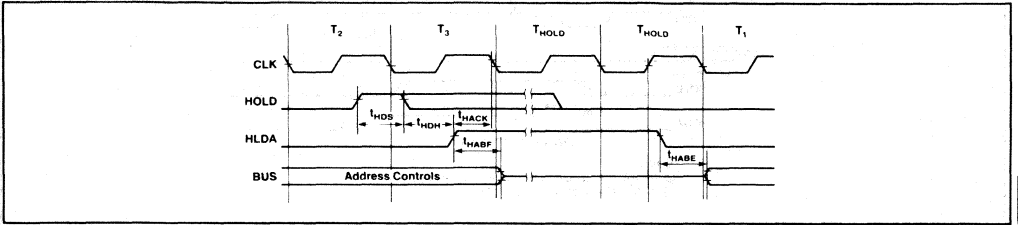
- Note:**
- N = Number of WAIT state.

## Timing Waveforms

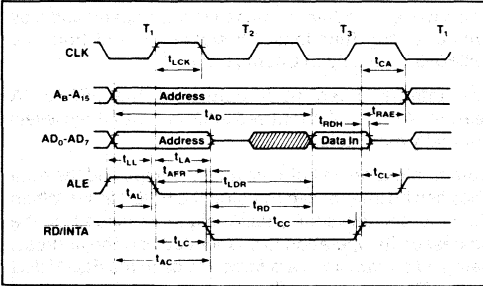
### Clock Timing Waveform



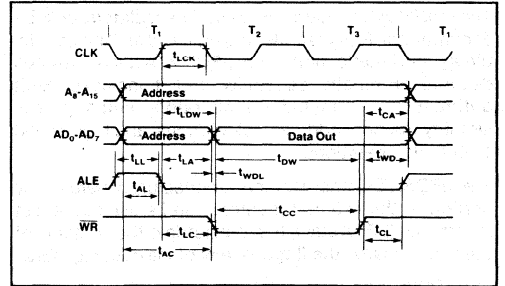
### Hold Timing



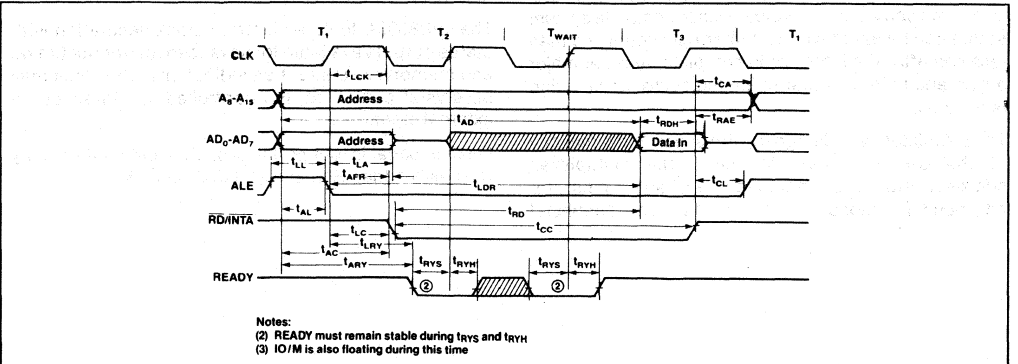
### 8085AH Bus Timing Read Operation



### Write Operation

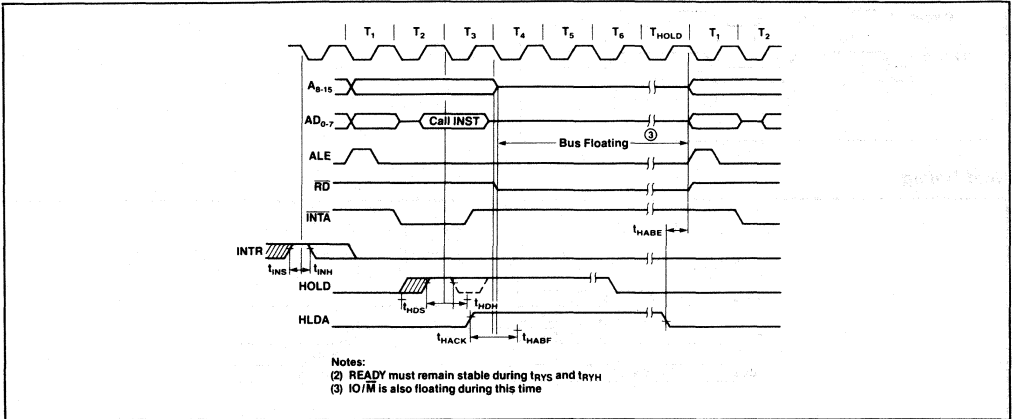


### Read Operation with Wait Cycle (same Ready Timing Applies to Write Operation)



Timing Waveforms (cont)

Interrupt and Hold Timing



Functional Description

The μPD8085A contains six 8-bit data registers, an 8-bit accumulator, four testable flag bits, and an 8-bit parallel binary arithmetic unit. The μPD8085A also provides decimal arithmetic capability and it includes 16-bit arithmetic and immediate operators which greatly simplify memory address calculations, and high speed arithmetic operations.

The μPD8085A has a stack architecture wherein any portion of the external memory can be used as a last in/first out (LIFO) stack to store/retrieve the contents of the accumulator, the flags, or any of the data registers.

The μPD8085A also contains a 16-bit stack pointer to control the addressing of this external stack. One of the major advantages of the stack is that multiple level interrupts can easily be handled since complete system status can be saved when an interrupt occurs and then restored after the interrupt is complete. Another major advantage is that almost unlimited subroutine nesting is possible.

The μPD8085A was designed with speed and simplicity of the overall system in mind. The multiplexed address/data bus increases available pins for advanced functions in the processor and peripheral

chips while providing increased system speed and less critical timing functions. All signals to and from the μPD8085A are fully TTL-compatible.

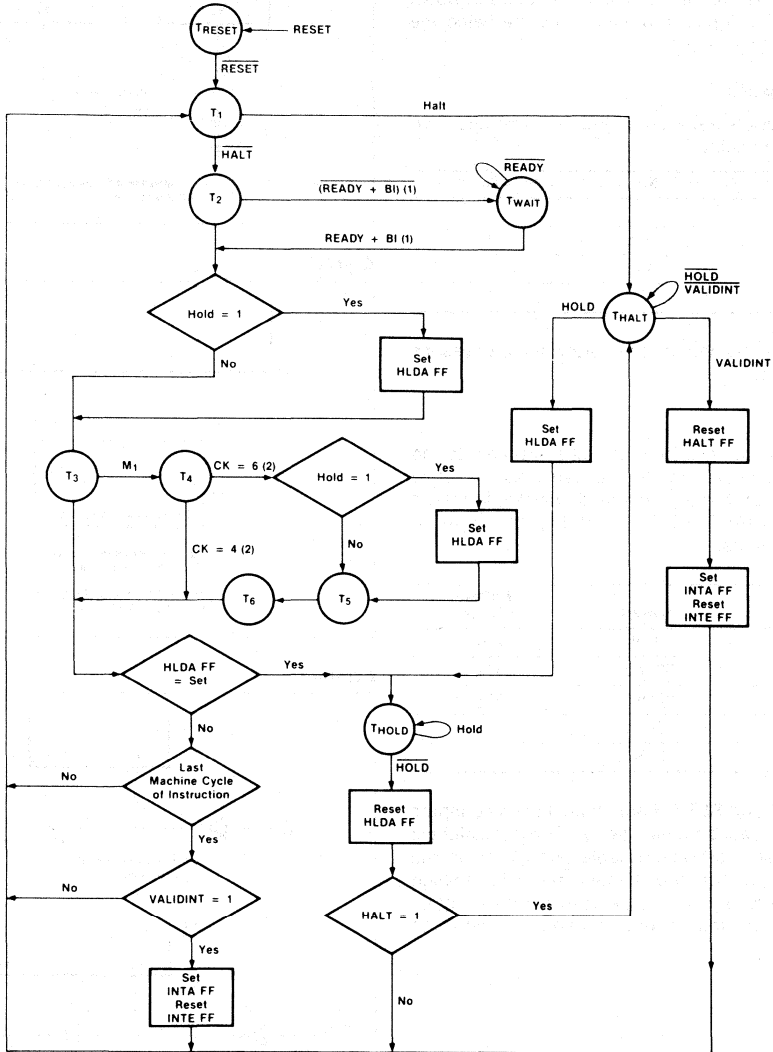
The internal interrupt structure of the μPD8085A features 4 levels of prioritized interrupt with three levels internally maskable.

Communication on both the address lines and the data lines can be interlocked by using the HOLD input. When the hold acknowledge (HLDA) signal is issued by the processor, its operation is suspended and the address, data and control lines are forced to be in the FLOATING state. This permits other devices, such as direct memory access channels (DMA), to be connected to the address and data buses.

The μPD8085A features internal clock generation with status outputs available for advanced read/write timing and memory/IO instruction indications. The clock may be crystal controlled, RC controlled, or driven by an external signal.

On-chip serial in/out port is available and controlled by the newly added RIM and SIM instructions.

Processor State Transition Diagram



- Notes:
1. BI indicates that the bus is idle during this machine cycle.
  2. CK indicates the number of clock cycles in this machine cycle.

**Clock Inputs**

As stated, the timing for the μPD8085A may be generated in one of two ways: crystal, or external clock. Recommendations for these methods are shown below. Note the input frequency must be twice the internal operating frequency.

**Status Outputs**

The status outputs are valid during ALE time and have the following meaning:

	S <sub>1</sub>	S <sub>0</sub>
Halt	0	0
Write	0	1
Read	1	0
Fetch	1	1

These pins may be decoded to portray the processor's data bus status.

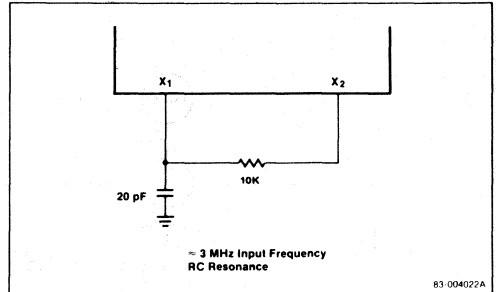
**Interrupts**

The μPD8085A has five interrupt pins available to the user. INTR is operationally the same as the 8080 interrupt request, three (3) internally maskable restart interrupts: RESTART 5.5, 6.5, and 7.5, and TRAP, a non-maskable restart.

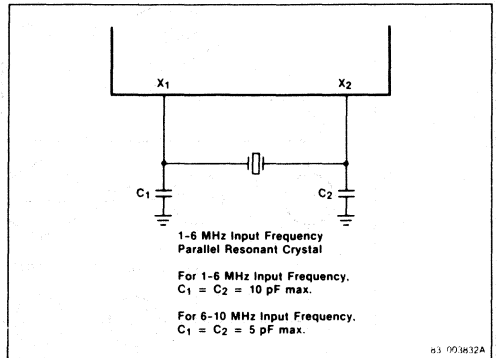
Priority	Interrupt	Restart Address
Highest	TRAP	24 <sub>16</sub>
	RST 7.5	3C <sub>16</sub>
	RST 6.5	34 <sub>16</sub>
	RST 5.5	2C <sub>16</sub>
Lowest	INTR	

INTR, RST 5.5 and RST 6.5 are all level sensing inputs while RST 7.5 is set on a rising-edge. TRAP, the highest priority interrupt, is non-maskable and is set on the rising-edge or positive level. It must make a low-to-high transition and remain high to be seen, but it will not be generated again until it makes another low-to-high transition.

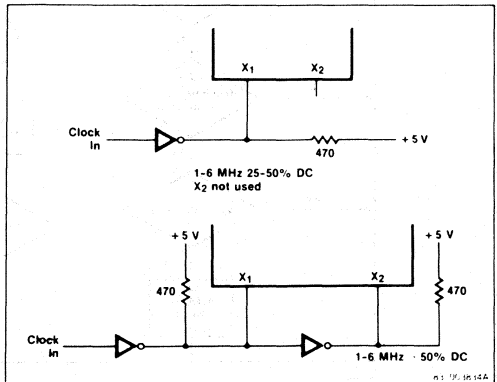
**RC**



**Crystal**



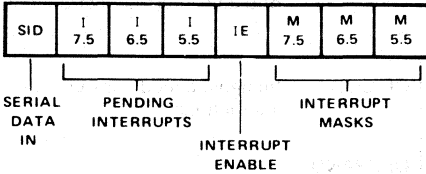
**External**



## Serial I/O

Serial input and output is accomplished with two new instructions not included in the 8080: RIM and SIM. These instructions serve several purposes: serial I/O, and reading or setting the interrupt mask.

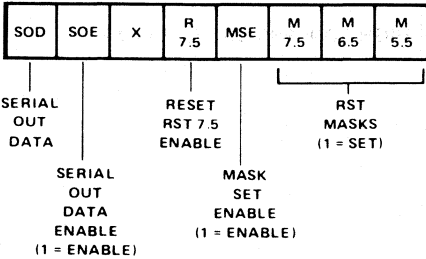
The RIM (Read Interrupt Mask) instruction is used for reading the interrupt mask and for reading serial data. After execution of the RIM instruction the ACC content is as follows:



**Note:**

(1) After the TRAP interrupt, the RIM instruction must be executed to preserve the status of IE.

The SIM (Set Interrupt Mask) instruction is used to program the interrupt mask and to output serial data. Presetting the ACC for the SIM instruction has the following meaning:



## Instruction Set

The instruction set includes arithmetic and logical operators with direct, register, indirect, and immediate addressing modes.

Move, load, and store instruction groups provide the ability to move either 8 or 16 bits of data between memory, the six working registers and the accumulator using direct, register, indirect, and immediate addressing modes.

The ability to branch to different portions of the program is provided with direct, conditional, or computed jumps. Also, the ability to call and return from subroutines is provided both conditionally and unconditionally. The RESTART (or single byte call instruction) is useful for interrupt vector operation.

Conditional jumps, calls and returns execute based on the state of the four testable flags (sign, zero, parity and carry). The state of each flag is determined by the result of the last instruction executed that affected flags. (See Instruction Set Table)

The sign flag is set (high) if bit 7 of the result is a "1"; otherwise it is reset (low). The zero flag is set if the result is "0"; otherwise it is reset. The parity flag is set if the modulo 2 sum of the bits of the result is "0" (even parity); otherwise (odd parity) it is reset. The carry flag is set if the last instruction resulted in a carry or a borrow out of the most significant bit (bit 7) of the result; otherwise it is reset.

In addition to the four testable flags, the μPD8085A has another flag (ACY) that is not directly testable. It is used for multiple precision arithmetic operations with the DAA instruction. The auxiliary carry flag is set if the last instruction resulted in a carry or a borrow from bit 3 into bit 4; otherwise it is reset.

Double precision operators such as stack manipulation and double add instructions extend both the arithmetic and interrupt handling capability of the μPD8085A. The ability to increment and decrement memory, the six general registers, and the accumulator are provided as well as extended increment and decrement instructions to operate on the register pairs and stack pointer. Further capability is provided by the ability to rotate the accumulator left or right through or around the carry bit.

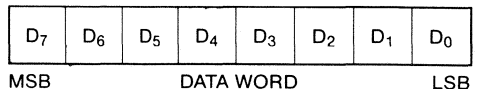
Input and output may be accomplished using memory addresses as I/O ports or the directly addressed I/O provided for in the μPD8085A instruction set.

Two instructions, RIM and SIM, are used for reading and setting the internal interrupt mask as well as input and output to the serial I/O port.

The special instruction group completes the μPD8085A instruction set: NOP, HALT stop processor execution; DAA provides decimal arithmetic capability; STC sets the carry flag; CMC complements it; CMA complements the contents of the accumulator; and XCHG exchanges the contents of two 16-bit register pairs directly.

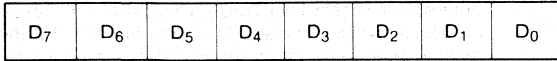
## Data and Instruction Formats

Data in the μPD8085A is stored as 8-bit binary integers. All data/instruction transfers to the system data bus are in the following format:



Instructions are one, two, or three bytes long. Multiple byte instructions must be stored in successive locations of program memory. The address of the first byte is used as the address of the instruction.

**One Byte Instructions**

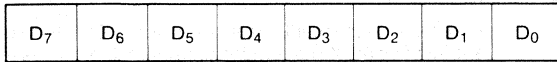


OP CODE

**Typical Instructions**

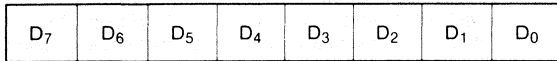
Register to register, memory reference, arithmetic or logical, rotate, return, push, pop, enable, or disable interrupt instructions

**Two Byte Instructions**



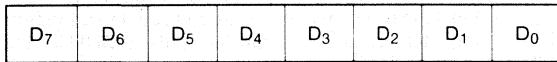
OP CODE

Immediate mode or I/O instructions



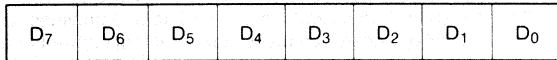
OPERAND

**Three Byte Instructions**

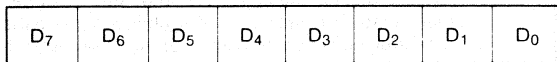


OP CODE

Jump, call or direct load and store instructions



LOW ADDRESS OR OPERAND 1



HIGH ADDRESS OR OPERAND 2

## Instruction Cycle Times

One to five machine cycles (M<sub>1</sub>-M<sub>5</sub>) are required to execute an instruction. Each machine cycle involves the transfer of an instruction or data byte into the processor or a transfer of a data byte out of the processor (the sole exception being the double add instruction). The first one, two or three machine cycles obtain the instruction from the memory or an interrupting I/O controller. The remaining cycles are used to execute the instruction. Each machine cycle requires from three to five clock times (T<sub>1</sub>-T<sub>5</sub>).

Machine cycles and clock states used for each type of instruction are shown below.

Instruction Type	Machine Cycles Executed Min/Max	Clock Status Min/Max
ALU R	1	4
CMC	1	4
CMA	1	4
DAA	1	4
DCR R	1	4
DI	1	4
EI	1	4
INR R	1	4
MOV R. R	1	4
NOP	1	4
ROTATE	1	4
RIM	1	4
SIM	1	4
STC	1	4
XCHG	1	4
HLT	1	5
DCX	1	6
INX	1	6
PCHL	1	6

Instruction Type	Machine Cycles Executed Min/Max	Clock Status Min/Max
RET COND.	1/3	6/12
SPHL	1	6
ALU I	2	7
ALU M	2	7
JNC	2/3	7/10
LDAX	2	7
MVI	2	7
MOV M. R	2	7
MOV R. M	2	7
STAX	2	7
CALL COND.	2/5	9/18
DAD	3	10
DCR M	3	10
IN	3	10
INR M	3	10
JMP	3	10
LOAD PAIR	3	10
MVI M	3	10
OUT	3	10
POP	3	10
RET	3	10
PUSH	3	12
RST	3	12
LDA	4	13
STA	4	13
LHLD	5	16
SHLD	5	16
XTHL	5	16
CALL	5	18



**Instruction Set**

Mnemonic(1)	Description	Operation Code(2)								Flags(4)			
		D7	D6	D5	D4	D3	D2	D1	D0	Cycles(3)	Sign	Zero	Parity
<b>Move</b>													
MOV d, s	Move register to register	0	1	d	d	d	s	s	s	4			
MOV M, s	Move register to memory	0	1	1	1	0	s	s	s	7			
MOV d, M	Move memory to register	0	1	d	d	d	1	1	0	7			
MVI d, D8	Move immediate to register	0	0	d	d	d	1	1	0	7			
MVI M, D8	Move immediate to memory	0	0	1	1	0	1	1	0	10			
<b>Increment / Decrement</b>													
INR d	Increment register	0	0	d	d	d	1	0	0	4	•	•	•
DCR d	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	•	•	•
<b>ALU — Register to Accumulator</b>													
ADD s	Add register to A	1	0	0	0	0	s	s	s	4	•	•	•
ADC s	Add register to A with carry	1	0	0	0	1	s	s	s	4	•	•	•
SUB s	Subtract register from A	1	0	0	1	0	s	s	s	4	•	•	•
SUBB s	Subtract register from A with borrow	1	0	0	1	1	s	s	s	4	•	•	•
ANA s	AND register with A	1	0	1	0	0	s	s	s	4	•	•	•
XRA s	Exclusive OR register with A	1	0	1	0	1	s	s	s	4	•	•	•
ORA s	OR register with A	1	0	1	1	0	s	s	s	4	•	•	•
CMP s	Compare register with A	1	0	1	1	1	s	s	s	4	•	•	•
<b>ALU — Memory to Accumulator</b>													
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7	•	•	•
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7	•	•	•
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	•	•	•
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	•	•	•
<b>ALU — Immediate to Accumulator</b>													
ADI D8	Add immediate to A	1	1	0	0	0	1	1	0	7	•	•	•
ACI D8	Add immediate to A with carry	1	1	0	0	1	1	1	0	7	•	•	•

## Instruction Set (cont)

Mnemonic(1)	Description	Operation Code(2)								Flags(4)				
		D7	D6	D5	D4	D3	D2	D1	D0	Cycles(3)	Sign	Zero	Parity	Carry
<b>ALU—Immediate to Accumulator (cont)</b>														
SUI D8	Subtract immediate from A	1	1	0	1	0	1	1	0	7	•	•	•	•
SBI D8	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7	•	•	•	•
ANI D8	AND immediate with A	1	1	1	0	0	1	1	0	7	•	•	•	0
XRI D8	Exclusive OR immediate with A	1	1	1	0	1	1	1	0	7	•	•	•	0
ORI D8	OR immediate with A	1	1	1	1	0	1	1	0	7	•	•	•	0
CPI D8	Compare immediate with A	1	1	1	1	1	1	1	0	7	•	•	•	•
<b>ALU—Rotate</b>														
RLC	Rotate A left, MSB to carry (8-bit)	0	0	0	0	0	1	1	1	4				•
RRC	Rotate A right, LSB to carry (8-bit)	0	0	0	0	1	1	1	1	4				•
RAL	Rotate A left through carry (9-bit)	0	0	0	1	0	1	1	1	4				•
RAR	Rotate A right through carry (9-bit)	0	0	0	1	1	1	1	1	4				•
<b>Jump</b>														
JMP ADDR	Jump unconditional	1	1	0	0	0	0	1	1	10				
JNZ ADDR	Jump on not zero	1	1	0	0	0	0	1	0	7/10				
JZ ADDR	Jump on zero	1	1	0	0	1	0	1	0	7/10				
JNC ADDR	Jump on no carry	1	1	0	1	0	0	1	0	7/10				
JC ADDR	Jump on carry	1	1	0	1	1	0	1	0	7/10				
JPO ADDR	Jump on parity odd	1	1	1	0	0	0	1	0	7/10				
JPE ADDR	Jump on parity even	1	1	1	0	1	0	1	0	7/10				
JP ADDR	Jump on positive	1	1	1	1	0	0	1	0	7/10				
JM ADDR	Jump on minus	1	1	1	1	1	0	1	0	7/10				
<b>Call</b>														
CALL ADDR	Call unconditional	1	1	0	0	1	1	0	1	18				
CNZ ADDR	Call on not zero	1	1	0	0	0	1	0	0	9/18				
CZ ADDR	Call on zero	1	1	0	0	1	1	0	0	9/18				
CNC ADDR	Call on no carry	1	1	0	1	0	1	0	0	9/18				
CC ADDR	Call on carry	1	1	0	1	1	1	0	0	9/18				
CPD ADDR	Call on parity odd	1	1	1	0	0	1	0	0	9/18				
CPE ADDR	Call on parity even	1	1	1	0	1	1	0	0	9/18				
CP ADDR	Call on positive	1	1	1	1	0	1	0	0	9/18				
CN ADDR	Call on minus	1	1	1	1	1	1	0	0	9/18				

**Instruction Set (cont)**

Mnemonic(1) Call (cont)	Description	Operation Code(2)								Flags(4)				
		D7	D6	D5	D4	D3	D2	D1	D0	Cycles(3)	Sign	Zero	Parity	Carry
Return														
RET	Return	1	1	0	0	1	0	0	1	10				
RNZ	Return on not zero	1	1	0	0	0	0	0	0	6/12				
RZ	Return on zero	1	1	0	0	1	0	0	0	6/12				
RNC	Return on no carry	1	1	0	0	1	0	0	0	6/12				
RC	Return on carry	1	1	0	1	1	0	0	0	6/12				
RPO	Return on parity odd	1	1	1	0	0	0	0	0	6/12				
RPE	Return on parity even	1	1	1	0	1	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				
<b>Load Register Pair</b>														
LXI B, D16	Load immediate register pair BC	0	0	0	0	0	0	0	1	10				
LXI D, D16	Load immediate register pair DE	0	0	0	1	0	0	0	1	10				
LXI H, D16	Load immediate register pair HL	0	0	1	0	0	0	0	1	10				
LXI SP, D16	Load immediate stack pointer	0	0	1	1	0	0	0	1	10				
<b>Push</b>														
PUSH B	Push register pair BC on stack	1	1	0	0	0	1	0	1	12				
PUSH D	Push register pair DE on stack	1	1	0	1	0	1	0	1	12				
PUSH H	Push register pair HL 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				
<b>Pop</b>														
POP B	Pop register pair BC off stack	1	1	0	0	0	0	0	1	10				
POP D	Pop register pair DE off stack	1	1	0	1	0	0	0	1	10				
POP H	Pop register pair HL 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				
<b>Double Add</b>														
DAD R	Add BC to HL	0	0	0	0	1	0	0	1	10				
DAD D	Add DE to HL	0	0	0	1	1	0	0	1	10				
DAD H	Add HL to HL	0	0	1	0	1	0	0	1	10				
DAD SP	Add stack pointer to HL	0	0	1	1	1	0	0	1	10				

## Instruction Set (cont)

Mnemonic(1)	Description	Operation Code(2)								Flags(4)					
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Cycles(3)	Sign	Zero	Parity	Carry	
<b>Increment Register Pair</b>															
INX B	Increment BC	0	0	0	0	0	0	0	1	1	1	6			
INX D	Increment DE	0	0	0	1	0	0	1	1	1	6				
INX H	Increment HL	0	0	1	0	0	0	1	1	1	6				
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	1	6				
<b>Decrement Register Pair</b>															
DCX B	Decrement BC	0	0	0	0	1	0	1	1	1	6				
DCX D	Decrement DE	0	0	0	1	1	0	1	1	1	6				
DCX H	Decrement HL	0	0	1	0	1	0	1	1	1	6				
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	1	6				
<b>Register Indirect</b>															
STAX B	Store A at ADDR in BC	0	0	0	0	0	0	1	0	7					
STAX D	Store A at ADDR in DE	0	0	0	1	0	0	1	0	7					
LDAX B	Load A at ADDR in BC	0	0	0	0	1	0	1	0	7					
LDAX D	Load A at ADDR in DE	0	0	0	1	1	0	1	0	7					
<b>Direct</b>															
STA ADDR	Store A direct	0	0	1	1	0	0	1	0	13					
LDA ADDR	Load A direct	0	0	1	1	1	0	1	0	13					
SHLD ADDR	Store HL direct	0	0	1	0	0	0	1	0	16					
LHLD ADDR	Load HL direct	0	0	1	0	1	0	1	0	16					
<b>Move Register Pair</b>															
XCHG	Exchange DE and HL register pairs	1	1	1	0	1	0	1	1	4					
XTHL	Exchange top of stack and HL	1	1	1	0	0	0	1	1	16					
SPHL	HL to stack pointer	1	1	1	1	1	0	0	1	6					
POHL	HL to program counter	1	1	1	0	1	0	0	1	6					
<b>Input / Output</b>															
IN A	Input	1	1	0	1	1	0	1	1	10					
OUT A	Output	1	1	0	1	0	0	1	1	10					
EI	Enable interrupts	1	1	1	1	1	0	1	1	4					
DI	Disable interrupts	1	1	1	1	0	0	1	1	4					
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					
RST A	Restart	1	1	A	A	A	1	1	1	12					

**Instruction Set (cont)**

Mnemonic(1)	Description	Operation Code(2)										Flags(4)						
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Cycles(3)	Sign	Zero	Parity	Carry				
<b>Miscellaneous</b>																		
CMA	Complement A	0	0	1	0	1	1	1	1	1	1	1	1	1	4			
STC	Set carry	0	0	1	1	0	1	1	1	1	1	1	1	1	4			
CMC	Complement carry	0	0	1	1	1	1	1	1	1	1	1	1	1	4			1/Cy
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	1	1	1	1	1	4	•	•	•
NOP	No operation	0	0	0	0	0	0	0	0	0	0	0	0	0	4			
HLT	Halt	0	1	1	1	0	1	1	0	1	1	0	1	0	5			

**Note:**

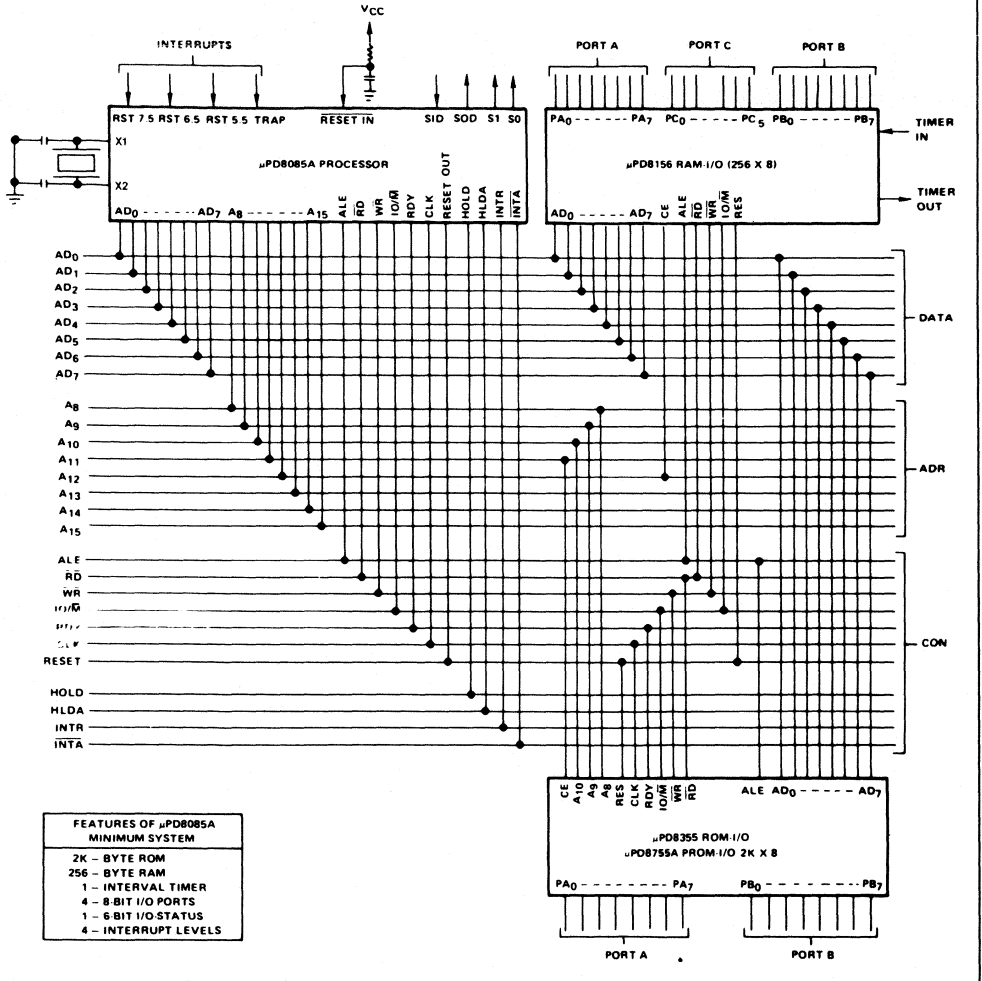
- (1) Operand symbols used
  - A = 8-bit address or expression
  - s = source register
  - d = destination register
  - PSW = Processor status word
  - SP = Stack pointer
  - DB = 8-bit data quantity, expression, or constant, always B<sub>2</sub> of instruction
  - DB = 16-bit data quantity, expression, or constant, always B<sub>2</sub>B<sub>1</sub> of instruction
  - ADDR = 16-bit memory address expression
- (2) ddd or sss = 000-B, 001-C, 010-D, 011-E, 100-H, 101-L, 110-Memory, 111-A
- (3) Two possible cycle times (7/10) indicate instruction cycles dependent on condition flags.
- (4) • = flag affected
  - = flag not affected
  - 0 = flag reset
  - 1 = flag set

## μPD8085A Family Minimum System Configuration

A minimum computer system consisting of a processor, ROM, RAM, and I/O can be built with only 3 40-pin

packs. This system is shown below with its address, data, control buses and I/O ports.

### Three Pack Computer System





## Description

The μPD8086 is a 16-bit microprocessor that has both 8-bit and 16-bit attributes. It has a 16-bit wide physical path to memory for high performance. Its architecture allows higher throughput than the 5 MHz μPD8085A-2.

The maximum operating frequency of the μPD8086 is 5 MHz. The μPD8086-2 is an 8-MHz version.

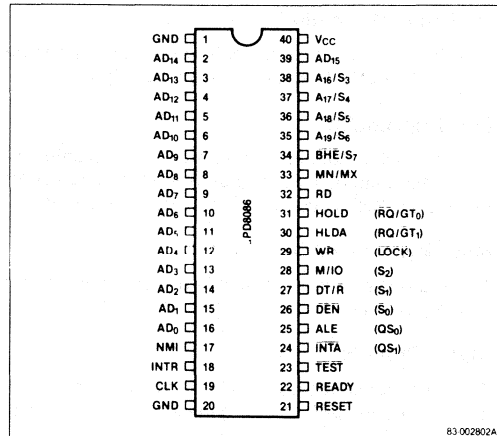
## Features

- Can directly address 1 megabyte of memory
- Fourteen 16-bit registers with symmetrical operations
- Bit, byte, word, and block operations
- 8- and 16-bit signed and unsigned binary or decimal arithmetic operations
- Multiply and divide instructions
- 24 operand addressing modes
- Assembly language compatible with the μPD8080/8085
- Complete family of components for design flexibility

## Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD8086D	40-pin ceramic DIP	5 MHz
μPD8086D	40-pin cerdip	5 MHz
μPD8086D-2	40-pin cerdip	8 MHz

## Pin Configuration



83 002802A

## Pin Identification

No.	Symbol	Function
1, 20	GND	Ground
2-16, 39	AD <sub>0</sub> -AD <sub>15</sub>	Address / data bus
17	NMI	Non-maskable interrupt
18	INTR	Interrupt request
19	CLK	Clock
21	RESET	Reset
22	READY	Ready
23	TEST	Test
24	INTA	Interrupt acknowledge
25	ALE	Address latch enable
26	DEN	Data enable
27	DT / R	Data transmit / receive
28	M / IO	Memory / IO status
29	WR	Write
30	HLDA	Hold acknowledge
31	HOLD	Hold
32	RD	Read
33	MN / MX	Minimum / maximum
34	BHE / S <sub>7</sub>	Bus / high enable
35-38	A <sub>16</sub> -A <sub>19</sub>	Most significant address bits
26-28, 34-38	S <sub>0</sub> -S <sub>7</sub>	Status outputs
24, 25	QS <sub>1</sub> , QS <sub>0</sub>	Queue status
29	LOCK	Lock
30, 31	RQ / GT <sub>0</sub> RQ / GT <sub>1</sub>	Request / grant
40	Vcc	Power supply



**Pin Functions****Ground**

Ground.

**Address/Data Bus**

Multiplexed address (T1) and data (T2, T3, TW, T4) bus. 8-bit peripherals tied to the lower 8 bits use  $A_0$  to condition chip select functions. These lines are three-state during interrupt acknowledge and hold states.

**Non-Maskable Interrupt**

This is an edge-triggered input causing a type 2 interrupt. A look-up table is used by the processor for vectoring information.

**Interrupt Request**

A level-triggered input sampled on the last clock cycle of each instruction. Vectoring is via an interrupt look-up table.  $\overline{INTR}$  can mask in software by resetting the interrupt enable bit.

**Clock**

The clock input is a  $1/3$  duty cycle input basic timing for the processor and bus controller.

**Reset**

This active high signal must be high for 4 clock cycles. When it returns low, the processor restarts execution.

**Ready**

An acknowledgement from memory or I/O that data will be transferred. Synchronization is done by the  $\mu$ PD8284 clock generator.

**Test**

This input is examined by the WAIT instruction, and if low, execution continues. Otherwise the processor waits in an idle state. Synchronized by the processor on the leading edge of CLK.

**Interrupt Acknowledge**

This is a read strobe for reading vectoring information. During T2, T3, and TW of each interrupt acknowledge cycle it is low.

**Address Latch Enable**

This is used in conjunction with the  $\mu$ PD8282/8283 latches to latch the address, during T1 of any bus cycle.

**Data Enable**

This is the output enable for the  $\mu$ PD8282/8287 transceivers. It is active low during each memory and I/O access and  $\overline{INTA}$  cycles.

**Data Transmit/Receive**

Used to control the direction of data flow through the transceivers.

**Memory/I/O Status**

This is used to separate memory access from I/O access.

**Write**

Depending on the state of the  $M/\overline{IO}$  line, the processor is either writing to I/O or memory.

**Hold Acknowledge**

A response to the HOLD input, causing the processor to three-state the local bus. The bus becomes active one cycle after HOLD goes low again.

**Hold**

When another device requests the local bus, driving HOLD high will cause the  $\mu$ PD8086 to issue a HLDA.

**Read**

Depending on the state of the  $M/\overline{IO}$  line, the processor is reading from either memory or I/O.

**Minimum/Maximum**

This input is to tell the processor which mode it is to be used in. This effects some of the pin descriptions.

**Bus/High Enable**

This is used in conjunction with the most significant half of the data bus. Peripheral devices on this half of the bus use BHE to condition chip select functions.

**Most Significant Address Bits**

These are the four most significant address bits for memory operations. Low during I/O operations.

**Status Outputs**

These are the status outputs from the processor. They are used by the  $\mu$ PD8288 to generate bus control signals.

### Queue Status

Used to track the internal μPD8086 instruction queue.

### Lock

This output is set by the LOCK instruction to prevent other system bus masters from gaining control.

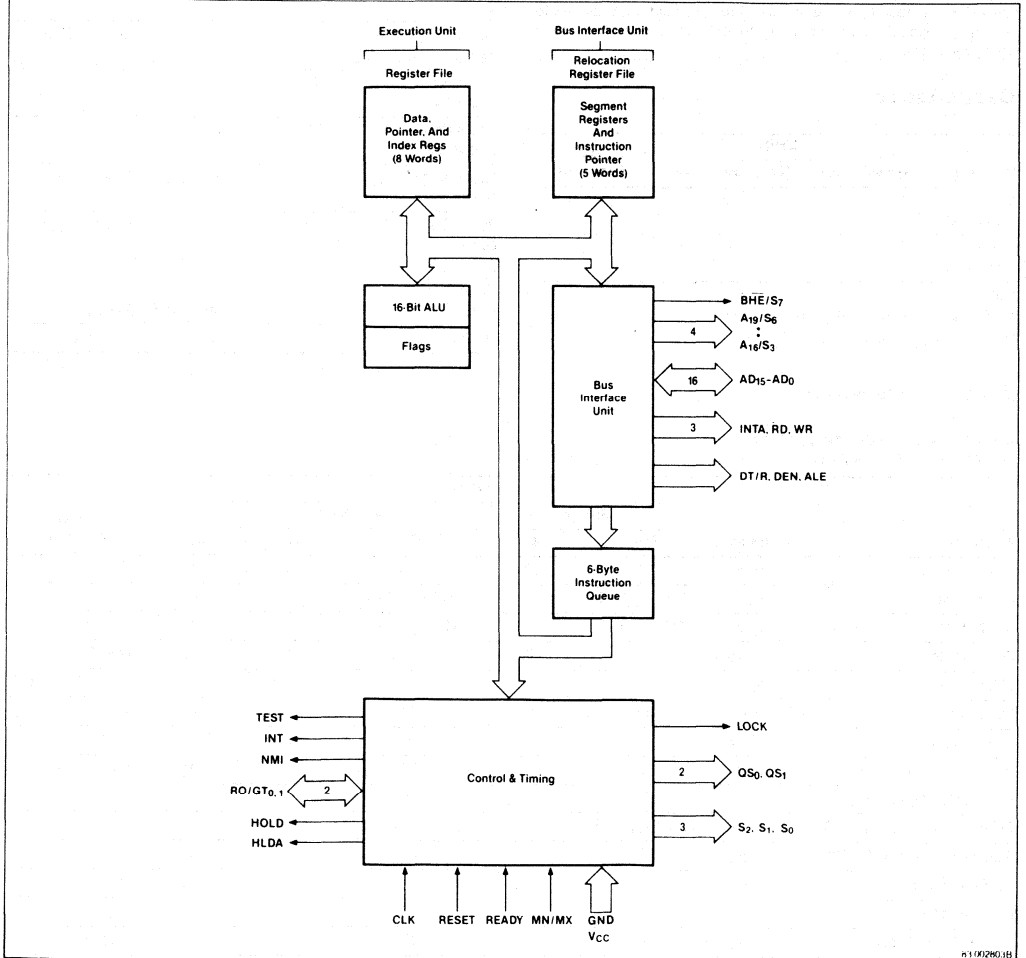
### Request/Grant

Other local bus masters can force the processor to rebase the local bus at the end of the current bus cycle.

### Vcc

This is the +5 V power supply.

### Block Diagram



### Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$	
Power supply voltage, $V_{CC}$	-1.0 V to +7 V
Operating temperature, $T_{OPT}$	0°C to +70°C
Storage temperature, $T_{STG}$	-65°C to +150°C
Power dissipation, $P_D$	2.5 W

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### Capacitance

$f_c = 1.0\text{ MHz}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	$C_I$		15		pF	(Note 1)
I/O capacitance	$C_{IO}$		15		pF	(Note 2)

**Note:**

- (1) All input pins except  $AD_0$ - $AD_{15}$  and  $\overline{RQ}/\overline{GT}$ .
- (2) Only input pins  $AD_0$ - $AD_{15}$  and  $\overline{RQ}/\overline{GT}$ .

### AC Characteristics

#### Minimum Complexity System

μPD8086:  $T_A = 0^\circ\text{C}$  to +70°C,  $V_{CC} = 5\text{ V} \pm 10\%$

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8086		μPD8086-2			
		Min	Max	Min	Max		
CLK cycle period	$t_{CLCL}$	200	500	125	500	ns	
CLK low time	$t_{CLCH}$	$(2/3 t_{CLCL}) - 15$		$(2/3 t_{CLCL}) - 15$		ns	
CLK high time	$t_{CHCL}$	$(1/3 t_{CLCL}) + 2$		$(1/3 t_{CLCL}) + 2$		ns	
CLK rise time	$t_{CH1CH2}$		10		10	ns	From 1.0 V to 3.5 V
CLK fall time	$t_{CL2CL1}$		10		10	ns	From 3.5 V to 1.0 V
Data in setup time	$t_{DVCL}$	30		20		ns	
Data in hold time	$t_{CLDX}$	10		10		ns	
READY setup time into μPD8284	$t_{RVCL}$	35		35		ns	(Notes 1 & 2)
READY hold time into μPD8284	$t_{CLR1X}$	0		0		ns	(Notes 1 & 2)
READY setup time into μPD8086	$t_{RYHCH}$	$(2/3 t_{CLCL}) - 15$		$(2/3 t_{CLCL}) - 15$		ns	
READY hold time into μPD8086	$t_{CHRYX}$	30		20		ns	
READY inactive to CLK	$t_{RYLCL}$	-8		-8		ns	(Note 3)
HOLD setup time	$t_{HVCH}$	35		20		ns	
INTR, NMI, TEST setup time	$t_{INVCH}$	30		15		ns	(Note 2)
Input rise time	$t_{ILIH}$		20		20	ns	From 0.8 V to 2.0 V
Input fall time	$t_{IHIL}$		12		12	ns	From 2.0 V to 0.8 V

### DC Characteristics

$T_A = 0^\circ\text{C}$  to +70°C,  $V_{CC} = +5\text{ V} \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage, low	$V_{IL}$	-0.5		+0.8	V	
Input voltage, high	$V_{IH}$	2		$V_{CC} + 0.5$	V	
Output voltage, low	$V_{OL}$			+0.45	V	$I_{OL} = 2.5\text{ mA}$
Output voltage, high	$V_{OH}$	2.4		V	V	$I_{OH} = -400\text{ }\mu\text{A}$
Input clock voltage, low	$V_{CL}$	-0.5		+0.6	V	
Output clock voltage, high	$V_{CH}$	3.9		$V_{CC} + 1.0$	V	
Input leakage current	$I_{LI}$			$\pm 10$	$\mu\text{A}$	$0\text{ V} < V_{IN} < V_{CC}$
Output leakage current	$I_{LO}$			$\pm 10$	$\mu\text{A}$	$0.45\text{ V} \leq V_{OUT} \leq V_{CC}$
Power supply current	$I_{CC}$					
μPD8086 /				340	mA	$T_A = 25^\circ\text{C}$
μPD8086-2				350	mA	$T_A = 25^\circ\text{C}$

## AC Characteristics (cont)

### Timing Responses

μPD8086: T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5V ± 10%

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8086		μPD8086-2			
		Min	Max	Min	Max		
Address valid delay	t <sub>CLAV</sub>	10	110	10	60	ns	(Note 4)
Address hold time	t <sub>CLAX</sub>	10		10		ns	(Note 4)
Address float delay	t <sub>CLAZ</sub>	t <sub>CLAX</sub>	80	t <sub>CLAX</sub>	50	ns	(Note 4)
ALE width	t <sub>LHLL</sub>	t <sub>CLCH</sub> - 20		t <sub>CLCH</sub> - 10		ns	(Note 4)
ALE active delay	t <sub>CLLH</sub>		80		50	ns	(Note 4)
ALE inactive delay	t <sub>CHLL</sub>		85		55	ns	(Note 4)
Address hold time to ALE inactive	t <sub>LLAX</sub>	t <sub>CHCL</sub> - 10		t <sub>CHCL</sub> - 10		ns	(Note 4)
Data valid delay	t <sub>CLDV</sub>	10	110	10	60	ns	(Note 4)
Data hold time	t <sub>CHDX</sub>	10		10		ns	(Note 4)
Data hold time after WR	t <sub>WHDX</sub>	t <sub>CLCH</sub> - 30		t <sub>CLCH</sub> - 30		ns	(Note 4)
Control active delay 1	t <sub>CVCTV</sub>	10	110	10	70	ns	(Note 4)
Control active delay 2	t <sub>CHCTV</sub>	10	110	10	60	ns	(Note 4)
Control active delay	t <sub>CVCTX</sub>	10	110	10	70	ns	(Note 4)
Address float to READ active	t <sub>AZRL</sub>	0		0		ns	(Note 4)
RD active delay	t <sub>CLRL</sub>	10	165	10	80	ns	(Note 4)
RD inactive delay	t <sub>CLRH</sub>	10	150	10	80	ns	(Note 4)
RD inactive to next address active	t <sub>RHAV</sub>	t <sub>CLCL</sub> - 45		t <sub>CLCL</sub> - 40		ns	(Note 4)
HLDA valid delay	t <sub>CLHAV</sub>	10	160	10	100	ns	(Note 4)
RD width	t <sub>RLRH</sub>	2t <sub>CLCL</sub> - 75		2t <sub>CLCL</sub> - 50		ns	(Note 4)
WR width	t <sub>WLWH</sub>	2t <sub>CLCL</sub> - 60		2t <sub>CLCL</sub> - 40		ns	(Note 4)
Address valid to ALE low	t <sub>AVAL</sub>	t <sub>CLCH</sub> - 60		t <sub>CLCH</sub> - 40		ns	(Note 4)
Output rise time	t <sub>OLOH</sub>		20		20	ns	From 0.8V to 2.0V
Output fall time	t <sub>O HOL</sub>		12		12	ns	From 2.0V to 0.8V

#### Note:

- (1) Signal at μPD8284 shown for reference only.
- (2) Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
- (3) Applies only to T2 state. (8 ns into T3)
- (4) C<sub>L</sub> = 20-100 pF for all μPD8086 outputs (in addition to μPD8086 self-load).

AC Characteristics (cont)

Maximum Mode System with μPB8288 Bus Controller

μPD8086: T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5V ± 10%

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8086		μPD8086-2			
		Min	Max	Min	Max		
CLK cycle period	t <sub>CLCL</sub>	200	500	125	500	ns	
CLK low time	t <sub>CLCH</sub>	(2/3 t <sub>CLCL</sub> ) - 15		(2/3 t <sub>CLCL</sub> ) - 15		ns	
CLK high time	t <sub>CHCL</sub>	(1/3 t <sub>CLCL</sub> ) + 2		(1/3 t <sub>CLCL</sub> ) + 2		ns	
CLK rise time	t <sub>CH1CH2</sub>		10		10	ns	From 1.0 V to 3.5 V
CLK fall time	t <sub>CL2CL1</sub>		10		10	ns	From 3.5 V to 1.0 V
Data in setup time	t <sub>DVCL</sub>	30		20		ns	
Data in hold time	t <sub>CLDX</sub>	10		10		ns	
READY setup time into μPD8284	t <sub>R1VCL</sub>	35		35		ns	(Notes 1 & 2)
READY hold time into μPD8284	t <sub>CLR1X</sub>	0		0		ns	(Notes 1 & 2)
READY setup time into μPD8086	t <sub>RYHCH</sub>	(2/3 t <sub>CLCL</sub> ) - 15		(2/3 t <sub>CLCL</sub> ) - 15		ns	
READY hold time into μPD8086	t <sub>CHRYX</sub>	30		20		ns	
READY inactive to CLK	t <sub>RYLCL</sub>	- 8		- 8		ns	(Note 5)
INTR, NMI, TEST setup time	t <sub>INVCH</sub>	30		15		ns	(Note 2)
RQ / GT setup time	t <sub>GVCH</sub>	30		15		ns	
RQ hold time into μPD8086	t <sub>CHGX</sub>	40		30		ns	
Input rise time	t <sub>L1IH</sub>		20		20	ns	From 0.8 V to 2.0 V
Input fall time	t <sub>L1IL</sub>		12		12	ns	From 2.0 V to 0.8 V

## AC Characteristics (cont)

### Timing Responses

μPD8086: T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5 V ± 10%

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8086		μPD8086-2			
		Min	Max	Min	Max		
Command active delay	t <sub>CLML</sub>	10	35	10	35	ns	(Notes 1 & 4)
Command inactive delay	t <sub>CLMH</sub>	10	35	10	35	ns	(Notes 1 & 4)
READY active to status passive	t <sub>RYHSH</sub>		110		65	ns	(Notes 3 & 4)
Status active delay	t <sub>CHSV</sub>	10	110	10	60	ns	(Note 4)
Status inactive delay	t <sub>CLSH</sub>	10	130	10	70	ns	(Note 4)
Address valid delay	t <sub>CLAV</sub>	10	110	10	60	ns	(Note 4)
Address hold time	t <sub>CLAX</sub>	10		10		ns	(Note 4)
Address float delay	t <sub>CLAZ</sub>	t <sub>CLAX</sub>	80	t <sub>CLAX</sub>	50	ns	(Note 4)
Status valid to ALE high	t <sub>SVLH</sub>		15		15	ns	(Notes 1 & 4)
Status valid to MCE high	t <sub>SVMCH</sub>		15		15	ns	(Notes 1 & 4)
CLK low to ALE valid	t <sub>CLLH</sub>		15		15	ns	(Notes 1 & 4)
CLK low to MCE high	t <sub>CLMCH</sub>		15		15	ns	(Notes 1 & 4)
ALE inactive delay	t <sub>CHLL</sub>		15		15	ns	(Notes 1 & 4)
MCE inactive delay	t <sub>CLMCL</sub>		15		15	ns	(Notes 1 & 4)
Data valid delay	t <sub>CLDV</sub>	10	110	10	60	ns	(Note 4)
Data hold time	t <sub>CHDX</sub>	10		10		ns	(Note 4)
Control active delay	t <sub>CVNV</sub>	5	45	5	45	ns	(Notes 1 & 4)
Control inactive delay	t <sub>CVNX</sub>	10	45	10	45	ns	(Notes 1 & 4)
Address float to READ active	t <sub>AZRL</sub>	0		0		ns	(Note 4)
RD active delay	t <sub>CLRL</sub>	10	165	10	100	ns	(Note 4)
RD inactive delay	t <sub>CLRH</sub>	10	150	10	80	ns	(Note 4)
RD inactive to next address active	t <sub>RHAV</sub>	t <sub>CLCL</sub> - 45		t <sub>CLCL</sub> - 40		ns	(Note 4)
Direction control active delay	t <sub>CHDTL</sub>		50		50	ns	(Notes 1 & 4)
Direction control inactive delay	t <sub>CHDTH</sub>		30		30	ns	(Notes 1 & 4)
GT active delay	t <sub>CLGL</sub>	0	85	0	50	ns	(Note 4)
GT inactive delay	t <sub>CLGH</sub>	0	85	0	50	ns	(Note 4)
RD width	t <sub>RLRH</sub>	2t <sub>CLCL</sub> - 50		2t <sub>CLCL</sub> - 50		ns	(Note 4)
Output rise time	t <sub>OLOH</sub>		20		20	ns	From 0.8 V to 2.0 V
Output fall time	t <sub>OHOL</sub>		12		12	ns	From 2.0 V to 0.8 V

#### Note:

- (1) Signal at μPB8284 or μPB8288 shown for reference only.
- (2) Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
- (3) Applies only to T3 and wait states.
- (4) C<sub>L</sub> = 20-100 pF for all μPD8086 outputs (in addition to μPD8086 self-load).
- (5) Applies only to T2 state. (8 ns into T3).





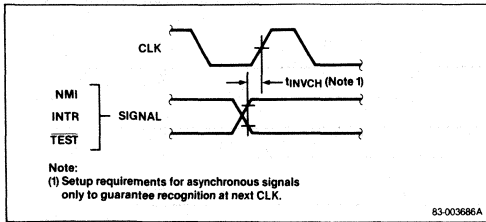




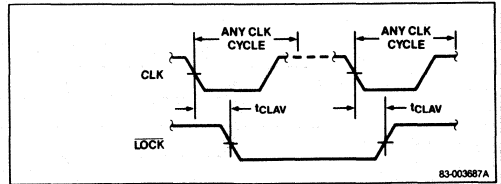


Timing Waveforms

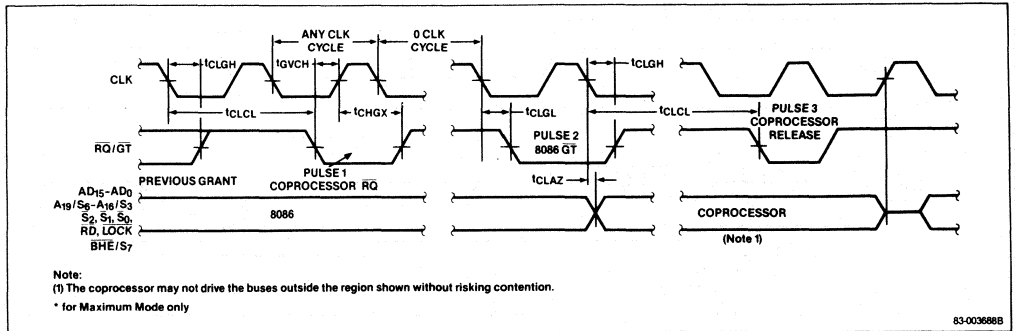
Asynchronous Signal Recognition



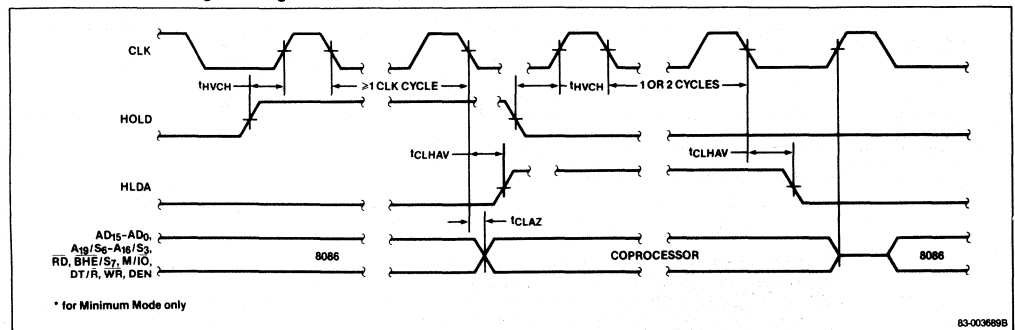
Bus Lock Signal Timing



Request/Grant Sequence Timing\*



Hold/Hold Acknowledge Timing\*



## Description

The μPD8088 and μPD8088-2 are powerful 8-bit microprocessors that are software-compatible with the μPD8086. They have the same bus interface signals as μPD8085A, allowing them to interface directly with multiplexed bus peripherals. Both having a 20-bit address space which can be divided into four segments of up to 64K bytes each.

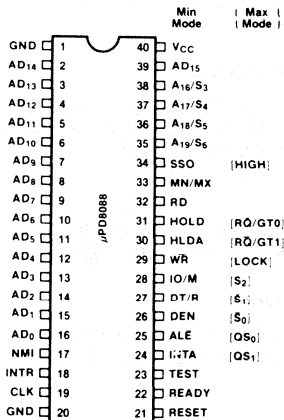
## Features

- 8-bit data bus interface
- 16-bit internal architecture
- Addresses 1 Mbyte of memory
- Software-compatible with the 8086
- Provides byte, word, and block operations
- Performs 8- and 16-bit signed and unsigned arithmetic in binary and decimal
- Multiply and divide instruction
- Directly interfaces to 8155, 8355, and 8755A multiplexed peripherals

## Ordering Information

Part Number	Package Type	Max Frequency of Operation
PD8088D	40-pin ceramic DIP	5 MHz
PD8088D-2	40-pin ceramic DIP	8 MHz

## Pin Configuration



## Pin Identification

No.	Symbol	Function
1, 20	GND	Ground
2-8, 35-39	A <sub>19</sub> -A <sub>8</sub>	Most significant address bits
9-16	AD <sub>7</sub> -AD <sub>0</sub>	Address/data bus
17	NMI	Non-maskable interrupt
18	INTR	Interrupt request
19	CLK	Clock
21	RESET	Reset
22	READY	Ready
23	TEST	Test
24	INTA	Interrupt acknowledge
25	ALE	Address latch enable
24, 25	QS <sub>1</sub> , QS <sub>0</sub>	Queue status
26	DEN	Data enable
27	DT/R	Data transmit/receive
28	IO/M	IO status/memory
29	WR	Write
29	LOCK	Lock
30	HLDA	Hold acknowledge
31	HOLD	Hold
30, 31	RQ/GT <sub>0</sub> RQ/GT <sub>1</sub>	Request/grant
32	RD	Read
33	MN/MX	Minimum/maximum
34	SS0	Status line
26-28	S <sub>0</sub> -S <sub>2</sub>	Status outputs
35-38	S <sub>3</sub> -S <sub>6</sub>	Status outputs
40	VCC	Power supply

83 990259A

**Pin Function****Ground**

Ground.

**Most Significant Address Bits**

Most significant bits for memory operations.

**Address/Data Bus**

Multiplexed address and data bus. 8-bit peripherals tied to these bits use  $A_0$  to condition chip select functions. These lines are three-state during interrupt acknowledge and hold states.

**Non-Maskable Interrupt**

This edge-triggered input causes a type 2 interrupt. The processor uses a look-up table for vectoring information.

**Interrupt Request**

This is a level-triggered interrupt sampled on the last clock cycle of each instruction. A look-up table is used for vectoring. INTR can be masked in software by resetting the interrupt enable bit.

**Clock**

The clock input is a  $1/3$  duty cycle input providing basic timing for the processor and bus controller.

**Reset**

This active high signal must be high for 4 clock cycles. When it returns low, the processor restarts execution.

**Ready**

An acknowledgement from memory or I/O that data will be transferred. Synchronization is done by the  $\mu$ PD8284 clock generator.

**Test**

This input is examined by the "WAIT" instruction, and if low, execution continues. Otherwise the processor waits in an "Idle" state. Synchronized by the processor on the leading edge of CLK.

**Interrupt Acknowledge**

This is a read strobe for reading vectoring information. During T2, T3, and TW of each interrupt acknowledge cycle it is low.

**Address Latch Enable**

This is used in conjunction with the  $\mu$ PD8282/8283 latches to latch the address, during T1 of any bus cycle.

**Queue Status**

(Max mode) tracks the internal  $\mu$ PD8088 instruction queue.

**Data Enable**

This is the output enable for the  $\mu$ PD8286/8287 transceivers. It is active low during memory and I/O access and  $\overline{INTA}$  cycles.

**Data Transmit/Receive**

Controls the direction of data flow through the transceivers.

**IO Status/ Memory**

Separates memory access from I/O access.

**Write**

Depending on the state of the  $IO/\overline{M}$  line, the processor is either writing to I/O or memory.

**Lock**

(Max mode) this output is set by the "LOCK" instruction to prevent other system bus masters from gaining control.

**Hold Acknowledge**

A response to the HOLD input, causing the processor to three-state the local bus. The bus becomes active one cycle after HOLD returns low.

**Hold**

When another device requests the local bus, HOLD is driven high, causing the  $\mu$ PD8088 to issue a HLDA.

**Request/Grant**

(Max mode) other local bus masters can force the processor to rebase the local bus at the end of the current bus cycle.

**Read**

Depending on the state of the  $IO/\overline{M}$  line, the processor is reading from either memory or I/O.

## Minimum/Maximum

This input tells the processor in which mode it is to be used. This affects some of the pin descriptions.

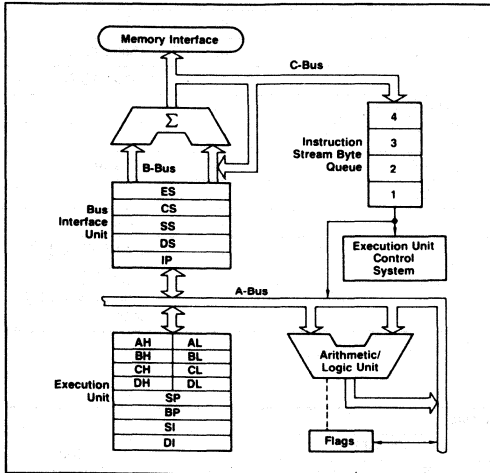
## Status Outputs

(Max mode) These are the status outputs from the processor. They are used by the μPD8288 to generate bus control signals.

## V<sub>CC</sub>

5 V power supply input.

## Block Diagram



## Absolute Maximum Ratings

T<sub>A</sub> = 25°C, Tentative

Power supply voltage, V <sub>DD</sub>	-0.5 V to +7 V
Input voltage, V <sub>I</sub>	-0.5 V to +7 V
Output voltage, V <sub>O</sub>	-0.5 V to +7 V
Operating temperature, T <sub>OPT</sub>	0°C to +70°C
Storage temperature, T <sub>STG</sub>	-65°C to +150°C
Power dissipation, P <sub>D</sub>	2.5 W

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## DC Characteristics

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = +5 V ± 10%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage low	V <sub>IL</sub>	-0.5		+0.8	V	
Input voltage high	V <sub>IH</sub>	2.0		V <sub>CC</sub> +0.5	V	
Output voltage low	V <sub>OL</sub>			+0.45	V	I <sub>O1</sub> = 2.0 mA
Output voltage high	V <sub>OH</sub>	2.4			V	I <sub>OH</sub> = -400 μA
Input clock voltage low	V <sub>CL</sub>	-0.5		+0.6	V	
Input clock voltage high	V <sub>CH</sub>	3.9		V <sub>CC</sub> +1.0	V	
Input leakage current	I <sub>LI</sub>		±10		μA	0 V < V <sub>I</sub> < V <sub>CC</sub>
Output leakage current	I <sub>LO</sub>		±10		μA	0.45 V ≤ V <sub>O</sub> ≤ V <sub>CC</sub>
Power supply current	I <sub>CC</sub>					
μPD8088 /				340	mA	T <sub>A</sub> = 25°C
μPD8088-2				350	mA	T <sub>A</sub> = 25°C

## Capacitance

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	C <sub>I</sub>			15	pF	(Note 1)
I/O capacitance	C <sub>I/O</sub>			15	pF	(Note 2)

### Note:

- (1) All input pins except AD<sub>0</sub>-AD<sub>7</sub> and RQ/GT.
- (2) Only input pins AD<sub>0</sub>-AD<sub>7</sub> and RQ/GT.

AC Characteristics

Minimum Complexity Systems

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5 V ± 10%

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8088		μPD8088-2			
		Min	Max	Min	Max		
CLK cycle period	t <sub>CLCL</sub>	200	500	125	500	ns	
CLK low time	t <sub>CLCH</sub>	( <sup>2</sup> / <sub>3</sub> t <sub>CLCL</sub> ) - 15		( <sup>2</sup> / <sub>3</sub> t <sub>CLCL</sub> ) - 15		ns	
CLK high time	t <sub>CHCL</sub>	( <sup>1</sup> / <sub>3</sub> t <sub>CLCL</sub> ) + 2		( <sup>1</sup> / <sub>3</sub> t <sub>CLCL</sub> ) + 2		ns	
CLK rise time	t <sub>CHCH2</sub>		10		10	ns	From 1.0 V to 3.5 V
CLK fall time	t <sub>CL2CL1</sub>		10		10	ns	From 3.5 V to 1.0 V
Data in setup time	t <sub>DVCL</sub>	30		20		ns	
Data in hold time	t <sub>CLDX</sub>	10		10		ns	
READY setup time into μPD8284	t <sub>RVCL</sub>	35		35		ns	(Notes 1 & 2)
READY hold time into μPD8284	t <sub>CLR1X</sub>	0		0		ns	(Notes 1 & 2)
READY setup time into μPD8088	t <sub>RVCH</sub>	( <sup>2</sup> / <sub>3</sub> t <sub>CLCL</sub> ) - 15		( <sup>2</sup> / <sub>3</sub> t <sub>CLCL</sub> ) - 15		ns	
READY hold time into μPD8088	t <sub>CHRYX</sub>	30		20		ns	
READY inactive to CLK	t <sub>RYLCL</sub>	- 8		- 8		ns	(Note 3)
HOLD setup time	t <sub>HVCH</sub>	35		20		ns	
INTR, NMI, TEST setup time	t <sub>INVCH</sub>	30		15		ns	(Note 2)
Input rise time	t <sub>ILIH</sub>		20		20	ns	From 0.8 V to 2.0 V, except clock
Input fall time	t <sub>IHL</sub>		12		12	ns	From 2.0 V to 0.8 V, except clock

Timing Responses

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5 V ± 10%

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8088		μPD8088-2			
		Min	Max	Min	Max		
Address valid delay	t <sub>CLAV</sub>	10	110	10	60	ns	(Note 4)
Address hold time	t <sub>CLAX</sub>	10		10		ns	(Note 4)
Address float delay	t <sub>CLAZ</sub>	t <sub>CLAX</sub>	80	t <sub>CLAX</sub>	50	ns	(Note 4)
ALE width	t <sub>LHLL</sub>	t <sub>CLCH</sub> - 20		t <sub>CLCH</sub> - 10		ns	(Note 4)
ALE active delay	t <sub>CLLH</sub>		80		50	ns	(Note 4)
ALE inactive delay	t <sub>CHLL</sub>		85		55	ns	(Note 4)
Address hold time to ALE inactive	t <sub>LAX</sub>	t <sub>CHCL</sub> - 10		t <sub>CHCL</sub> - 10		ns	(Note 4)
Data valid delay	t <sub>CLDV</sub>	10	110	10	60	ns	(Note 4)
Data hold time	t <sub>CHDX</sub>	10		10		ns	(Note 4)
Data hold time after WR	t <sub>WHDX</sub>	t <sub>CLCH</sub> - 30		t <sub>CLCH</sub> - 30		ns	(Note 4)
Control active delay 1	t <sub>CVCTV</sub>	10	110	10	70	ns	(Note 4)
Control active delay 2	t <sub>CHCTV</sub>	10	110	10	70	ns	(Note 4)
Control inactive delay	t <sub>CVCTX</sub>	10	110	10	70	ns	(Note 4)
Address float to READ active	t <sub>AZRL</sub>	0		0		ns	(Note 4)
$\overline{RD}$ active delay	t <sub>CLRL</sub>	10	165	10	80	ns	(Note 4)

## AC Characteristics (cont)

### Timing Responses (cont)

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5 V ± 10%

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8088		μPD8088-2			
		Min	Max	Min	Max		
R <sub>D</sub> inactive delay	t <sub>CLR<sub>H</sub></sub>	10	150	10	80	ns	(Note 4)
R <sub>D</sub> inactive to next address active	t <sub>RH<sub>AV</sub></sub>	t <sub>CLCL</sub> - 45		t <sub>CLCL</sub> - 40		ns	(Note 4)
HLDA valid delay	t <sub>CLH<sub>AV</sub></sub>	10	160	10	100	ns	(Note 4)
R <sub>D</sub> width	t <sub>RL<sub>RH</sub></sub>	2t <sub>CLCL</sub> - 75		2t <sub>CLCL</sub> - 50		ns	(Note 4)
WR width	t <sub>WL<sub>WH</sub></sub>	2t <sub>CLCL</sub> - 60		2t <sub>CLCL</sub> - 40		ns	(Note 4)
Address valid to ALE low	t <sub>AV<sub>AL</sub></sub>	t <sub>CLCH</sub> - 60		t <sub>CLCH</sub> - 40		ns	(Note 4)
Output rise time	t <sub>OL<sub>OH</sub></sub>		20		20	ns	From 0.8 V to 2.0 V
Output fall time	t <sub>OH<sub>OL</sub></sub>		12		12	ns	From 2.0 V to 0.8 V

#### Note:

- (1) Signal at μPD8284 shown for reference only.
- (2) Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
- (3) Applies only to T2 state. (8 ns into T3)
- (4) C<sub>L</sub> = 20-100 pF for all μPD8088 outputs (in addition to μPD8088 self-load).

### Maximum Mode System with μPB8288 Bus Controller

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5 V ± 10%

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8088		μPD8088-2			
		Min	Max	Min	Max		
CLK cycle period	t <sub>CLCL</sub>	200	500	125	500	ns	
CLK low time	t <sub>CLCH</sub>	( <sup>2</sup> / <sub>3</sub> t <sub>CLCL</sub> ) - 15		( <sup>2</sup> / <sub>3</sub> t <sub>CLCL</sub> ) - 15		ns	
CLK high time	t <sub>CHCL</sub>	( <sup>1</sup> / <sub>3</sub> t <sub>CLCL</sub> ) + 2		( <sup>1</sup> / <sub>3</sub> t <sub>CLCL</sub> ) + 2		ns	
CLK rise time	t <sub>CH<sub>CH</sub>2</sub>		10		10	ns	From 1.0 V to 3.5 V
CLK fall time	t <sub>CL<sub>2</sub>CL1</sub>		10		10	ns	From 3.5 V to 1.0 V
Data in setup time	t <sub>DV<sub>CL</sub></sub>	30		20		ns	
Data in hold time	t <sub>CL<sub>D</sub>X</sub>	10		10		ns	
READY setup time into μPD8284	t <sub>R<sub>IV</sub>CL</sub>	35		35		ns	(Notes 1 & 2)
READY hold time into μPD8284	t <sub>CL<sub>R</sub>IX</sub>	0		0		ns	(Notes 1 & 2)
READY setup time into μPD8088	t <sub>R<sub>Y</sub>HCH</sub>	( <sup>2</sup> / <sub>3</sub> t <sub>CLCL</sub> ) - 15		( <sup>2</sup> / <sub>3</sub> t <sub>CLCL</sub> ) - 15		ns	
READY hold time into μPD8088	t <sub>CH<sub>R</sub>YX</sub>	30		20		ns	
READY inactive to CLK	t <sub>RY<sub>L</sub>CL</sub>	-8		-8		ns	(Note 5)
INTR, NMI, TEST setup time	t <sub>INV<sub>CH</sub></sub>	30		15		ns	(Note 2)
RQ / GT setup time	t <sub>GV<sub>CH</sub></sub>	30		15		ns	
RQ hold time into μPD8088	t <sub>CH<sub>G</sub>X</sub>	40		30		ns	
Input rise time	t <sub>IL<sub>I</sub>H</sub>		20		20	ns	From 0.8 V to 2.0 V, except clock
Input fall time	t <sub>I<sub>H</sub>L</sub>		12		12	ns	From 2.0 V to 0.8 V, except clock



**AC Characteristics (cont)**

**Timing Responses**

μPD8088: T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5 V ± 10%

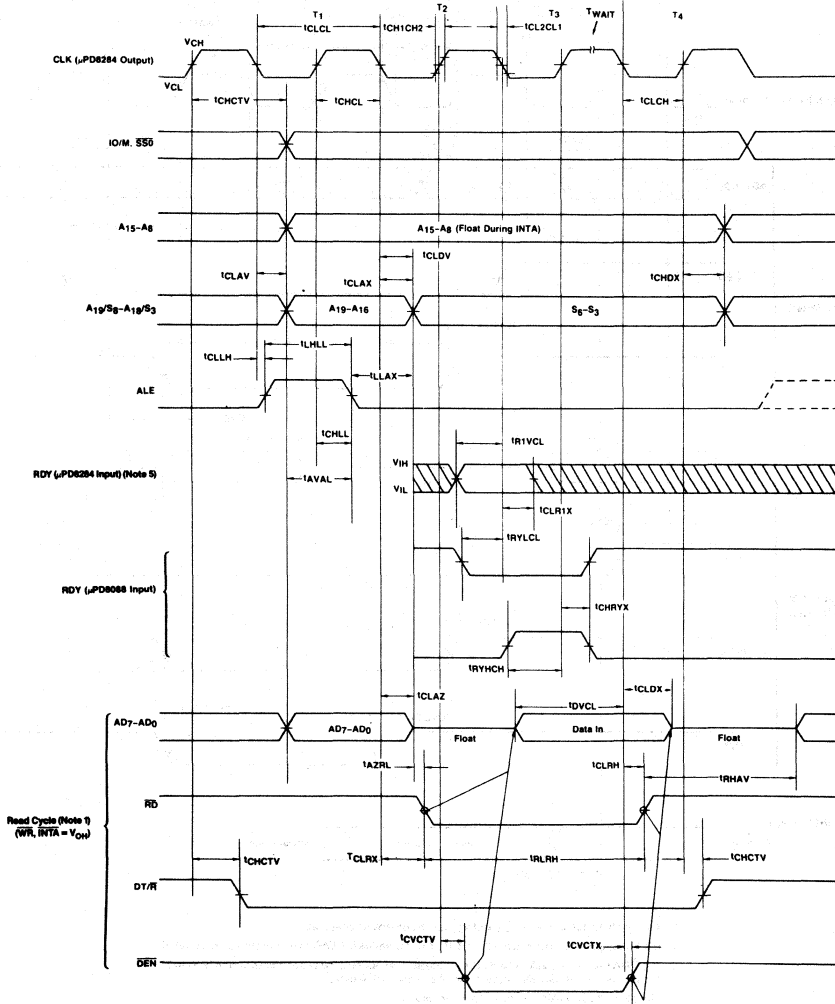
Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8088		μPD8088-2			
		Min	Max	Min	Max		
Command active delay	t <sub>CLML</sub>	10	35	10	35	ns	(Notes 1 & 4)
Command inactive delay	t <sub>CLMH</sub>	10	35	10	35	ns	(Notes 1 & 4)
READY active to status passive	t <sub>RYHSH</sub>		110		65	ns	(Notes 3 & 4)
Status active delay	t <sub>CHSV</sub>	10	110	10	60	ns	(Note 4)
Status inactive delay	t <sub>CLSH</sub>	10	130	10	70	ns	(Note 4)
Address valid delay	t <sub>CLAV</sub>	10	110	10	60	ns	(Note 4)
Address hold time	t <sub>CLAX</sub>	10		10		ns	(Note 4)
Address float delay	t <sub>CLAZ</sub>	t <sub>CLAX</sub>	80	t <sub>CLAX</sub>	50	ns	(Note 4)
Status valid to ALE high	t <sub>SVLH</sub>		15		15	ns	(Notes 1 & 4)
Status valid to MCE high	t <sub>SVMCH</sub>		15		15	ns	(Notes 1 & 4)
CLK low to ALE valid	t <sub>CLLH</sub>		15		15	ns	(Notes 1 & 4)
CLK low to MCE high	t <sub>CLMCH</sub>		15		15	ns	(Notes 1 & 4)
ALE inactive delay	t <sub>CHLL</sub>		15		15	ns	(Notes 1 & 4)
MCE inactive delay	t <sub>CLMCL</sub>		15		15	ns	(Notes 1 & 4)
Data valid delay	t <sub>CLDV</sub>	10	110	10	60	ns	(Note 4)
Data hold time	t <sub>CHDX</sub>	10		10		ns	(Note 4)
Control active delay	t <sub>CVNV</sub>	5	45	5	45	ns	(Notes 1 & 4)
Control inactive delay	t <sub>CVNX</sub>	10	45	10	45	ns	(Notes 1 & 4)
Address float to READ active	t <sub>AZRL</sub>	0		0		ns	(Note 4)
RD active delay	t <sub>CLRH</sub>	10	165	10	100	ns	(Note 4)
RD inactive delay	t <sub>CLRH</sub>	10	150	10	80	ns	(Note 4)
RD inactive to next address active	t <sub>RHAV</sub>	t <sub>CLCL</sub> - 45		t <sub>CLCL</sub> - 40		ns	(Note 4)
Direction control active delay	t <sub>CHDTL</sub>		50		50	ns	(Notes 1 & 4)
Direction control inactive delay	t <sub>CHDTH</sub>		30		30	ns	(Notes 1 & 4)
GT active delay	t <sub>CLGL</sub>	0	85	0	50	ns	(Note 4)
GT inactive delay	t <sub>CLGH</sub>	0	85	0	50	ns	(Note 4)
RD width	t <sub>RLRH</sub>	2t <sub>CLCL</sub> - 75		2t <sub>CLCL</sub> - 50		ns	(Note 4)
Output rise time	t <sub>OLOH</sub>		20		20	ns	From 0.8 V to 2.0 V
Output fall time	t <sub>OHOH</sub>		12		12	ns	From 2.0 V to 0.8 V

**Note:**

- (1) Signal at μPB8284 or μPB8288 shown for reference only.
- (2) Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
- (3) Applies only to T3 and wait states.
- (4) C<sub>L</sub> = 20–100 pF for all μPD8088 outputs (in addition to μPD8088 self-load).
- (5) Applies only to T2 state. (β ns into T3).

## Timing Waveforms

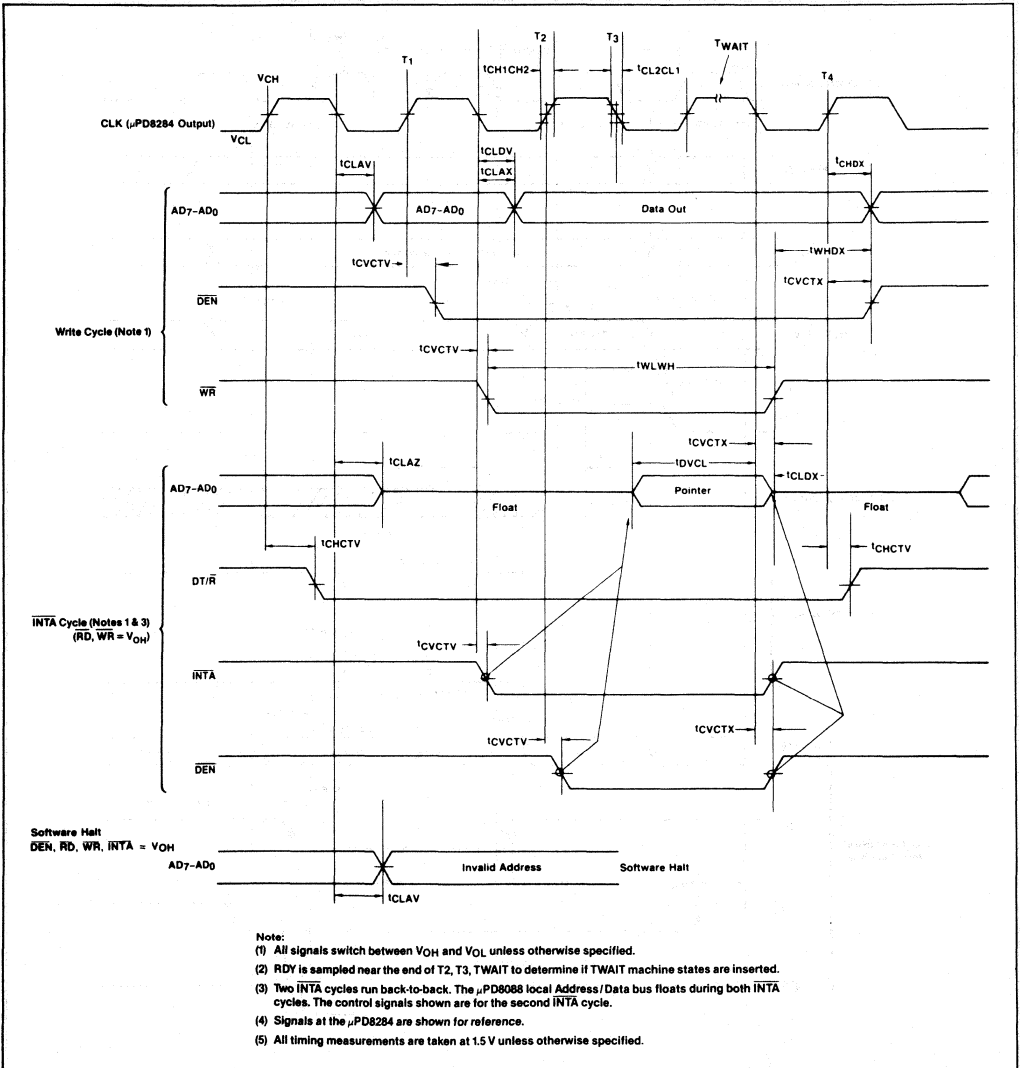
### Minimum Complexity Systems (Note 5)



- Note:**
- (1) All signals switch between  $V_{OH}$  and  $V_{OL}$  unless otherwise specified.
  - (2) RDY is sampled near the end of T2, T3, TWAIT to determine if TWAIT machine states are inserted.
  - (3) Two INTA cycles run back-to-back. The μPD8088 local Address/Data bus floats during both INTA cycles. The control signals shown are for the second INTA cycle.
  - (4) Signals at the μPD8284 are shown for reference.
  - (5) All timing measurements are taken at 1.5V unless otherwise specified.

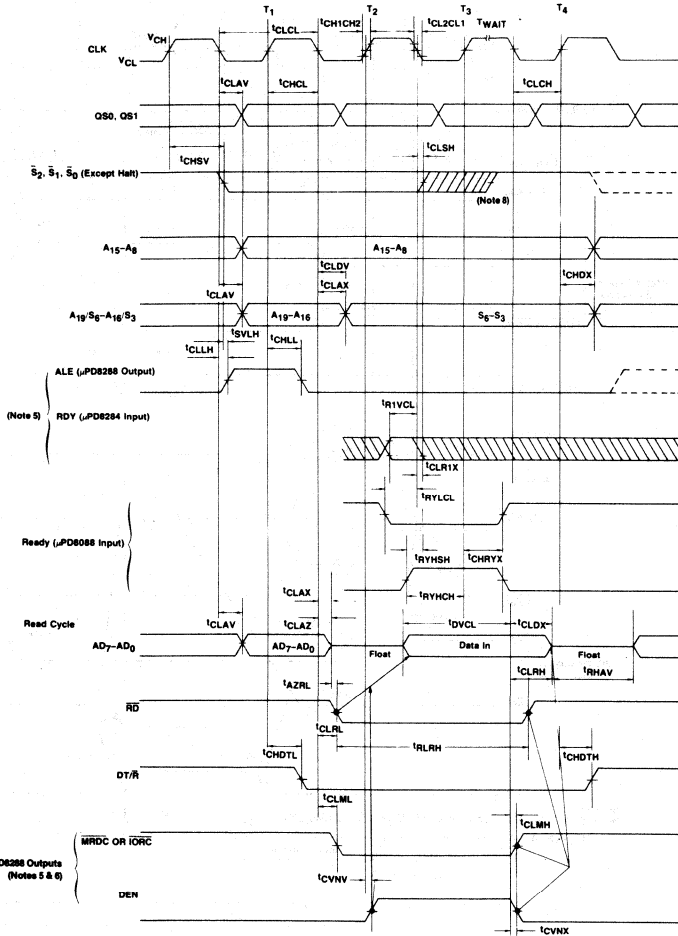
Timing Waveforms (cont)

Minimum Complexity Systems (Note 5)



## Timing Waveforms (cont)

### Maximum Mode System Bus Timing Using μPB8288 Bus Controller (Note 7)

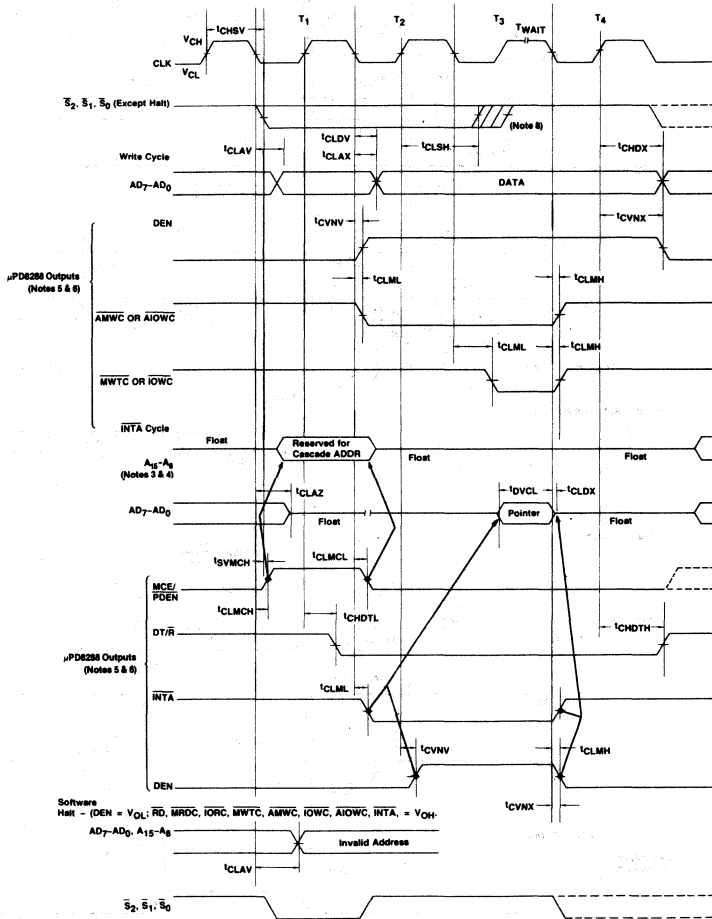


**Note:**

- (1) All signals switch between  $V_{OH}$  and  $V_{OL}$  unless otherwise specified.
- (2) RDY is sampled near the end of T2, T3, TWAIT to determine if TWAIT machine states are inserted.
- (3) The cascade address is valid between the first and second  $\overline{INTA}$  cycles.
- (4) Two  $\overline{INTA}$  cycles run back-to-back. The μPD8088 local Address/Data bus floats during both  $\overline{INTA}$  cycles. The control signals shown are for the second  $\overline{INTA}$  cycle.
- (5) Signals at the μPD8284 are shown for reference.
- (6) The μPD8288 active-high CEN lags when the μPD8288 issues command and control signals (MRDC, MWTC, AMWC, IORC, IOWC, A10WC, INTA, and DEN).
- (7) All timing measurements are taken at 1.5 V unless otherwise specified.
- (8) Status is inactive prior to T4.

Timing Waveforms (cont)

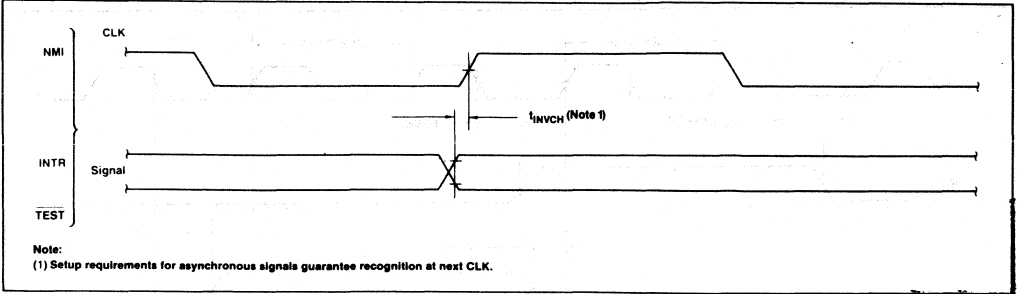
Maximum Mode System Bus Timing Using μPB8288 Bus Controller (cont) (Note 7)



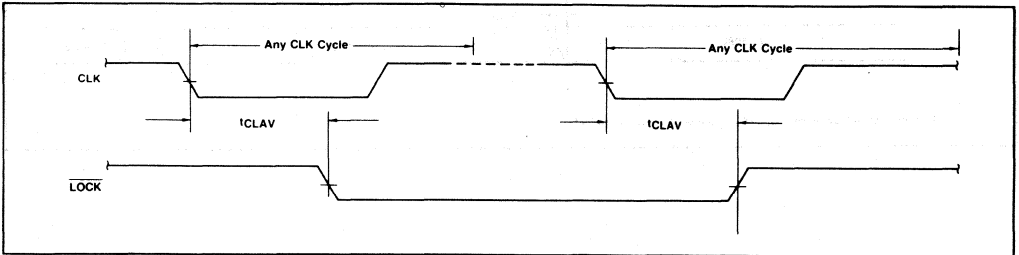
- Note:
- (1) All signals switch between  $V_{OH}$  and  $V_{OL}$  unless otherwise specified.
  - (2) RDY is sampled near the end of T2, T3, TWAIT to determine if TWAIT machine states are inserted.
  - (3) The cascade address is valid between the first and second INTA cycles.
  - (4) Two INTA cycles run back-to-back. The μPD8088 local Address/Data bus floats during both INTA cycles. The control signals shown are for the second INTA cycle.
  - (5) Signals at the μPD8284 are shown for reference.
  - (6) The μPD8288 active-high CEN lags when the μPD8288 issues command and control signals (MRDC, MWTC, AMWC, IORC, IOWC, AIOWC, INTA, and DEN).
  - (7) All timing measurements are taken at 1.5 V unless otherwise specified.
  - (8) Status is inactive prior to T4.

## Timing Waveforms (cont)

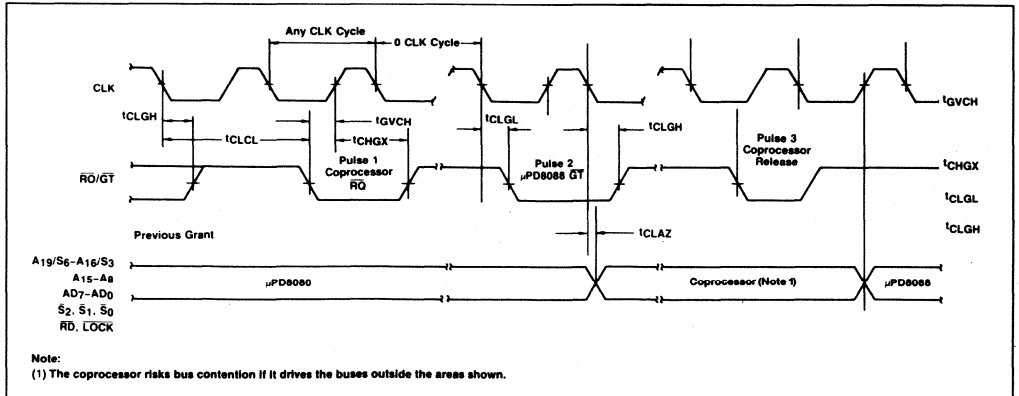
### Asynchronous Input Recognition



### Maximum Mode Bus Lock Signal Timing

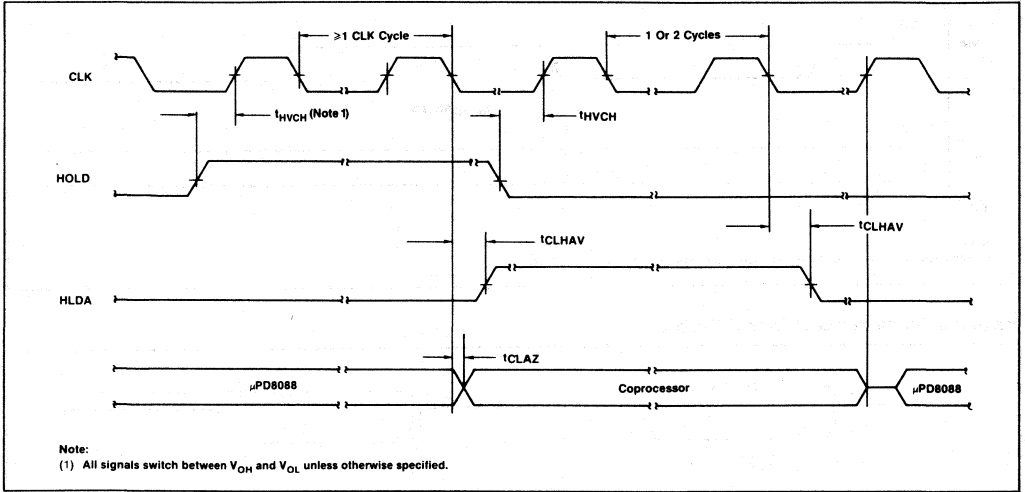


### Maximum Mode Request/Grant Sequence Timing



Timing Waveforms (cont)

Minimum Mode Hold Acknowledge Timing



## INTELLIGENT PERIPHERAL CONTROLLERS

5



## Section 5 – Intelligent Peripheral Controllers

### Magnetic Media Controllers

μPD765A/7265	Single/Double Density Floppy Disk Controllers .....	5.3
μPD72065/66		
μPB9201	Floppy-Disk Interface .....	5.61
μPD71065/66	Floppy-Disk Interface .....	5.69
μPD72067	Advanced Floppy Disk Controller .....	5.77
μPD72068	Single Chip Floppy Disk Controller with Digital Data Separator .....	5.87
μPD7260	Hard and Floppy Disk Controller .....	5.91
μPD7261A/B	Hard-Disk Controllers .....	5.111
μPD9306/A	CMOS Hard-Disk Interface .....	5.147
μPD7262	Enhanced Small Device Interface Controller .....	5.155

### Communications Controllers

μPD7201A	Multiprotocol Serial Communications Controller .....	5.167
μPD72001	CMOS Multiprotocol Serial Communications Controller .....	5.181
μPD72105	OMNINET Local Area Network Controller .....	5.199
μPD7210	Intelligent GPIB Controller .....	5.215

### Display/Graphics Controllers

μPD7220A	High-Performance Graphics Display Controller .....	5.227
μPD72022	Intelligent Display Processor .....	5.257
μPD72120	Advanced Graphic Display Controller .....	5.275
μPD7225	LCD 7/14 Segment Controller/Driver .....	5.297
μPD7227	LCD 8 x 40 Dot Matrix Controller/Driver .....	5.311
μPD7228/A	LCD 8 x 50/16 x 42 Dot Matrix Controller/Driver .....	5.315
μPD72030	LCD 640 x 256 Dot Panel Controller .....	5.339
μPD6307	LCD Row Driver for μPD72030 .....	5.374
μPD6308	LCD Column Driver for μPD72030 .....	5.383

### Image Processing

μPD7281	Data-flow based Image Pipelined Processor .....	5.389
μPD9305	Support chip for μPD7281 .....	5.431

## Introduction

NEC offers a family of floppy disk controller chips (FDCs) capable of driving different disk drives with various formats. While μPD765A, μPD765A-2 and μPD7265 are produced with NMOS technology the new μPD72065 and μPD72066 are compatible CMOS devices. The advantage of the CMOS device is a much lower power consumption resulting in a big flexibility regarding various package types including PLCC. All FDCs can be driven by NEC's V-Series CMOS microprocessors and most other standard microprocessors very easily. Powerful commands and control interface drastically reduce an overhead in both hardware and software.

Format \ FDC		μPD72065/μPD765A		μPD72066/μPD7265	
		8 MHz	4 MHz	8 MHz	4 MHz
8 inch	IBM	○		△	
	ECMA/ISO	△		○	
5.25 inch	IBM	○	○	△	△
	ECMA/ISO	△	△	○	○
3.5 inch	IBM	○	○	△	△
	ECMA/ISO	△	△	○	○
3 inch	IBM		○		△
	ECMA/ISO		△		○

- — Capable of read, write, and formatting
- △ — Capable of only read and write

**NOTE:** In descriptions of the functional timing in this manual, the standard floppy is that which requires the 8 MHz functions to the FDC, and the mini-floppy is that which requires the 4 MHz functions.

Clock frequencies are selected as follows depending on the data transfer rate.

Clock Frequency	Data Transfer Rate [K bits/s]		Disk
	Single-density (FM)	Double-density (MFM)	
8 MHz	250	500	8", 5¼", 3.5"
4 MHz	125	250	5¼", 3"
*4.8 MHz		300	5¼"

\* IBM PC-AT Format

## 1. Features

### μPD72065/μPD765A/μPD765A-2

- Compatible with IBM diskette 1 (single-sided; 128, 256, or 512 bytes/sector).
- Compatible with IBM diskette 2 (double-sided; 256 bytes/sector).
- Compatible with IBM diskette 2D (double-sided, double-density; 256 or 1,024 bytes/sector).

### μPD72066/μPD7265

- Compatible with ECMA/ISO mini-floppy format.
- ECMA66 (ISO/TC 97/SC11 N419) ——— Single-sided, single-density (256 bytes/sector).
- ECMA70 (ISO/TC 97/SC11 N475) ——— Double-sided, double-density (256 bytes/sector).

### All FDCs

- FM and MFM control (specified by the command for read/write operation).
- Variable recording length: 128, 256, 512, 1,024, 2,048, 4,096, or 8,192 bytes/sector.
- Multi-sector function\*<sup>1</sup>
- Multi-track function\*<sup>2</sup>
- Capable of controlling up to four floppy disk drives.
- Capable of parallel seek operation on up to four disk drives. (While the floppy disk drive(s) are in seek operation, the FDC accepts the seek command to other floppy disk drives.)
- Capable of specifying read/write even up to the middle of the sector (only in 128 bytes/sector, FM mode).
- Built-in CRC generating and check functions ( $X^{16} + X^{12} + X^5 + 1$ ).
- Programmable step rate (specified by the SPECIFY command).
- Programmable head load time and head unload time (specified by the SPECIFY command).
- Data scan function (detects the sector with data equal to or larger or smaller than data in the main memory).
- DMA or Non-DMA (interrupt) data transfer (specified by the SPECIFY command).
- Built-in control signal generator for write data compensation.
- Data bus and control bus compatible with 8080 family.
- Single-phase clock—8 MHz (standard floppy disk), 4MHz (mini-floppy disk).
- Single power supply +5V

\*1 This is a function which continuously reads or writes two or more sectors in the same track by a single command (controlled by the TC signal).

\*2 This is a function which continuously reads or writes both tracks in side 0 and 1 in the same cylinder by a single command (specified by the MT bit of the command).

### Comparison between μPD72065/72066 and μPD765A/7265

Items	μPD72065	μPD72066	μPD765A	μPD7265
Track Format	IBM	ECMA/ISO	IBM	ECMA/ISO
Recalibratable Tracks	255		77	255
Skipping Time After Detection of an Index Pulse (4 MHz)	0.2 ms		Approx. 1.2 ms	Approx. 0.2 ms.
DRQ↑ → RD↓	$\phi_{CY}=125ns$	1 $\phi_{CY}$ *	0.8 $\mu s$	
Response Time	$\phi_{CY}=250ns$	1 $\phi_{CY}$	1.6 $\mu s$	
Wait for FDD Response after	$\phi_{CY}=125ns$	2.5 $\mu s$	0.5 $\mu s$	
Output of US signals	$\phi_{CY}=250ns$	5.0 $\mu s$	1.0 $\mu s$	
Multi-Track Write on Tunnel Erase Head	OK		NG	
Standby Function (Standby Command)	Yes		No	
SOFTWARE RESET Command	Yes		No	

\*: Clock Cycle Time

### Comparison between μPD765A and μPD765A-2

The NMOS device μPD765A is preferably available as μPD765A-2 in an improved 3 μm fabrication technology. While the functionality remains the same some minor differences are detailed in the specification tables (Chapter 8).

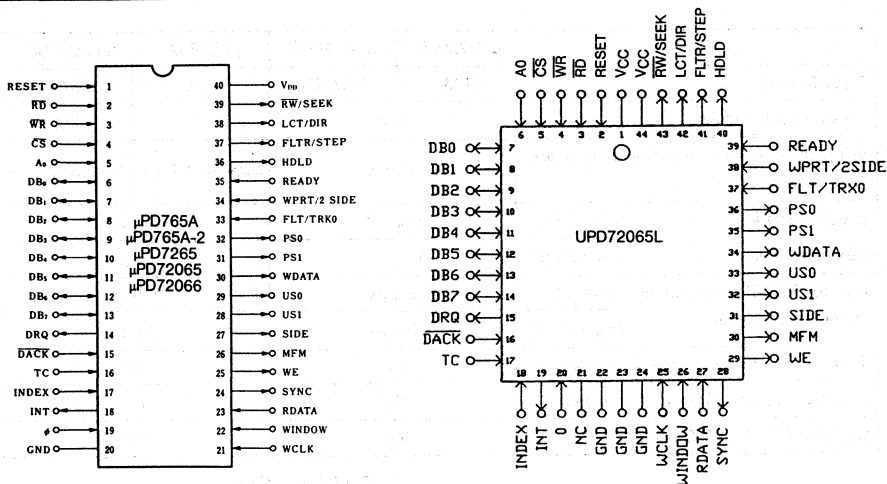
For new designs only μPD765A-2 or the CMOS type μPD72065 is recommended.

### Ordering Information

Order Code	Description	Package
μPD765AC	Floppy Disk Controller, IBM-Format*	40 pin DIL plastic
μPD765AD	Floppy Disk Controller, IBM-Format	40 pin DIL ceramic
μPD7265	Floppy Disk Controller, ISO-ECMA Format	40 pin DIL plastic
μPD765AC-2	Floppy Disk Controller, IBM-Format	40 pin DIL plastic
μPD72065C	Floppy Disk Controller, CMOS, IBM-Format	40 pin DIL plastic
μPD72065G	Floppy Disk Controller, CMOS, IBM-Format	52 pin flat pack
μPD72065L	Floppy Disk Controller, CMOS, IBM-Format	44 pin PLCC
μPD72066C	Floppy Disk Controller, CMOS, ISO-ECMA Format	40 pin DIL plastic
μPD72066G	Floppy Disk Controller, CMOS, ISO-ECMA	52 pin flat pack

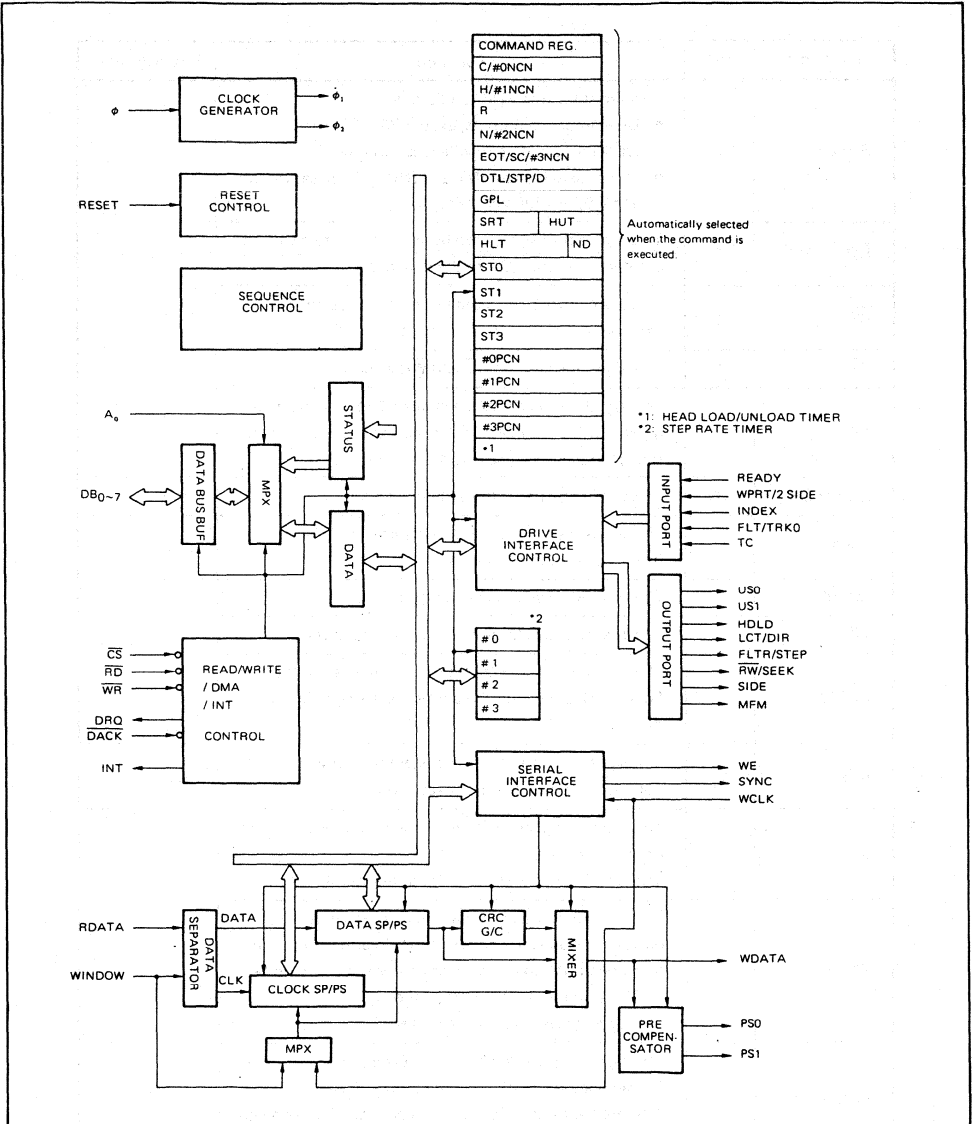
\* = not to use for new designs

Pin Configurations



NC: Non-Connection  
 IC : Internally Connected

## Block Diagram



**2. Pin Functions**

Pin Name	I/O	Function	Status at reset
RESET	I	This signal sets the FDC to idle state. <ul style="list-style-type: none"> <li>• Drive interface outputs, except PS0, PS1, and WDATA (undefined), will be set to low.</li> <li>• In the main system, INT and DRQ will be set to low, and DB7 to DB0 will be set to input.</li> </ul>	-
$\overline{RD}$	I	This is a control signal used by the main system to read out data from the FDC to the data bus.	
$\overline{WR}$	I	This is a control signal used by the main system to write data placed on the data bus to the FDC.	
$\overline{CS}$	I	This signal enables the $\overline{RD}$ and $\overline{WR}$ signals.	
A0	I	This signal selects either the status register or the data register inside the FDC via the address bus. 0 selects the status register, and 1 selects the data register.	
DB7 ~ DB0	I/O	This is a bidirectional three-state data bus.	Input
DRQ	O	This is a request signal for data transfer in DMA mode. A pull-up resistor is required.	Low
$\overline{DACK}$	I	This is a DMA cycle acknowledge signal.	-
TC	I	This is a data transfer termination request signal.	
INDEX	I	This signal indicates that the read/write head of the floppy disk drive is on the physical starting point of the track.	
INT	O	This signal requests the main system to deal with transfer of data or result of execution.	Low
$\phi$	I	This is a single phase TTL level clock input and requires a pull-up resistor. Standard floppy: 8 MHz, Mini-floppy: 4 MHz	-
WCLK	I	This is a timing signal for data transfer in write operation. However, this signal should also be input in read operation. The rising edge of this signal should be synchronized to the rising edge of $\phi$ . FM: 16 $\phi$ cycles, MFM: 8 $\phi$ cycles	
WINDOW	I	This signal is generated by the VFO circuit, and is used for sampling the clock and data bits of RDATA. Whether the bit is a clock bit or a data bit is determined inside the FDC.	

## Pin Functions (cont)

Pin Name	I/O	Function	Status at reset																				
RDATA	I	This signal is the data read out from the drive and consists of clock and data bits. Unless both WINDOW and RDATA are input at read operation, the FDC will enter the deadlock state.	—																				
SYNC	O	This signal indicates the functional mode of the FDC. 1 indicates read operation, and 0 indicates read operation is inhibited.	Low																				
WE	O	This signal requests the write operation to the drive.																					
MFM	O	This signal specifies the function mode of the VFO circuits. 1 specifies MFM mode, and 0 specifies FM mode.																					
SIDE	O	This signal selects either head 0 or 1 in a double-sided FDD. 0 selects the head 0, 1 selects the head 1.																					
US0, 1	O	These are drive select signals. One of four FDDs is selected by decoding these signals.																					
WDATA	O	This is write data to the drive and consists of the clock and data bits.	Undefined																				
PS0, 1	O	These are request signals to shift the WDATA bit in the opposite direction of the expected peak shift in order to cancel out the peak shift created when writing in MFM mode. <table border="1" style="margin: 10px auto;"> <thead> <tr> <th>PS0</th> <th>PS1</th> <th>FM</th> <th>MFM</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>No shift</td> <td>No shift</td> </tr> <tr> <td>0</td> <td>1</td> <td>—</td> <td>Delays the WDATA bit</td> </tr> <tr> <td>1</td> <td>0</td> <td>—</td> <td>Advances the WDATA bit</td> </tr> <tr> <td>1</td> <td>1</td> <td>—</td> <td>—</td> </tr> </tbody> </table>		PS0	PS1	FM	MFM	0	0	No shift	No shift	0	1	—	Delays the WDATA bit	1	0	—	Advances the WDATA bit	1	1	—	—
PS0	PS1	FM		MFM																			
0	0	No shift		No shift																			
0	1	—	Delays the WDATA bit																				
1	0	—	Advances the WDATA bit																				
1	1	—	—																				
FLT/ TRK0	I	When $\overline{RW/SEEK}$ signal specifies $\overline{RW}$ , this pin becomes FLT, and inputs the signal indicating whether the drive is in FAULT state. When $\overline{RW/SEEK}$ signal specifies $\overline{SEEK}$ , this pin becomes TRK0, and inputs the signal indicating whether the read/write head of the drive is positioned at cylinder 0.																					
WPRT/ 2 SIDE	I	When $\overline{RW/SEEK}$ signal specifies $\overline{RW}$ , this pin becomes WPRT and inputs the signal indicating whether the media is in the write inhibit state. When $\overline{RW/SEEK}$ signal specifies $\overline{SEEK}$ , this pin becomes 2 SIDE and inputs the signal indicating whether a double-sided floppy disk is inserted.																					



**Pin Functions (cont)**

Pin Name	I/O	Function	Status at reset
READY	I	This signal indicates the drive is in the ready state.	—
HDLD	O	This signal sets the read/write head of the drive to the load state.	Low
FLTR/ STEP	O	When $\overline{RW/SEEK}$ signal specifies $\overline{RW}$ , this pin becomes FLTR and releases the FAULT state of the drive. When $\overline{RW/SEEK}$ signal specifies SEEK, this pin becomes STEP and generates seek pulses.	
LCT/DIR	O	When $\overline{RW/SEEK}$ signal specifies $\overline{RW}$ , this pin becomes LCT and indicates that the read/write head of the drive is selecting a cylinder beyond the 43rd cylinder. When $\overline{RW/SEEK}$ signal specifies SEEK, this pin becomes DIR and specifies the direction of the seek operation. 0 indicates the direction toward the outside, 1 indicates the direction toward the inside.	
$\overline{RW/SEEK}$	O	This signal discriminates the drive interface signals combining for read/write and for seek. 0 indicates $\overline{RW}$ , 1 indicates SEEK.	
V <sub>CC</sub>	—	+5V power supply	—
GND	—	Ground	

### 3. Register Organization

#### 3.1 Registers for Interfacing in the Main System Side

The FDC uses the data register (DATA) and the status register (STATUS) for interfacing to the main system. Each register is selected by  $\overline{CS}$ , A0,  $\overline{RD}$ , and  $\overline{WR}$  control signals. Table 3-1 shows these relationships.

**Table 3-1** (x: Don't Care)

$\overline{CS}$	A0	$\overline{RD}$	$\overline{WR}$	Function
0	0	0	1	Status Register Read
		1	0	Standby Control, SOFTWARE RESET Command Write
	1	0	1	Data Register Read
		1	0	Data Register Write
1	x	x	x	Unselectable

**NOTE:** When  $\overline{DACK}$  input is active (0), the data register will be selected regardless of the status of  $\overline{CS}$  and A0. Writing ( $\overline{WR} = 0$ ) any code other than standby control or SOFTWARE RESET command code is inhibited when CS and A0 are 0.

**(1) Data register (DATA)**

This is an 8-bit register that temporarily stores various information (commands, parameters, data, and result status) transferred between the FDC and the main system.

**(2) Status register (STATUS)**

This is an 8-bit register that indicates the status of the FDC. Table 3-2 shows the status register organization. The main system can read the status register contents at any time.

Table 3-2 Status Register Organization

Bit	Name	Symbol	Function
D7	Request for Master	RQM	<p>Indicates to the main system that the data communication is ready. The following functions will be performed according to the status of the DIO bit (bit D6).</p> <p>When DIO = 0;</p> <p>This status indicates that data are sent from the main system to the FDC. When data are written to the FDC by the main system, RQM is set to 0; RQM then becomes 1 when the FDC accepts the data.</p> <ul style="list-style-type: none"> <li>● C-phase, Idle state</li> <li>● E-phase of the Non-DMA write</li> <li>● E-phase of seek operation</li> </ul> <p>When DIO = 1;</p> <p>This status indicates that data are sent from the FDC to the main system. When the FDC set data to the data register, RQM is set to 1; RQM is then set to 0 when the main system read the data.</p> <ul style="list-style-type: none"> <li>● R-phase</li> <li>● E-phase of the Non-DMA read (except READ ID)</li> </ul>
D6	Data Input/Output	DIO	<p>Indicates the direction of the data transferred between the main system and the FDC.</p> <p>0 indicates that the data flow is from the main system to the FDC, 1 indicates that the data flow is from the FDC to the main system.</p>
D5	Non-DMA mode	NDM	<p>Indicates that data transfer is being performed (E-phase) in Non-DMA mode.</p> <p>This bit is reset in the C-phase and in the R-phase.</p>
D4	FDC Busy	CB	<p>This bit indicates the FDC is in the C-phase, R-phase, or in the E-phase of read/write command. (This bit will not be set in the E-phase of the seek group command.)</p> <p>When this bit is set, the FDC will not accept the next command.</p>
D3	FD3 Busy	D3B	<p>This bit indicates either device No. 3 is in seek operation or the interrupt of seek end is reserved (E-phase).</p> <p>When this bit is set, a read/write command should not be written.</p>
D2	FD2 Busy	D2B	<p>This bit is for device No. 2; the functions are the same as the D3 bit.</p>
D1	FD1 Busy	D1B	<p>This bit is for device No. 1; the functions are the same as the D3 bit.</p>
D0	FD0 Busy	D0B	<p>This bit is for device No. 0; the functions are the same as the D3 bit.</p>

## 3.2 Registers for Interfacing in the FDD Side

### (1) Data separator (DATA SEPARATOR)

The data separator separates read data into data bits and clock bits using the WINDOW signal in the read mode.

Separated data are sent to the data shift register, and separated clocks are sent to the clock shift register.

### (2) Data shift register (DATA SP/PS)

In read mode, the data shift register converts serial data separated by the data separator to parallel data, and outputs the converted data onto the internal 8-bit bus.

In write mode, the data shift register converts 8-bit parallel data input from the internal bus to serial data and outputs to the CRC generator/checker and the mixer.

### (3) Clock shift register (CLOCK SP/PS)

In read mode, the clock shift register converts serial clocks separated by the data separator to parallel clocks and outputs them onto the internal 8-bit bus.

In write mode, the clock shift register converts 8-bit parallel clocks input from the internal bus to serial clocks and outputs them to the mixer.

### (4) CRC generator/checker (CRC G/C)

The CRC generator/checker generates and checks the CRC byte. The multiple formula for generating CRC is:  $X^{16} + X^{12} + X^5 + 1$

In read mode, CRC of read data is calculated and compared with the CRC bytes follow the ID and data of each sector. If these CRC bytes do not match, an error is indicated.

In write mode, CRC is calculated, and two bytes of CRC are written after the ID and data of each sector.

### (5) Precompensator (PRECOMPENSATOR)

The timing of read out data is out of synchronization with the timing when the data were written because of magnetic hysteresis. This deviation can be estimated from the data pattern; therefore, by shifting write data in the direction opposite the estimated direction, the deviation at read operation can be offset. This compensation becomes more important in MFM mode because the data window width in MFM mode is a half that of FM mode.

According to the write data pattern only in MFM mode, the precompensator generates preshift signals (PS0, 1) to control the external delay circuit that shifts write data. If the VFO circuit generates DATA WINDOW signal accurately enough to eliminate the problem of bit deviation of read data, the preshift function is not necessary.

## 4. Commands

### 4.1 Description of Command Functions

The FDC executes commands given from the main system in three phases in the following order:

#### (1) Command phase (C-phase)

When the FDC is in idle state (waiting for a command) and if the command is given from the main system, the FDC receives the parameters specifying the command function. The main system must write every parameter in a specified order.

#### (2) Execution phase (E-phase)

The command is then executed according to the parameters.

- Read/write group commands (except READ ID):  
Perform data transfer between the floppy disk drive and the main system (processed by DMA or INT).
- Seek group commands:  
Generate seek pulses (STEP).

#### (3) Result phase (R-phase)

In this phase, data are set that indicate the result of the command execution (such as result status) to the data register.

The main system should read all of these data.

The phases for each command are organized as shown in the following table.

Command	C-phase	E-phase	R-phase	INT
Read/write group	○	○	○	YES
Seek group	○	○		YES
Sense status group	○		○	NO
INVALID				
SPECIFY	○			NO
*SOFTWARE RESET	○			NO
*SET STANDBY	○			NO
*RESET STANDBY	○		○	NO

\*For the CMOS FDCs μPD72065 and μPD72066 only

## 4.2 Internal Block Functions

Fig. 4-1 shows a block diagram simplified from the one shown in section 1.9. The sequence control section shown in the figure in section 1.9 controls functional timings in other internal blocks; however, this is abbreviated in Fig. 4-1. Figs. 4-2 through 4-5 show the general functions of each block. In these figures, solidly filled arrows indicate data flow, and arrows filled with diagonal lines indicate the flow of control signals. As shown in Fig. 4-2, the host CPU utilizes  $A_0$ ,  $\overline{CS}$ , and  $\overline{WR}$  signals to control writing of the command and its parameters, and data on the data bus are taken into the data register and stored in the command register. Fig. 4-3 shows general functions of each block for each command.

In the case of seek group commands, the drive interface controller handles the control signals to or from the floppy disk drives.

In the case of the read/write group commands (except READ ID), in DMA mode, by receiving or sending DRQ, DACK, and  $\overline{RD}$ , or  $\overline{WR}$  signals, the host system controls data transfer. In Non-DMA mode, by receiving or sending  $A_0$ ,  $\overline{CS}$ , INT, and  $\overline{RD}$ , or  $\overline{WR}$  signals, the host system controls data transfer.

In the case of READ DATA, READ DELETED DATA, and READ DIAGNOSTIC commands, serial data from the drive are converted to parallel and set in the data register one byte after another, then output to the host system.

On the contrary, in the case of the WRITE DATA, WRITE DELETED DATA commands, data from the host system is taken in one byte after another, converted to serial, and then output to the drive.

In the case of the READ ID command, ID bytes are read out from the drive, converted to parallel, and then stored in the result register.

In the case of the WRITE ID command, the format is internally generated and output to the drive. Only the ID bytes are transferred from the host system to the drive.

In the case of the SCAN command, data from the host system are taken into the data register and compared with the parallel-converted data that were taken in from the drive.

As shown in Fig. 4-4, with  $A_0$ ,  $\overline{CS}$ , and  $\overline{RD}$  signals, the host system controls reading out of result status and its parameters, and places the contents of the result register in the data register. The data register contents are then output to the host system.

Fig. 4-5 shows the reading out of the status. This is always performed when the command and its parameter are written, during data transfer in Non-DMA mode, and during read out of the result status and its parameter.

The host system controls read out of the status using  $A_0$ ,  $\overline{CS}$ , and  $\overline{RD}$  signals, and the contents of the status register is output to the host system.

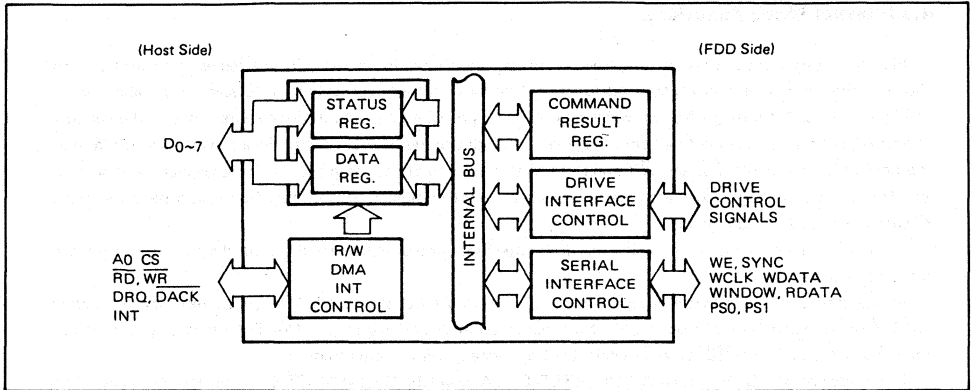


Fig. 4-1 Simplified Internal Block Diagram

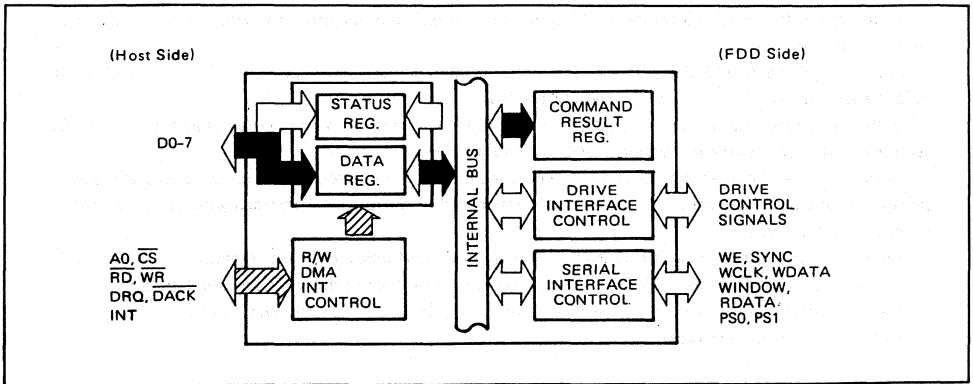


Fig. 4-2 Writing of Command

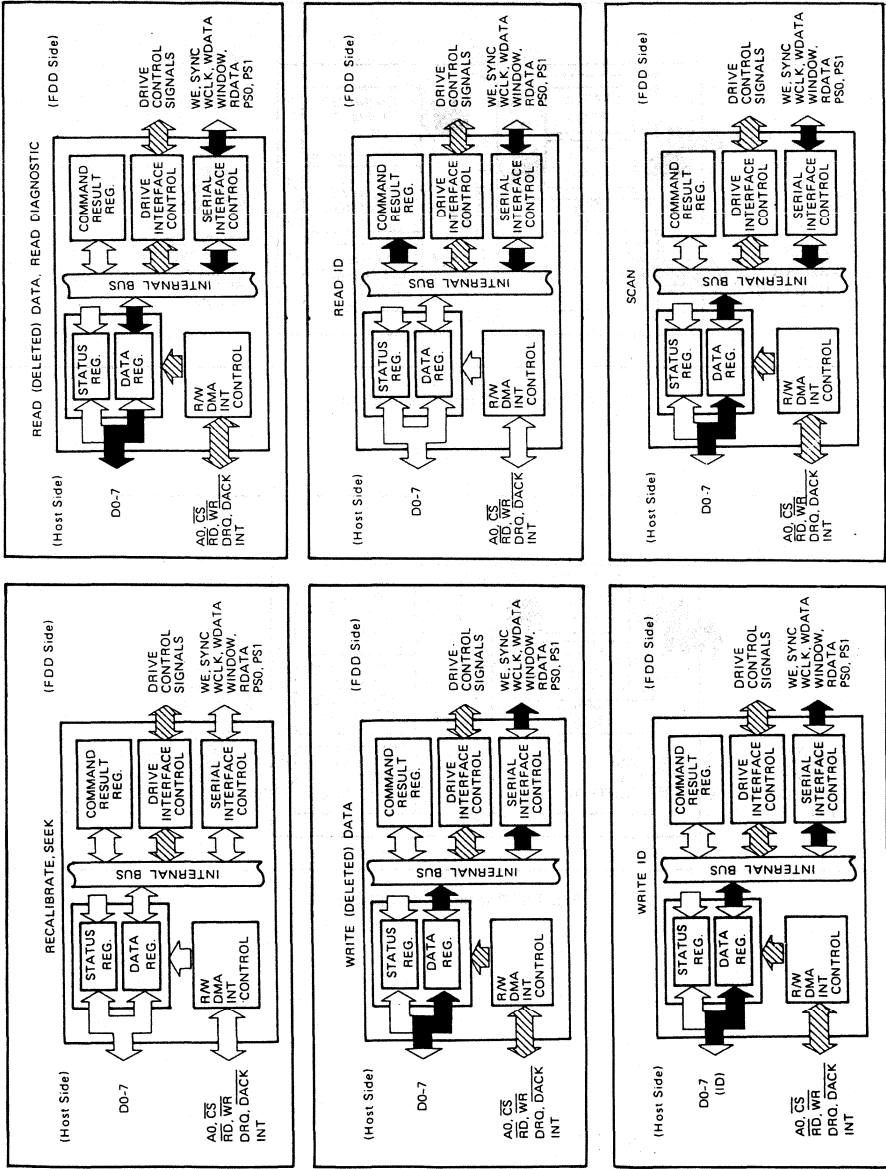


Fig. 4-3 Command Execution



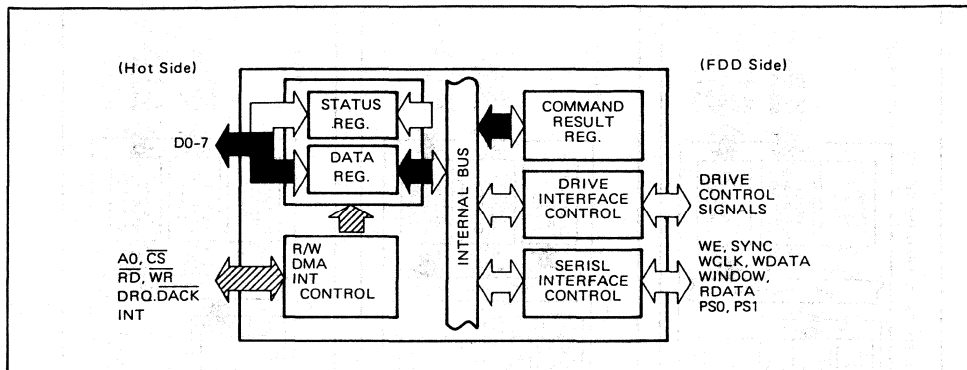


Fig. 4-4 Reading Out of Result Status

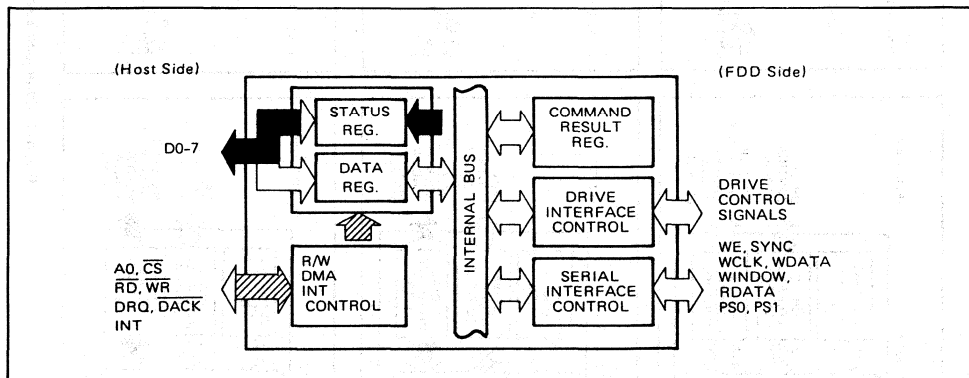


Fig. 4-5 Reading Out of Status

### 4.3 Command Functions

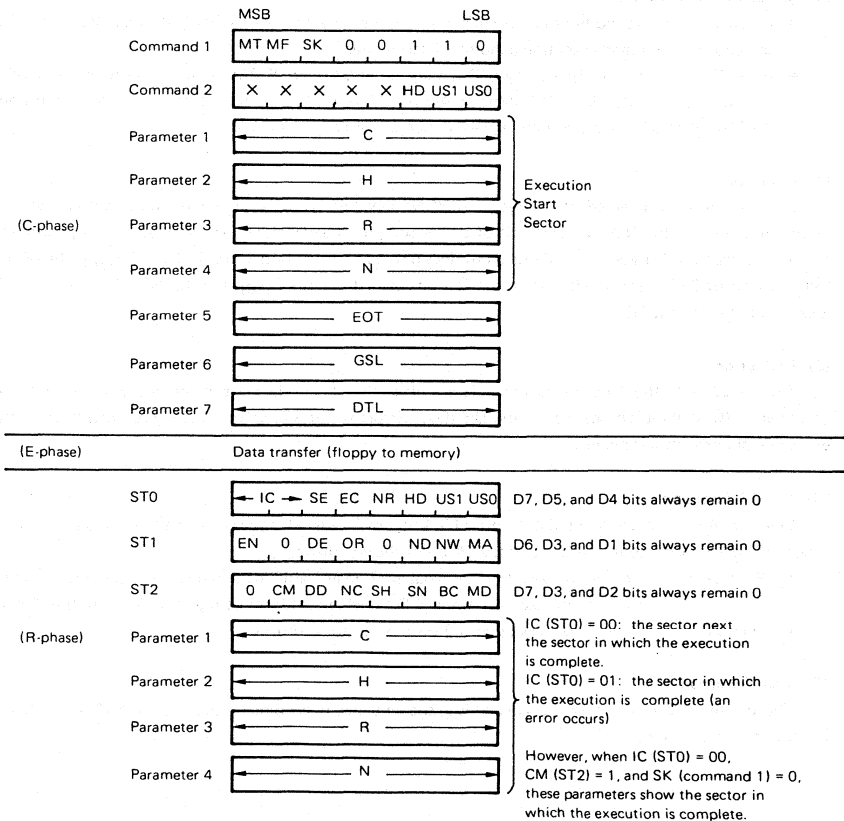
In C-phase, the host system sets  $\overline{CS}$  to 0, and A0 to 1 to select the data register of the FDC and writes the command and its parameters ( $\overline{WR} = 0$ ).

In R-phase, in the same manner, the host system sets  $\overline{CS}$  to 0, and A0 to 1 to select the data register of the FDC and reads the result status and its parameters ( $\overline{RD} = 0$ ).

However, before performing writing or reading data register in C-phase or R-phase, the host system should set  $\overline{CS}$  to 0, and A0 to 0 to select the status register of the FDC. It then reads out the status ( $\overline{RD} = 0$ ) and confirms that RQM is set to 1. These read/write procedures should follow the figures shown in the beginning portions of sections 4.3.1 through 4.3.13.

However the μPD72065/μPD765A write IAM (Index Address Mark) to floppy disk in WRITE ID command, all the FDCs ignore it in the case of the other READ/WRITE group commands.

#### 4.3.1 READ DATA



### General outline

Data in the sector specified by ID bytes (IDR) given from the main system are read from the disk and set in the data register one byte after another. The main system reads out those data in Non-DMA mode or in DMA mode. (refer to the appendix 2(5))

#### (1) Not Ready

If the device is in Not Ready state at the start of command execution or during the SYNC bytes search of ID field, NR (Not Ready) of the ST0 is set and the command execution abnormally terminates.

#### (2) Head Load

At the start of disk access, if the head is in the unloaded state after being selected by the HD bit, the head is loaded and the access starts immediately after the head load time (specified by the HLT of SPECIFY command) elapses.

#### (3) MA error and MD error

- If IDAM is not found before two INDEX pulses are detected, MA (Missing Address Mark) of ST1 is set, and the command execution abnormally terminates.
- After ID of the sector specified by IDR is detected, if DAM or DDAM cannot be found in the specified time period (1 ms/8 MHz), MD (Missing Address Mark in Data Field) bit of ST2 is set in addition to the MA bit of ST1, and the command execution abnormally terminates.

#### (4) ND error

Even if IDAM is detected, if the sector (CHRN) specified by IDR cannot be found before two index pulses are detected, the ND (No Data) bit of ST1 is set, and the command execution abnormally terminates. In the case that the C byte of the ID read out does not match the C byte of IDR, if it is FF<sub>(16)</sub>, the BC (Bad Cylinder) bit of ST2 is set in addition to the ND bit, or if it is not FF<sub>(16)</sub>, the NC (No Cylinder) bit of ST2 is set in addition to the ND bit.

#### (5) CRC error

After read out, the CRC bytes immediately after the ID or data are compared with the internally generated CRC; if these do not match, the DE (Data Error) bit of ST1 is immediately set, and the command execution abnormally terminates.

#### (6) DDAM detection

When DDAM is detected, the CM (Control Mark) bit of ST2 is set, and one of the following two processes is performed according to the contents of the SK bit of the command.

- When SK = 0: Command execution terminates normally after data transfer of that sector. The ID bytes of the sector in which DDAM is detected will be the value for CHRN in R-phase.
- When SK = 1: That sector is skipped and the next sector is processed.

#### (7) Transfer capacity

The transfer capacity for each command is as shown below depending on the N byte of IDR, and MT and MF bit of the command.

MT	MF	N(16)	Transfer Capacity (bytes)		Last Sector
0	0	00	(1 to 128) × n	n=1 to 26 sectors	Sector 26 of head 0 or sector 26 of head 1
	1	01	256 × n		
1	0	00	(1 to 128) × n	n=1 to 52 sectors	Sector 26 of head 1 (Multi-track)
	1	01	256 × n		
0	0	01	256 × n	n=1 to 15 sectors	Sector 15 of head 0 or sector 15 of head 1
	1	02	512 × n		
1	0	01	256 × n	n=1 to 30 sectors	Sector 15 of head 1 (Multi-track)
	1	02	512 × n		
0	0	02	512 × n	n=1 to 8 sectors	Sector 8 of head 0 or sector 8 of head 1
	1	03	1,024 × n		
1	0	02	512 × n	n=1 to 16 sectors	Sector 8 of head 1 (Multi-track)
	1	03	1,024 × n		

**NOTE:** The value n in the table is for standard floppy disks. In mini-floppy disks, there are fewer sectors per track than standard floppy disks; therefore, these values will be less than n in this table for mini-floppy disks.

### (8) Overrun

At data transfer, if the main system does not perform the service (in active state of  $\overline{RD}/\overline{WR}$  or  $\overline{DACK}$ ) within the following specified time from request by INT or DRQ, OR (Over Run) bit of ST1 is set after the data of that sector are transferred and the command execution abnormally terminates.

- For standard floppy – FM: 27 μs, MFM: 13 μs
- For mini-floppy – FM: 54 μs, MFM: 26 μs

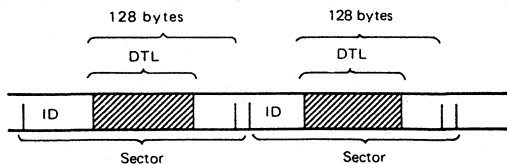
Without  $\overline{DACK}$  input, DRQ is not reset.  $\overline{DACK}$  must be input for the last DRQ in DMA mode.

### (9) Data length

When the N byte of IDR is 00, the DTL byte specifies the data length to be processed within each sector. DTL = 01 specifies 1 byte/sector, and  $DTL \geq 80(16)$  specifies 128 bytes/sector.

When DTL specifies to the middle of data in the sector ( $DTL < 80(16)$ ), for all sectors, the following data are read out from the disk; however, these data are not set in the data register and only CRC check is performed.

When  $N \neq 00$ , the DTL byte has no meaning.



**(10) Multi-sector read**

After data for one sector are transferred, if no termination request (TC) comes from the main system, R byte of IDR is modified ( $R \leftarrow R + 1$ ), and the FDC searches for the sector specified by the new IDR. However, modification of R byte will be performed within the range of  $R \leq EOT$ .

**(11) Sector format and parameter**

The reference value of EOT and GSL for data length is shown in the following table. GSL is a parameter used for skipping the splice points between data field and the following ID field on multi-sector access.

Parameter		N <sub>(16)</sub>	EOT <sub>(16)</sub>	GSL <sub>(16)</sub>	Note
IBM Format					
FM	128 bytes/sector	00	1A	07	IBM diskette 1
	256 bytes/sector	01	0F	0E	IBM diskette 2
	512 bytes/sector	02	08	1B	
	1,024 bytes/sector	03	04	Not determined	
	2,048 bytes/sector	04	02	Not determined	
	4,096 bytes/sector	05	01	Not determined	
MFM	256 bytes/sector	01	1A	0E	IBM diskette 2D
	512 bytes/sector	02	0F	1B	
	1,024 bytes/sector	03	08	35	IBM diskette 2D
	2,048 bytes/sector	04	04	Not determined	
	4,096 bytes/sector	05	02	Not determined	
	8,192 bytes/sector	06	01	Not determined	

Parameter		N <sub>(16)</sub>	EOT <sub>(16)</sub>	GSL <sub>(16)</sub>	Note
ECMA Format					
FM	128 bytes/sector	00	01	07	ECMA 66 (single-sided)
	256 bytes/sector	01	09	0E	
MFM	256 bytes/sector	01	10	0E	ECMA 70 (double-sided)

In double-density (MFM) disks of both IBM and ECMA formats, track 00 on side 0 is single-density (FM) with 128 bytes/sector.

**(12) TC timing**

At data transfer, when the TC signal is input in the middle of the sector, data in the remaining portion of the sector are read out, but the data are not to be set in the data register. And if a CRC error does not occur, the command execution terminates normally.

The timing to receive the TC signal is within the following specified time after the data transfer request by INT or DRQ.

- For standard floppy : (1 byte-time) – 5 μs
- For mini-floppy : (1 byte-time) – 10 μs

If the TC signal is input during this period, data transfer request will not be generated after the data transfer requested by the INT or the DRQ is performed.

However as for transfer request for the last byte in the sector, if TC is input within 2 bytes-time, the command execution will be terminated in that sector.

1 byte-time means, as shown in the following table, the time required to transfer one byte of data.

	FM	MFM
Standard floppy	32 μs	16 μs
Mini-floppy	64 μs	32 μs

**(13) EN error**

Unless the TC signal is input even if access to the last sector is complete (R = EOT), the EN (End of Cylinder) bit of ST1 is set and the command execution is abnormally terminated.

**(14) IDR in normal termination**

When the command execution terminates normally, the value indicated in the following table is set to IDR in R-phase according to the MT bit and the EOT byte.

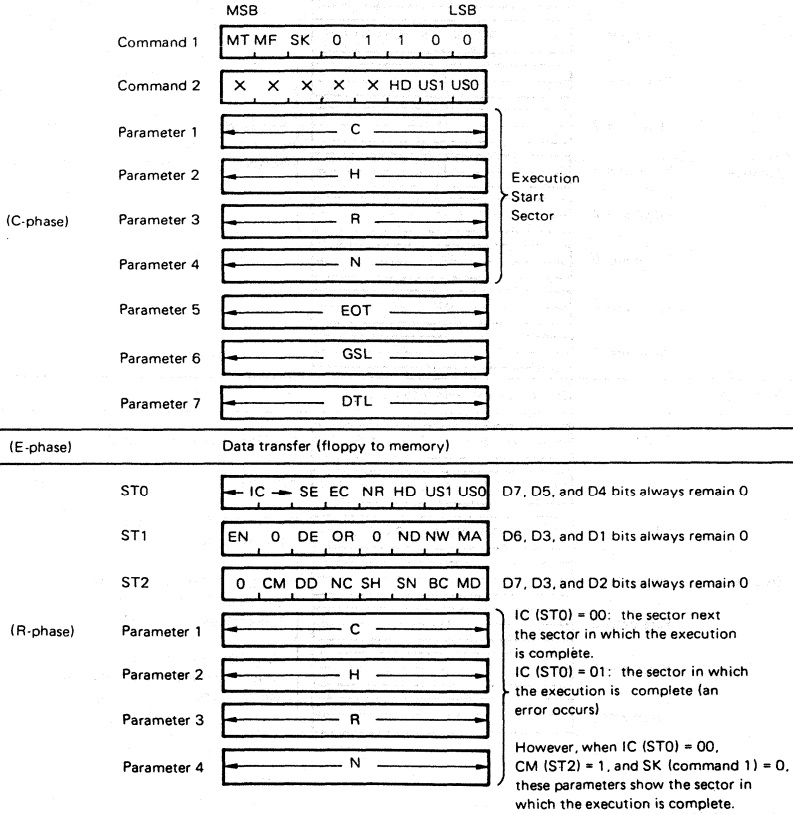
MT	EOT(16)	Sector to which the last transfer byte belongs			IDR of R-phase			
					C	H	R	N
0	1A 0F 08	Head 0	1 to 25	R < EOT	No change	No change	R + 1	No change
			1 to 14					
	1A 0F 08	Head 0	1 to 7	R = EOT	C + 1	No change	01	No change
			26					
1	1A 0F 08	Head 1	1 to 25	R < EOT	No change	No change	R + 1	No change
			1 to 14					
	1A 0F 08	Head 1	1 to 7	R = EOT	C + 1	No change	01	No change
			26					
1	1A 0F 08	Head 0	1 to 25	R < EOT	No change	No change	R + 1	No change
			1 to 14					
	1A 0F 08	Head 0	1 to 7	R = EOT	No change	Reversed LSB	01	No change
			26					
1	1A 0F 08	Head 1	1 to 25	R < EOT	No change	No change	R + 1	No change
			1 to 14					
	1A 0F 08	Head 1	1 to 7	R = EOT	C + 1	Reversed LSB	01	No change
			26					

**(15) Head unload**

The head is set to unloaded state (HDLD signal OFF) after the head unload time (specified by HUT of the SPECIFY command) has elapsed from the termination of command execution. If the read/write group command for the same device is given within this head unload time, the next command can be executed without waiting the head load time because the head is in loaded state.

However, even within the head unload time, if the command for other device or the seek group command for other cylinder is activated, the head will be immediately set to unloaded state.

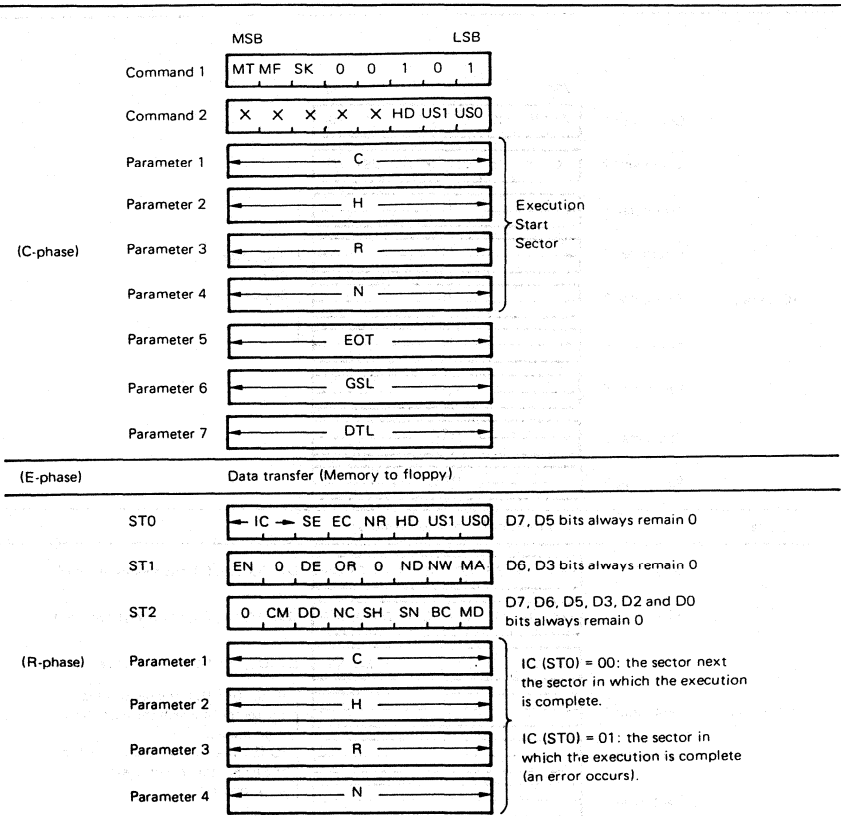
### 4.3.2 READ DELETED DATA



The difference from the function of READ DATA is that DAM and DDAM are alternated. (refer to the appendix II(5))



4.3.3 WRITE DATA



**General description**

After the DAM is written to the sector specified by IDR, data which the main system writes to the data register byte by byte are transferred to that sector. (refer to appendix 11(6))

**(1) Same functions as READ DATA**

The following functions are the same as READ DATA:

- (1) Not ready      (2) Head load      (4) ND error      (7) Transfer capacity
- (11) Sector format and parameter      (13) EN error      (14) IDR at normal termination
- (15) Head unload

**(2) NW error**

If WPRT signal is input at the start of command execution, NW (Not Writable) bit of the ST1 will be immediately set and the command execution will abnormally terminate.

**(3) MA error**

If IDAM is not found before two INDEX pulses are detected, MA (Missing Address Mark) of ST1 is set, and the command execution abnormally terminates.

**(4) CRC error**

When the CRC bytes immediately after the ID is read out and compared with internally generated CRC but they do not match, the DE bit of ST1 will be immediately set and the command execution will abnormally terminate.

**(5) Overrun**

At data transfer, if the main system does not perform the service within the following specified time from the request by INT or DRQ, the OR (Over Run) bit of ST1 is set after data are transferred to that sector and command execution abnormally terminates.

- For standard floppy – FM: 31 μs, MFM: 15 μs
- For mini-floppy – FM: 62 μs, MFM: 30 μs

Without  $\overline{\text{DACK}}$  input, DRQ is not reset.  $\overline{\text{DACK}}$  must be input for the last DRQ in DMA mode.

**(6) Data length**

When N byte of IDR is 00, DTL byte specifies the data length to be processed within each sector. DTL = 01 specifies 1 byte/sector, and  $\text{DTL} \geq 80(16)$  specifies 128 bytes/sector. When DTL specifies to the middle of data in the sector ( $\text{DTL} < 80(16)$ ), for all sectors, 00 is written in each byte of the following data.

When  $N \neq 0$ , DTL byte has no meaning.

**(7) EC error**

If FLT signal is input after writing CRC of data and 1 byte of Gap, EC (Equipment Check) bit of ST0 is set and command execution abnormally terminates.

**(8) Multi-sector write**

After data of one sector are transferred and if the TC signal is not input from the main system, R byte of IDR is modified ( $R \leftarrow R + 1$ ) and the sector specified by the renewed IDR is searched for.

However, the modification of R byte will be performed within the range of  $R \leq \text{EOT}$ .

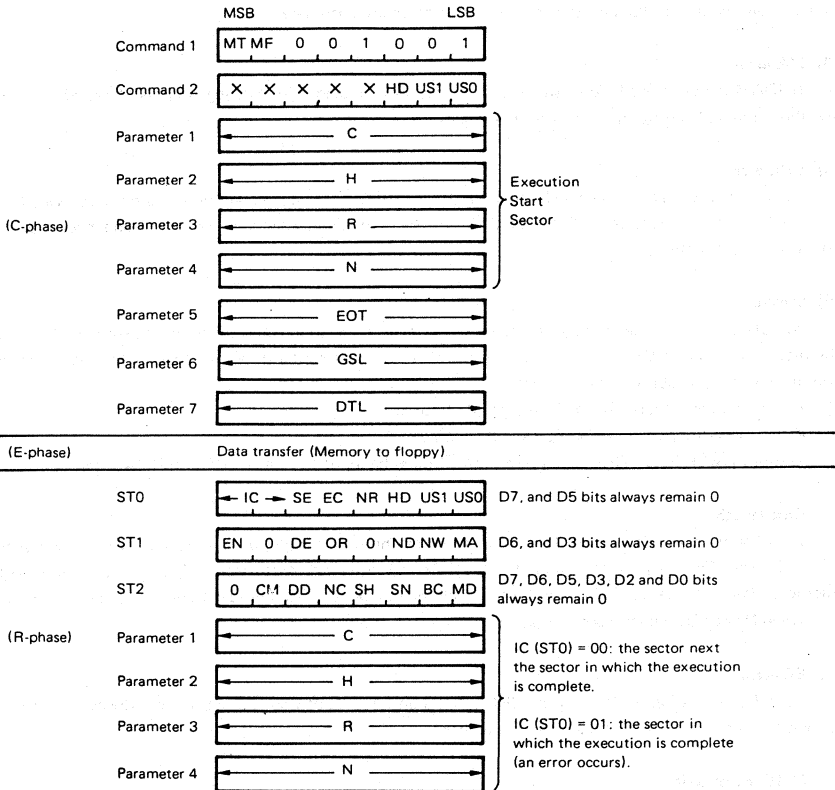
**(9) TC timing**

At data transfer, when the TC signal is input in the middle of the sector, 00 is written in the remaining portion of the sector, and the command execution terminates normally. The timing to receive the TC signal is within the following specified time after the data transfer request by INT or DRQ.

- For standard floppy : (1 byte-time) – 6 μs
- For mini-floppy : (1 byte-time) – 12 μs

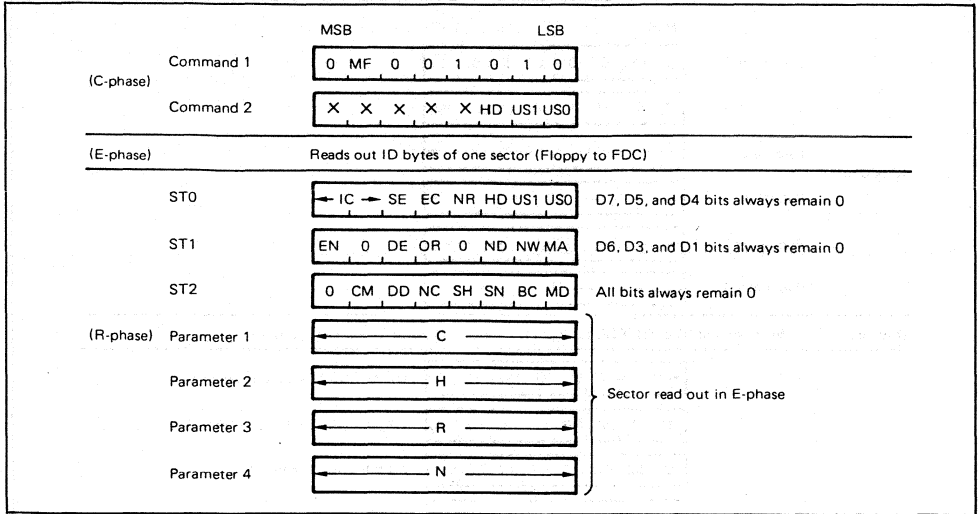
If the TC signal is input during this period, data transfer request will not be generated after the data transfer requested by the INT or the DRQ is performed. However, as for a transfer request for the last byte in the sector, if TC is input within 2 bytes-time, the command execution is terminated in that sector.

4.3.4 WRITE DELETED DATA



Same as WRITE DATA except DDAM is written instead of DAM. (refer to appendix I(6))

## 4.3.5 READ ID



### General description

ID of the first detected sector (with no DE error or MA error) after the head load time has elapsed is set to the data register as IDR of R-phase. (refer to the appendix II(8)) The sector to be detected cannot be specified and is undefined.

#### (1) Same functions as READ DATA

The following functions are the same as those of READ DATA.

- (1) Not ready      (2) Head load      (15) Head unload

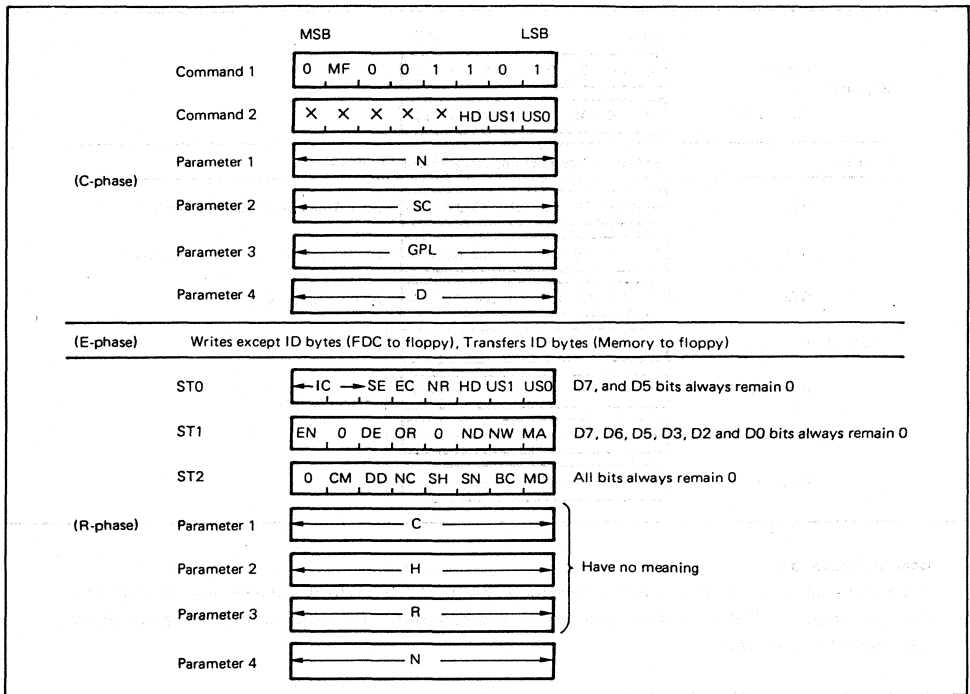
#### (2) MA error

If IDAM is not found before two INDEX pulses are detected, the MA bit of ST1 is set and the command execution abnormally terminates.

#### (3) ND error

Even if IDAM is found, unless the ID not having a CRC error is found before two INDEX pulses are detected, the ND bit of ST1 is set and the command execution abnormally terminates.

4.3.6 WRITE ID [Format Write]



**General description**

Data for one track are written according to the format. Data from the main system are transferred to the ID field, and D byte of the parameter is written to the data field as the data pattern. (refer to appendix II(9))

**(1) Same functions as READ DATA**

The following functions are the same as those of READ DATA.

- (1) Not ready      (2) Head load      (15) Head unload

**(2) Same functions as WRITE DATA**

The following functions are the same as those of WRITE DATA.

- (2) NW error      (5) Overrun

**(3) Transfer data**

(Four bytes of CHRN) x (number of sectors), that is, only IDs for one track are transferred. Therefore, the sequence IDs (setting R bytes at intervals) or IDs for bad cylinder (setting C bytes to FF(16)) can be written by those data from the main system.

**(4) Format**

The FDC internally sets and writes data according to the format (Gap, Sync, Address Mark, Data, and CRC) except ID. However, data length, number of sectors, length of Gap 3, and data pattern can be programmed by the parameters.

**(5) Sector format and parameter**

The following table shows reference values of SC and GPL for data length. GPL specifies the length (number of bytes) of Gap 3 written at the end of sectors.

Parameter		$N_{(16)}$	$SC_{(16)}$	$GPL_{(16)}$	Note
IBM Format					
FM	128 bytes/sector	00	1A	1B	IBM diskette 1
	256 bytes/sector	01	0F	2A	IBM diskette 2
	512 bytes/sector	02	08	3A	
	1,024 bytes/sector	03	04	Not determined	
	2,048 bytes/sector	04	02	Not determined	
	4,096 bytes/sector	05	01	Not determined	
MFM	256 bytes/sector	01	1A	36	IBM diskette 2D
	512 bytes/sector	02	0F	54	
	1,024 bytes/sector	03	08	74	IBM diskette 2D
	2,048 bytes/sector	04	04	Not determined	
	4,096 bytes/sector	05	02	Not determined	
	8,192 bytes/sector	06	01	Not determined	

Parameter		$N_{(16)}$	$SC_{(16)}$	$GPL_{(16)}$	Note
ECMA Format					
FM	128 bytes/sector	00	10	18	ECMA66 (single-sided)
	256 bytes/sector	01	09	26	
MFM	256 bytes/sector	01	10	32	ECMA70 (double-sided)

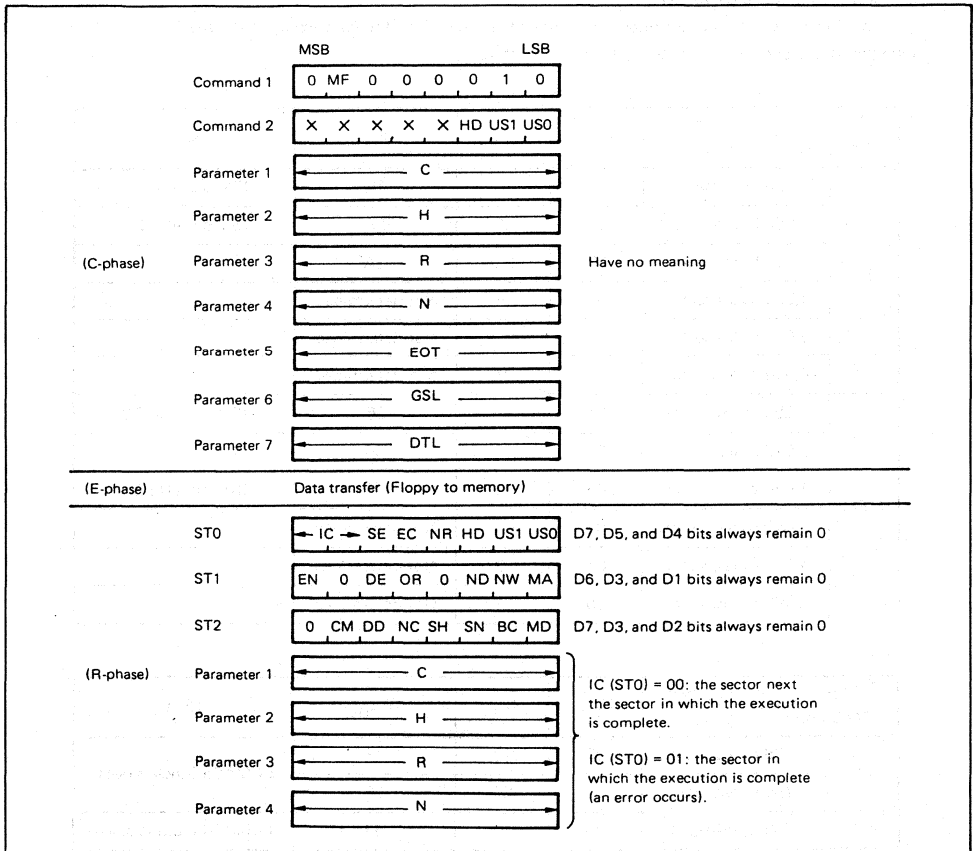
**(6) Start of formatting**

After the start of command execution, when an INDEX pulse is input, formatting starts. After the one sector is written, if TC is not input and the number of processed sectors does not match the SC parameter, Gap 3 will be written and then the next sector format will be processed.

**(7) Completion of formatting**

After one sector is written, if TC is already input or the number of processed sectors matches the SC parameter, then only Gap is written. When the INDEX pulse is input, writing operation ends.

**4.3.7 READ DIAGNOSTIC**



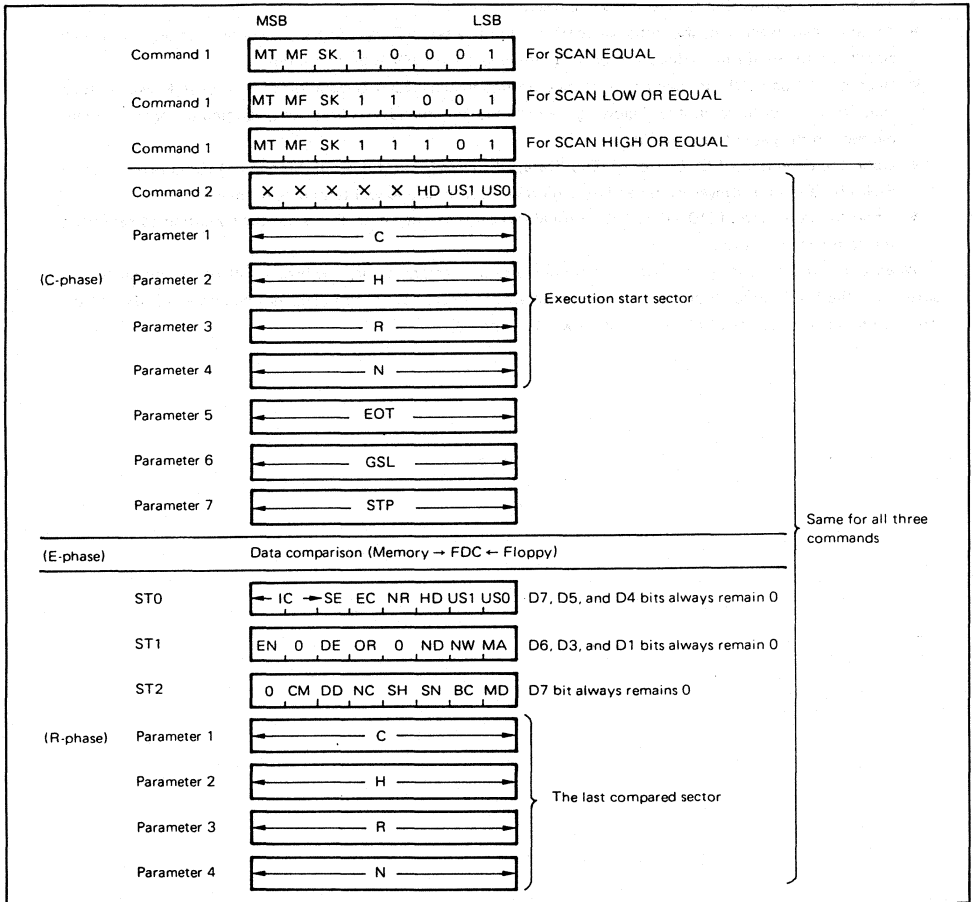
Same as READ DATA except for the following differences. (refer to appendix II(7))

- There is no specification for multi-track and skip.
- At command execution, the initial value 01 is internally set to R byte of the parameter, and processing will start from the sector immediately after the point where the INDEX pulse is input.
- Read out ID and IDR are compared; if they do not match, ND (No Data) bit of ST1 is set. However, data transfer will continue in the following sectors and command execution will terminate normally (This happens in the case of sequence ID.).
- Upon the detection of CRC error of ID or Data, DE (Data Error) bit or/DE bit and DD (Data Error in Data Field) bit are set; however, command execution will continue and will terminate normally.
- Upon the detection of DDAM, CM (Control Mark) bit is set; however, this is not the condition to terminate the command execution.

When command execution is normally terminated and if errors occur, the errors will be accumulated and output for the result status; however, the sectors in which errors have occurred cannot be known (CHRN of the result is the renewed sector in which execution was terminated).



4.3.8 SCAN EQUAL/SCAN LOW OR EQUAL/SCAN HIGH OR EQUAL



**General description**

By comparing data on each sector with the same amount of data in the main system, the sector which satisfies the condition (equal, high, or low) will be searched. (refer to appendix 11(10))

**(1) Same functions as READ DATA**

The following functions are the same as those of READ DATA:

- (1) Not ready      (2) Head load      (3) MA error and MD error      (4) ND error
- (5) CRC error      (6) DDAM detection      (8) Overrun
- (11) Sector format and parameter      (15) Head unload

**(2) Data not to be compared**

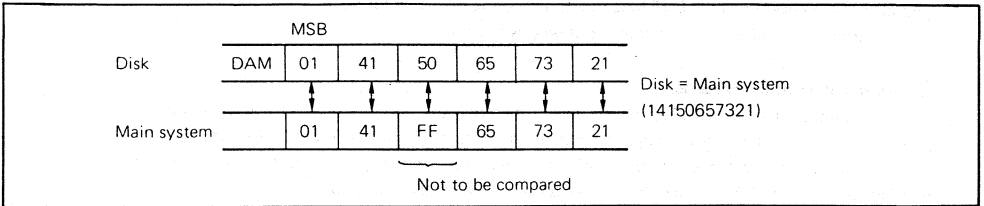
When data from the main system are FF(16), these data are not compared (they are assumed to match).

**(3) Comparison method**

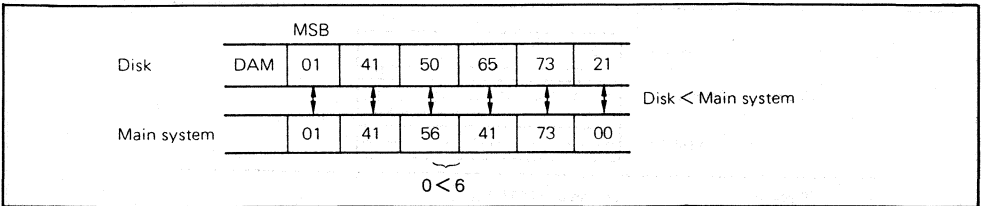
- **SCAN EQUAL:** Checks whether the data in the sector to be compared match the data of the main system. If they match, the command execution will terminate normally.
- **SCAN LOW OR EQUAL:** When the data in the sector to be compared are smaller than or equal to the data of the main system, the command execution will terminate normally.
- **SCAN HIGH OR EQUAL:** When the data in the sector to be compared are larger than or equal to the data of the main system, the command execution will terminate normally.

Comparison of equal, larger, and smaller is performed so that the first byte to be compared is made the most significant byte and the data group in one sector is handled as one numeric value (In the same way, a data group on the main system side is also handled as one numeric value.).

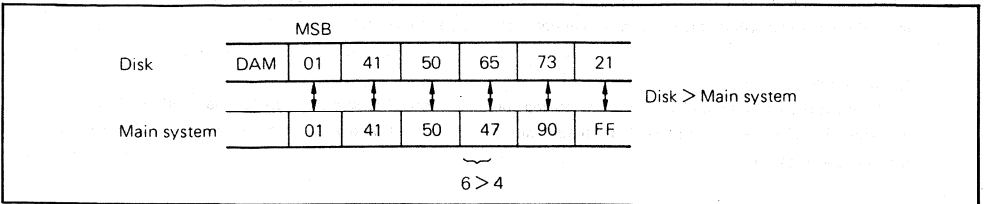
- **EQUAL (matching) data string example**



- **Low (smaller) data string example**



- **HIGH (larger) data string example**



**(4) TC signal**

Whenever the last byte in the each sector is transferred, input the TC signal (for instance, by autoloading of the DMA controller). When all bytes in each sector do not need to be compared, TC can be input in the middle of the sector and bytes beyond that point will not be compared; only data bytes up to the point of TC input will be compared in all sectors. For this command, TC input is not the condition to terminate command execution.

**(5) Sector modification**

When data in one sector are compared to the data of the main system, and if the condition does not meet, R byte of IDR is modified ( $R \leftarrow R + STP$ ;  $STP = 1$  or  $2$ ), and the sector specified by the new R byte is compared.

Therefore, the main system needs to send repeatedly the same data for each sector.

**(6) STP**

STP byte of the parameter specifies 01 or 02. However, when 02 is specified, R byte of the parameter (initial sector number) and the EOT byte should satisfy the following formula:

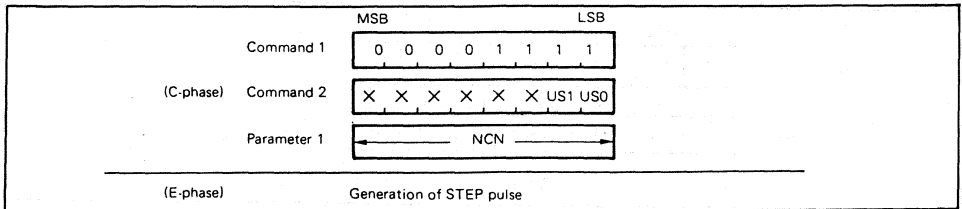
$$R + 2(n - 1) = EOT \quad n: \text{Number of sectors to be processed}$$

**(7) Termination of the execution**

If a compared sector satisfies the condition, the command execution will terminate normally at that point. In this case, if the EQUAL condition is satisfied, SH (Scan Equal Hit) bit of ST2 is set.

If the sector that satisfies the condition is not found even though the last sector is compared, SN (Scan Not Satisfied) bit of ST2 is set and the command execution will terminate normally.

**4.3.9 SEEK**



**General description**

Assuming the NCN (New Cylinder Number) byte of the parameter as the cylinder number to be sought, the read/write head will be moved to this cylinder. (refer to appendix 11(4))

**(1) Not ready**

At the start of command execution or during the execution of seek operation (E-phase), if the device is not ready, SE (Seek End) and NR (Not Ready) bits of ST0 are set, and the command execution abnormally terminates at that point.

**(2) Seek operation**

The PCN (Present Cylinder Number) byte indicating the cylinder at which the read/write head is located is compared with the NCN byte. When they are not equal, either one of the following two operations is performed.

- When  $NCN > PCN$ : DIR signal is set, a STEP pulse (STEP) is output, and PCN is incremented by one ( $PCN \leftarrow PCN + 1$ ).
- When  $NCN < PCN$ : DIR signal is reset, a STEP pulse is output, and PCN is decremented by one ( $PCN \leftarrow PCN - 1$ ).

After this, the above will be repeated at each step rate time specified by the SPECIFY command.

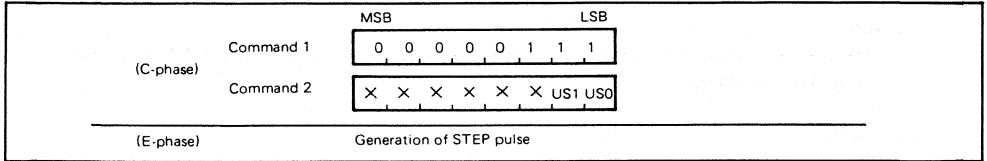
**(3) Termination of seek operation**

When PCN and NCN become equal, SE (Seek End) of ST0 is set, and the command execution will terminate normally. However, the contents of ST0 will be processed by the SENSE INTERRUPT STATUS command.

**(4) Parallel seek**

Because the FDC is in Non-Busy state in the E-phase (FD Busy), SEEK or RECALIBRATE command for other devices can be accepted, and the seek operation can be performed simultaneously in up to four devices. However, the read/write group command should not be written at this time (FD Busy).

**4.3.10 RECALIBRATE [Return to cylinder 0]**



**General description**

Seek operation towards the outer track (DIR = 0) is made until the TRK0 signal is input. (refer to appendix II(4))

**(1) Functions same as SEEK**

The following functions are the same as those of SEEK operation.

- (1) Not ready
- (2) Parallel seek

**(2) Recalibrate operation**

After PCN byte is cleared, the TRK0 signal is checked. If the TRK0 signal is 0, the DIR signal is reset and a STEP pulse is generated.

After this, a STEP pulse is generated at every step rate time specified by the SPECIFY command.

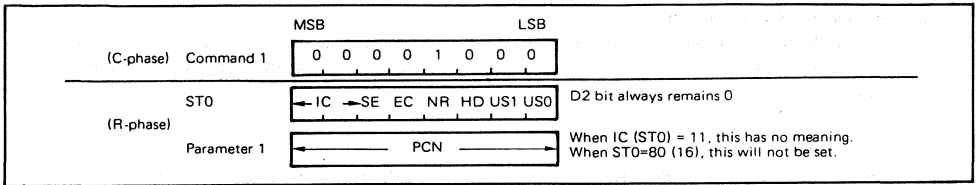
**(3) Termination of recalibrate operation**

When TRK0 signal becomes 1, SE (Seek End) bit of ST0 is set and command execution terminates normally. However, the contents of ST0 will be processed by the SENSE INTERRUPT STATUS command.

**(4) EC error**

If TRK0 signal is not input even if the recalibrate operation is repeated 255 times, SE bit and EC (Equipment Check) bit of ST0 are set, and command execution abnormally terminates.

**4.3.11 SENSE INTERRUPT STATUS**



**General description**

The result status of seek operation by the SEEK or RECALIBRATE command or the result status at a status change (Not Ready to Ready, or Ready to Not Ready) of the device not under the command execution is set to the data register. (refer to appendix II(3))

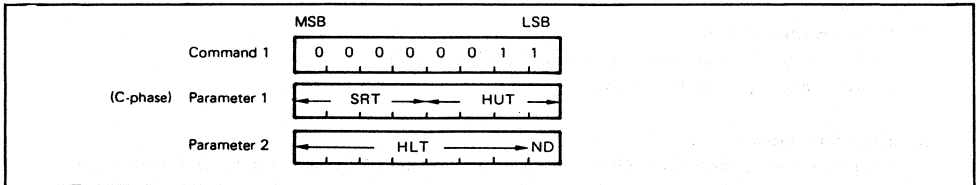
**(1) PCN**

PCN byte of R-phase indicates the cylinder number at the end of the seek operation. PCN byte has no meaning at the status change.

**(2) Invalid**

If the termination of seek operation or a status change does not occur in any device, this command will be processed as an invalid command (refer to 4.3.14 INVALID).

**4.3.12 SPECIFY**



### General description

The initial value of various internal timers and functional mode are defined (refer to appendix II(2)).

#### (1) HUT

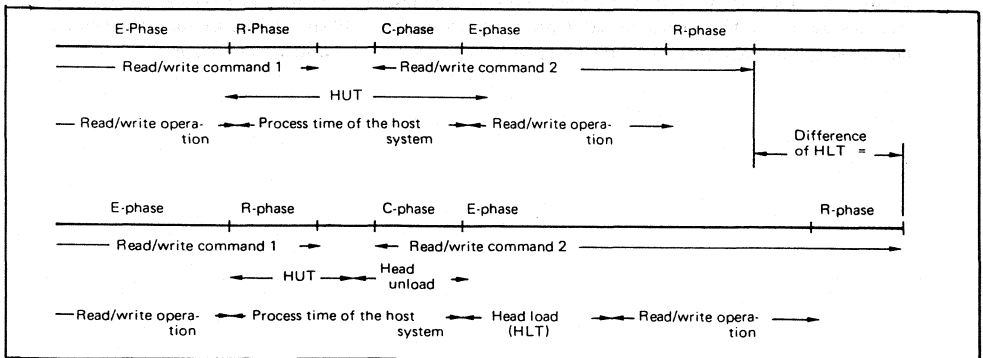
HUT (Head Unload Time) specifies the time to set the read/write head to unloaded state after the read/write group command is executed (SYNC/WE reset).

- Standard floppy: 16 to 240 ms (16 ms per unit)
- Mini-floppy: 32 to 480 ms (32 ms per unit)

HUT <sub>(16)</sub>	Time (ms)
0	Inhibited
1	16/32
2	32/64
3	48/96
4	64/128
5	80/160
6	96/192
7	112/224

HUT <sub>(16)</sub>	Time (ms)
8	128/256
9	144/288
A	160/320
B	176/352
C	192/384
D	208/416
E	224/448
F	240/480

When read/write commands are executed continuously, the access time can be shortened by setting HUT longer than the process time of the host system between the E-phases because the head remains in loaded state and HLT will be ignored.



**(2) SRT**

SRT (Step Rate Time) specifies the interval of step pulses generated by seek group commands.

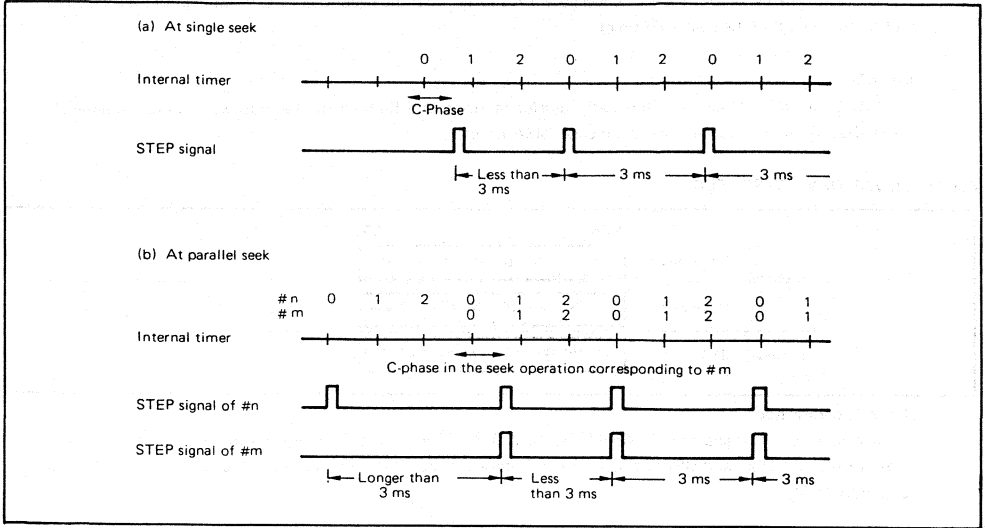
- For standard floppy: 1 to 16 ms (1 ms per unit)
- For mini-floppy: 2 to 32 ms (2 ms per unit)

SRT <sub>(16)</sub>	Time (ms)
0	16/32
1	15/30
2	14/28
3	13/26
4	12/24
5	11/22
6	10/20
7	9/18

SRT <sub>(16)</sub>	Time (ms)
8	8/16
9	7/14
A	6/12
B	5/10
C	4/8
D	3/6
E	2/4
F	1/2

If C-phase occurs over the STEP signal generating timing, the STEP signal is delayed until the end of the phase, causing a shortened step signal interval for the next STEP signal. This occurs when the duration of C-phase of the seek group command is longer than the step rate time (Fig. 4-6(a)) or when another command is written (for example, parallel seek operation) while the seek group command is being executed (Fig. 4-6(b)). In these cases, the minimum value of the step pulse interval will be shorter than the values shown in the table by 1 ms (for standard floppy) and 2 ms (for mini-floppy).

To avoid seek errors, shorten as much as possible the duration of C-phase of the seek group command or the duration of C-phase when another command is written in E-phase. If neither is possible, set a longer step rate time. As long as the duration of each C-phase is less than  $20 \mu\text{s} \times [\text{Step Rate}]$ , seek error will not occur. For instance, if the step rate time is 3 ms, the parameter of C-phase can be written within  $60 \mu\text{s}$ .



**Fig. 4-6 Shortened Step Rates**

**(3) HLT**

HLT (Head Load Time) specifies the stabilizing time of the read/write head after loading at the start of the read/write group commands (set to the specification of the drive).

- For standard floppy: 2 to 254 ms (2 ms per unit)
- For mini-floppy: 4 to 508 ms (4 ms per unit)

However, if the head is in loaded state (within the HUT of the previous command) at the start of command execution, HLT is meaningless.

HLT <sub>(16)</sub>	Time (ms)
00	Inhibited
01	2/4
02	4/8
03	6/12
04	8/16
05	10/20
06	12/24
07	14/28

HLT <sub>(16)</sub>	Time (ms)
08	16/32
09	18/36
⋮	
7D	250/500
7E	252/504
7F	254/508

The minimum value of the head load time will be shorter than the values shown in the table by 1 ms (for standard floppy) and 2 ms (for mini-floppy).

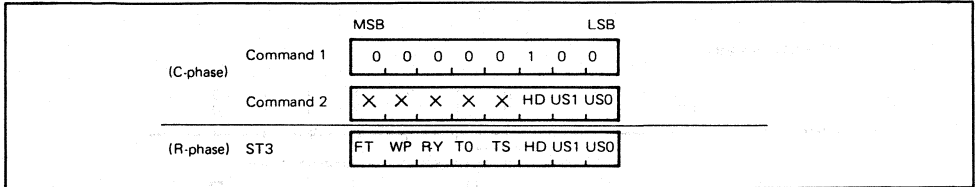


Settling time need not be specified because it is shorter than the head load time (assuming that the settling will be completed in the head load time).

(4) ND

ND (Non-DMA Mode) specifies data transfer mode in the E-phase of the read/write group command. 1 specifies Non-DMA transfer, and 0 indicates DMA transfer.

4.3.13 SENSE DEVICE STATUS



General description

The state of the signals (FLT, WPRT, READY, TRKO, and 2 SIDE) indicating the device status and the state of the HD, US0, and US1 bits specified in C-phase are set to ST3 and loaded to the data register. (refer to appendix II(3))

4.3.14 INVALID

General description

In the following two cases, 10<sub>(2)</sub> (IC: Invalid command) is set to the Interrupt Code (highest two bits) of ST0, and 0 is set to all remaining bits (ST0 = 80H).

This is then loaded to the data register. (refer to appendix II(1))

- When an undefined command code is given; or
- When the SENSE INTERRUPT STATUS command is activated even though an INT request by termination of the seek group command or by a drive status change is not generated.

4.3.15 SET STANDBY\*

FDC internal clocks are stopped, and the FDC is set into standby mode.

The contents of RAM and the state of output pins are held in standby mode.

The SET STANDBY command can be written when CB = 0, DIO = 0 and RQM = 1 on status register as usual. The FDC is set into standby mode about 3μs later than the accepting of this command. The standby mode can be released by RESET STANDBY or SOFTWARE RESET command or RESET input.

This command has only C-phase.

4.3.16 RESET STANDBY\*

The FDC is released from standby mode. This command is written any time in standby mode.

This command is processed as invalid command. Therefore this command has C-phase and R-phase.

4.3.17 SOFTWARE RESET\*

The FDC is reset in the same way as the RESETsignal input. This command can be written any time.

This command has only C-phase.

\*For CMOS FDCs μPD72065 and μPD72066 only

### 4.4 Normal and Abnormal Displaying Bits of the Result Status (1 Indicates the Bit to be Set)

Command	Conditions at the end of execution	ST0					ST1					ST2							
		NT	AT	SE	EC	NR	EN	DE	OR	ND	NW	MA	CM	DD	NC	SH	SN	BC	MD
READ DATA	Normal termination	00																	
	Not ready		01			1													
	ID field	IDAM not detected	01									1							
		C not matched (≠FFH)	01							1				1					
		C not matched (=FFH)	01								1							1	
		H not matched	01								1								
		R not matched (in one track)	01								1								
		N not matched	01								1								
	Data field	CRC not matched	01					1											
		DAM not detected	01									1							1
		DDAM detected	00										1						
		CRC not matched	01						1					1					
		Overrun	01						1										
		Incomplete at the last sector	01					1											
READ DELETED DATA	Normal termination	00																	
	Not ready		01			1													
	ID field	IDAM not detected	01									1							
		C not matched (≠FFH)	01								1				1				
		C not matched (=FFH)	01									1						1	
		H not matched	01									1							
		R not matched (in one track)	01									1							
		N not matched	01									1							
		CRC not matched	01					1											
	Data field	DDAM not detected	01										1						1
		DAM detected	00											1					
		CRC not matched	01						1						1				
		Overrun	01							1									
		Incomplete at the last sector	01					1											
WRITE DATA	Normal termination	00																	
	Not ready		01			1													
	Write protect	01										1							
	ID field	IDAM not detected	01										1						
		C not matched (≠FFH)	01								1				1				
		C not matched (=FFH)	01									1						1	
		H not matched	01									1							
		R not matched (in one track)	01									1							
		N not matched	01										1						
		CRC not matched	01						1										
	Data field	Fault	01			1													
		Overrun	01							1									
		Incomplete at the last sector	01					1											

**Normal and Abnormal Displaying Bits of the Result Status (cont)**  
**(1 Indicates the Bit to be Set)**

Command	Conditions at the end of execution	ST0				ST1				ST2									
		NT	AT	SE	EC	NR	EN	DE	OR	ND	NW	MA	CM	DD	NC	SH	SN	BC	MD
WRITE DELETED DATA	Normal termination	00																	
	Not ready	00			1														
	Write protect	01								1									
	ID field	IDAM not detected	01									1							
		C not matched (#FFH)	01							1				1					
		C not matched (=FFH)	01							1								1	
		H not matched	01							1									
		R not matched (in one track)	01							1									
		N not matched	01							1									
		CRC not matched	01						1										
	Data field	Fault	01		1														
		Overrun	01						1										
	Incomplete at the last sector	01				1													
READ ID	Normal termination	00																	
	Not ready	01			1														
	ID field	IDAM not detected	01									1							
		CRC not matched	01							1									
WRITE ID	Normal termination	00																	
	Not ready	01			1														
	Write protect	01								1									
	Fault	01		1															
	Overrun	01						1											
READ DIAGNOS- TIC	Normal termination	00																	
	Not ready	01			1														
	ID field	IDAM not detected	01									1							
		C not matched (Incomplete termination)	00							1									
		H not matched (Incomplete termination)	00							1									
		R not matched (Incomplete termination)	00							1									
		N not matched (Incomplete termination)	00							1									
		CRC not matched (Incomplete termination)	00						1										
	Data field	DAM not detected	01									1						1	
		DDAM detected (Incomplete termination)	00										1						
		CRC not matched (Incomplete termination)	00						1					1					
		Overrun	01						1										
		Incomplete at the last sector	01				1												

**Normal and Abnormal Displaying Bits of the Result Status (cont)**  
**(1 Indicates the Bit to be Set)**

Command	Conditions at the end of execution	ST0					ST1					ST2								
		NT	AT	SE	EC	NR	EN	DE	OR	ND	NW	MA	CM	DD	NC	SH	SN	BC	MD	
SCAN EQUAL/ SCAN LOW OR EQUAL/ SCAN HIGH OR EQUAL	LOW/HIGH condition met	00																		
	EQUAL condition met	00													1					
	Condition not met	00															1			
	ID field	Not ready	01				1													
		IDAM not detected	01									1								
		C not matched (≠FFH)	01								1				1					
		C not matched (=FFH)	01								1								1	
		H not matched	01								1									
		R not matched (in one track)	01								1									
		N not matched	01								1									
	Data field	CRC not matched	01								1									
		DAM not detected	01											1						1
		DDAM detected	00											1						
		CRC not matched	01								1				1					
SEEK (accepted by the SENSE INTERRUPT STATUS)	Overrun	01								1										
	Normal termination	00			1															
	Not ready		01		1		1													
RECALI- BRATE (accepted by the SENSE INTERRUPT STATUS)	Normal termination	00			1															
	Not ready		01		1		1													
	Track 0 not detected		01		1	1														

**4.5 Symbols in the Command Table (Shown in 4.6)**

Symbol (Name)	Function
R/W (Read/Write)	Indicates that $\overline{RD}$ or $\overline{WR}$ signal will be set to active state.
D7 – D0 (Data Bus)	Indicates correspondence to the 8-bit data bus.
MT (Multi-track)	When this is 1, the multi-track mode is specified.
MF (MEM Mode)	1 specifies MFM mode, 0 specifies FM mode.
SK (Skip)	Specifies to skip the sector with the DDAM or DAM.
HD (Head)	Specifies physical head number, either 0 (side 0) or 1 (side 1).
US0, 1 (Unit Select)	Specifies the drive number (0 to 3).
C (Cylinder Number)	Indicates cylinder number.
H (Head Number)	Indicates logical head number (data on the media).
R (Record Number)	Indicates sector number.
N (Record Length)	Code indicating the data length in one sector.
EOT (End of Track)	Indicates the last sector number to be accessed on the track.
GPL (Gap Length)	Indicates number of bytes to be written for Gap 3.
GSL (Gap Skip Length)	Indicates number of bytes to be skipped in Gap 3.
DTL (Data Length)	Specifies the data length per sector to be processed.
ST0, 1, 2, 3 (Status)	Indicates contents of result status.
SC (Sector)	Indicates number of sectors per track to be formatted by the WRITE ID command.
D (Data)	Indicates the data pattern to be written to the data field by WRITE ID command.
STP (Step)	During execution of SCAN command, if STP is 1, the next sector is processed; if STP is 2, every other sector is processed.
NCN (New Cylinder Number)	Indicates the cylinder number being sought.
PCN (Present Cylinder Number)	Indicates the cylinder number at the time of completion of the SENSE INTERRUPT STATUS command.
SRT (Step Rate Time)	Specifies the interval of the step pulses.
HUT (Head Unload Time)	Specifies head unloading time.
HLT (Head Load Time)	Specifies head loading time.
ND (Non-DMA Mode)	When this is 1, Non-DMA mode is specified.

### 4.6 Command Table (X: Don't care, Refer to Section 4.5 for Symbols)

Command	Phase	R/W	D7 _____ D0	Note	
READ DATA	C	W	1	MT MF SK 0 0 1 1 0	SK: Skip DDAM  ID of execution start sector
			2	X X X X X HD US1 US0	
			3	C	
			4	H	
			5	R	
			6	N	
			7	EOT	
			8	GSL	
			9	GTL	
	E	R		Data transfer	
	R	R	1	ST0	When normally terminated (except CM), ID of the sector next to execution end sector; when abnormally terminated or when CM (Table 4-6), ID of execution end sector
			2	ST1	
			3	ST2	
			4	C	
			5	H	
			6	R	
7			N		
READ DELETED DATA	C	W	1	MT MF SK 0 1 1 0 0	SK: Skip DMA  Same as READ DATA
			2	X X X X X HD US1 US0	
			3	C	
			4	H	
			5	R	
			6	N	
			7	EOT	
			8	GSL	
			9	DTL	
	E	R		Data transfer	
	R	R	1	ST0	Same as READ DATA
			2	ST1	
			3	ST2	
			4	C	
			5	H	
			6	R	
7			N		

**Command Table (cont) (X: Don't care, Refer to Section 4.5 for Symbols)**

Command	Phase	R/W	D7					D0	Note				
READ ID	C	W	1	0	MF	0	0	1	0	1	0		
			2	X	X	X	X	X	HD	US1	US0		
	E	-										Stores first read ID having no error (No data transfer)	
	R	R	1								ST0	} Same as READ DATA	
			2								ST1		
			3								ST2		
			4								C	} ID read in the E-phase	
5										H			
6									R				
7									N				
WRITE ID	C	W	1	0	MF	0	0	1	1	0	1		
			2	X	X	X	X	X	HD	US1	US0		
			3										N
			4										SC
			5										GPL
	6									D			
	E	W										Transfers ID (SC X 4 bytes) of sectors for one track	
	R	R	1								ST0	} Same as READ DATA	
			2								ST1		
			3								ST2		
4											C	} Have no meaning	
5											H		
6										R			
7										N	N byte specified in C-phase		

**Command Table (cont) (X: Don't care, Refer to Section 4.5 for Symbols)**

Command	Phase	R/W	D7 _____ D0	Note	
WRITE DATA	C	W	1	MT MF 0 0 0 1 0 1	Same as READ DATA
			2	X X X X X HD US1 US0	
			3	C	
			4	N	
			5	R	
			6	N	
			7	EOT	
			8	GSL	
			9	GTL	
	E	W		Data transfer	
	R	R	1	ST0	Same as READ DATA
			2	ST1	
			3	ST2	
			4	C	
5			H		
6			R		
7			N		
WRITE DELETED DATA	C	W	1	MT MF 0 0 1 0 0 1	Same as READ DATA
			2	X X X X X HD US1 US0	
			3	C	
			4	H	
			5	R	
			6	N	
			7	EOT	
			8	GSL	
			9	DTL	
	E	W		Data transfer	
	R	R	1	ST0	Same as READ DATA
			2	ST1	
			3	ST2	
			4	C	
5			H		
6			R		
7			N		

5



**Command Table (cont) (X: Don't care, Refer to Section 4.5 for Symbols)**

Command	Phase	R/W	D7	D0				D0		Note			
READ DIAG- NOSTIC	C	W	1	0	MF	0	0	0	1	0	Same as READ DATA (except that R has no meaning)		
			2	X	X	X	X	X	HD	US1		US0	
			3									C	
			4									H	
			5									R	
			6									N	
			7									EOT	
			8									GSL	
			9									DTL	
	E	R								Data transfer			
	R	R	1							ST0	Same as READ DATA		
			2							ST1			
			3							ST2			
			4							C			
5									H				
6									R				
7									N				
SCAN EQUAL	C	W	1	MT	MF	SK	1	0	0	0	1	Same as READ DATA	
			2	X	X	X	X	X	HD	US1	US0		
			3										C
			4										H
			5										R
			6										N
			7										EOT
			8										GSL
			9										STP
	E	W									Data comparison		
	R	R	1								ST0	Last sector compared	
			2								ST1		
			3								ST2		
			4								C		
5										H			
6										R			
7									N				

**Command Table (cont) (X: Don't care, Refer to Section 4.5 for Symbols)**

Command	Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Note		
SCAN LOW OR EQUAL	C	W	1	MT	MF	SK	1	1	0	0	1	} Same as READ DATA	
			2	X	X	X	X	X	HD	US1	US0		
			3										C
			4										H
			5										R
			6										N
			7										EOT
			8										GSL
			9										STP
	E	W										Data comparison	
	R	R	1									ST0	} Last sector compared
			2									ST1	
			3									ST2	
			4									C	
5											H		
6											R		
7											N		
SCAN HIGH OR EQUAL	C	W	1	MT	MF	SK	1	1	1	0	1	} Same as READ DATA	
			2	X	X	X	X	X	HD	US1	US0		
			3										C
			4										H
			5										R
			6										N
			7										EOT
			8										GSL
			9										STP
	E	W									Data comparison		
	R	R	1									ST0	} Last sector compared
			2									ST1	
			3									ST2	
			4									C	
5											H		
6											R		
7											N		

**Command Table (cont) (X: Don't care, Refer to Section 4.5 for Symbols)**

Command	Phase	R/W	D7	D0				D0		Note			
SEEK	C	W	1	0	0	0	0	1	1	1	1		
			2	X	X	X	X	X	X	US1	US0		
			3	NCN									
	E	—	—									Seek operation	
RECALI- BRATE	C	W	1	0	0	0	0	0	1	1	1		
			2	X	X	X	X	X	X	US1	US0		
			E	—	—								
SENSE INTER- RUPT STATUS	C	W	1	0	0	0	0	1	0	0	0	Cylinder number at the end of command execu- tion (has no meaning at the drive status change)	
	R	R	1	ST0									
			2	PCN									
SENSE DEVICE STATUS	C	W	1	0	0	0	0	0	1	0	0	Status of the drive	
			2	X	X	X	X	X	HD	US1	US0		
	R	R	ST3										
SPECIFY	C	W	1	0	0	0	0	0	0	1	1		
			2	SRT				HUT					
			3	HLT									
INVALID	C	W	1	Other code									ST0 = 80 <sub>(16)</sub>
	R	R	1	ST0									
SET STANDBY*	C	W	1	0	0	1	1	0	1	0	1		
RESET STANDBY*	C	W	1	0	0	1	1	0	1	0	0	ST0 = 80 <sub>(16)</sub>	
	R	R	1	ST0									
SOFTWARE RESET*	C	W	1	0	0	1	1	0	1	1	0		

\*For the CMOS FDCs μPD72065 and μPD72066 only

## 4.7 Result Status

Table 4-2 Result Status 0 (ST0)

Bit	Name	Symbol	Function
D7	Interrupt Code	IC	Indicates cause of INT request <u>D7 D6</u>
			0 0 Normal termination of command execution (NT)
			0 1 Abnormal termination of command execution (AT)
D6			1 0 Indicates the command was not executed because it was invalid (IC)
			1 1 Indicates a change in device status
D5	Seek End	SE	This bit is set when seek operation by SEEK or RECALIBRATE command is terminated normally or abnormally.
D4	Equipment Check	EC	This bit is set when the Fault signal from the device is accepted or when the Track 0 signal cannot be detected within a certain period in execution of the RECALIBRATE command.
D3	Not Ready	NR	This bit is set when the specified device is not ready.
D2	Head Address	HD	Indicates the head status at the time of INT request. This bit is set to 0 when SENSE INTERRUPT STATUS command is executed.
D1	Unit Select 1	US1	Indicates the device number at the time of INT request
D0	Unit Select 0	US0	

NT: Normal Terminate  
IC: Invalid Command

AT: Abnormal Terminate  
AI: Attention Interrupt

Table 4-3 Result Status 1 (ST1)

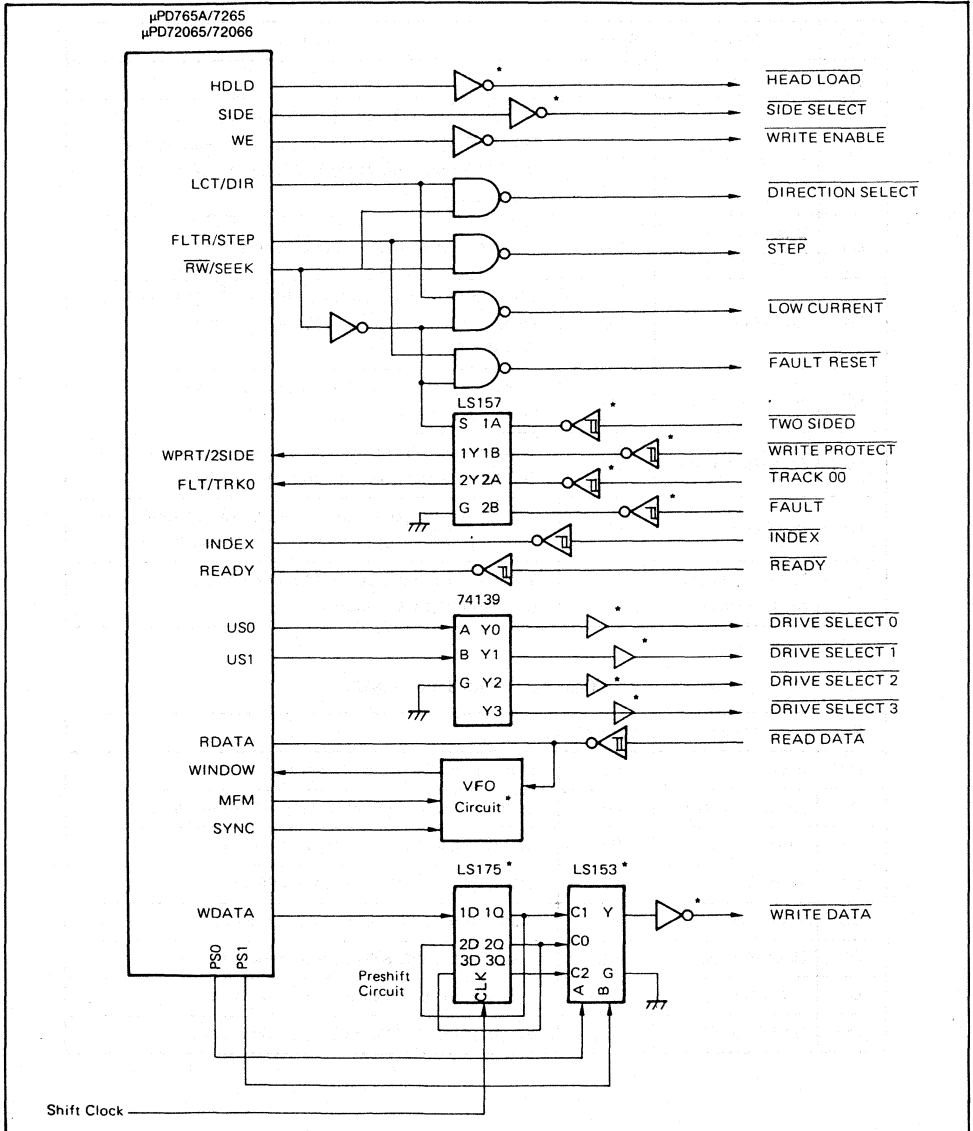
Bit	Name	Symbol	Function
D7	End of Cylinder	EN	This bit is set when read or write is attempted beyond the last sector specified by the EOT
D6	—	—	0
D5	Data Error	DE	This bit is set when the CRC error of ID or data is detected (except READ ID). DD bit (D5) of ST2 specifies either ID or data.
D4	Overrun	OR	This bit is set when service by the main system is not performed within the specified amount of time at the data transfer.
D3	—	—	0
D2	No Data	ND	<ol style="list-style-type: none"> <li>This bit is set if the sector specified by the IDR cannot be detected on the track when any one of the following five commands is executed: <ul style="list-style-type: none"> <li>● READ DATA</li> <li>● READ DELETED DATA</li> <li>● WRITE DATA</li> <li>● WRITE DELETED DATA</li> <li>● SCAN</li> </ul> </li> <li>This bit is set when an ID with no CRC error is not detected on the track in the execution of READ ID command.</li> <li>This bit is set when the sector ID and the contents of the specified IDR are not matched at the READ DIAGNOSTIC command execution.</li> </ol>
D1	Not writable	NW	This bit is set when the write protect signal is detected by execution of a write group command.
D0	Missing Address Mark	MA	<ol style="list-style-type: none"> <li>This bit is set when the IDAM cannot be found before two index pulses are detected by the execution of a command that accesses the ID of the disk.</li> <li>This bit is set when the DAM or DDAM cannot be found after the IDAM is found. MD bit of ST2 is also set at this time.</li> </ol>

**Table 4-4 Result Status 2 (ST2)**

Bit	Name	Symbol	Function
D7	—	—	0
D6	Control Mark	CM	This bit is set when the DDAM is detected at the READ DATA, READ DIAGNOSTIC or SCAN execution or when the DAM is detected at the READ DELETED DATA execution.
D5	Data Error in Data Field	DD	This bit is set when a CRC error is detected.
D4	No Cylinder	NC	When neither C byte of ID matches nor FF(16), this bit is set together with ND bit of ST1 (except READ DIAGNOSTIC).
D3	Scan Equal Hit	SH	This bit is set when the Equal condition is satisfied at the SCAN command execution.
D2	Scan Not Satisfied	SN	This bit is set when the condition is not satisfied at the SCAN command execution up to the last sector.
D1	Bad Cylinder	BC	This bit is set together with the ND bit of ST1 when the C byte of ID is FF(16) (except READ DIAGNOSTIC)
D0	Missing Address Mark in Data Field	MD	This bit is set in the case of [2.] where MA bit of ST1 is set (DAM, DDAM cannot be found).

**Table 4-5 Result Status 3 (ST3)**

Bit	Name	Symbol	Function
D7	Fault	FT	Status of the Fault signal from the device
D6	Write Protected	WP	Status of the Write Protected signal from the device
D5	Ready	RY	Status of the Ready signal from the device
D4	Track 0	T0	Status of the Track 0 signal from the device
D3	Two Side	TS	Status of the 2 Side signal from the device
D2	Head Address	HD	Status of the Side Select signal to the device
D1	Unit Select 1	US1	Status of the Unit Select 1 signal to the device
D0	Unit Select 0	US0	Status of the Unit Select 0 signal to the device



**Fig. 7-2 Interfacing of FDC to FDD (Reference Circuit)**

\*Included in the Floppy Disk Interface μPB9201  
 Output drivers can sink 24 mA

## SPECIFICATION

### Absolute Maximum Ratings

T<sub>A</sub> = 25°C

Power supply voltage, V <sub>CC</sub>	-0.5 to +7V
Input voltage, V <sub>I</sub>	-0.5 to +7V
Output Voltage, V <sub>O</sub>	-0.5 to +7V
Operating temperature, T <sub>OPT</sub>	-10°C to +70°C
Storage temperature, T <sub>STG</sub> (μPD765A/7265)	-40°C to +125°C
Storage temperature, T <sub>STG</sub> (μPD765A-2/72065/72066)	-65°C to +150°C

**Comment:** Exposing the device to stresses above those listed in the Absolute Maximum Ratings could cause permanent damage. The device should not be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### Capacitance

T<sub>A</sub> = 25°C, f<sub>C</sub> = 1MHz, V<sub>CC</sub> = 0V

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input clock capacitance	C <sub>IN(O)</sub>			20	pF	(Note 1)
Input capacitance	C <sub>IN</sub>			10	pF	(Note 1)
Output capacitance	C <sub>OUT</sub>			20	pF	(Note 1)

### Note:

(1) All pins except pin under test tied to AC ground

### DC Characteristics

T<sub>A</sub> = -10°C to +70°C, V<sub>CC</sub> = +5V ± 5% unless otherwise specified for μPD765A and μPD7265 else V<sub>CC</sub> = +5V ± 10%

Parameter	Symbol	μPD765A/7265		μPD765A-2/7265-2		μPD72065/72066		Unit	Test Conditions
		Limits		Min	Max	Min	Max		
		Min	Max						
Input voltage low	V <sub>IL</sub>	-0.5	+0.8	-0.5	+0.8	-0.5	0.8	V	
Input voltage high	V <sub>IH</sub>	2.0	V <sub>CC</sub> +0.5	2.0	V <sub>a</sub> +0.5	2.2	V <sub>CC</sub> +0.5	V	
Output voltage low	V <sub>OL</sub>		0.45		0.45		0.45	V	I <sub>OL</sub> = 2.0 mA
Output voltage high	V <sub>OH</sub>	2.4	V <sub>CC</sub>	2.4	V <sub>CC</sub>	2.4	V <sub>CC</sub>	V	I <sub>OH</sub> = -200 μA
Input voltage low (CLK + WR clock)	V <sub>IL(φ)</sub>	-0.5	0.65	-0.5	+0.65	-0.5	+0.65	V	
Input voltage high (CLK + WCLK)	V <sub>IH(φ)</sub>	2.4	V <sub>CC</sub> +0.5	2.4	V <sub>CC</sub> +0.5	2.4	V <sub>CC</sub> +0.5	V	
Supply current (V <sub>CC</sub> )	I <sub>CC</sub>		150		140		10	mA	φ = 125ns
Input load current high	I <sub>LIH</sub>		10		10		10	μA	V <sub>IN</sub> = V <sub>CC</sub>
Input load current low	I <sub>LIL</sub>		-10		-10		-10	μA	V <sub>IN</sub> = 0V
Output leakage current high	I <sub>LOH</sub>		10		10		10	μA	V <sub>OUT</sub> = V <sub>CC</sub>
Output leakage current low	I <sub>LOL</sub>		-10		-10		-10	μA	V <sub>OUT</sub> = +0.45V
Supply current (Stand-by)	I <sub>CC1</sub> I <sub>CC2</sub> I <sub>CC3</sub>		-		-		500 250 100		φ = 125ns φ = 250ns clock stopped



**AC Characteristics**

T<sub>A</sub> = -10°C to +70°C, V<sub>CC</sub> = +5V ± 5% for μPD765A/7265, otherwise V<sub>CC</sub> = +5V ± 10%

Parameter	Symbol	765A, 7265			765A-2, 7265-2			72065, 72066			Unit	Test Conditions
		Min	Typ (1)	Max	Min	Typ (1)	Max	Min	Typ (1)	Max		
Clock period	Φ <sub>CY</sub>	120	125	500	120	125	500	120	125	500	ns	(Note 4)
			125			125			125		ns	8" FDD
			250			250			250		ns	5¼" FDD
			125			125			125		ns	3½" Sony (3)
Clock active (high)	Φ <sub>0</sub>	40			40			40			ns	
Clock rise time	Φ <sub>r</sub>			20			20			20	ns	
Clock fall time	Φ <sub>f</sub>			20			20			20	ns	
A <sub>0</sub> , CS, DACK setup time to RD <sup>†</sup>	t <sub>Ar</sub>	0			0			0			ns	
A <sub>0</sub> , CS, DACK hold time from RD <sup>†</sup>	t <sub>RA</sub>	0			0			0			ns	
RD width	t <sub>RR</sub>	250			200			200			ns	
Data access time from RD <sup>†</sup>	t <sub>RD</sub>			200			140			140	ns	C <sub>L</sub> = 100 pF
DB to float delay time from RD <sup>†</sup>	t <sub>DF</sub>	20		100	10		85	10		85	ns	C <sub>L</sub> = 100 pF
A <sub>0</sub> , CS, DACK setup time to WR <sup>†</sup>	t <sub>AW</sub>	0			0			0			ns	
A <sub>0</sub> , CS, DACK hold time to WR <sup>†</sup>	t <sub>WA</sub>	0			0			0			ns	
WR width	t <sub>WW</sub>	250			200			200			ns	
Data setup time to WR <sup>†</sup>	t <sub>DW</sub>	150			100			100			ns	
Data hold time from WR <sup>†</sup>	t <sub>WD</sub>	5			0			0			ns	
INT delay time from RD <sup>†</sup>	t <sub>RI</sub>			500			400			400	ns	
INT delay time from WR <sup>†</sup>	t <sub>WI</sub>			500			400			400	ns	
DRQ cycle time	t <sub>MCY</sub>	13			13			13			μs	Φ <sub>CY</sub> = 125ns (4)
DACK <sup>†</sup> → DRQ <sup>†</sup> delay	t <sub>AM</sub>			200			140			140	ns	
DRQ <sup>†</sup> → DACK <sup>†</sup> delay	t <sub>MA</sub>	200			200			200			ns	Φ <sub>CY</sub> = 125ns (4)
DACK width	t <sub>AA</sub>	2			2			2			Φ <sub>CY</sub>	
TC width	t <sub>TC</sub>	1 Φ <sub>CY</sub>			1 Φ <sub>CY</sub>			60ns			Φ <sub>CY</sub>	
Reset width	t <sub>RST</sub>	14			14			14			Φ <sub>CY</sub>	
Clock hold time at standby	t <sub>WC</sub>							32			Φ <sub>CY</sub>	
Clock hold time release	t <sub>CW</sub>							16			Φ <sub>CY</sub>	
WCLK cycle time	t <sub>CY</sub>	16			16			16			Φ <sub>CY</sub>	MFM = 0
		8			8			8			Φ <sub>CY</sub>	MFM = 1
WCLK active time (high)	t <sub>0</sub>		2			2		2			Φ <sub>CY</sub>	
WCLK rise time	t <sub>r</sub>			20			20			20	ns	
WCLK fall time	t <sub>f</sub>			20			20			20	ns	
Preshift delay time from WCLK <sup>†</sup>	t <sub>CP</sub>	20		100	20		100	10		80	ns	
WCLK <sup>†</sup> → WE <sup>†</sup> delay	t <sub>CWE</sub>	20		100	20		100	10		80	ns	
WDA delay time from WCLK <sup>†</sup>	t <sub>CD</sub>	20		100	20		100	10		80	ns	
RDD active time (high)	t <sub>RDD</sub>	40			40			40			ns	

## AC Characteristics (cont)

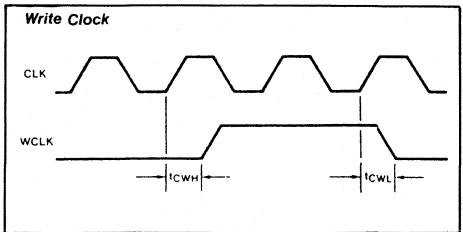
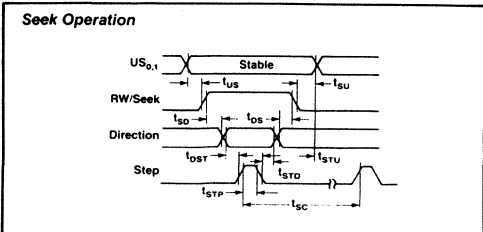
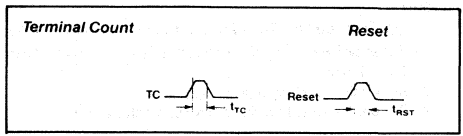
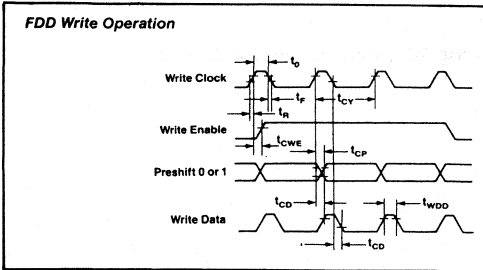
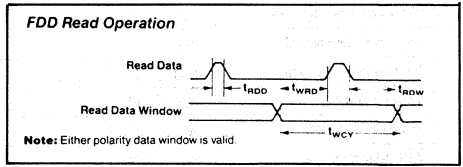
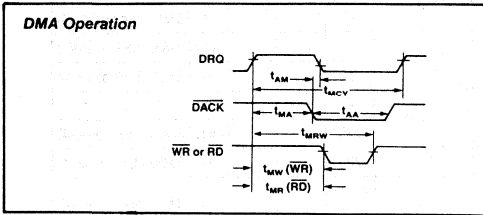
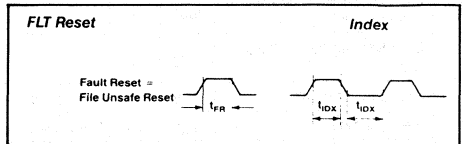
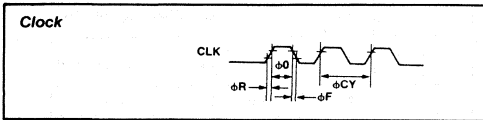
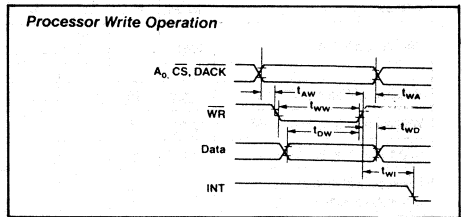
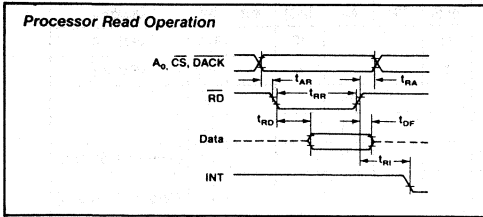
T<sub>A</sub> = -10°C to 70°C, V<sub>CC</sub> = +5V ± 5% for μPD765A/7265, otherwise V<sub>CC</sub> = +5V ± 10%

Parameter	Symbol	765A, 7265			765A-2, 7265-2			72065, 72066			Unit	Test Conditions
		Min	Typ (1)	Max	Min	Typ (1)	Max	Min	Typ (1)	Max		
Window cycle time	t <sub>WCY</sub>		16			16			16		°CY	MFM = 0
			8			8			8		°CY	MFM = 1
Window hold time to RDATA	t <sub>RDW</sub>	15			15			15			ns	
Window hold time from RDATA	t <sub>WRD</sub>	15			15			15			ns	
US <sub>0,1</sub> hold time to $\overline{RW}$ /seek†	t <sub>US</sub>	12			12			12			μs	8 MHz clock period(4)
$\overline{RW}$ /seek hold time to low current/direction†	t <sub>SD</sub>	7			7			7			μs	8 MHz clock period(4)
Low current/direction hold time to fault reset/step†	t <sub>DST</sub>	1			1			1			μs	8 MHz clock period(4)
US <sub>0,1</sub> hold time from fault reset/step 1	t <sub>STU</sub>	5			5			5			μs	8 MHz clock period(4)
Step active time (high)	t <sub>STP</sub>	6	7	8	6	7	8	6	7	8	μs	(Note 4)
Step cycle time	t <sub>SC</sub>	33	(Note 2)	(Note 2)	33	(Note 2)	(Note 2)	33			μs	(Note 4)
Fault reset active time (high)	t <sub>FR</sub>	8		10	8		10	8		10	μs	(Note 4)
Write data width	t <sub>WDD</sub>	t <sub>0</sub> -50			t <sub>0</sub> -50			t <sub>0</sub> -50			ns	
US <sub>0,1</sub> hold time after seek	t <sub>SU</sub>	15			15			15			μs	8 MHz clock period(4)
Seek hold time from DIR	t <sub>DS</sub>	30			30			30			μs	8 MHz clock period(4)
DIR hold time after step	t <sub>STD</sub>	24			24			24			μs	8 MHz clock period(4)
Index pulse width	t <sub>IDX</sub>	10			4			4			°CY	
$\overline{RD}$ delay from DRQ	t <sub>MR</sub>	800			800			125			ns	8 MHz clock period(4)
$\overline{WR}$ delay from DRQ	t <sub>MW</sub>	250			250			250			ns	8 MHz clock period(4)
$\overline{WR}$ or $\overline{RD}$ response time from DRQ†	t <sub>MRW</sub>			12			12				μs	8 MHz clock period(4)

### Note:

- (1) Typical values for T<sub>A</sub> = 25°C and nominal supply voltage.
- (2) Under software control. The range is form 1 ms to 16 ms at 8 MHz clock period, and 2 ms to 32 ms at 4 MHz clock period.
- (3) Sony Micro Floppydisk 3½" drive.
- (4) Double these values for a 4 MHz clock period.

**Timing Waveforms**



## Description

The μPB9201 floppy disk interface (FDI) is an LSI device that provides a wide range of functions commonly needed in a floppy disk controller design. A floppy disk controller design using the μPD765A and the μPB9201 requires only four to five chips, depending on individual requirements.

The digital phase lock loop implemented in the FDI simulates the function of an analog PLL. If higher resolution is required, the device provides for the addition of an external VCO chip. This essentially converts the digital PLL to an analog one. The external VCO is seldom required, however, due to the excellent performance of the digital PLL.

The FDI generates the write clock and processor clock for the μPD765A. The clocks are automatically switched in frequency when the 8" or 5-1/4" mode is selected. These clocks are changed synchronously so that random clock edges are not generated.

The FDI includes a precompensation circuit that allows delays of 0 ns, 125 ns, 187.5 ns, and 250 ns.

The on-chip drive select logic combined with the head load (HDL) signal eliminates the normally required selection logic. The on-chip buffers allow direct connections from  $\overline{DS}_0$ - $\overline{DS}_3$  and  $\overline{HS}_0$ - $\overline{HS}_3$  to the FDD.

The FDI provides the designer with the ability to delay the DRQ signal that normally goes from the FDC to the host DMA controller. The minimum delay is either 0.75 μs or 1.5 μs, depending on the selection of 8" or 5-1/4" mode. This allows the use of fast DMA controllers such as the μPD8237A-5.

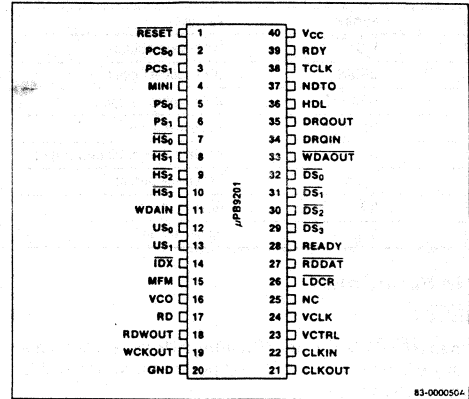
## Features

- Programmable digital write precompensation
- Write clock generation for 5-1/4" and 8" drives
- Data separation
- 5-1/4" and 8" drives select
- External VCO hook-up provision (optional)
- Processor clock generation
- Internal buffers capable of sinking 24 mA
- TTL-compatible
- Drive select logic
- Head select logic
- DRQ delay
- No data time out

## Ordering Information

Part Number	Package Type
μPB9201C	40-pin plastic DIP

## Pin Configuration



## Pin Identification

No.	Symbol	Function
1	RESET	Reset input
2, 3	PCS <sub>0</sub> , PCS <sub>1</sub>	Precompensation select input
4	MINI	Mode select
5, 6	PS <sub>0</sub> , PS <sub>1</sub>	Precompensation input
7-10	HS <sub>0</sub> -HS <sub>3</sub>	Head select
11	WDAIN	Write data input
12, 13	US <sub>0</sub> , US <sub>1</sub>	Unit select input
14	IDX	Index output
15	MFM	MFM mode input
16	VCO	VCO sync input
17	RD	Read data output
18	RDWOUT	Read data window output
19	WCKOUT	Write clock output
20	GND	Ground
21	CLKOUT	Clock output
22	CLKIN	Clock input
23	VCTRL	VCO control
24	VCLK	VCO clock input
25	NC	No connect
26	LDCR	Load control register input
27	RDDAT	Read data input
28	READY	Ready input
29-32	DS <sub>3</sub> -DS <sub>0</sub>	Drive select outputs

**Pin Identification (cont)**

No.	Symbol	Function
33	WDAOUT	Write data output
34	DRQIN	DMA request input
35	DRQOUT	DMA request output
36	HDL	Head load input
37	NDTO	No data time out input/output
38	TCLK	Test clock output
39	RDY	Ready output
40	V <sub>CC</sub>	Power supply

**Pin Functions****RESET**

When  $\overline{\text{RESET}}$  is low, the FDI internal logic is reset. This feature is used mainly for test purposes. Normally this pin is pulled high.

**HS<sub>0</sub>-HS<sub>3</sub>**

These head select outputs are derived from the head load and the US<sub>0</sub> - US<sub>1</sub> signals from the μPD765A. Each of these open collector output sinks 24 mA.

**PCS<sub>0</sub>, PCS<sub>1</sub>**

These inputs select the precompensation delay according to the following table:

PCS <sub>1</sub>	PCS <sub>0</sub>	Delay
0	0	0 ns
0	1	125 ns
1	0	187.5 ns
1	1	250 ns

**PS<sub>0</sub>, PS<sub>1</sub>**

These are the precompensation input signals from the μPD765A.

**WDAIN**

Write data from the μPD765A is input at this pin. It passes through the circuitry which is controlled by PS<sub>0</sub>, PS<sub>1</sub> and the FDI control register to provide various precompensation levels.

**MINI**

When this input is high, 5-1/4" mode is selected. When it is low, 8" mode is selected.

**US<sub>0</sub>, US<sub>1</sub>**

These are the unit select input pins. The μPD765A uses them to select up to four double-sided drives.

**IDX**

The FDI uses this signal to generate index pulses to the μPD765A when there is no data coming from the disk drive.

**MFM**

This signal controls the read data window to conform to MFM (double density) or FM (single density) recording modes. It also controls the frequency of the WCKOUT signal. MFM is input from the μPD765A.

**VCO**

This is the VCO sync input from the μPD765A. It is used for internal control.

**RD**

The read data output signal is the same as the data coming from the FDD but it has been shaped and synchronized to the 16 MHz clock. RD is directly connected to the RD signal of the μPD765A.

**RDWOUT**

This signal is generated by the FDI PLL circuitry. It is controlled by the MFM signal from the μPD765A and by the selection of 5-1/4" or 8" mode.

**WCKOUT**

This write clock output signal is output to the WCK pin of the μPD765A.

**CLKOUT**

This signal provides the processor clock for the μPD765A and is programmable via the FDI control register for an 8 MHz or 4 MHz square wave output. The switching between 4 MHz and 8 MHz is synchronous.

**CLKIN**

This input signal should be a 16 MHz TTL-compatible square wave. All timing for the FDI is derived from this signal.

**VCLK**

If an external VCO chip is used, this pin should be connected to the output of the VCO. If an external VCO is not used, then this pin should be connected to the 16 MHz clock input.

## VCTRL

This three-state signal controls the external VCO frequency. It is the equivalent of combined pump-up and pump-down signals.

## TCLK

This signal is used to test different modes of the FDI. Depending upon the mode, this pin outputs a 4 MHz, 8 MHz or 16 MHz square wave. It is not used in controller design.

## LDCR

This input signal is level triggered. When  $\overline{\text{LDCR}}$  is low,  $\text{PSC}_0$ ,  $\text{PSC}_1$ , and  $\text{MINI}$  are transferred to the internal control register. When  $\overline{\text{LDCR}}$  goes high, the data on pins 6-8 will remain latched. Pins 6-8 may be connected to a data bus and  $\overline{\text{LDCR}}$  may be used as a strobe, or they may be driven from external latches by connecting  $\overline{\text{LDCR}}$  to GND.

## RDDAT

This input is directly connected to the read data signal from the floppy disk interface.

## READY

This input signal is connected through an inverter to the FDD. The RDY output signal is generated by this signal.

## RDY

This output signal is directly connected to the RDY pin of the μPD765A. When the 8" mode is selected, the READY signals from the floppy disk drive is sent directly to the μPD765A. When the FDI is in the 5-1/4" mode, RDY is set to 1 at all times.

## DRQIN

This is an input from the μPD765A. DRQIN is delayed 3 to 4 clock pulses before being output (DRQOUT). This achieves the DRQ to RD delay that is required by the μPD765A.

## DRQOUT

This is the output of the delayed DRQIN signal.

## WDAOUT

This open collector output is directly connected to the floppy disk drive and writes data to it. WDAOUT sinks 24 mA.

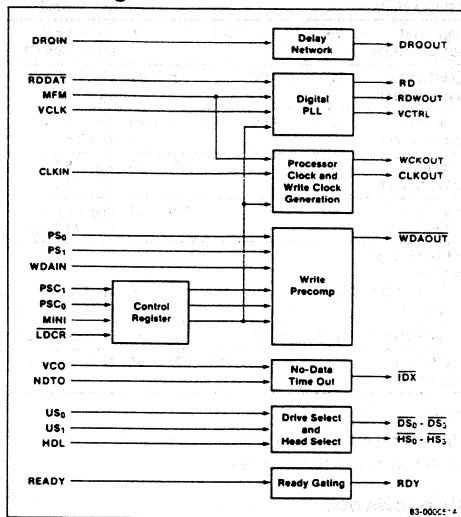
## NDTO

The FDI uses this pin to generate a time out when there is no data coming from the floppy disk drive. External RC components are required for the timing.

## HDL

The head load input is used in conjunction with the  $\text{US}_1$  and  $\text{US}_0$  signals from the μPD765A to generate the drive and head select signals.

## Block Diagram



## Absolute Maximum Ratings

$T_A = +25^\circ\text{C}$

Operating temperature, $T_{\text{OPT}}$	0 to $+70^\circ\text{C}$
Storage temperature, $T_{\text{STG}}$	$-65$ to $+150^\circ\text{C}$
All output voltages, $V_O$	$-5$ to $+5.5\text{ V}$
All input voltages, $V_I$	$-5$ to $+7\text{ V}$
Power supply voltage, $V_{\text{CC}}$	$-5$ to $+7\text{ V}$
Power dissipation, $P_D$	1.5 W

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**DC Characteristics**

T<sub>A</sub> = 0 to +70°C; V<sub>CC</sub> = +5 V ±10% unless otherwise specified

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Low level input voltage	V <sub>IL</sub>			0.8	V	
High level input voltage	V <sub>IH</sub>	2.0			V	
Input clamp voltage	V <sub>IC</sub>	1.5			V	V <sub>CC</sub> = 4.5 V I <sub>IH</sub> = -18 mA
Low level output voltage	V <sub>OL</sub>		0.3	0.5	V	V <sub>CC</sub> = 4.5 V I <sub>OL</sub> = 12 mA
High level output voltage	V <sub>OH</sub>	2.5	3.4		V	V <sub>CC</sub> = 4.5 V I <sub>OH</sub> = 1 mA (1)
Short circuit output current	I <sub>OS</sub>	-100		-25	mA	V <sub>CC</sub> = 5.5 V V <sub>O</sub> = 0 V
Low level input current	I <sub>IL</sub>		-100		μA	V <sub>CC</sub> = 5.5 V V <sub>I</sub> = 0.4 V
High level input current	I <sub>IH</sub>			20	μA	V <sub>CC</sub> = 5.5 V V <sub>I</sub> = 2.7 V
High level output current	I <sub>OH</sub>			100	μA	V <sub>CC</sub> = 4.5 V V <sub>O</sub> = 4.5 V (2)
Off state output current						
Three state output	I <sub>OZ1</sub>		-20		μA	V <sub>CC</sub> = 5.5 V V <sub>O</sub> = 0.4 V/2.7 V
Bidirectional	I <sub>OZ2</sub>		-100	+40	μA	(VCO CNTRL pin)
V <sub>CC</sub> supply current	I <sub>CC</sub>		170	296	mA	T <sub>A</sub> = +25°C

**Note:**

- (1) Does not apply to open collector outputs.
- (2) For open collector outputs only.

**Capacitance**

T<sub>A</sub> = +25°C; f<sub>C</sub> = 1 MHz

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Clock input	C <sub>IN</sub> (φ)			20	pF	All pins except those under test
Input	C <sub>IN</sub>			10	pF	Input tied to AC GND
Output	C <sub>OUT</sub>			15	pF	

**AC Characteristics**

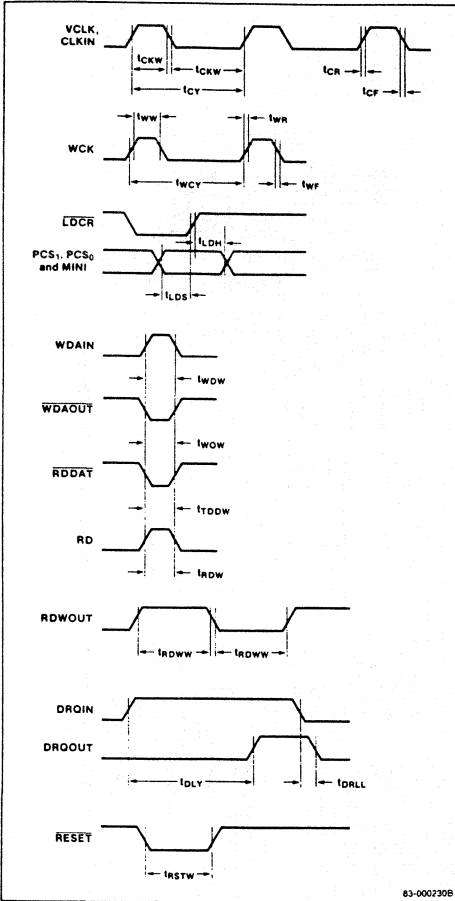
T<sub>A</sub> = 0 to +70°C; V<sub>CC</sub> = +5 V ±10% unless otherwise specified

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
CLKIN high and low width	t <sub>CKW</sub>	20			ns	
CLKIN period	t <sub>CY</sub>	55	62.5		ns	
CLKIN rise time	t <sub>CR</sub>			10	ns	
CLKIN fall time	t <sub>CF</sub>			10	ns	
WCK cycle time	t <sub>WCY</sub>		1		μs	MFM, 8"
			2		μs	FM, 8"
			2		μs	MFM, 5-1/4"
			4		μs	FM, 5-1/4"
						CLKIN = 16 MHz
WCK high width	t <sub>WW</sub>		250		ns	
WCK rise time	t <sub>WR</sub>			20	ns	
WCK fall time	t <sub>WF</sub>			20	ns	
PCS <sub>0</sub> , PCS <sub>1</sub> , MINI set up time to LDCR	t <sub>LDS</sub>	10			ns	
PCS <sub>0</sub> , PCS <sub>1</sub> , MINI hold time from LDCR	t <sub>LDH</sub>	10			ns	
WDAIN high width	t <sub>WDW</sub>	25			ns	
WDAOUT low width	t <sub>WOW</sub>		4t <sub>CY</sub>			t <sub>WOW</sub> = 250 ns where CLKIN = 6 MHz
RDDAT high width	t <sub>RDDW</sub>	25			ns	
RD high width	t <sub>RDW</sub>		2t <sub>CY</sub>			MINI = 0 MINI = 1
			4t <sub>CY</sub>			
RDWOUT width	t <sub>RDWW</sub>		1		μs	MFM, 8"
			2		μs	FM, 8"
			2		μs	MFM, 5-1/4"
			4		μs	FM, 5-1/4"
						CLKIN = 16 MHz
DRQOUT delay time from DRQIN	t <sub>DLY</sub>	0.75		1	μs	MINI = 0
		1.5		2	μs	MINI = 1
DRQOUT low from DRQIN low	t <sub>DRLL</sub>			30	ns	
RESET low width	t <sub>RSTW</sub>	250			ns	
VCLK period	t <sub>CY</sub>	55	62.5		ns	
VCLK high and low width	t <sub>CKW</sub>	20			ns	

**Note:**

The FDI is designed to run at 16 MHz, and all of the test conditions for signals generated by the FDI are at 16 MHz.

## Timing Waveforms



## Interfacing

Figure 1 shows all the required interconnections between the FDI and a typical FDC chip such as the μPD765A. An external 16 MHz clock input to the CLKIN pin is required. The FDI generates all the internal timing from this input clock.

An alternate method of utilizing the μPB9201 is shown in figure 2. This method minimizes the parts count and fully utilizes all of the FDI features.

The type of the drive can be selected by setting the value of the MINI pin; ie, MINI = 0: 8" and MINI = 1: 5-1/4". This can be achieved by either a jumper or a peripheral port.

The PCS<sub>0</sub> and PCS<sub>1</sub> pins are used to program the device for a desired amount of precompensation. The PS<sub>0</sub> and PS<sub>1</sub> signals from the μPD765A inform the FDI whether the bit shift is late, normal, or early.

The LDCR (load control register) pin can be used as a strobe to latch the values of MINI, PCS<sub>1</sub>, and PCS<sub>0</sub> into the control register of the FDI. Whenever LDCR is low, the control register is updated. If the strobing of LDCR is not preferred, then LDCR should be connected to ground and MINI, PCS<sub>1</sub>, and PCS<sub>0</sub> should be connected either to logic 1 or 0, depending upon the desired mode of operation.

The FDI uses the US<sub>1</sub>, US<sub>0</sub>, and HDSL signals from the μPD765A to generate the DS<sub>0</sub>-DS<sub>3</sub> (drive select) and the HS<sub>0</sub>-HS<sub>3</sub> (head select) signals. All these output signals are capable of sinking 24 mA and can be directly connected to the corresponding FDD signals. (This assumes that the FDD contains 220/330 termination resistors. Some drives contain 150-ohm pull-up resistors, which will require the use of a buffer external to the μPB9201.) The designer has two options available when using the head select signals. The first option is to connect all the head select signals together to the HEAD LOAD 0 signal of the FDD interface. This method generates one common "head load" signal for all drives. The second option is to add external delay circuits to each head select signal. This causes the head for the particular drive to stay loaded for the amount of specified time delay when the drive is deselected. The advantage of this method, as compared to the former one, is that it eliminates redundant head loading and unloading when copying diskettes from one to another.



Figure 1. Typical Personal Computer Application of the μPB9201

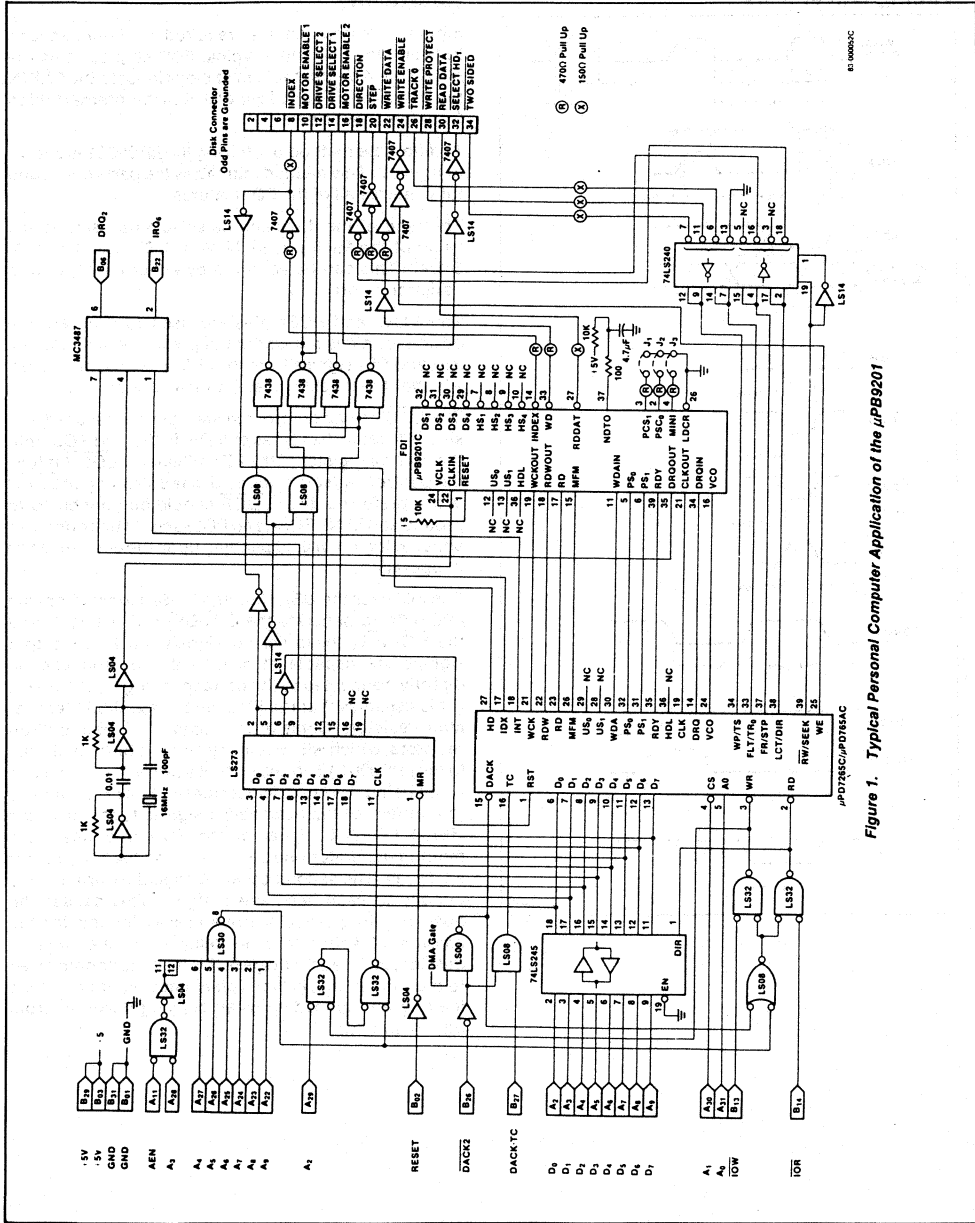
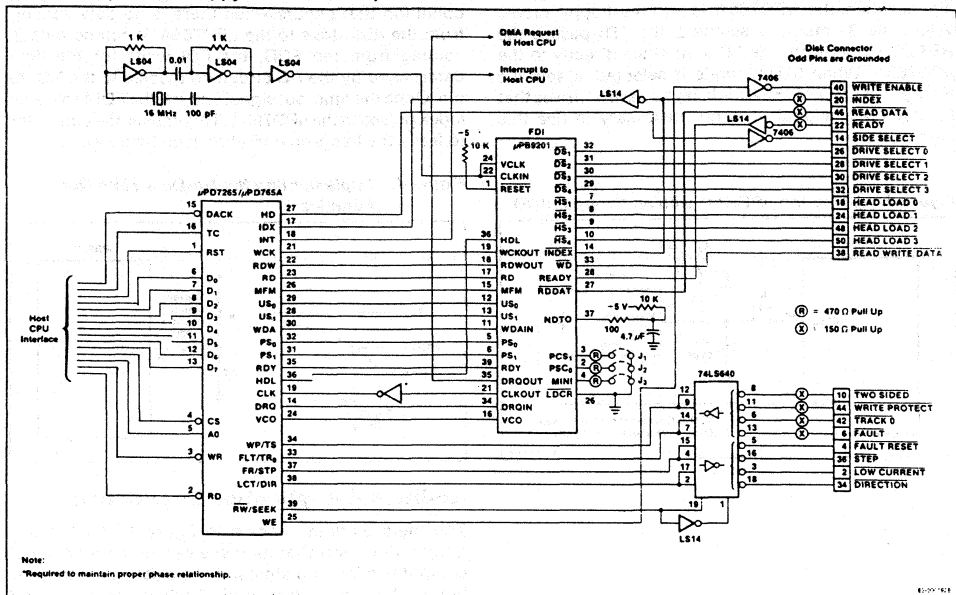


Figure 1. Typical Personal Computer Application of the μPB9201

Figure 2. Typical 8" Floppy Disk Controller; Minimum Parts Count



Optionally, an external VCO chip can be added to achieve better performance. As an example, figure 3 illustrates the necessary interconnections between the 74LS624 VCO chip and the FDI. The input frequency control of the VCO is connected to the VCTRL pin of the FDI through an integrator (a simple RC circuit). The VCTRL signal is the output of the internal digital phase comparator. When there is no data bit coming in, this pin stays at approximately 2.0 volts (high impedance state). Since the frequency control pin of 74LS624 is also at 2.0 volts (adjusted by R2), the voltage across R1 will be 0 volts. As a result of this, C1 is neither charged nor discharged and the VCO will be running at its nominal frequency (16 MHz).

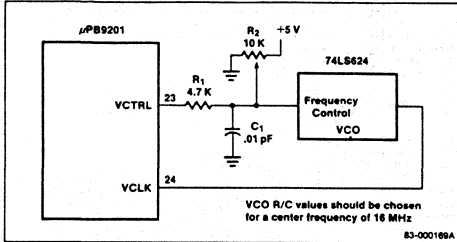
When a data bit occurs, the VCTRL pin goes first to a high state, then to a low state, and finally back to the high impedance state. The high and low states correspond to ramp-up and ramp-down respectively. The duration of ramp-up and ramp-down are determined by the position of the data bit in the read window.

If the data always arrives early, then ramp-up will have a longer period than ramp-down, causing an increase in VCO frequency. If the data arrives late, the converse is true. The integrator averages the frequency changes of the signal coming from the VCTRL pin. The values of R1 and C1 determine the time constant for the integrator. These values can be selected so that the VCO follows the slow speed variations of the disk drive. The VCLK pin should be connected to the output of the VCO when using the external VCO. If the VCO is not used, then the VCLK pin should be connected to the 16 MHz input clock.

The μPD765A requires a fairly long delay from DRQ going high to the issuance of a READ pulse to the chip. It is usually necessary to delay the DRQ signal going to the host DMA controller so that the READ pulse does not arrive early. The FDI is capable of delaying the DRQ from the μPD765A controller for approximately 1 μs (8" drive), or 2 μs for a 5-1/4" drive. In figure 1, the DRQ from the μPD765A is connected to the DRQIN pin of the FDI and the DRQOUT is connected to the host DMA controller. DRQOUT is automatically reset when DRQIN goes low.

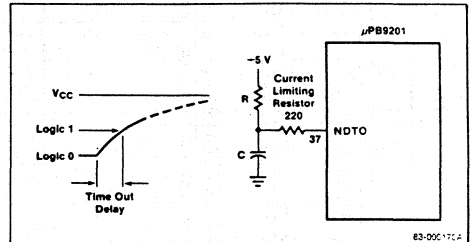
The FDI provides the necessary logic for the READY signal when the μPD765A is in mini-floppy mode. When the 8" mode is selected, the FDI passes the READY signal from the FDD interface directly to the μPD765A. When 5-1/4" mode is selected, it sets the RDY pin of the μPD765A high. If you have 5" drives that have a ready signal, it is not necessary to use this signal.

**Figure 3. Using the μPB9201 with an External VCO**



The FDI is capable of correcting a rare hang-up condition that occurs when there is no data coming from the disk drive to the μPD765A. When no data is coming from the FDD, the FDI waits for the time determined by the RC circuit connected to the NDTO pin. Once the time-out signal occurs, the FDI generates index pulses to the μPD765A. This causes the controller to leave the hang-up condition (see figure 4).

**Figure 4. Implementing the No-Data Time Out Function**



**Additional Application Information**

The logic diagram, shown in figure 1, illustrates a floppy disk controller as implemented on a personal computer. It is compatible with the existing controllers, but has the ability to control 8" drives and single and double density as well.

### Description

The μPD71065 and μPD71066 are CMOS devices that interface a floppy-disk drive (FDD) with a floppy-disk controller (FDC). The controller can be μPD765A, μPD7265, μPD72065, μPD72066, μPD7260, or one of the FD179X series.

The floppy-disk interface can operate at various data rates, including the 300-kb/s rate that results from using high-density 5-inch drives with media formatted at the standard 250-kb/s rate. Also, the μPD71065/66 generates the write clock needed by the selected controller and provides synchronous switching when changing data rates.

### Features

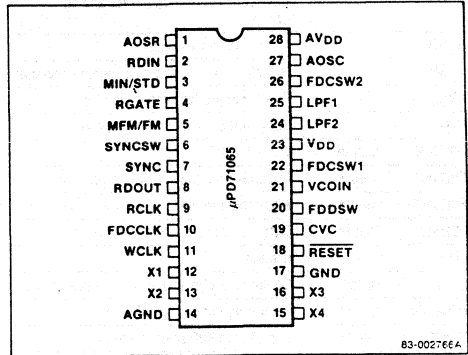
- Compatible with all industry-standard controllers
- Multiple data rates: 500/300/250/150/125 kb/s
- Internal or external sync field detection logic
- Head-loading timer for FD179X-series controllers
- No analog adjustments required
- CMOS, low power consumption
- 5-volt power supply

### Ordering Information

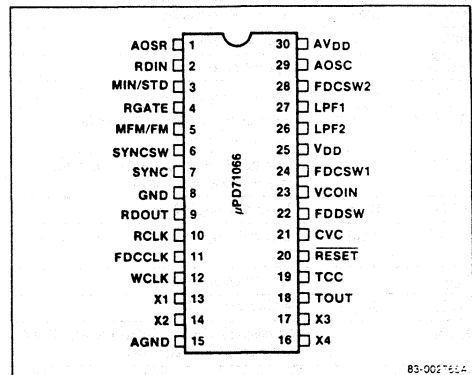
Part Number	Package	Internal Timer
μPD71065G	28-pin plastic SO	Not included
μPD71066CT	30-pin plastic shrink DIP	Implemented to FD179X-series controllers as head-loading timer.

### Pin Configurations

#### 28-Pin Plastic SO



#### 30-Pin Plastic Shrink DIP



**Pin Identification**

Symbol	Input/Output	Function
ACOS		Capacitor connection pin for analog one-shot
AGND		Ground for analog circuits
AOSR		Resistor connection pin for analog one-shot
AVDD		Power supply for analog circuits
CVC		Capacitor connection pin for VCO
FDCCLK	Output	Clock to FDC
FDCSW1	Input*	FDC selection pin or timer trigger input
FDCSW2	Input*	FDC selection pin
FDDSW	Input*	Data transfer rate selection pin
GND		Ground
LPF1, LPF2	Output	Connection pins to external lowpass filter
MFM/FM	Input*	Recording density selection pin
MIN/STD	Input*	5- or 8-inch FDD selection pin
RCLK	Output	Read data sampling clock
RDOUT	Output	Read data to FDC
RGATE	Input*	Read enable/disable
RDIN	Input*	Read data from FDD
RESET	Input*	System reset
SYNC	Input*	External PLL gain selection
SYNCW	Input*	Determines whether gain selection is internal or external
TCC		External RC time constant connection to internal timer (μPD71066)
TOUT	Output	Timer signal (μPD71066)
VCOIN	Input	External lowpass filter output to internal VCO
VDD		+5-volt power supply
WCLK	Output	Write clock to FDC
X1, X2		Connection pins for 16-MHz crystal (X1, X2) or external clock input (X1)
X3, X4		Connection pins for 19.2-MHz crystal (X3, X4) or external clock input (X3)

\*Input pin has an on-chip pull-up resistor

**Pin Functions**

The following paragraphs supplement the brief descriptions of certain pins in the preceding table. Pin symbols are in alphabetical order.

**FDCSW1 and FDCSW2.** The μPD71065/66 is configured for the applicable FDC by applying logic levels L and H (or open) to these pins.

FDCSW1	FDCSW2	Floppy-Disk Controller
Open or H	Open or H	μPD765A/7265
L	Open or H	μPD7260
*	L	FD179X series

\* FDCSW1 is the trigger input to the timer circuit when FDCSW2 is low.

**FDDSW.** The logic level applied to this pin selects the data transfer rate of the FDD.

FDDSW	Data Transfer Rate
Open or H	500/250/125 kb/s
L	500/250/300/150 kb/s

**MFM/FM Pin.** The logic level applied to this pin and the FDCSW2 pin selects the modulation type. Double-density and single-density recording use MFM (modified FM) and FM modulation, respectively.

FDCSW2	MFM/FM	Modulation
H	H	MFM
H	L	FM
L	H	FM
L	L	MFM

**MIN/STD.** Logic level L on this pin selects a 5-inch FDD. An open or H selects an 8-inch FDD.

**RDIN.** This is a composite read data and clock signal input from the FDD.

**RDOUT.** The read data output from this pin is synchronized with the read clock (RCLK) derived from the RDIN composite signal.

**RGATE.** In conjunction with FDCSW2, RGATE enables or disables the read operation that is sent from the FDC.

FDCSW2	RGATE	Read Operation
H	H	Enable
H	L	Disable
L	H	Disable
L	L	Enable

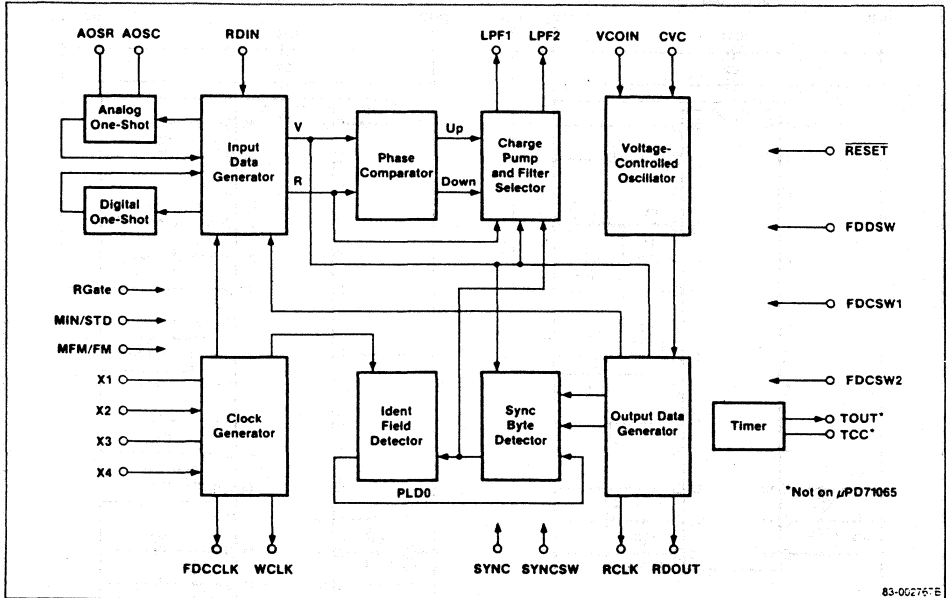
**SYNC and SYNC<sub>SW</sub>.** The PLL gain is determined by the input signal at the SYNC pin and the logic levels at the FDCSW1 and SYNC<sub>SW</sub> pins.

FDCSW1	SYNC <sub>SW</sub>	SYNC	PLL Gain
Open or H	Open or H	H (1)	Low
		L (1)	High
L	L	H (2)	Low
		L (2)	High

**Note:**

- (1) Input signal at SYNC is the PLL gain selection signal between the ID and DATA fields.
- (2) Input signal at SYNC is the SYNC field detection signal from the FDC.

### Block Diagram



Functions of the block diagram components are explained below.

**Clock Generator.** Using both 16-MHz and 19.2-MHz oscillators, outputs clock signals corresponding to the mode used to the FDCCLK and WCLK pins.

**Input Data Generator.** According to the input data, generates the R and V signals to be input to the phase comparator. In addition to this, the input data generator determines whether the analog one-shot circuit or the digital one-shot circuit is used.

**Charge Pump and Filter Selector.** According to the PLL (phase-locked loop) gain selection signal, enables or disables the LPP2 side charge pump to control the PLL gain.

**Output Data Generator.** Generates the window signal (RCLK) and read data signal (RDOUT) depending on the mode and FDC to be used.

**Sync Byte Detector.** Detects the sync field within 16 to 20 pulses regardless of FM or MFM mode.

**Ident Field Detector.** Determines whether the sync field detected by the sync byte detector is ID or DATA field and sets the PLL gain.

**Basic External Circuit**

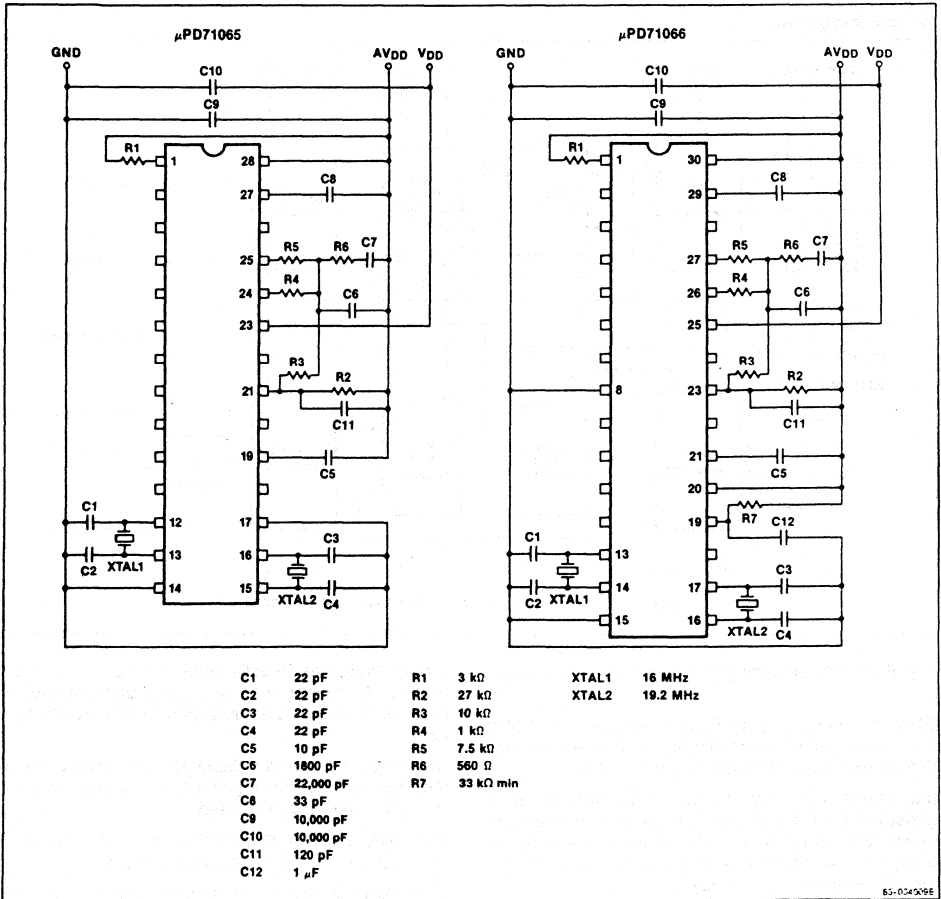
Figure 1 shows the basic external circuit including the lowpass filter and crystals. The data transfer rate is selected by strapping pins FDDSW, MIN/STD, and MFM/FM to L (low) or open (high). See table 1.

The VCO frequency and the phase delay between RDIN and RDOUT can be optimized by adjusting resistors R2 and R1, respectively.

**VCO Frequency**

For this procedure, the data transfer rate is undefined. Strap RGATE to H and RDIN to L. Adjust resistor R2 to set the VCO frequency at the RCLK pin to the same numerical value as the data transfer rate; for example, 500 kHz and 500 kb/s.

**Figure 1. Basic External Circuit**



85-0042X06E

### Data Read Phase Delay

For this procedure, set the data transfer rate to 500 kb/s, set the RDIN signal to a 2-μs cycle time, and strap RGATE to H. Adjust resistor R1 to set the value of  $t_{STW}$  (figure 2) to 950 ns.

Figure 2. Read Data Timing Diagram

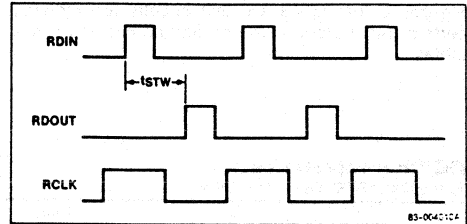


Table 1. Data Transfer Rate Selection

Floppy-Disk Controllers	Data Transfer Rate (kb/s)	Clock Output Frequencies from μPD71065/71066			Selection Pins (Note 1)		
		FDCLK (MHz)	RCLK (kHz)	WCLK (kHz)	FDCSW	MIN/STD	MFM/FM
μPD765A, μPD7265, μPD72065, μPD72066 (Note 2)	250	4	250	500	Open	Open	Open
	125	4	125	250	Open	Open	L
	500	8	500	1 MHz	Open	L	Open
	250	8	250	500	Open	L	L
	300	4.8	300	600	L	Open	Open
	150	4.8	150	300	L	Open	L
	500	8	500	1 MHz	L	L	Open
μPD7260 (Note 3)	250	8	250	500	L	L	L
	250	4	500	500	Open	Open	Open
	125	4	250	250	Open	Open	L
	500	8	1 MHz	1 MHz	Open	L	Open
	250	8	500	500	Open	L	L
	300	4.8	600	600	L	Open	Open
	150	4.8	300	300	L	Open	L
FD179X Series (Note 4)	500	8	1 MHz	1 MHz	L	L	Open
	250	8	500	500	L	L	L
	250	1	250	500	Open	Open	L
	125	1	125	250	Open	Open	Open
	500	2	500	1 MHz	Open	L	L
	250	2	250	500	Open	L	Open
	300	1.2	300	600	L	Open	L
	150	1.2	150	300	L	Open	Open
	500	2	500	1 MHz	L	L	L
	250	2	250	500	L	L	Open

**Note:**

- (1) Selection pin states: L = low; Open = open or H (high)
- (2) μPD765A/7265/72065/72066: FDCSW1 and FDCSW2 = Open
- (3) μPD7260: FDCSW1 = L and FDCSW2 = Open. FDCLK clock is not used
- (4) FD179X Series: FDCSW1 = Don't care and FDCSW2 = L. WCLK clock is not used.



**Electrical Characteristics**

Figures 3 through 8 are test circuits for verifying certain parameters in the dc and ac characteristics tables.

**Absolute Maximum Ratings**

T<sub>A</sub> = +25°C

Power supply voltage, V <sub>DD</sub>	-0.3 to +6 V
Input voltage, V <sub>I</sub>	-0.3 to V <sub>DD</sub> + 0.3 V
Output voltage, V <sub>O</sub>	-0.3 to V <sub>DD</sub> + 0.3 V
Operation temperature, T <sub>OP</sub> T	-10 to +70°C
Storage temperature, T <sub>ST</sub> G	-40 to +125°C

**DC Characteristics**

T<sub>A</sub> = -10 to +70°C; V<sub>DD</sub> = +5 V ±10%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage, low	V <sub>IL</sub>	-0.3		0.8	V	
Input voltage, high	V <sub>IH</sub>	2.2		V <sub>DD</sub> + 0.3	V	
Output voltage, low	V <sub>OL</sub>			0.45	V	I <sub>OL</sub> = 2 mA
Output voltage, high	V <sub>OH</sub>	0.7 V <sub>DD</sub>		V <sub>DD</sub>	V	I <sub>OH</sub> = -200 μA
Clock input level	V <sub>KD-P</sub>	1		V <sub>DD</sub>	V	
Input leakage current, low	I <sub>LIL</sub>	-150		-50	μA	V <sub>I</sub> = 0 V
Input leakage current, high	I <sub>LIH</sub>	-10		+10	μA	V <sub>I</sub> = V <sub>DD</sub>
Output leakage current, low	I <sub>LOL</sub>	-10			μA	V <sub>O</sub> = 0.45 V
Output leakage current, high	I <sub>LOH</sub>			+10	μA	V <sub>O</sub> = V <sub>DD</sub>
Power supply current	I <sub>DD</sub>			25	mA	XTAL: 16 MHz, 19.2 MHz
				20	mA	XTAL: 16 MHz

**AC Characteristics**

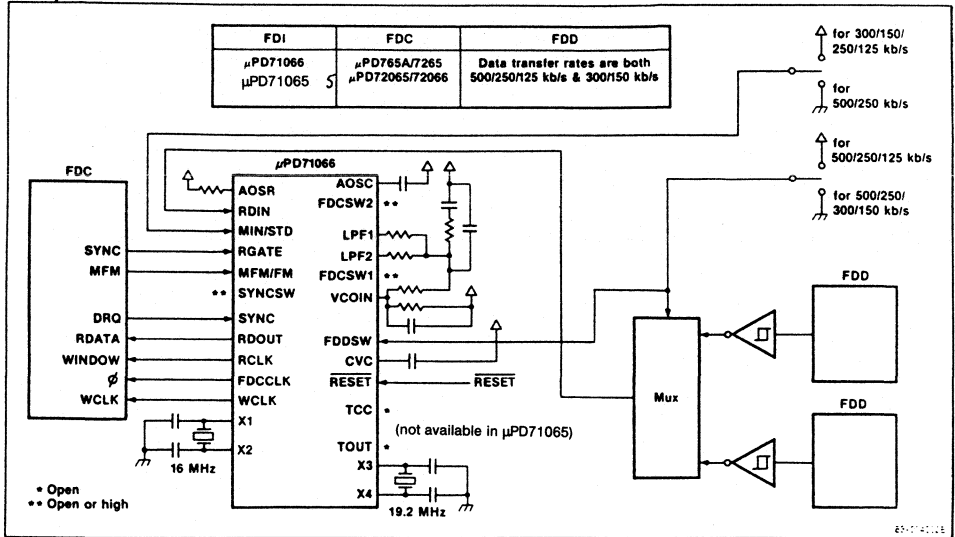
T<sub>A</sub> = -10 to +70°C; V<sub>DD</sub> = +5 V ±10%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Rise time	t <sub>R</sub>	0		20	ns	
Fall time	t <sub>F</sub>	0		20	ns	
RDOU <sub>T</sub> setup time to RCLK ↑	t <sub>SRR</sub>	40			ns	For μPD7260
CLK high/low level width	t <sub>KK</sub>	20			ns	
VCO oscillation frequency	f <sub>o</sub>			8	MHz	V <sub>F</sub> = V <sub>DD</sub>
VCO free-run frequency	f <sub>i</sub>	3.6	4	4.4	MHz	FDDSW = H, V <sub>F</sub> = open
		2.1	2.4	2.7	MHz	FDDSW = L, V <sub>F</sub> = open
VCO control voltage sensitivity	K <sub>V</sub>	2.5	3.5	4.6	MHz/V	(V <sub>DD</sub> /2) - V <sub>F</sub>   ≤ 0.5 V
K <sub>V</sub> voltage coefficient	ΔK <sub>V</sub> /V <sub>DD</sub>	-1	-19	-22	%/V	
f <sub>i</sub> power supply voltage coefficient	Δf <sub>i</sub> /V <sub>DD</sub>	0		5	%/V	
f <sub>i</sub> temperature coefficient	Δf <sub>i</sub> /T <sub>A</sub>	0	-500	-1000	ppm/°C	
Phase detect sensitivity	K <sub>P</sub>	0.7	0.8	0.9	V/rad	
RCLK jitter	t <sub>j</sub>	0	30	50	ns	500-kb/s mode
RDIN ↑ to RDOU <sub>T</sub> ↑ delay time	t <sub>DRR</sub>	900	950	1000	ns	
Capture range (Note 1)	f <sub>CAP</sub>	537		427	kHz	500-kb/s mode
		286		213	kHz	250-kb/s mode
		143		107	kHz	125-kb/s mode
		343		256	kHz	300-kb/s mode
		172		128	kHz	150-kb/s mode

**Note:**

(1) The frequencies in the Max and Min columns are the lower and upper limits, respectively, of the capture range. For example, in the 500-kb/s mode, the capture range is from 427 kHz (or lower) to 537 kHz (or higher).

Figure 3. System Example μPD71066 FDI and μPD765A FDC





## 1.1 System Outline

The μPD72067 Advanced Floppy Disk Controller (AFDC) creates a new generation of intelligent disk controllers integrating nearly all parts necessary in a floppy disk controller system. It includes:

- digital data separation
- clock generation circuit
- write precompensation logic

The data separator uses a new principle with a multi-level quantised phase comparator and a frequency tracking algorithm. Without any external components it provides about 85% of jitter margin and ±10% of speed tolerance. The write precompensation circuit is programmable with pins or software. Also the clock selection for data transfer rate specification is done in that way. Two crystals with 32MHz for standard data rates and 19.2MHz for the 300 kbit/s data rate are directly connected.

The AFDC can be used in internal and external mode. After poweron or reset, the chip enters external mode. All functions are selected by hardware. The chip is compatible with the industry standard μPD765A or its CMOS equivalent μPD72065. By switching the ISO/ECMA format pin, also this disk format is supported. When using a special command, the AFDC enters into internal mode where all functions are software selectable with additional auxiliary commands. For special applications, an external high precision PLL can be supported in this mode.

The host interface and the disk drive interface are designed in the same way as in the μPD765A or μPD72065. A typical block diagram with 4 disk drives connected is shown in fig. 1.

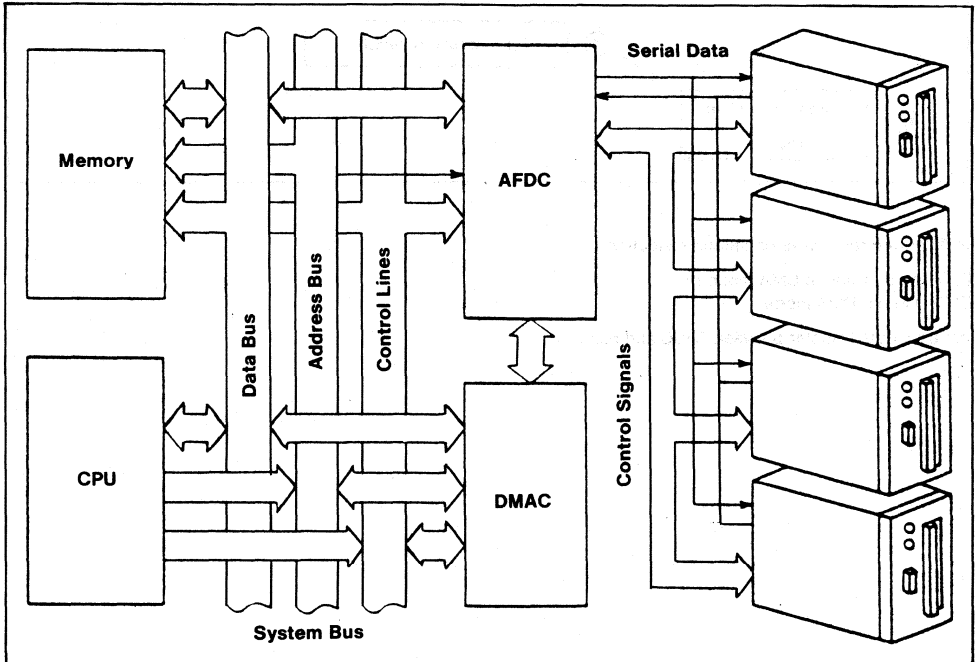
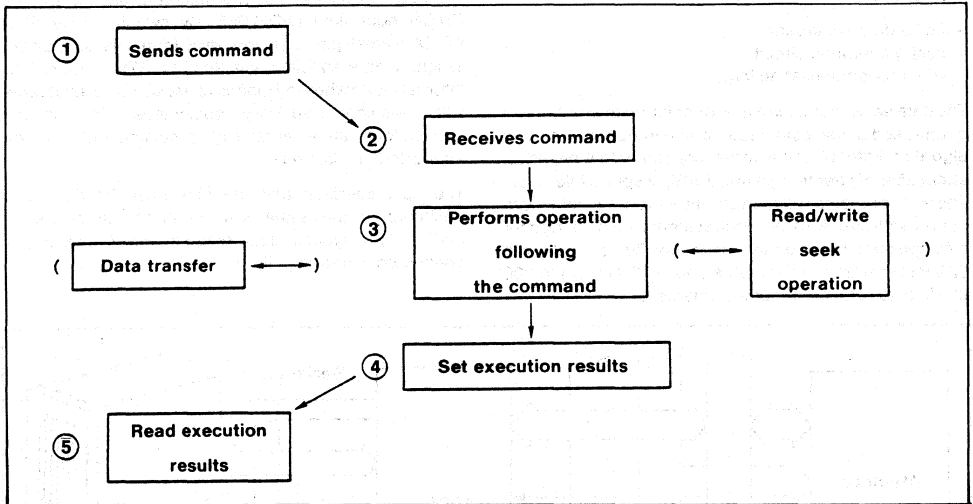


Fig. 1.1 System Configuration Example

**1.2 System Operation Outline**

The μPD72067 system operation is as follows:

(Host)



In ③, during read/write command execution,

- DMA controller in DMA mode or
- CPU in nonDMA mode

controls data transfer between AFDC and memory.

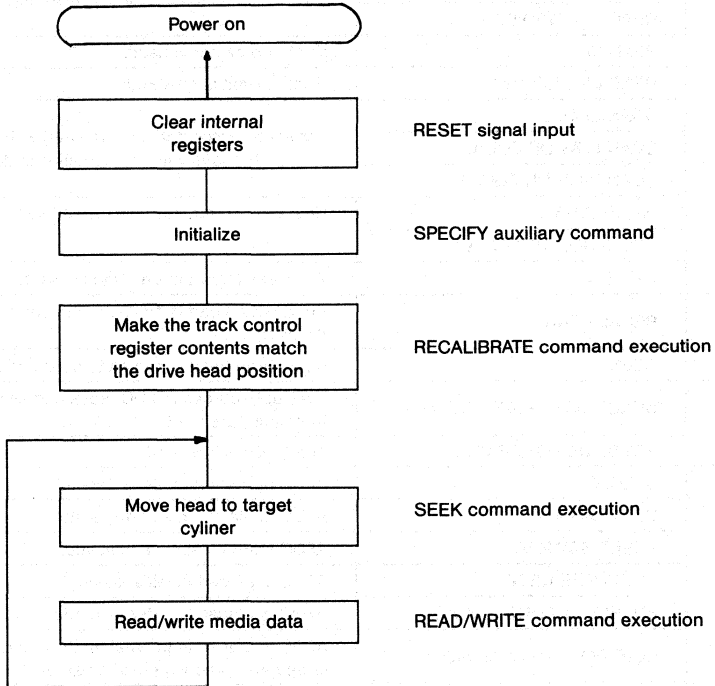
### 1.3 Command Types

The μPD72067 contains the following 15 commands and eight auxiliary commands.

Classification	Command name	Outline
Read	READ DATA	Data on specified sector is transferred to the host.
	READ DELETED DATA	
	READ ID	ID of one sector is used.
	READ DIAGNOSTIC	Track format is checked.
	SCAN EQUAL	For each sector, data is compared with host data and sector satisfying the condition is detected.
	SCAN LOW OR EQUAL	
	SCAN HIGH OR EQUAL	
Write	WRITE DATA	Host data is transferred to specified sector.
	WRITE DELETED DATA	
	WRITE ID	The track format of one track is written.
Seek	RECALIBRATE	Read/write head is moved to the outermost track (track 0).
	SEEK	Read/write head is moved to specified cylinder.
Sense	SENSE INTERRUPT STATUS	Interrupt sources in FDC (seek end and state transition) are read.
	SENSE DEVICE STATUS	The FDD state is read.
Initialize	SPECIFY	AFDC operation mode is defined.
Auxiliary	SET STANDBY	AFDC is placed in standby state.
	RESET STANDBY	AFDC standby state is released.
	SOFTWARE RESET	AFDC is placed in initial state.
	ENABLE EXTERNAL MODE	AFDC is placed in external mode.
	CONTROL INTERNAL MODE	AFDC is placed in internal mode, and data transfer speed and precompensation amount are set.
	ENABLE MOTORS	On/off control of spindle motor.
	SELECT FORMAT	IBM or ECMA/ISO format is selected.
	START CLOCK	Clock generation circuit operation is initiated.

1.4 Basic Operation

Access media (floppy disk) by using the μPD72067 in the following sequence:



The operation steps to RECALIBRATE command execution shown above must always be performed in the sequence as shown (containing the case where reset is made before completion of operation).

## 1.5 Features

- μPD765A/7265 and μPD72065/72066 command compatible
- IBM Diskette 1 (single side 128, 256, or 512 bytes/sector compatible)
- IBM Diskette 2 (double-sided 256 bytes/sector) compatible
- IBM Diskette 2D (double-sided double-density 256 or 1024 bytes/sector) compatible
- ECMA/ISO minifloppy disk format compatible
- ECMA66 (ISO/TC 97/SC11 N419): Single-sided single density (256 bytes/sector)
- ECMA70 (ISO/TC 97/SC11 N475): Double-sided double-density (256 bytes/sector)
- Data transfer speed: 300 kbps (for compatibility with high-density floppy disk drives (Note 1))
- Standby function
- Internal window signal generation VFO (DPLL)
- Internal write precompensation circuit
- Internal system clock generation circuit
- Internal write clock generation circuit
- External VFO (such as μPD71065/71066) can be connected
- Spindle motor control function
- FM, MFM control (specified in each command during read/write)
- Variable recording length: 128, 256, 512, 1024, 2048, 4096, or 8192 bytes/sector
- Multisector function (Note 2)
- Multitrack function (Note 3)
- Up to four floppy disk drives can be connected. (Single-sided single and double-sided double-density floppy disk drives can be mixed.)
- Seek operation can be performed on a maximum of four floppy disk drives at a time. (During drive seeking, a seek command can be issued to another drive.)
- Incomplete sector read/write can be specified (only when 128 bytes/sector = FM)
- Internal CRC generation and check function ( $x^{10} + x^{12} + x^5 + 1$ )
- Programmable step speed (specified by using SPECIFY command)
- Programmable head-load time and head-unload time (specified by using SPECIFY command)

- Data scan function (detection of sector satisfying equal-to, greater-than, or less-than condition with main memory data)
- DMA or nonDMA (interrupt) data transfer (as specified by using SPECIFY command)
- CMOS
- Single +5V power supply

## Ordering information

Order name	Package
μPD72067C	48-pin plastic dual-in-line package (600 mil)
μPD72067GC-3B6	52-pin plastic flatpack (normal bend)
μPD72067L	52-pin PLCC



**SPECIFICATION**

**Absolute Maximum Ratings**

TA = 25°C

Power supply voltage, VCC	-0.5 to + 7V
Input voltage, VI	-0.5 to + 7V
Output voltage, VO	-0.5 to + 7V
Operating temperature, T <sub>OPT</sub>	-10°C to + 70°C
Storage temperature, T <sub>STG</sub> (μPD765A/7265)	-40°C to + 125°C
Storage temperature, T <sub>STG</sub> (μPD765A-2/72065/72066)	-65° C to +150° C

**Comment:** Exposing the device to stresses above those listed in the Absolute Maximum Ratings could cause permanent damage. The device should not be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**DC Characteristics**

TA = -10°C to +70° C, VCC = +5V ±10%

Parameter	Symbol	Min	Max	Unit	Test Conditions
Input voltage low	V <sub>IL</sub>	- 0.5	0.8	V	
Input voltage high	V <sub>IH</sub>	2.2	V <sub>CC</sub> +0.5	V	
Output voltage low	V <sub>OL</sub>		0.45	V	I <sub>OL</sub> = 2.0 mA
Output voltage high	V <sub>OH</sub>	2.4	V <sub>CC</sub>	V	I <sub>OH</sub> = - 200μA
Input voltage low (CLK + WR clock)	V <sub>IL</sub> ( )	- 0.5	0.65	V	
Input voltage high (CLK + WCLK)	V <sub>IH</sub> ( )	2.2	V <sub>CC</sub> +0.5	V	
Supply current (VCC)	I <sub>CC</sub>		70	mA	= 125ns
Input load current high	I <sub>LIH</sub>		10	μA	V <sub>IN</sub> = V <sub>CC</sub>
Input load current low	I <sub>LIL</sub>		- 10	μA	V <sub>IN</sub> = 0V
Output leakage current high	I <sub>LOH</sub>		10	μA	V <sub>OUT</sub> = V <sub>CC</sub>
Output leakage current low	I <sub>LOL</sub>		- 10	μA	V <sub>OUT</sub> = +0.45V

**Capacitance**

TA = 25°C, f<sub>C</sub> = 1 MHz, VCC = 0V

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input clock capacitance	C <sub>IN</sub> (0)			20	pF	(Note 1)
Input capacitance	C <sub>IN</sub>			10	pF	(Note 1)
Output capacitance	C <sub>OUT</sub>			20	pF	(Note 1)

**Note:**

(1) All pins except pin under test tied to AC ground

## AC Characteristics

T<sub>A</sub> = -10°C to +70°C, V<sub>CC</sub> = +5V ±10%

Parameter	Symbol	32MHz (DR1 = 0)			19.2MHz (DR1 = 1)			Unit	Test Conditions
		Min	Typ(1)	Max	Min	Typ(1)	Max		
Clock period	ØCY	2.5	31.25	125	25	52.08	125	ns	
Clock active (high)	0	7			7			ns	
Clock rise time	r	5			5			ns	
Clock fall time	f	5			5			ns	
A <sub>0</sub> , CS, DACK setup time to RD↓	tAr	0			0			ns	
A <sub>0</sub> , CS, DACK hold time from RD↑	tRA	0			0			ns	
RD width	tRR	200			200			ns	
Data access time from RD↓	tRD			140			140	ns	C <sub>L</sub> = 100 pF
DB to float delay time from RD↑	tDF	10		85	10		85	ns	C <sub>L</sub> = 100 pF
A <sub>0</sub> , CS DACK setup time to WR↓	tAW	0			0			ns	
A <sub>0</sub> , CS, DACK hold time to WR↑	tWA	0			0			ns	
WR width	tWW	200			200			ns	
Data setup time to WR↑	tDW	100			100			ns	
Data hold time from WR↑	tWD	0			0			ns	
INT delay time from RD↑ (2)	tRI			400			400	ns	
INT delay time from WR↑ (2)	tWI			400			400	ns	
DRY cycle time	tMCY	13						μs	(3)
DACK↓→DRQ↓ delay	tAM			140			140	ns	
DRQ↑→DACK↓ delay	tMA	200			333.3			ns	(3)
DACK width	tAA	8			8			CY	(3)
DRQ↑→RD↓	tMR	125			208.3			ns	(3)
DRQ↑→WR↓	tMW	250			416.7			ns	(3)
DRQ↑→WR/RD↓	tMRW			12			12	ns	
TC width	tTC	60			60				
Reset width	tRST	56			56			CY	(3)
Clock hold time at standby (4)	tWC	128			128			CY	(3)
Clock hold time release (4)	tCW	64			64			CY	(3)
Window cycle time (5)	tWCY		2			3.33		ns	MFM = 0 (3)
			1			1.67		ns	MFM = 1 (3)
Window hold time to RDATA (5)	tRDW	15			15			ns	
Window hold time from RDATA (5)	tWRD	15			15			ns	
US <sub>0,1</sub> hold time to RW/seek†	tUS	12			20			μs	(3)
RW/seek hold time to low current/direction†	tSD	7			11.7			μs	(3)
Low current/direction hold time to fault reset/step†	tDST	1			1.7			μs	(3)

### AC Characteristics (cont.)

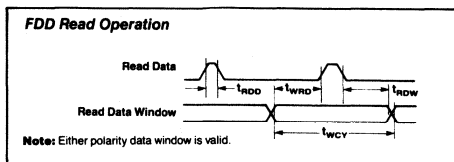
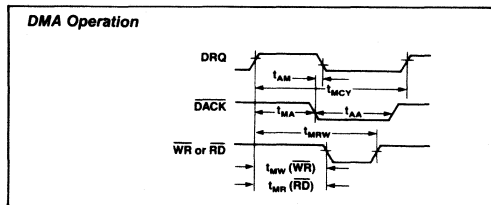
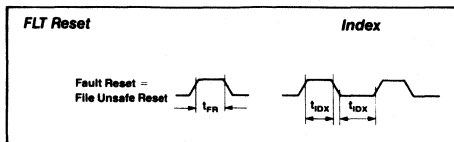
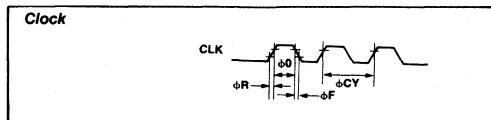
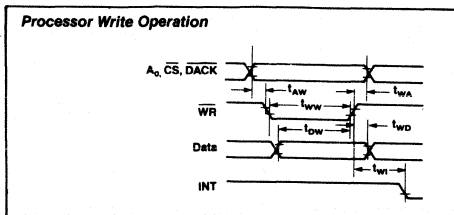
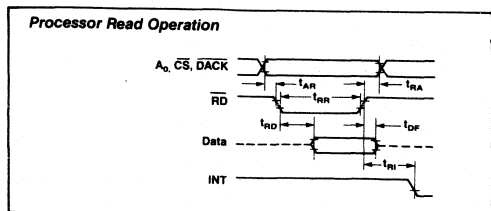
T<sub>A</sub> = - 10°C to + 70°C, V<sub>CC</sub> = + 5V ±10%

Parameter	Symbol	32MHz (DR1 = 0)			19.2MHz (DR1 = 1)			Unit	Test Conditions
		Min	Typ (1)	Max	Min	Typ (1)	Max		
US <sub>0,1</sub> hold time from fault reset/ step 1	t <sub>STU</sub>	5			8.3			μs	(3)
Step active time (high)	t <sub>STP</sub>	6	7	8	10	11.7	13.3	μs	(3)
Step cycle time	t <sub>SC</sub>	33			55			μs	(3)
Fault reset active time (high)	t <sub>FR</sub>	8		10	13.3		16.7	μs	(3)
Write data width	t <sub>WDD</sub>	t <sub>0-50</sub>			t <sub>0-50</sub>			ns	
US <sub>0,1</sub> hold time after seek (6)	t <sub>SU</sub>	15			25			μs	(3)
Seek hold time from DIR	t <sub>DS</sub>	30			50			μs	(3)
DIR hold time after step	t <sub>STD</sub>	24			40			μs	(3)
Index pulse width	t <sub>IDX</sub>	16			16			CY	(3)

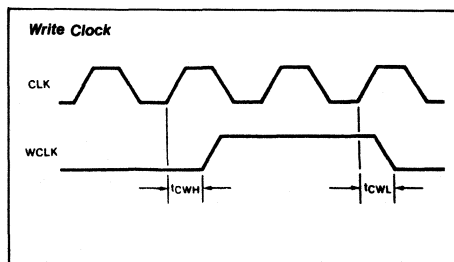
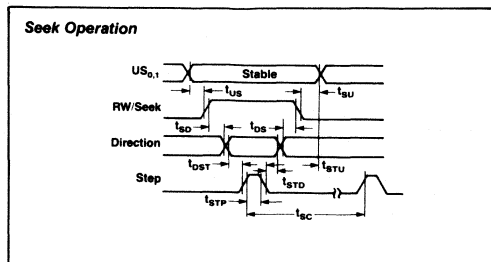
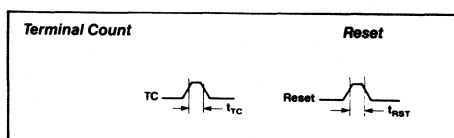
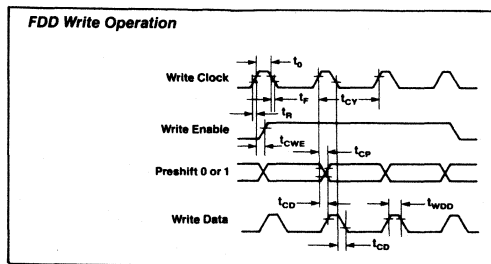
#### Note:

- (1) Typical values for T<sub>a</sub> = 25°C and nominal supply voltage.
- (2) Data transfer in NON DMA mode
- (3) Double these values for 32MHz clock and 250Kbps operation (DR0 = 0 DR0 = 1)
- (4) When external clock is used
- (5) When external VFO is used
- (6) SENSE DEVICE STATUS command being executed for the other devices during certain device being under seek operation.

## Timing Waveforms



5





### Description

The μPD72068 is a new Floppy Disk Controller specially designed for PC applications. Based on the original firmware of the μPD765 industry standard controller it is fully compatible to all PC software with MS-DOS. The new chip contains a high performance digital PLL resulting in excellent data separation with about 85% jitter margin and +/- 10% speed tolerance. This is about the same as any unadjusted analog PLL, however no external circuit is necessary, and stable operation independent of voltage drop or temperature or degradation of analog components is guaranteed. Additionally, the new chip integrates registers and decode logic required for a standard PC, AT, or PS-2 like system. For the drive side, the high current drivers (24mA) and Schmitt trigger inputs, all fully demultiplexed, are integrated. On the host side, the registers for drive select, motor control, data transfer rate selection are included. So the μPD72068 replaces a board type application with a single chip. Even more flexibility is provided by selectable output polarity (active low, active high), selectable diskette format (IBM or ECMA) and by support of the new high speed data rate of 600 Kbit/s.

### Features

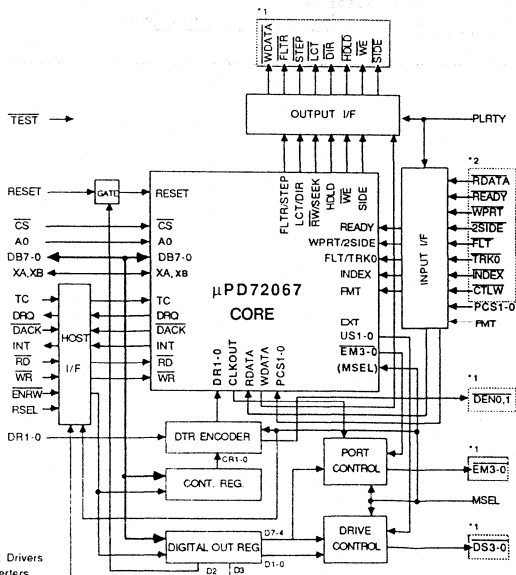
- Compatible with μPD765A, 7265, 72065, 72066 and 72067

- On-chip digital phase locked loop for data separator
- On-chip write precompensator
- Can support both IBM and ECMA format
- Controls up to 4 floppy disk drives
- On-chip clock generator
- Data transfer rate 600 Kbps
- Needs 32 MHz and 38.4 MHz crystals for on-chip DPLL (38.4 MHz in case 600 Kbps is used, otherwise 19.2 MHz crystal)
- Motor on/off control logic
- 15 disk commands and 8 auxiliary commands
- High current drivers (24 mA) and Schmitt trigger input buffers for device interface
- IBM-PC host interface logic and registers
- Generate density exchange signal based on data transfer rate
- CMOS single +/- 5V

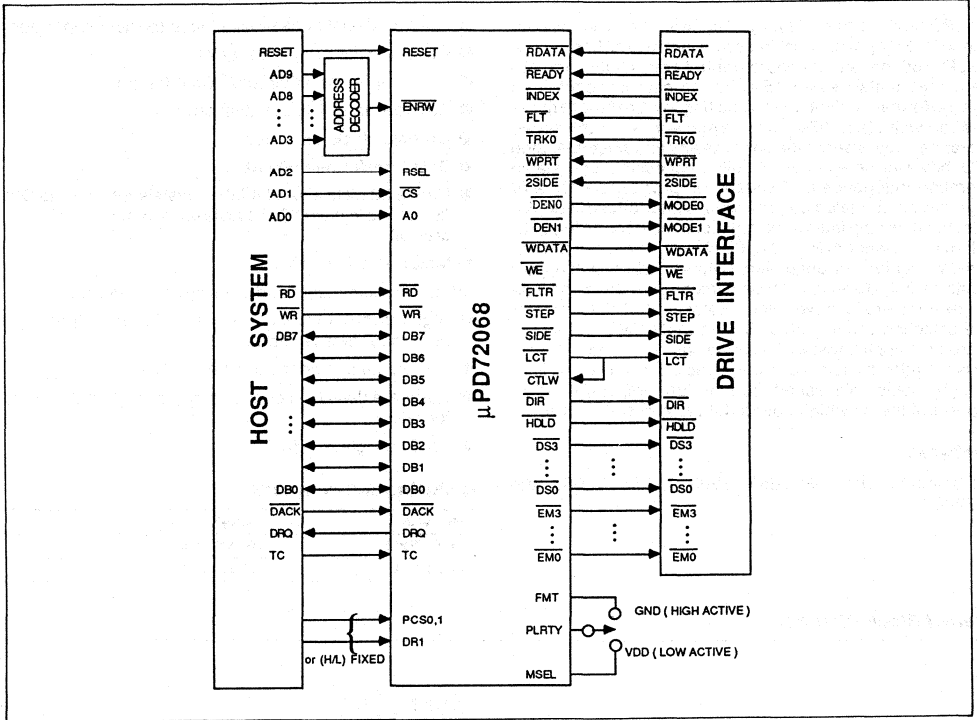
### Ordering Informations

Order Code	Package
μPD72068GF	80-pin flat pack
μPD72068L	84-pin PLCC

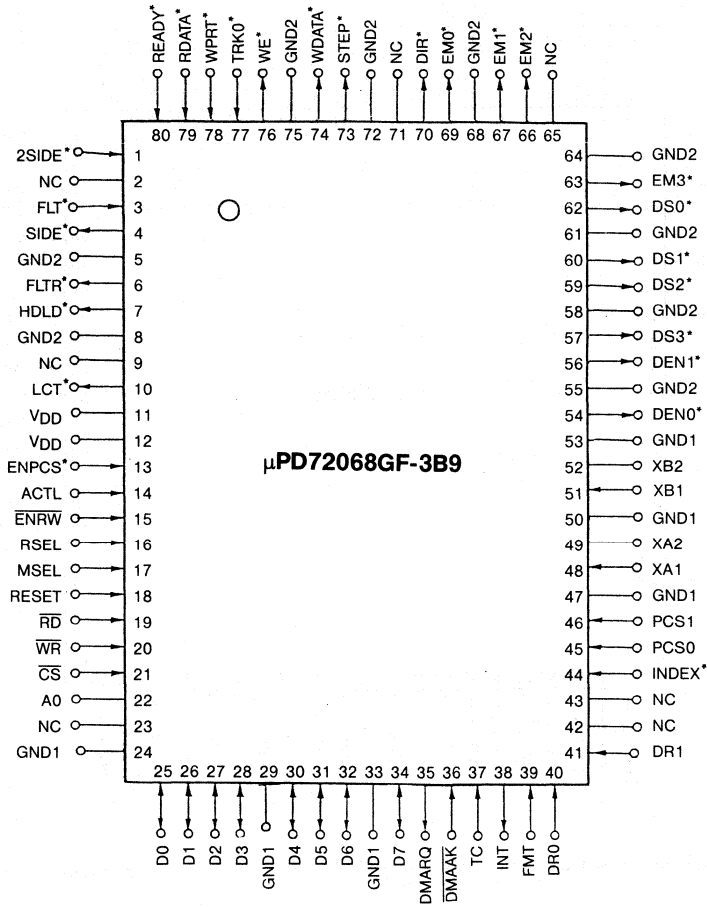
### Internal Block Diagram



System Configuration Example



### 80 Pin flat package



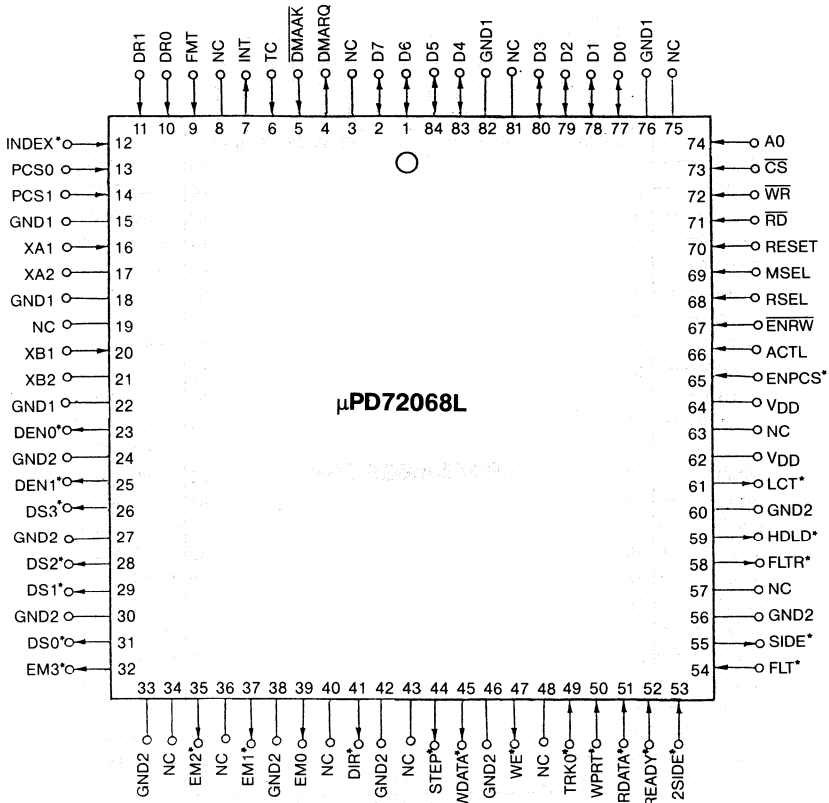
The output level is selectable (active high or active low)

NC: Non-Connection

GND1 and GND2 must be connected on shortest way



84 Pin PLCC



The output level is selectable (active high or active low)

NC: Non-Connection

GND1 and GND2 must be connected on shortest way

## Description

The μPD7260 is a single-chip disk controller that is capable of interfacing to a maximum of four floppy or hard disks in any combination. The chip utilizes the ST-506 defacto standard for the Winchester disks and is compatible with 8-inch, 5-1/4-inch and 3-1/2-inch floppy disks. The μPD7260 is based on the μPD7261A architecture, but with changes to enhance performance and flexibility. The μPD7260 can generate both IBM- and ECMA-compatible floppy disks and hard disks with the standard format. ECC and CRC capabilities along with many high-level commands provide excellent system throughput, and the single-chip design provides for efficient board space utilization.

## Features

- Hard and floppy disk interface
- Controls four drives (any combination) simultaneously
- Programmable track format
- Transfer rate 6 MHz maximum
- 16 high-level disk commands
- Parallel seek capability
- Multi-sector, -track, -cylinder read/write capability
- Implied seek function
- CRC error detection
- ECC error detection and correction
- DMA data transfer
- Single +5 volt supply
- NMOS 40-pin ceramic DIP

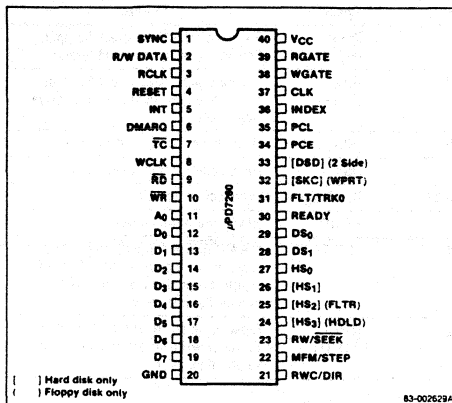
## μPD7260 Commands

Check	Sense Intr. Status
Detect Error	Sense Status
Read Data	Specify1
Read Diagnostic	Specify2
Read ID	Verify Data
Recalibrate	Verify ID
Scan	Write Data
Seek	Write Format

## Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD7260D	40-pin ceramic DIP	12 MHz

## Pin Configuration



## Pin Identification

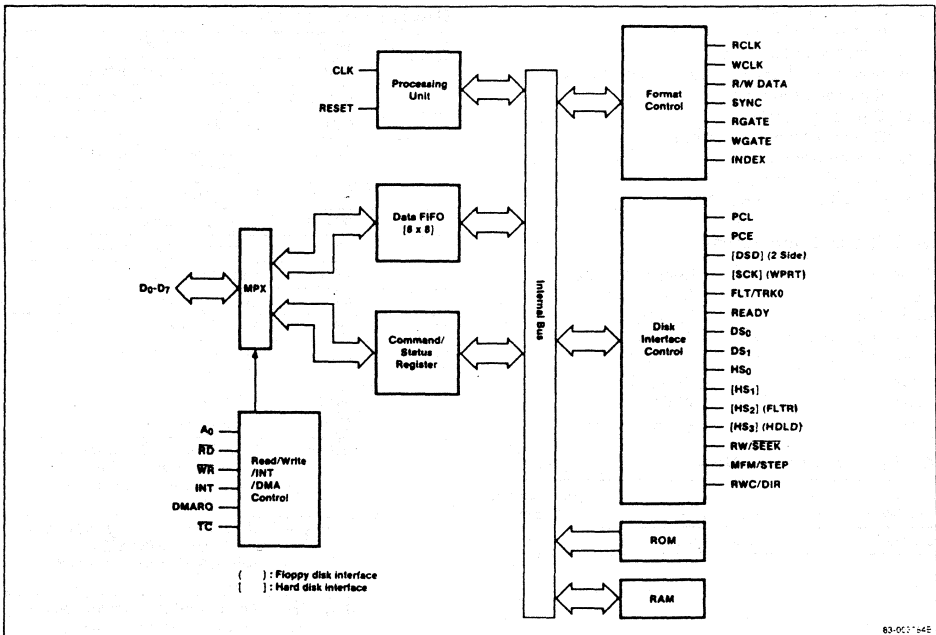
No.	Symbol	Function
1	SYNC	PLL synchronization output
2	R/W DATA	Read data input or write data output
3	RCLK	Read clock input
4	RESET	System reset input from host computer
5	INT	Interrupt request output
6	DMARQ	DMA request output
7	TC	Terminal count input from DMA
8	WCLK	Write clock input
9	RD	Host computer read control input
10	WR	Host computer write control input
11	A <sub>0</sub>	Status/command register or FIFO select pin
12-19	D <sub>0</sub> -D <sub>7</sub>	System data bus connections
20	GND	System ground
21	RWC/DIR	If RW/SEEK = 1, outputs read/write current decrease signal. If RW/SEEK = 0, outputs the direction RW head is to move.
22	MFM/STEP	If RW/SEEK = 1, outputs MFM signal to VCO circuit. If RW/SEEK = 0, outputs STEP signal to move RW head.
23	RW/SEEK	Output signal that specifies function of some multiplexed signals

**Pin Identification (cont)**

No.	Symbol	Function
24	[HS <sub>3</sub> ] (HOLD)	For hard disk, head select 3 output. For floppy disk, head load output.
25	[HS <sub>2</sub> ] (FLTR)	For hard disk, head select 2 output. For floppy disk, output to clear drive fault state.
26	[HS <sub>1</sub> ]	Head select output to disk drive.
27	HS <sub>0</sub>	Head select output to disk drive.
28-29	DS <sub>1</sub> -DS <sub>0</sub>	Drive select outputs.
30	READY	Ready input from disk drive.
31	FLT/TRK0	If RW/SEEK = 1, inputs a fault flag from the disk drive. If RW/SEEK = 0, inputs a signal indicating R/W head is over cylinder zero.
32	[SKC] (WPRT)	Seek complete input from hard disk drive, or write protected input from floppy disk drive.

No.	Symbol	Function
33	[DSD] (2 Side)	Drive selected input from hard disk drive, or double-sided disk input from floppy disk drive.
34-35	PCE, PCL	Precompensation early/late output to disk drive
36	INDEX	Index hole detect input from disk drive
37	CLK	System clock input from host computer
38	WGATE	Write gate output to disk drive
39	RGATE	Read gate output to disk drive
40	V <sub>CC</sub>	+5 V (typical)

**Block Diagram**



## 1. Pin Functions

The pins of the μPD7260 may be divided into two groups: those for the CPU interface and those for the disk drive interface.

The following describes the pins of each group:

### 1.1 CPU Interface

(1) CLK (Clock) . . . Input

Inputs external clock. This input is used as the operation clock for the processing unit.

Hard disk:

$0.6 \times (\text{RCLK, WCLK frequency}) \leq (\text{CLK frequency}) \leq 1.5 \times (\text{RCLK, WCLK frequency})$

Floppy disk:

$4 \times (\text{RCLK, WCLK frequency}) \leq (\text{CLK frequency}) \leq 44 \times (\text{RCLK, WCLK frequency})$

(2) D0 to D7 (Data Bus) . . . Three-state input/output

These pins constitute an 8-bit bidirectional data bus and are used for data transfer between the μPD7260 and the host system, or for the communication of commands, parameters, result status and status.

(3)  $\overline{\text{RD}}$  (Read) . . . Input

Inputs an active low control signal to read μPD7260 data/status via the data bus. Because the μPD7260 has no chip select input, read operations can only be performed upon input of the  $\overline{\text{RD}}$  signal.

(4)  $\overline{\text{WR}}$  (Write) . . . Input

Inputs an active low control signal to write data/commands to the μPD7260 via the data bus. Because the μPD7260 has no chip select input, write operations can only be performed upon input of the  $\overline{\text{WR}}$  signal.

(5) A0 (Address 0) . . . Input

Inputs a signal to select access of the status/command register or the data FIFO.

A0	$\overline{\text{RD}}$	$\overline{\text{WR}}$	Function
0	0	1	Read data FIFO
	1	0	Write data FIFO
1	0	1	Read status register
	1	0	Write command register

(6) INT (Interrupt Request) . . . Output

Outputs an interrupt request signal to the host system. This signal becomes high level when the μPD7260 requests service from the host.

(7)DMARQ (DMA Request) . . . Output

Outputs a DMA request signal to the host system. This signal becomes high level when the μPD7260 requests DMA processing.

(8)  $\overline{\text{TC}}$  (Terminal Count) . . . Input

Inputs an active low signal from the host system that requests termination of μPD7260 data transfer.

The  $\overline{\text{TC}}$  signal is accepted in one byte time after the  $\overline{\text{RD}}/\overline{\text{WR}}$  signal access for the last transfer byte (one byte time refers to the time required to send or receive one byte of serial data).

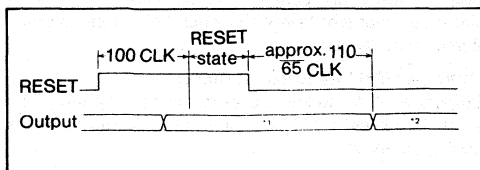
Data transfer rate	One-byte time
5 Mbps	1.6 μs
500 K	16
250 K	32
125 K	64

5

Pin No.	Signal name	Reset state *1	After reset release *2	Pin No.	Signal name	Reset state *1	After reset release *2
1	SYNC	Low	Unchanged	25	(HS2) (FLTR)	High	Low
2	R/W DATA	Hi-Z					
5	INT	Low					
6	DMARQ	Low					
12-19	D0-27	Hi-Z		29	DS0		
21	RWC/DIR	High	Low	34	PCE	Low	Unchanged
22	MFM/STEP						
23	RW/SEEK						
24	(HS3) (HLDL)						
				35	PCL		
				38	WGATE		
				39	RGATE		

(9) RESET (Reset) . . . Input

Inputs a reset signal. While CLK and WCLK are both being input, inputting a high-level signal to the RESET pin for more than 100 CLK cycles clears all μPD7260 registers and causes the μPD7260 to enter the idle state. The following table shows the output and input/output pin status occurring when be in reset state and approximately 110 clocks (approx. 65 clocks for DS0) are input after release of reset state.



Note: After the reset input is set to low level (reset is released), confirm that the CB bit in the status register is "0" before writing a disk command to the μPD7260.

**1.2 Disk Drive Interfaces**

This section describes the pin function for both hard disk (floppy-like) and floppy disk devices.

(1) R/W DATA (Read/Write data) . . . Three-state input/output

Inputs read data and outputs write data to/from the disk drive. Input and output data both consist of clock bits and data bits. This pin becomes an output pin only when the WGATE signal is asserted.

(2) SYNC (PLL Synchronization) . . . Output

Outputs a signal indicating that a continuous series of 16 zeros has been detected in the read data input to the R/W DATA pin.

(3) RCLK (Read Clock) . . . Input

Inputs the clock signal from the VFO circuit used to sample read data from the disk drive.

(4) WCLK (Write Clock) . . . Input

Inputs the clock signal used to create the output timing to write data to the disk drive and for the PCL and PCE signals. This clock must be input any time, because it is also used for other than write operation.

(5) RGATE (Read Gate) . . . Output

Outputs the control signal used to read data from the disk drive.

(6) WGATE (Write Gate) . . . Output

Outputs the control signal used to write data to the disk drive.

(7) INDEX (Index) . . . Input

Inputs the index hole detect signal from the disk drive.

(8) PCL, PCE (Precompensation Late/Early) . . . output

To compensate for peak-shift of read data, these pins output signals that causes a specified time delay or advance of the bit that would create the preshift when the data are written to the disk drive.

This signal is output only in MFM mode.

The write data bits that causes the PCL or PCE to be output is the bits indicated in the shaded part in the table below. The output timing is synchronous with the write data bits to be shifted.

Write data bit string	PCL	PCE
0 0 1 0 0	0	0
0 0 1 0 1	1	0
1 0 1 0 0	0	1
1 0 1 0 1	0	0

(9) DS0, DS1 (Drive Select) . . . Output

Outputs disk drive select signals. One of four drives may be selected with decoding these signals.

(10) READY (ready) . . . Input

Inputs a signal that indicates the disk drive is ready.

(11) RW/SEEK (Read Write/Seek) . . . Output

Outputs a signal that specifies the use of those disk drive interface signals that are multiplexed for seek and read/write. When this signal is low level, seek operation is indicated, while high level indicates read/write.

(12) MFM/STEP (MFM Mode/Step) . . . Output

When the RW/SEEK signal level is high (R/W operation), this pin outputs an MFM signal that specifies the operation mode of the VFO circuit. A low-level MFM signal indicates FM mode while a high-level MFM signal indicates MFM mode. When the RW/SEEK signal level is low (seek operation), this pin outputs the STEP signal that moves the read/write head.

(13) RWC/DIR (Read Write Current/Direction) . . . Output

When the RW/SEEK signal level is high, this pin outputs an RWC signal directing the device to decrease write current to the disk or increase read current. This signal becomes high level if the cylinder number of the cylinder where the read/write head is currently located is equal to or greater (closer to the center of the disk) than that specified by the parameters of the SPECIFY2 command. These parameters are RWCH, RWCL for hard disk and FRWC for floppy disk.

When the RW/SEEK signal level is low, this pin outputs the DIR signal that specifies the direction of read/write head movement. A low-level DIR signal specifies outward head movement while a high-level DIR specifies inward head movement.

(14) FLT/TRK0 (Fault/Track 0) . . . Input

When the RW/SEEK signal is high level, this pin inputs an FLT signal indicating that the disk drive is in a fault state.

When executing a READ/WRITE command, the μPD7260 latches the FLT signal if the RGATE or WGATE signal is active, and clears the latch upon INT signal output. The FLT signal is latched provided the high level width is 48 clock cycles or more, however.

When RW/SEEK signal level is low, this pin inputs a TRK0 signal indicating the read/write head is located at cylinder 0.

The following is a functional description of pins used only for (floppy-like) hard disk drives:

(1) HSO to HS3 (Head Select) . . . Output

These pins output the read/write head select signals. One of 16 read/write heads may be selected decoding these signals.

(2) DSD (Drive Selected) ... Input

Inputs a signal indicating that the disk drive corresponding to the disk drive select signals has been selected.

(3) SKC (Seek Complete) ... Input

This pin accepts the SKC signal which indicates that the seek operation is completed and the read/write head is positioned to the specified cylinder.

The following is a functional description of pins used only for floppy disk drives:

(1) HSO (Head Select) ... Output

Outputs a read/write head select signal.

(2) 2SIDE (Two Side) ... Input

Inputs a signal indicating that a double-sided disk is in place in the drive.

(3) FLTR (Fault Reset) ... Output

Outputs a signal that clears the fault state of the disk drive.

(4) HDLD (Head Load) ... Output

Outputs a signal that loads the read/write head on the disk.

(5) WPRT (Write Protected) ... Input

This pin accepts the Wprt signal which indicates that the disk drive or medium is write protected.

2. Disk Command List

Command	Phase	R/W	A0	Hard disk		Floppy Disk		Remarks	
				D7	D0	D7	D0		
SPECIFY1	C	W	0	1	Mode		Same as left.		
				2	HESN				
				3	← HEHN → 0 ← DDTL →				
				4	FESN				
				5	HGPL2				
				6	HCOH				
				7	000 ← FWRT →				
	R	R	1	1	0 0 1 0 0 000				
SPECIFY2	C	W	0	1	← HNSR → 0 HBSR		FGPL2 can be omitted.		
				2	HRWCH				
				3	HRWCL				
				4	FRWC				
				5	← FNSR → ← HST →				
				6	HLT				
				7	HUT				
	8	FGPL2							
R	R	1	1	0 0 1 0 1 000					
RECALIBRATE	C	W	0	1	SNLH	SNLH		SNLH,L can be omitted.	
				2	SNLL	SNLL			
				1	01010 0 DS1 DS0	0101 HL DS1 DS0			
	E	—			—	—		Recalibrate	
	R	R	1	1	STR	STR			
SEEK	C	W	0	1	PCNH	PCNH			
				2	PCNL	PCNL			
				1	0110 BS0DS1 DS0	01100 HL DS1 DS0		Seek	
	R	R	1	1	STR	STR			
				1	0	IST			
WRITE FORMAT	C	W	0	1	0000 ← PHN →		0000000 PHN		
				2	SCT		SCT		
				3	DPAT		DPAT		
				4	HGPL1		FGPL3		
				5	HGPL3		00000 ← FDTL →		
				1	011100DS1 DS0	0111 EF MF DS1 DS0			
	E	W		0				Transfers IDs (SCT x 4 bytes) of the sectors on a track.	
	R	R	1	1	STR	STR			
			1	0	EST				
			2	0	SCT				

Command	Phase	R/W	A0	Hard Disk		Floppy Disk		Remarks
				D7	D0	D7	D0	
VERIFY ID	C	W	1	0	0000 · PHN→		0000000 PHN	Compares IDs.
			2		SCT		SCT	
			3		—		00000 · FDTL→	
		1	100000 DS1 DS0		100000MF DS1 DS0			
	E	W		0				
	R	R		1	STR		STR	
READ ID	C	W	1	0	0000 · PHN→		0000000 PHN	LCNH is omitted if IM = 0.
			2		SCT		SCT	
			3		LCNH		00000 · FDTL→	
		1	1001 1MODS1 DS0		1001 1M DS1 DS0			
	E	R		0			Transfers ID.	
	R	R		1	STR		STR	
READ DIAGNOSTIC	C	W	1	0	0000 · PHN→		0000000 PHN	Transfers data.
			2		SCT		SCT	
			3		—		00000 · FDTL→	
		1	101000 DS1 DS0		10100 MF DS1 DS0			
	E	R		0				
	R	R		1	STR		STR	
SENSE UNIT STATUS	C	W	1	1	001100 DS1 DS0		Same as left.	
					1	STR		
			R		R	1		UST
		2	0	PCNH				
		3	0	PCNL				
	C	W		1	00010000			
SENSE INTERRUPT STATUS	R	R		1	STR			
				1	0	IST		
				1	0	IST		
READ DATA	C	W	1	0	· ISO→ · PHN→		· ISO→ SK00PHN	FESN is Omissible
			2		LCNH		LCN	
			3		LCNL		0000000 LHN	
			4		0000 · LHN →		LSN	
			5		LSN		00000 · FDTL →	
			6		SCTH		SCTH	
			7		SCTL		SCTL	
			8		—		FESN	
					1	1011IB0DS1 DS0		

5



Command	Phase	R/W		A0	Hard Disk		Floppy Disk		Remarks
		D7	D0		D7	D0			
READ DATA (Cont'd)	E	R		0					Transfer data.
	R	R		1	STR	STR			The ID of the sector next to the sector where the command execution was completed successfully; or the ID of the sector where the command execution aborted.
				2	EST	EST			
				3	· ISO→ · PHN→	· ISO→ SK00PHN			
				4	LCNH	LCN			
				5	LCNL	0000000 LHN			
				6	0000 · LHN→	LSN			
				7	LSN	00000 · FDTL→			
	8	SCTH	SCTL						
				0	SCTL	SCTL			
CHECK	C	C		1	· ISO→ · PHN→	· ISO→SK00PHN			Same as READ DATA.
				2	LCNH	LCN			
				3	LCNL	0000000 LHN			
				4	0000 · LHN→	LSN			
				5	LSN	00000 · FDTL→			
				6	SCTH	SCTL			
				7	SCTL	SCTL			
				8	—	FESN			
	E	—		1	1100 IBxDS1 DS0	1100 Damf DS1 DS0			Data check
	R	R		0	1	STR	STR		
2					EST	EST			
3					· ISO→ · PHN→	· ISO→ SK00 PHN			
4					LCNH	LCN			
5					LCNL	0000000 LHN			
6					0000 · LHN→	LSN			
7					LSN	00000 · FDTL→SCTH			
8					SCTH	SCTL			

Command	Phase	R/W	AO	Hard disk		Floppy Disk		Remarks
				D7	D0	D7	D0	
SCAN	C	W	0	1	- ISO - - PHN -	- ISO - SKOOPHN	Same as READ DATA.	
				2	LCNH	LCN		
				3	LCNL	0000000 LHN		
				4	0000 - LHN -	LSN		
				5	LSN	00000 - FDTL -		
				6	SCTH	SCTH		
				7	SCTL	SCTL		
				8	-	FESN		
	E	W	0	1	1101 IB0DS1 DS0	1101 DAMF DS1 DS0	Compares data.	
				1	STR	STR		
	R	R	0	1	EST	EST	Last sector to be compared. For NCI, however, the ID of the sector next to the last sector to be compared.	
				2	- ISO - - PHN -	- ISO - SK00 PHN		
				3	LCNH	LCN		
				4	LCNL	0000000 LHN		
				5	000 - LHN -	LSN		
				6	LSN	00000 - FDTL -		
7				SCTH	SCTH			
8				SCTL	SCTL			
VERIFY DATA	C	W	0	1	- ISO - - PHN -	- ISO - SKOOPHN	Same as READ DATA.	
				2	LCNH	LCN		
				3	LCNL	0000000 LHN		
				4	0000 - LHN -	LSN		
				5	LSN	00000 - FDTL -		
				6	SCTH	SCTH		
				7	SCTL	SCTL		
				8	-	FESN		
	E	W	0	1	1110 IB0DS1 DS0	1110 DAMF DS1 DS0	Compares data.	
				1	STR	STR		
	R	R	0	1	EST	EST	Same as READ DATA.	
				2	- ISO - - PHN -	- ISO - SK00 PHN		
				3	LCNH	LCN		
				4	LCNL	0000000 LHN		
				5	0000 - LHMN -	LSN		
				6	LSN	00000 - FDTL -		
7				SCTH	SCTH			
8				SCTL	SCTL			

Command	Phase	R/W	AO	Hard Disk		Floppy Disk		Remarks
				D7	D0	D7	D0	
WRITE DATA	C	W	0	1	- ISO - - PHN -	- ISO - SKOOPHN	Same as READ DATA.	
				2	LCNH	LCN		
				3	LCNL	0000000 LHN		
				4	0000 - LHN -	LSN		
				5	LSN	00000 - FDTL -		
				6	SCTH	SCTH		
				7	SCTL	SCTL		
				8	-	FESN		
	E	W	0	1	1111 IB0 DS1 DS0	1111 DAMF DS1 DS0	Transfers data.	
				0				
	R	R	0	1	STR	STR	Same as READ DATA.	
				2	EST	EST		
				3	- ISO - - PHN -	- ISO - SKOO PHN		
				4	LCNH	LCN		
				5	LCNL	0000000 LHN		
				6	0000 - LHN -	LSN		
7				LSN	00000 - FDTL -			
8				SCTH	SCTH			
DETECT ERROR	C	W	1	1	010000 DS1 DS0			
				1	STR			
	R	R	0	1	EAOH			
				2	EAOL			
				3	EPAT1			
				4	EPAT2			
5	EPAT3							

**Subcommands**

Command Name	Description	Command Code							
		D7	D6	D5	D4	D3	D2	D1	D0
CHIP RESET	Resets the HFDC	0	0	0	0	0	0	0	1
CLEAR DATA FIFO	Clears the data FIFO	0	0	0	0	0	0	1	0
MASK SRQ INTERRUPT	Masks SRQ interrupt	0	0	0	0	0	1	0	0
CLEAR COMMAND END BIT	Clears a disk command execution completion condition	0	0	0	0	1	0	0	0

### 3. Electrical Specifications

#### Absolute Maximum Ratings (Ta = 25 °C)

Parameter	Symbol	Test Conditions	Ratings	Units
Power Supply Voltage	V <sub>CC</sub>		-0.5 to +7.0	V
Input Voltage	V <sub>I</sub>		-0.5 to +7.0	V
Output Voltage	V <sub>O</sub>		-0.5 to +7.0	V
Operating Temperature	T <sub>opt</sub>		0 to +70	°C
Storage Temperature	T <sub>stg</sub>		-65 to +150	°C

#### DC Characteristics (Ta = 0 to +70 °C, V<sub>CC</sub> = +5.0V ~ 10%)

Parameter	Symbol	Test Conditions	MIN	TYP	MAX	Units
Input Low Voltage	V <sub>IL1</sub>	Except CLK, RCLK, WCLK	-0.5		+0.8	V
	V <sub>IL2</sub>	CLK, RCLK, WCLK	-0.5		+0.6	V
Input High Voltage	V <sub>IH1</sub>	Except CLK, RCLK, WCLK	+2.2		V <sub>CC</sub> +0.5	V
	V <sub>IH2</sub>	CLK, RCLK, WCLK	+3.3		V <sub>CC</sub> +0.5	V
Output Voltage Low	V <sub>OL</sub>	I <sub>OL</sub> = +2.0mA			+0.45	V
Output Voltage High	V <sub>OH1</sub>	I <sub>OH</sub> = -100μA, Except Pin 21-29, 34	+2.4			V
	V <sub>OH2</sub>	I <sub>OH</sub> = -50μA Pin 21-29, 34	+2.4			V
Input Leakage Current	I <sub>LI1</sub>	V <sub>I</sub> = 0.45V to V <sub>CC</sub> Except Pin 30-33			±10	μA
	I <sub>LI2</sub>	V <sub>I</sub> = 0.45V to V <sub>CC</sub> Pin 30-33			-500	μA
Output Leakage Current	I <sub>LO</sub>	V <sub>O</sub> = 0.45V to V <sub>CC</sub>			±30	μA
Supply Current	I <sub>CC</sub>			250	320	mA

#### Capacitance (Ta = 25 °C, V<sub>CC</sub> = GND = 0V)

Parameter	Symbol	Test Conditions	MIN	TYP	MAX	Units
Input Capacitance	C <sub>I</sub>	f = 1MHz Unmeasured pin returned to 0V			15	pF
Output Capacitance	C <sub>O</sub>				15	pF
I/O Capacitance	C <sub>IO</sub>				20	pF

**AC Characteristics** (Ta=0°C to +70°C, VCC=+5.0 + 10%)

**Main System side:**

Parameter	Symbol	Test Conditions	MIN	MAX	Units
CLK Cycle Time	tCY		83		ns
CLK Active High	tCH		30		ns
CLK Active Low	tCL		30		ns
CLK Rise Time	tCR			10	ns
CLK Fall Time	tCF			10	ns
A0 Setup Time to $\overline{RD}$	tAR		0		ns
A0 Hold Time from $\overline{RD}$	tRA		0		ns
RD Pulse Width	tRR		180		ns
Data Output Delay from A0	tAD			150	ns
Data Output Delay from $\overline{RD}$	tRD			130	ns
Data Float Delay from $\overline{RD}$	tRDF		0	100	ns
A0 Setup Time to $\overline{WR}$	tAW		0		ns
A0 Hold Time from $\overline{WR}$	tWA		0		ns
$\overline{WR}$ Pulse Width	tWW		190		ns
Data Setup Time to $\overline{WR}$	tDW		100		ns
Data Hold Time from $\overline{WR}$	tWD		5		ns
Recovery Time from $\overline{RD}$ , $\overline{WR}$	tRV		160		ns
INT Delay Time from $\overline{WR}\uparrow$	tWI			200	ns
DREQ Delay Time from $\overline{WR}\uparrow$	tWRQ			250	ns
DREQ Delay Time from $\overline{RD}\uparrow$	tRRQ1	*1		250	ns
DREQ Delay Time from $\overline{RD}\downarrow$	tRRQ2	*2		150	ns
$\overline{TC}$ Pulse Width	tTC		100		ns
RESET Pulse Width	tRES		100		tCY

\*1: During Disk Read Operation

\*2: After Disk Read Operation

**Disk Drive Interface side:**

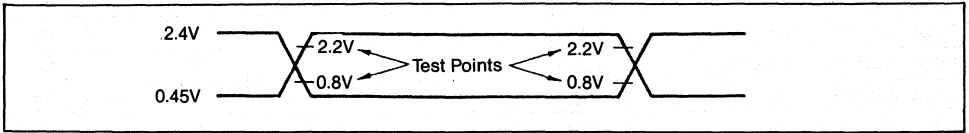
Parameter	Symbol	Test Conditions	MIN	MAX	Units	
RCLK, WCLK Cycle Time	tRWCY		83	5000	ns	
RCLK, WCLK Pulse Width High	tRWCH		30		ns	
RCLK, WCLK Pulse Width Low	tRWCL		30		ns	
RCLK, WCLK Rise Time	tRWCR			10	ns	
RCLK, WCLK Fall Time	tRWCF			10	ns	
Data Setup Time to RCLK	tRDRC		40		ns	
Data Hold Time from RCLK	tRCRD		5		ns	
Data Output Delay from WCLK	tWCWD		5	83	ns	
RGATE Delay Time from RCLK	tRCRG			300	ns	
WGATE Delay Time from WCLK	tWCWG			150	ns	
PCE, PCL Delay Time from WCLK	tRWPCP		5	83	ns	
SYNC Delay Time from RCLK	tRCSY			150	ns	
INDEX Pulse Width	tIDXF		8		tRWCY	
DSO, 1 Setup Time to STEP	tDSST		4000		tCY	
DIR Setup Time to STEP	tDIST		80		tCY	
STEP Pulse Width	tSTEP		69	85	tCY	
DSO, 1 Hold Time from STEP	tSTDS		2700		tCY	
DIR Hold Time from STEP	tSTDI		6600		tCY	
DSO, 1 Setup Time to DIR	tDSDI		3900		tCY	
DIR Hold Time from SKC	tSKDI	Hard Disk Drive	590		tCY	
HSO, 1, 2, 3 Hold Time from WGATE	tWGHS		3800		tCY	
DSO, 1 Setup Time to HDLD	tDSDI	Floppy Disk Drive	2100		tCY	
STEP Cycle Time	tSTCY	Hard Disk Buffered Step Mode	HBSR=1	190	290	tCY
			HBSR=0	700	920	tCY

5

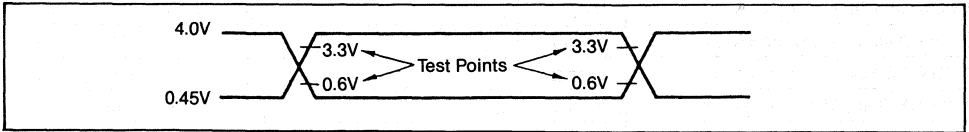
## $\mu$ PD7260

### AC Timing Test Points ( $C_L = 100\text{pF}$ )

(Except CLK, RCLK, WCLK)

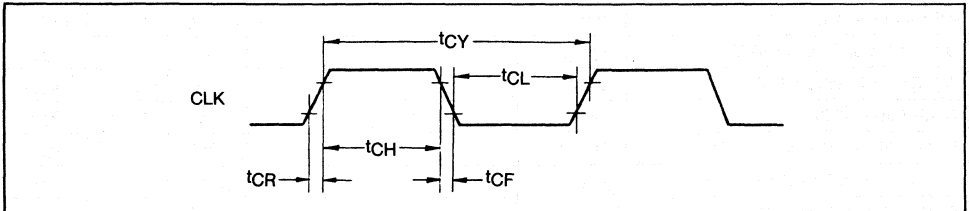


(CLK, RCLK, WCLK)

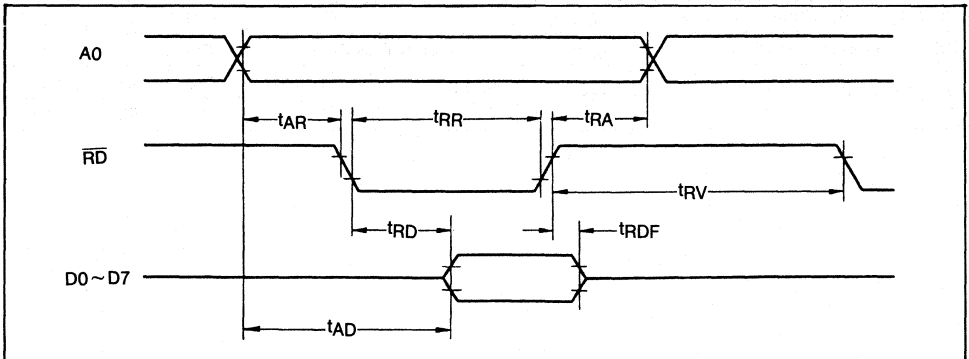


### Timing Waveforms

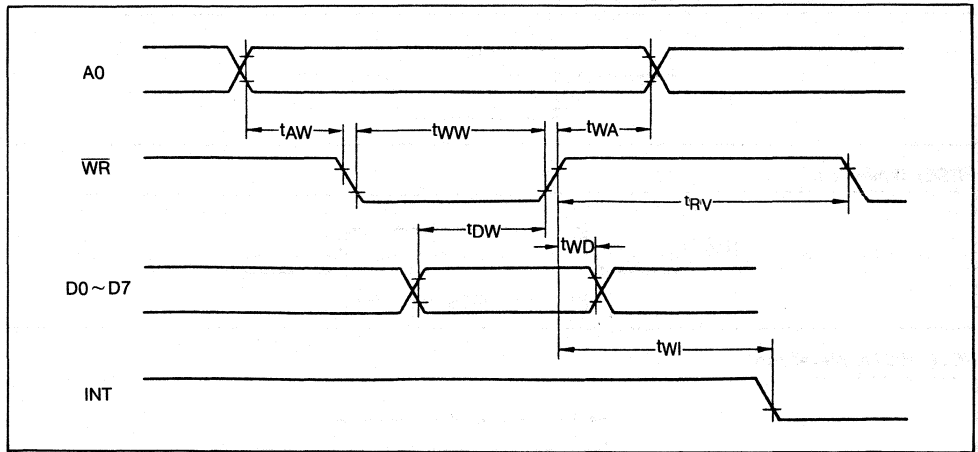
CLK Waveform



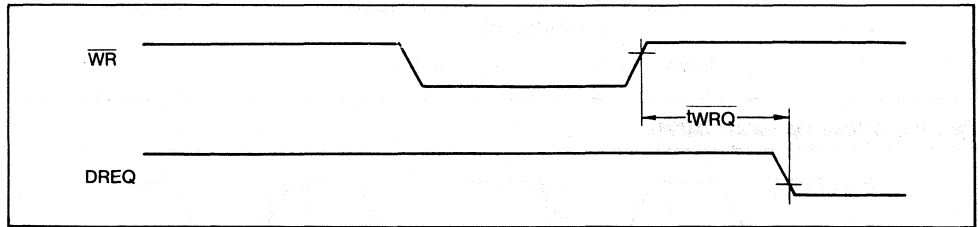
Read Timing



### Write Timing

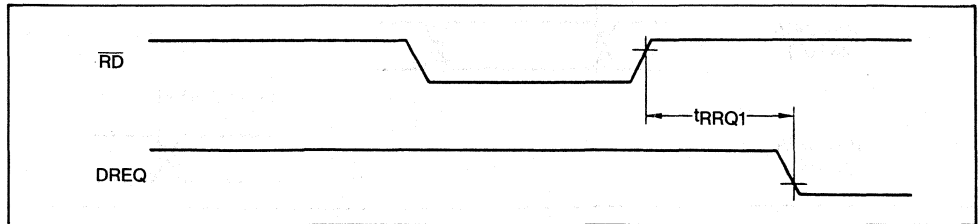


### DMA Write Timing

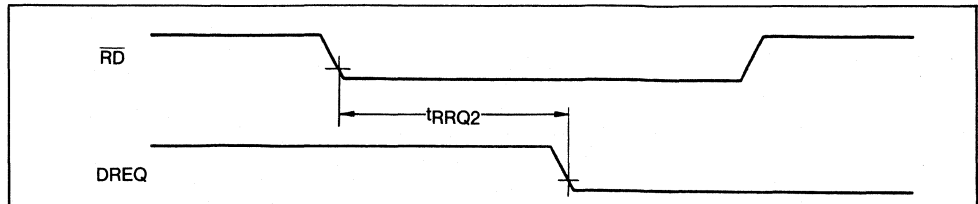


5

### DMA Write Timing

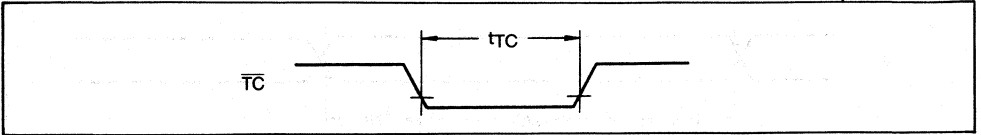


### DMA Read Timing (During Disk Read Operation)

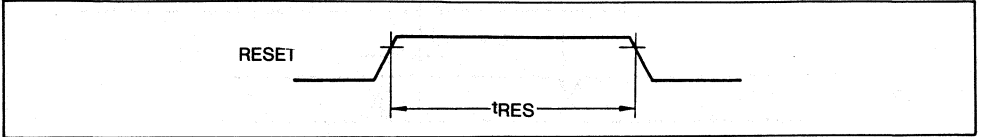




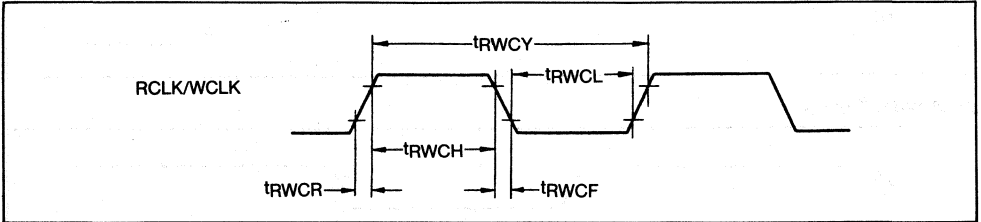
**DMA Read Timing (After Disk Read Operation)**



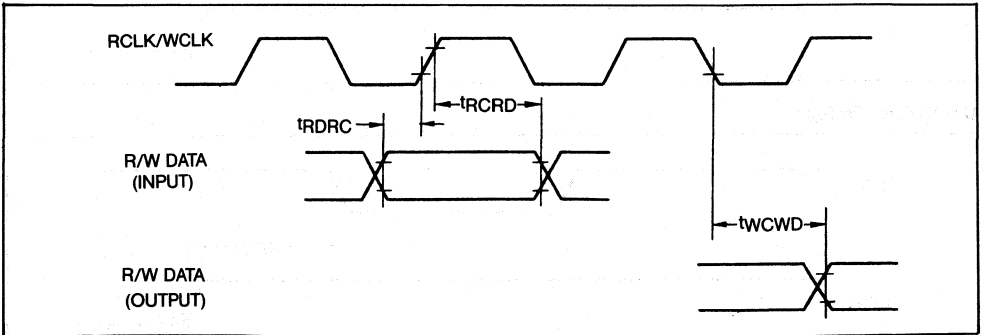
**RESET Waveform**



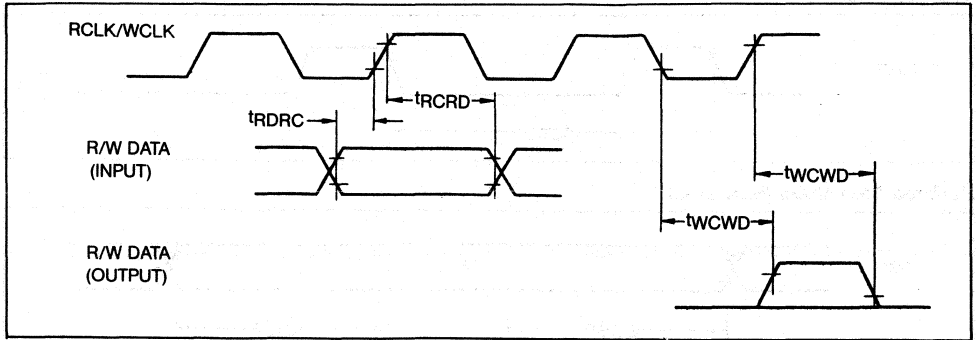
**RCLK, WCLK Waveform**



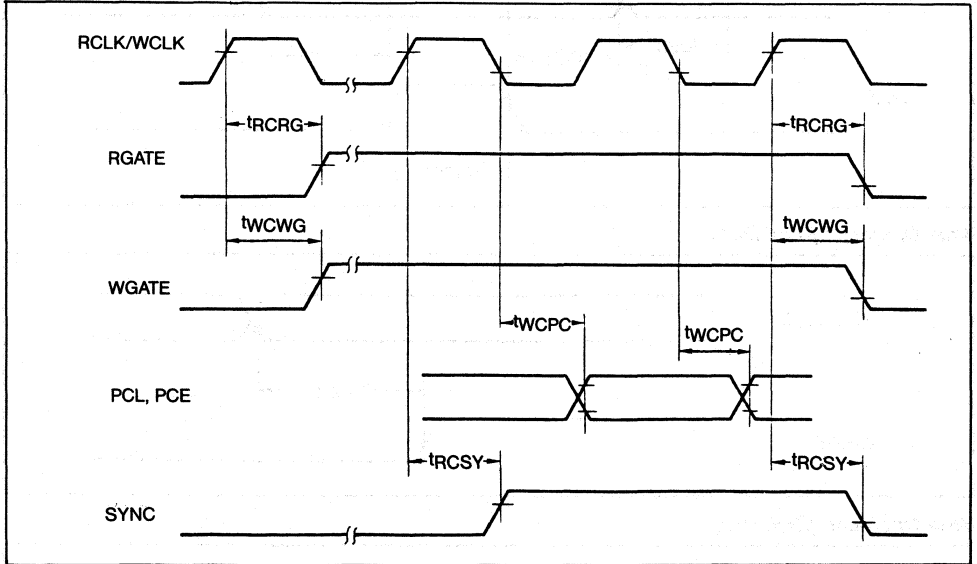
**Data Read/Write Operation (MFM)**



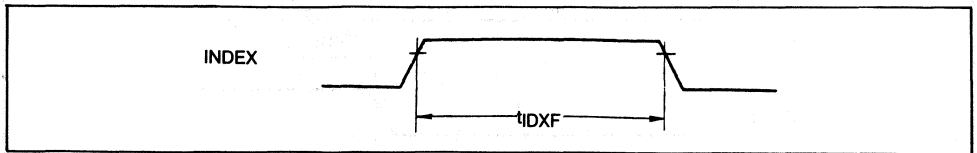
### Data Read/Write Operation (FM)



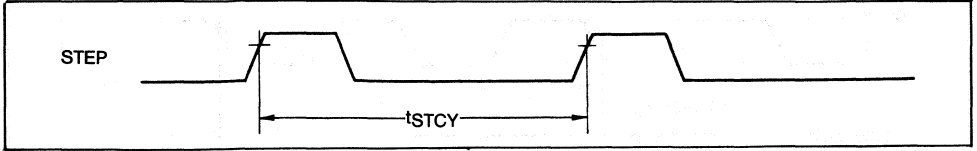
### Read/Write Operation



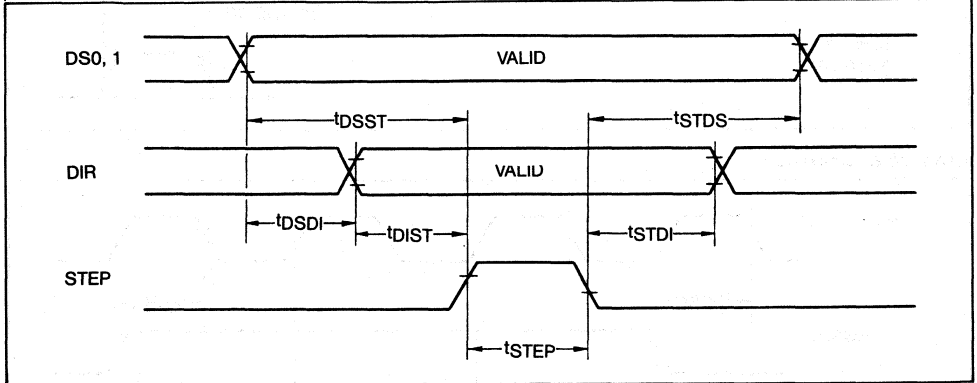
### INDEX Waveform



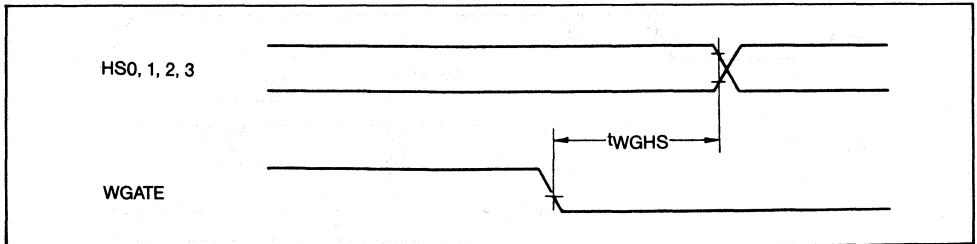
**Seek Operation**



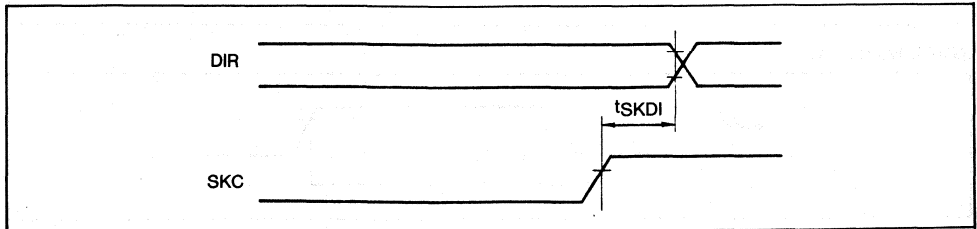
**Buffered Step Mode (Hard Disk)**



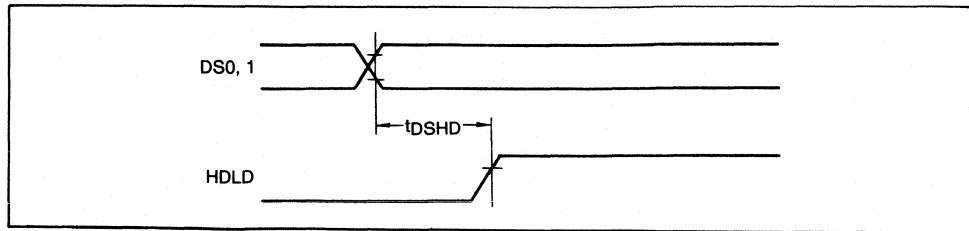
**Write Operation (Hard Disk)**



**Seek Operation (Hard Disk)**



## Head Load Operation (Floppy Disk)





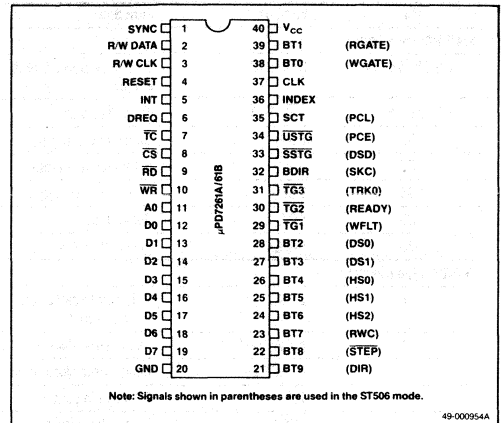
## Description

The μPD7261A and μPD7261B hard-disk controllers are intelligent microprocessor peripherals designed to control a number of different types of disk drives. They are capable of supporting either hard-sector or soft-sector disks and provide all control signals that interface the controller with either SMD disk interfaces or ST506-type drives. The sophisticated instruction set minimizes the software overhead for the host microprocessor. By using the DMA controller, the microprocessor needs only to load a few command bytes into the μPD7261A/7261B and all the data transfers associated with read, write, or format operations are done by the μPD7261A/7261B and the DMA controller. Extensive error reporting, verify commands, ECC, and CRC data error checking assure reliable controller operation. The μPD7261A/7261B provides internal address mark detection, ID verification, and CRC or ECC checking and verification. An eight-byte FIFO is used for loading command parameters and obtaining command results. This makes the structuring of software drivers a simple task. The FIFO is also used for buffering data during DMA read/write operations.

## Features

- Flexible interface to various types of hard disk drives
- Programmable track format
- Controls up to 8 drives in SMD mode; up to 4 drives in ST506-type mode
- Parallel seek operation capability
- Multi-sector and multi-track transfer capability
- Data scan and data verify capability
- High-level commands, including:
  - Read Data, Read ID, Write Data, Format, Scan Data, Verify Data, Verify ID, Check, Seek (normal or buffered), Recalibrate (normal or buffered), Read Diagnostic (SMD only), Specify, Sense Interrupt Status, Sense Unit Status, and Detect Error
- NRZ or MFM data format
- Maximum R/W CLK frequency:
  - 12 MHz (7261A)
  - 18 MHz (7261B-18)
  - 23 MHz (7261B-23)
- Error detection and correction capability
- Simple I/O structure: compatible with most microprocessors
- All inputs and outputs except clock pins are TTL-compatible (clock pins require pullup)
- Data transfers under DMA control
- NMOS
- Single +5-volt power supply
- 40-pin ceramic DIP

## Pin Configuration



## Ordering Information

Device Number	Package Type	Max Freq. of Operation
μPD7261AD	40-pin ceramic DIP	12 MHz
μPD7261BD-18	40-pin ceramic DIP	18 MHz
μPD7261BD-23	40-pin ceramic DIP	23 MHz

## Pin Identification

No.	Symbol	Function
<b>Host Interface</b>		
4	RESET	Reset input
5	INT	Interrupt request output
6	DREQ	DMA request output
7	TC	Terminal count input
8	CS	Chip select input
9	RD	Read strobe input
10	WR	Write strobe input
11	A <sub>0</sub>	Register select input
12-19	D <sub>0</sub> -D <sub>7</sub>	Data I/O bus
20	GND	Ground
37	CLOCK	External clock input
40	V <sub>CC</sub>	+5 V power supply
<b>SMD Interface</b>		
1	SYNC	PLL synchronization output
2	R/W DATA	Read/write data I/O
3	R/W CLK	Read/write clock input

**Pin Identification (cont)**

No.	Symbol	Function
<b>SMD Interface (cont)</b>		
21-28, 38, 39	BT9-BT0	Bit 9-0 outputs / Status inputs
29-31	TG1-TG3	Tag 1-3 output
32	BDIR	Bit direction output
33	SSTG	SR select tag output
34	USTG	Unit select tag output
35	SCT	Sector input
36	INDEX	Sector zero input
<b>ST506-Type Interface</b>		
1	SYNC	PLL lock / Read clock enable output
2	R/W DATA	Read / write data I/O
3	R/W CLK	Read / write clock input
21	DIR	Direction in output
22	STEP	Step pulse output
23	RWC	Reduced write current output
24-26	HS2-HS0	Head select outputs 2-0
27, 28	DS1, DS0	Drive select outputs 1, 0
29	WFLT	Write fault input
30	READY	Ready input
31	TRK0	Track zero input
32	SKC	Seek complete input
33	DSD	Drive selected input
34	PCE	Precomp early output
35	PCL	Precomp late output
36	INDEX	Index input
38	WGATE	Write gate output
39	RGATE	Read gate output

**Pin Functions — Host Interface****RESET (Reset)**

When the RESET input is pulled high, it forces the device into an idle state. The device remains idle until a command is issued to the system.

**INT (Interrupt Request)**

The  $\mu$ PD7261A/7261B pulls the INT output high to request an interrupt.

**DREQ (DMA Request)**

The  $\mu$ PD7261A/7261B pulls the DREQ output high to request a DMA transfer between the disk controller and the memory.

**TC (Terminal Count)**

The TC input goes low to signal the final DMA transfer.

**CS (Chip Select)**

When the CS input is low, it enables reading from or writing to the register selected by A<sub>0</sub>.

**RD (Read Strobe)**

When the RD strobe is low, data is read from the selected register.

**WR (Write Strobe)**

When the WR input is low, data is written to the selected register.

**A<sub>0</sub> (Register Select)**

The A<sub>0</sub> input is connected to a non-multiplexed address bus line. When A<sub>0</sub> is high, it selects the command or status register. When it is low, it selects the data buffer.

**D<sub>0</sub>-D<sub>7</sub> (Data Bus)**

D<sub>0</sub>-D<sub>7</sub> are connected to the system data bus.

**CLOCK (Clock)**

The CLOCK input is the timing clock for the on-chip processor.

**Pin Functions — SMD Interface****SYNC (PLL Synchronization)**

This output goes high after the read gate signal (BT1 when TG3=0) is high and a given number of bytes (GPL2-2) has elapsed.

**R/W DATA (Read/Write Data)**

The R/W DATA pin outputs the write data to the drive, and inputs the read data from the drive.

**R/W CLK (Read/Write Clock)**

R/W CLK is the input for the read and write clocks.

**BT9-BT0 (Bit 9-0)**

BT9-BT0 output the bit signals, bit 9-0. The bit 9-0 outputs send cylinder and unit addresses to the drives. BT9-BT2 also act as inputs for status signals from the drives as shown in table 1.

**Table 1. Bit and Control Information**

No.	Bit	Control
21	BT9	Unit Selected
22	BT8	Seek End
23	BT7	Write Protected
24	BT6	
25	BT5	Unit Ready
26	BT4	On Cylinder
27	BT3	Seek Error
28	BT2	Fault

BT7-BT2 also read the device status 2 (SR7-SR2) and device type (DT7-DT2). The index and SCT pins read SR0, SR1 and DT0, DT1.

### BDIR (Bit Direction)

The BDIR output determines whether pins 28-21 will output BT2-BT9 or input drive status signals.

### TG3-TG1 (Tag 3-1)

The TG outputs define the use of the BT pins. When TG1 is low, BT9-BT0 output the cylinder address. When TG2 is low, BT7-BT0 select a head address. When TG3 is low, BT9-BT0 output control signals for the disk drive.

### SSTG (SR Select Tag)

When the SSTG output is low, BT7-BT2, INDEX and SCT will be inputting SR7-SR0 or DT7-DT0.

### USTG (Unit Select Tag)

When the USTG output is low, BT4-BT2 will be outputting a unit address.

### INDEX (Index)

The INDEX input goes high when the drive detects an index mark. INDEX also acts as the SR0 and DT0 input pin.

### SCT (Sector)

The SCT input goes high when the drive detects a sector mark. SCT also acts as the SR1 and DT1 input pin.

## Pin Functions — ST506-Type Interface

### SYNC (Read Clock Enable)

SYNC indicates that a sync pattern has been detected and that synchronization has been achieved.

### R/W DATA (Read/Write Data)

The R/W DATA pin outputs the write data to the drive, and inputs the read data from the drive.

### R/W CLK (Read/Write Clock)

R/W CLK is the input for the read and write clocks.

### DIR (Direction In)

The DIR output determines the direction the read/write head will move in when it receives a step pulse. DIR high will cause the head to move inward, DIR low will move the head outward.

### STEP (Step Pulse)

STEP outputs the head step pulses.

### RWC (Reduced Write Current)

The RWC output signals that the read/write head of the disk drive has selected a cylinder address larger than that specified in the SPECIFY command. This signal is used to reduce the write current.

### HS2-HS0 (Head Select 2-0)

The HS2-HS0 outputs select the head. Up to 8 read/write heads can be selected per drive.

### DS1, DS0 (Drive Select 1,0)

The DS1 and DS0 outputs select one of up to 4 drives.

### WFLT (Write Fault)

The WFLT input detects write faults.

### READY (Ready)

The READY input detects the drive's ready state.

### TRK0 (Track 0)

The TRK0 input signals that the head is at track 0.

### SKC (Seek Complete)

The SKC input signals that a seek is complete.

### DSD (Drive Selected)

The DSD input signals that the drive is selected.

### PCE (Precomp Early)

When the PCE output is high, early write precompensation is required.

### PCL (Precomp Late)

When the PCL output is high, late write precompensation is required.



## μPD7261A/B

### INDEX (Index)

The INDEX input goes high when the drive detects the index mark.

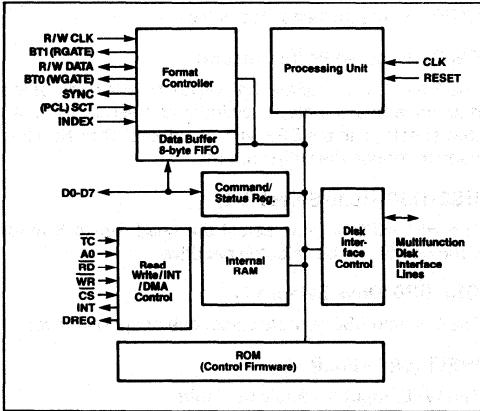
### WGATE (Write Gate)

WGATE output goes high when the μPD7261A/7261B is writing data.

### RGATE (Read Gate)

The RGATE output goes high when the μPD7261A/7261B is reading from the disk.

### Block Diagram



### Absolute Maximum Ratings

Operating temperature, $T_{OP}$	0°C to +70°C
Storage temperature, $T_{STG}$	-65°C to +150°C
Voltage on any pin with respect to ground, $V_{CC}$	-0.5 to +7.0 V*
Input voltage, $V_I$	-0.5 to +7.0 V*
Output voltage, $V_O$	-0.5 to +7.0 V*

**Comment:** Exposing the device to stresses above those listed in the Absolute Maximum Ratings could cause permanent damage. The device should not be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

\* $T_A = 25^\circ\text{C}$

### DC Characteristics

\*μPD7261B specifications are preliminary  
 $T_A = 0$  to +70°C,  $V_{CC} = +5.0\text{V} \pm 10\%$  unless otherwise specified

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage low	$V_{IL1}$	-0.5		+0.8	V	All except CLK, R/WCLK
Input voltage low	$V_{IL2}$	-0.5		+0.6	V	CLK, R/WCLK
Input voltage high	$V_{IH1}$	+2.0		$V_{CC} + 0.5$ V	V	All except CLK, R/WCLK
Input voltage high	$V_{IH2}$	+3.3		$V_{CC} + 0.5$ V	V	CLK, R/WCLK
Output voltage low	$V_{OL}$			+0.45	V	$I_{OL} = +2.0$ mA
Output voltage high	$V_{OH1}$	+2.4			V	$I_{OH} = -100$ μA, all except pins 21-34
Output voltage high	$V_{OH2}$	+2.4			V	$I_{OH} = -50$ μA, pins 21-34
Input leakage current	$I_{LI1}$			±10	μA	$V_{IN} = V_{CC}$ to 0.45 V, all except pins 21-34
Input leakage current	$I_{LI2}$			-500 (7261A) -700 (7261B)	μA	$V_{IN} = V_{CC}$ to 0.45 V; pins 21-34
Output leakage current	$I_{LO}$			±10	μA	$V_{OUT} = V_{CC}$ to 0.45 V
Supply current	$I_{CC}$		250	320	mA	

### Capacitance

$T_A = 25^\circ\text{C}$ ,  $V_{CC} = 0\text{V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	$C_{IN}$			15	pF	(Note 1)
Output capacitance	$C_{OUT}$			15	pF	(Note 1)
Input / Output capacitance	$C_{I/O}$			20	pF	(Note 1)

### Note:

(1)  $f = 1\text{MHz}$ . All unmeasured pins tied to GND.

## AC Characteristics

T<sub>A</sub> = 0 to +70°C; V<sub>CC</sub> = +5V ± 10%; C<sub>L</sub> = 100 pF (50 pF for 7261B-23)

Parameter	Symbol	Limits						Unit	Test Conditions	
		7261A		7261B-18		7261B-23				
		Min	Max	Min	Max	Min	Max			
<b>Processor Interface</b>										
Clock cycle	t <sub>CY</sub>	83		55				43		ns
Clock time, low	t <sub>CL</sub>	30			20			15		ns
Clock time, high	t <sub>CH</sub>	30			20			17		ns
Clock rise time	t <sub>CR</sub>		10			10			10	ns
Clock fall time	t <sub>CF</sub>		10			10			10	ns
A <sub>0</sub> , $\overline{\text{CS}}$ setup to $\overline{\text{RD}}$	t <sub>AR</sub>	0		0				0		ns
A <sub>0</sub> , $\overline{\text{CS}}$ hold from $\overline{\text{RD}}$	t <sub>RA</sub>	0		0				0		ns
$\overline{\text{RD}}$ pulse width	t <sub>RR</sub>	200			100			100		ns
Data delay from $\overline{\text{RD}}$	t <sub>RD</sub>		150			85			85	ns
Output float delay	t <sub>RDF</sub>	0	100	0	75	0	75			ns
Data delay from A <sub>0</sub> , $\overline{\text{CS}}$	t <sub>AD</sub>		150			85			85	ns
A <sub>0</sub> , $\overline{\text{CS}}$ setup to $\overline{\text{WR}}$	t <sub>AW</sub>	0		0				0		ns
A <sub>0</sub> , $\overline{\text{CS}}$ hold from $\overline{\text{WR}}$	t <sub>WA</sub>	0		0				0		ns
$\overline{\text{WR}}$ pulse width	t <sub>WW</sub>	200			100			100		ns
Data setup to $\overline{\text{WR}}$	t <sub>DW</sub>	100			55			55		ns
Data hold from $\overline{\text{WR}}$	t <sub>WD</sub>	5			5			5		ns
Recovery time from $\overline{\text{RD}}$ , $\overline{\text{WR}}$	t <sub>RV</sub>	200			70			70		ns
Reset pulse width	t <sub>RES</sub>	100			100			100		t <sub>CY</sub>
$\overline{\text{TC}}$ pulse width	t <sub>TC</sub>	100			100			80		ns
INT delay from $\overline{\text{WR}}$ ↑	t <sub>WI</sub>		200			200			200	ns
DREQ delay from $\overline{\text{WR}}$ ↑	t <sub>WRQ</sub>		250			125			125	ns
DREQ delay from $\overline{\text{RD}}$ ↑	t <sub>RRQ1</sub>		250			160			160	ns
DREQ delay from $\overline{\text{RD}}$ ↓	t <sub>RRQ2</sub>		150			130			100	ns
<b>ST506-Type Interface</b>										
R/W CLK cycle period	t <sub>RWCY</sub>	83			83			83		ns
R/W CLK time, low	t <sub>RWCL</sub>	30			30			30		ns
R/W CLK time, high	t <sub>RWCH</sub>	30			30			30		ns
R/W CLK rise time	t <sub>RWCR</sub>		10			10			10	ns
R/W CLK fall time	t <sub>RWCF</sub>		10			10			10	ns
R/W DATA setup to R/W CLK	t <sub>RDRC</sub>	40			35			35		ns
R/W DATA hold from R/W CLK	t <sub>RCRD</sub>	5			5			5		ns
R/W DATA delay from R/W CLK	t <sub>WCWD</sub>	35	90	10	60	10	60	10	60	ns
RGATE delay from R/W CLK	t <sub>RCRG</sub>		300			300			300	ns
WGATE delay from R/W CLK	t <sub>WCWG</sub>		150			150			150	ns
PCE / PCL delay from R/W CLK	t <sub>RWCPC</sub>	35	110	10	80	10	80	10	80	ns
SYNC delay from R/W CLK	t <sub>RWCSY</sub>		150			150			150	ns

**AC Characteristics (cont)**

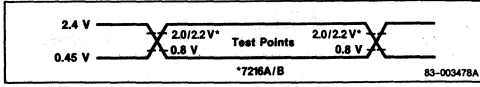
Parameter	Symbol	Limits						Unit	Test Conditions
		7261A		7261B-18		7261B-23			
		Min	Max	Min	Max	Min	Max		
<b>ST506-Type Interface (cont)</b>									
DS0, DS1 setup to STEP	t <sub>DSST</sub>	250		250		250		t <sub>CY</sub>	Normal seek mode
DIR setup to STEP	t <sub>DIST</sub>	200		200		200		t <sub>CY</sub>	
STEP pulse width	t <sub>STEP</sub>	69	85	69	85	69	85	t <sub>CY</sub>	
DS0, DS1 hold from STEP	t <sub>STDS</sub>	750		750		750		t <sub>CY</sub>	Normal seek mode; polling mode
DIR hold from STEP	t <sub>STDI</sub>	750		750		750		t <sub>CY</sub>	
DS0, DS1 hold from SKC	t <sub>SKDS</sub>	100		100		100		t <sub>CY</sub>	Normal seek mode; nonpolling
DIR hold from SKC	t <sub>SKDI</sub>	100		100		100		t <sub>CY</sub>	
DS0, DS1 setup to STEP	t <sub>DSSTB</sub>	250		250		250		t <sub>CY</sub>	Buffered seek mode
DIR setup to STEP	t <sub>DISTB</sub>	200		200		200		t <sub>CY</sub>	
STEP pulse width	t <sub>STEPB</sub>	69	85	69	85	69	85	t <sub>CY</sub>	
STEP cycle period	t <sub>STCY</sub>	500	660	500	660	500	660	t <sub>CY</sub>	
DS0, DS1 hold from STEP	t <sub>STDSB</sub>	200		200		200		t <sub>CY</sub>	Buffered seek mode; polling mode
DIR hold from STEP	t <sub>STDIB</sub>	200		200		200		t <sub>CY</sub>	
DS0, DS1 hold from SKC	t <sub>SKDSB</sub>	100		100		100		t <sub>CY</sub>	Buffered seek mode; nonpolling
DIR hold from SKC	t <sub>SKDIB</sub>	100		100		100		t <sub>CY</sub>	
Index pulse width	t <sub>IDXF</sub>	8		8		8		t <sub>RWCY</sub>	
<b>SMD interface</b>									
R/W CLK cycle period	t <sub>RWCY</sub>	83		55		43		ns	
R/W CLK time, low	t <sub>RWCL</sub>	30		20		15		ns	
R/W CLK time, high	t <sub>RWCH</sub>	30		20		17		ns	
R/W CLK rise time	t <sub>RWCR</sub>		10		10		10	ns	
R/W CLK fall time	t <sub>RWCF</sub>		10		10		10	ns	
R/W DATA setup to R/W CLK	t <sub>RDRC</sub>	40		35		35		ns	
R/W DATA hold from R/W CLK	t <sub>RCRD</sub>	5		5		5		ns	
R/W DATA delay from R/W CLK	t <sub>WCWD</sub>	35	90	10	60	10	50	ns	
BT1 delay from R/W CLK	t <sub>RCRG</sub>		300		300		300	ns	
BT0 delay from R/W CLK	t <sub>WCWG</sub>		150		150		150	ns	
SYNC delay from R/W CLK	t <sub>RWCYSY</sub>		150		150		150	ns	
BDIR setup to USTG	t <sub>BDUT</sub>	60		60		60		t <sub>CY</sub>	Unit select operation
BDIR hold from USTG	t <sub>UTBD</sub>	15		15		15		t <sub>CY</sub>	
Unit ADR setup to USTG	t <sub>UAUT</sub>	38	52	38	52	38	52	t <sub>CY</sub>	
Unit ADR hold from USTG	t <sub>UTUA</sub>	15		15		15		t <sub>CY</sub>	
BDIR setup to TG1	t <sub>BDT1</sub>	27	48	27	48	27	48	t <sub>CY</sub>	Cylinder select operation
BDIR hold from TG1	t <sub>TBD</sub>	60		60		60		t <sub>CY</sub>	

## AC Characteristics (cont)

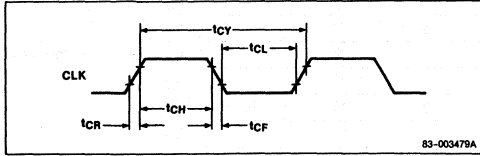
Parameter	Symbol	Limits						Unit	Test Conditions
		7261A		7261B-18		7261B-23			
		Min	Max	Min	Max	Min	Max		
<b>SMD Interface (cont)</b>									
CYL. ADR setup to $\overline{TG1}$	$t_{CAT1}$	27	48	27	48	27	48	$t_{CY}$	Cylinder select operation
CYL. ADR hold from $\overline{TG1}$	$t_{T1CA}$	24		24		24		$t_{CY}$	
$\overline{TG1}$ pulse width	$t_{TG1}$	24	36	24	36	24	36	$t_{CY}$	Head select operation
BDIR setup to $\overline{TG2}$	$t_{BDT2}$	15		15		15		$t_{CY}$	
BDIR hold from $\overline{TG2}$	$t_{T2BD}$	70		70		70		$t_{CY}$	
HEAD ADR setup $\overline{TG2}$	$t_{HAT2}$	15	70	15	70	15	70	$t_{CY}$	
HEAD ADR hold from $\overline{TG2}$	$t_{T2HA}$	24		24		24		$t_{CY}$	RT2, FAULT CLR, SERVO, DATA STB control timing
$\overline{TG2}$ , pulse width	$t_{TG2}$	24	36	24	36	24	36	$t_{CY}$	
BDIR setup to $\overline{TG3}$	$t_{BDT3}$	24		24		24		$t_{CY}$	
BDIR hold from $\overline{TG3}$	$t_{T3BD}$	24	36	24	36	24	36	$t_{CY}$	
$\overline{TG3}$ , pulse width	$t_{TG3}$	56	100	56	100	56	100	$t_{CY}$	
BT2, 3, 4, 6, 7, 8 setup to $\overline{TG3}$	$t_{BT3}$		56		56		56	$t_{CY}$	
BT4, 6 hold from $\overline{TG3}$	$t_{T3BT1}$	24		24		24		$t_{CY}$	Sense unit status timing
BT2, 3, 7, 8 hold from $\overline{TG3}$	$t_{T3BT2}$	75		75		75		$t_{CY}$	
BDIR delay from $\overline{SSTG}$	$t_{STBD}$	24		24		24		$t_{CY}$	
BDIR high time	$t_{BDIR}$	54	66	54	66	54	66	$t_{CY}$	
BT9 setup to BDIR	$t_{BTBD}$	38	52	38	52	38	52	$t_{CY}$	
BT9 hold from BDIR	$t_{BDT}$	24	33	24	33	24	33	$t_{CY}$	
$\overline{SSTG}$ pulse width	$t_{SSTG}$		370		370		370	$t_{CY}$	
Index pulse width	$t_{IDXH}$	8		8		8			
SCT pulse width	$t_{SCT}$	8		8		8			

### Timing Waveforms — Host System Interface

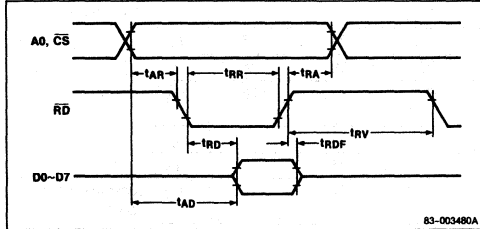
#### AC Test Points (Except R/W CLK, CLK)



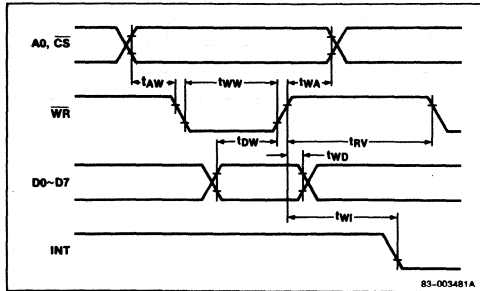
#### CLK Waveform



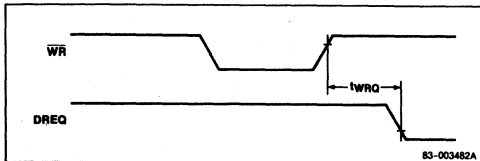
#### Read Timing



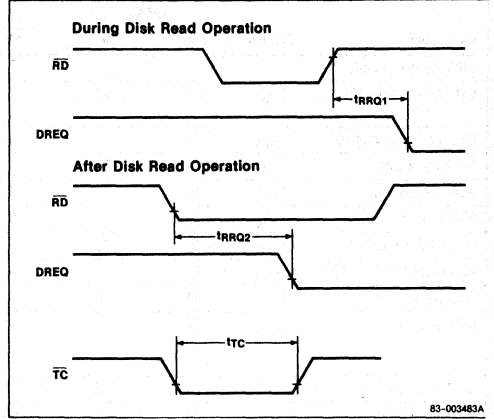
#### Write Timing



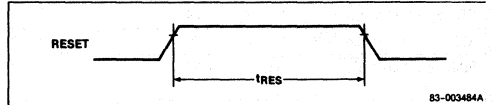
#### DMA Write Timing



#### DMA Read Timing

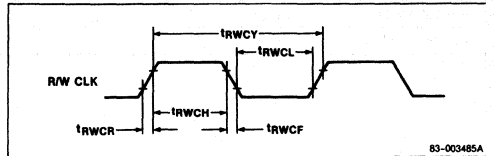


#### Reset Waveform

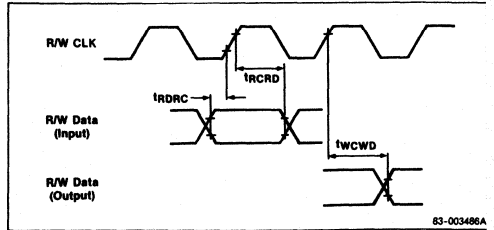


### Timing Waveforms — SMD Interface

#### R/W CLK Waveform

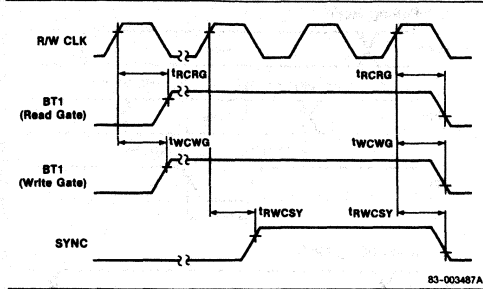


#### Data Read/Write Timing

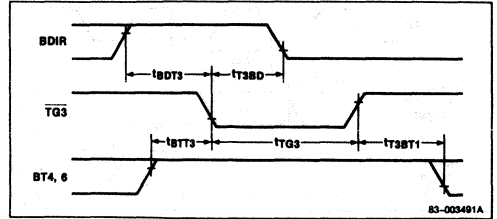


## Timing Waveforms — SMD Interface (cont)

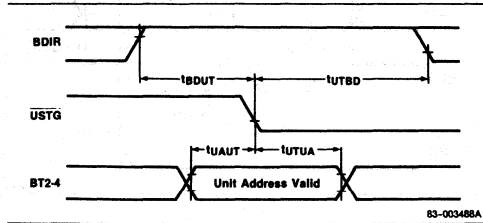
### Read/Write Timing



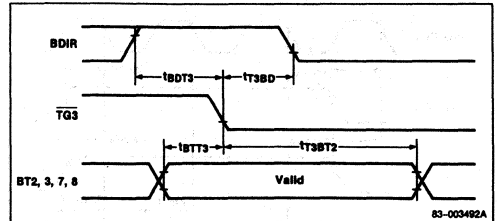
### Bit Bus Timing, Fault Clear/Return-to-Zero



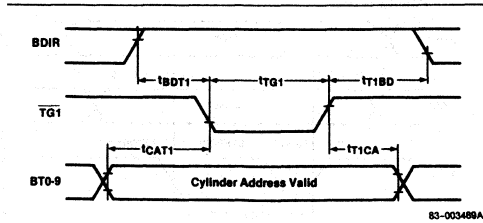
### Unit Select Timing



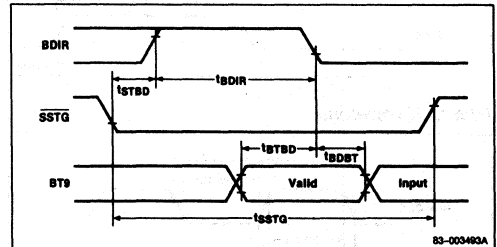
### Bit Bus Timing, Servo Offset/Data Strobe



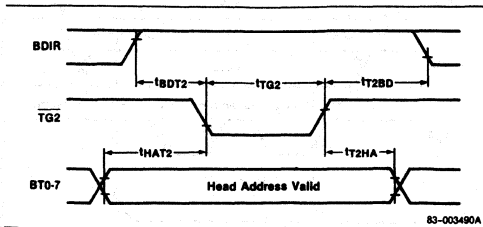
### Seek Timing



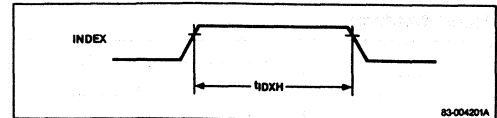
### Bit Bus 9 Timing



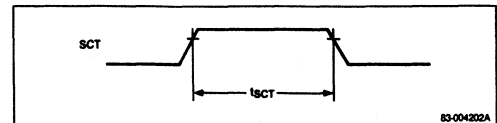
### Head Select Timing



### Index Waveform



### Sector Waveform

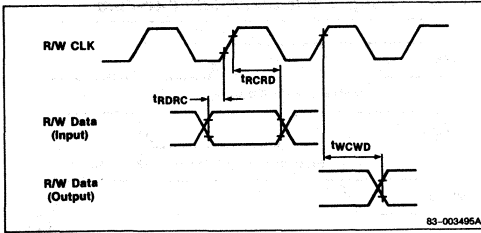


5

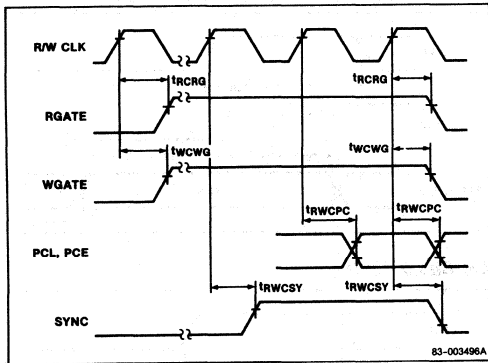
## $\mu$ PD7261A/B

### Timing Waveforms — ST506-Type Interface

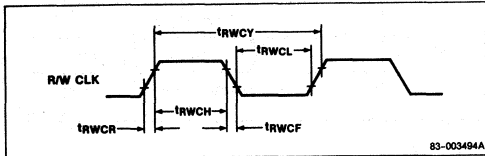
#### Data Read/Write Operation



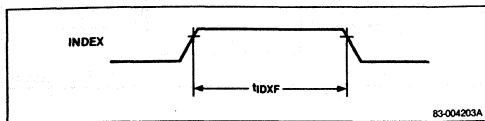
#### Read/Write Operation



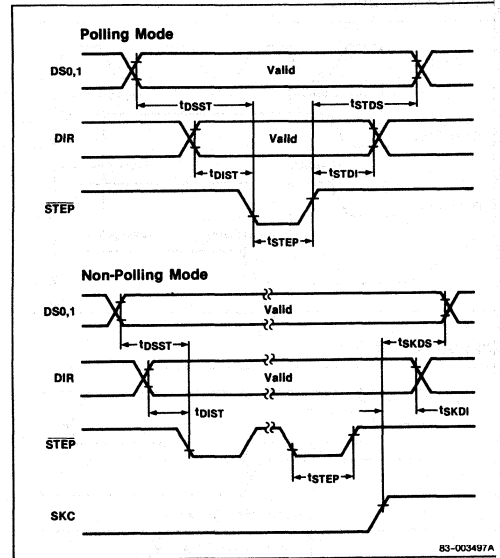
#### R/W CLK Waveform



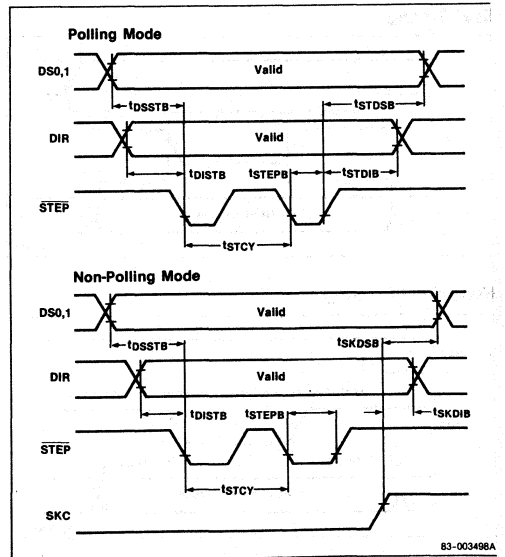
#### Index Waveform



#### Normal Seek Operation

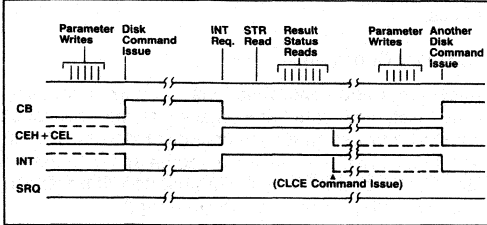


#### Buffered Seek Operation

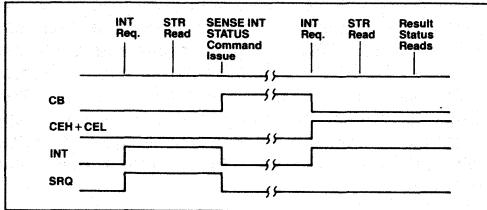


## Timing Waveforms — ST506-Type Interface (cont)

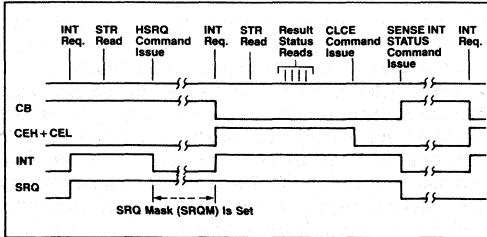
### Read/Write Sequence (Disk Command Issue)



### Sense Interrupt Status Request When Controller Not Busy



### Sense Interrupt Status Request When Controller Busy



## High-Level Commands

### Specify

Allows user to select SMD or ST506-type mode data block length, ending track number, end sector number, gap length, track at which write current is reduced, ECC or CRC function, choice of polynomial, and polling mode enable.

### Sense Interrupt Status

When a change of disk status occurs, the HDC will interrupt the host CPU. This command will reveal the cause of interrupt, such as seek end, disk ready change, seek error, or equipment check. The disk unit address is also supplied.

### Sense Unit Status

The host CPU specifies the drive numbers and the HDC will return information such as write fault, ready, track 000, seek complete and drive selected, or for SMD units fault, seek error, on cylinder, unit ready, AM found, write protected, seek end, and unit selected.

### Detect Error

Used after a read operation where ECC has been employed. The detect error command supplies the information needed to allow the host CPU to execute an error correction routine. (Only allowed when an actual correctable error is detected by the HDC.)

### Recalibrate

Returns the disk drive heads to the home position or track 000 position. Has four modes of operation: SMD, normal, buffered, or nonpolling.



**Seek**

Moves the disk drive heads to the specified cylinder. As in recalibrate, seek has four modes of operation.

**Format**

This command is used to initialize the medium with the ~~desired~~ format which includes various gap lengths, data patterns, and CRC codes. This command is used in conjunction with the specify command.

**Verify ID**

Used to verify the ID bytes with data from memory. Performs the operation over a specified number of sectors.

**Read ID**

Used to verify the position of the read/write heads.

**Read Diagnostic**

Used in SMD mode only, the command allows the programmer to read a sector of data even if the ID portion of the sector is defective. Only one sector at a time can be read.

**Read Data**

Reads and transfers to the system memory the number of sectors specified. The HDC can read multiple sectors and multiple tracks with one instruction.

**Scan**

Compares a specified block of memory with specified sectors on the disk. The 7261A/7261B continues until a sector with matching data is found, until the sector count reaches zero, or the end of the cylinder is reached.

**Verify Data**

Makes a sector-by-sector comparison of data in the system memory by DMA transfer. As in read operation, multiple sectors and tracks may be verified with this command.

**Write Data**

Data from the system memory, transferred by DMA, is written onto the specified disk unit. As in the read command, data may be written onto successive sectors and tracks.

**Auxiliary Command**

Allows four additional functions to be executed: software reset, clear data buffer, mask interrupt request bit (masks interrupts caused by change of status of drives), and reset interrupt caused by command termination (used when no further disk commands will be issued, which would normally reset the interrupt).

**Command Operation**

There are three phases for most of the instructions that the μPD7261A/7261B can execute: command phase, execution phase, and result phase. During the command phase the host CPU loads preset parameters into the μPD7261A/7261B FIFO via the data bus and by successive write pulses to the part with A<sub>0</sub> and CS true low. Once the required parameter bytes are loaded the appropriate command is initiated by issuing a write pulse with A<sub>0</sub> high and CS low and the command code on the data bus.

The μPD7261A/7261B is now in the execution phase. This can be verified by examining the status register bit 7 (the controller busy bit). The execution phase is ended when a normal termination or an abnormal termination occurs. An abnormal termination can occur due to a read or write error, or a change of status in the addressed disk drive. A normal termination occurs when the command given is correctly completed. (This is indicated by bits in the status register.) The result phase is then entered. The host CPU may read various result parameters from the FIFO. These result parameters may be useful in determining the cause of an interrupt, or the location of a sector causing a read error, for example.

The chart shown in table 2 illustrates the preset parameters and result parameters that are associated with each command. The abbreviations are defined at the end of table 2.

**Table 2. Preset Parameters and Result Status Byte**

Disk Command	Command Code	Preset Parameters / Result Status							
		1st	2nd	3rd	4th	5th	6th	7th	8th
Detect error	0100X	EADH	EADL	EPT1	EPT2	EPT3			
Recalibrate	0101[B]	IST*							
Seek	0110[B]	PCNH	PCNL						
Format	0111(S)	PHN EST	(PSN) SCNT	SCNT	DPAT	GPL1	[GPL3]		
Verify ID	1000(S)	PHN EST	(PSN) SCNT	SCNT					
Read ID	1001(S)	PHN EST	(PSN) SCNT	SCNT					
(Read diagnostic)	1010X	PHN EST	PSN						
Read data	1011X	PHN EST	(FLAG) PHN	LCNH (FLAG)	LCNL	LHN	LSN	SCNT LSN	SCNT
Check	1100X	PHN EST	(FLAG) PHN	LCNH (FLAG)	LCNL	LHN	LSN	SCNT LSN	SCNT
Scan	1101X	PHN EST	(FLAG) PHN	LCNH (FLAG)	LCNL	LHN	LSN	SCNT LSN	SCNT
Verify data	1110X	PHN EST	(FLAG) PHN	LCNH (FLAG)	LCNL	LHN	LSN	SCNT LSN	SCNT
Write data	1111X	PHN EST	(FLAG) PHN	LCNH (FLAG)	LCNL	LHN	LSN	SCNT LSN	SCNT
Sense interrupt status	0001X	IST							
Specify	0010X	MODE	DTLH	DTLL	ETN	ESN	GPL2	(MGPL1) [RWCH]	[RWCL]
Sense unit status	0011X	UST							

**Note:**

- ( ): These are omitted for soft-sector disks.
- [ ]: These are omitted for hard-sector disks.
- \*: IST available as a result byte only when in nonpolling mode.
- B: Indicates buffered mode when set.
- S: Indicates Skewed mode (SMD only) when set.
- X: Indicates don't care.

**Mnemonic Definitions**

- EADH Error address, high byte
- EADL Error address, low byte
- EPT1 Error pattern, byte one
- EPT2 Error pattern, byte two
- EPT3 Error pattern, byte three
- PCNH Physical cylinder number, high byte

**Mnemonic Definitions (cont)**

- PCNL Physical cylinder number, low byte
- PHN Physical head number
- PSN Physical sector number
- SCNT Sector count
- DPAT Data pattern
- GPL1 Gap length one
- GPL3 Gap length three
- EST Error status byte
- FLAG Flag byte
- LCNH Logical cylinder number, high byte

**Mnemonic Definitions (cont)**

LCNL	Logical cylinder number, low byte
LHN	Logical head number
LSN	Logical sector number
IST	Interrupt status byte
MODE	Mode
DTLH	Data length, high byte
DTLL	Data length, low byte
ETN	Ending track number
ESN	Ending sector number
GPL2	Gap length two
RWCL	Write current cylinder, low byte
RWCH	Write current cylinder, high byte
UST	Unit status byte
MGPL1	Modified gap length 1

**Status Register**

This register is a read only register and may be read by asserting  $\overline{RD}$  and  $\overline{CS}$  with  $A_0$  high. The status register may be read at any time. It is used to determine controller status and partial result status. See table 3.

**Table 3. Status Register Bits**

Pin		Function
No.	Name	
D7	CB (Controller busy)	Set by a disk command issue. Cleared when the command is completed. (This bit is also set by an external reset signal or an RST command, but will be cleared at the completion of the reset function.) When this bit is set, a new disk command will not be accepted.
D6, D5	CEH, CEL (Command end)	<p>CEH = 0 and CEL = 0 A disk command is in process, or no disk command is issued after the last reset signal or the last CLCE auxiliary command. Both the CEH and CEL bits are cleared by a disk command, a CLCE auxiliary command, or a reset signal.</p> <p>CEH = 0 and CEL = 1 Abnormal termination of a disk command. Execution of a disk command was started, but was not successfully completed.</p> <p>CEH = 1 and CEL = 0 Normal termination of a disk command. The execution of a disk command was completed and properly executed.</p> <p>CEH = 1 and CEL = 1 Invalid command issue.</p>

**Table 3. Status Register Bits (cont)**

Pin		Function
No.	Name	
D4	SRQ (Sense interrupt status request)	When a seek end, an equipment check condition, or a ready signal state change is detected, this bit is set. Requesting a sense interrupt status command be issued to take the detailed information. This bit is cleared by an issue of that command or by a reset signal.
D3	RRQ (Reset request)	Set when controller has lost control of the format controller (missing address mark, for example). An auxiliary RST command or RESET signal will clear this bit.
D2	IER (ID error)	Set when a CRC error is detected in the ID field. An auxiliary RST or another disk command will reset this bit.
D1	NCI (Not coincident)	Set if the controller cannot find a sector on the cylinder which meets the comparison condition during the execution of a scan command. This bit is also set if data from the disk does not coincide with the data from the system during a verify ID or a verify data command. This bit is cleared by a disk command or a reset signal.
D0	DRQ (Data request)	During execution of write ID, verify ID scan, verify data, or a write data command, this bit is set to request the data be written into the data buffer. During execution of read ID, read diagnostic, or read data command, this bit is set to request that data be read from the data buffer.

## Error Status Byte

This byte is available to the host at the termination of a read, write, or data verification command and provides additional error information to the host CPU. If the status register indicates a normal command termination, it can be assumed that the command was executed without error and it is not necessary to read this byte. When it is necessary to determine the cause of an error this byte may be read by issuing an  $\overline{RD}$  pulse with  $\overline{CS}$  and  $A_0$  low. The remaining result bytes associated with a particular command may be read by issuing additional  $\overline{RD}$  pulses. Data transfer from or to the FIFO is asynchronous and may occur at rates up to 2.5 Mbytes per second. See table 4.

**Table 4. Error Status Bits**

Pin		
No.	Name	Function
D <sub>7</sub>	ENC (End of cylinder)	Set when the controller tries to access a sector beyond the final sector of a cylinder. Cleared by a disk command or an auxiliary RST command.
D <sub>6</sub>	OVR (Overrun)	When set, indicates that the FIFO became full during a read operation, or empty during a write operation.
D <sub>5</sub>	DER (Data error)	A CRC or an ECC error was detected in the data field.
D <sub>4</sub>	EQC (Equipment check)	A fault signal from the drive has been detected or a track 0 signal has not been returned within a certain time interval after the recalibrate command was issued.
D <sub>3</sub>	NR (Not ready)	The drive is not in ready state.
D <sub>2</sub>	ND (No data)	The sector specified by ID parameters was not found on the track.
D <sub>1</sub>	NWR (Not writable)	Set if write protect signal is detected when the controller tries to write on the disk. It is cleared by a disk command or by an auxiliary RST command.
D <sub>0</sub>	MAM (Missing address mark)	This bit is set if during execution of read data, check, scan, or verify data commands, no address mark was found in the data field or if during execution of a read ID or verify ID command, no address mark was detected in the ID field.

## Interrupt Status Byte

This byte is made available to the host CPU by executing the Sense Interrupt Status command. This command should be issued only when the μPD7261A/7261B requests it, as indicated by bit D<sub>4</sub> of the status register. This byte reveals changes in disk drive status that have occurred. See table 5.

**Table 5. Interrupt Status Bits**

Pin		
No.	Name	Function
D <sub>7</sub>	SEN (Seek end)	A seek end or seek complete signal has been returned after a seek or a recalibrate command was issued.
D <sub>6</sub>	RC (Ready change)	The state of the ready signal from the drives has changed. The state itself is indicated by the NR bit.
D <sub>5</sub>	SER (Seek error)	Seek error has been detected on seek end.
D <sub>4</sub>	EQC (Equipment check)	Identical to bit 4 of the error status byte.
D <sub>3</sub>	NR (Not ready)	Identical to bit 3 of the error status byte.
D <sub>2</sub> -D <sub>0</sub>	UA <sub>2</sub> -UA <sub>0</sub> (Unit address)	The unit address of the drive which caused an interrupt request on any of the above conditions.

5

## Drive Interface

The μPD7261A/7261B has been designed to implement two of the more popular types of interfaces: the SMD (Storage Module Drive) and the floppy-like Winchester drive which has come to be known as the ST506 interface. The desired interface mode is selected by the Specify command.

## ST506-Type Interface

In the ST506 mode the μPD7261A/7261B performs MFM encoding and decoding at data rates to 6 MHz and provides all necessary drive interface signals. Included internally is circuitry for address mark detection, sync area recognition, serial-to-parallel-to-serial conversion, an 8-byte FIFO for data buffering, and circuitry for logical addressing of the drives. External circuitry required consists of control signal buffering, a delay network for precompensation, a phase-lock loop, a write clock oscillator and a differential transceiver for drive data. The floppy-like interface can be implemented with as few as 7 IC's using NEC's hard-disk interface chip, the μPD9306A, or with 12 to 14 SSI ICs. See figure 1.

## μPD7261A/B

Figure 1. μPD7261A/7261B ST506-Type Interface

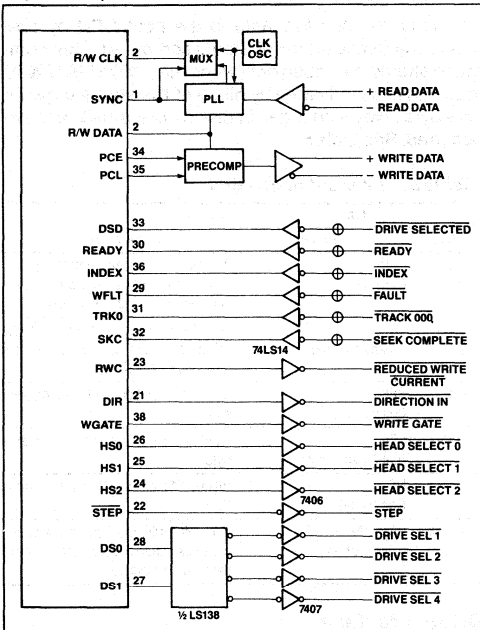
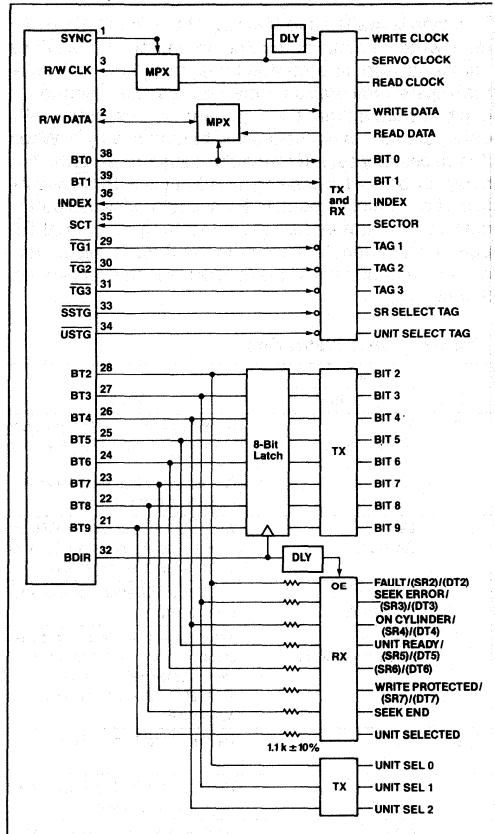


Figure 2. μPD7261A/7261B SMD Interface



### SMD Interface

In the SMD mode the μPD7261A/7261B will support data rates to 10 MHz/15MHz in the NRZ format. All control functions necessary for an SMD interface are implemented on-chip with de-multiplexing of 8 data lines performed externally by a single 8-bit latch. A small amount of logic is required to multiplex the data and clock lines, and differential drivers and receivers are required to implement the actual interface. Depending on individual logic design and the number of drives used, the SMD interface may be implemented with as few as 12 ICs. See figure 2.

#### Note:

CLK (pin 37) frequency must be a minimum of  $1.1 \times \text{NRZ data rate}$ .

### Internal Architecture

The μPD7261A/7261B can be divided into three major internal logic blocks: command processor; format controller; microprocessor interface.

### Command Processor

The command processor is an 8-bit microprocessor with its own instruction set, program ROM, scratchpad RAM, ALU, and I/O interface. Its major functions are:

- To decode the commands from the host microcomputer that are received through the 8-bit data bus
- To execute seek and recalibrate commands
- To interface to the drives and read the drive status lines
- To load the format controller with the appropriate microcode, enabling it to execute the various read/write data commands.

The command processor microprocessor is idle until it receives the command from the host microcomputer. It then reads the parameter bytes from the FIFO, and loads them into its RAM. The command byte is decoded and, depending on its opcode, the appropriate subroutine from the 2.6K internal ROM is selected and executed. Some of these commands are executed by the command processor without involvement of the format

controller. When data transfers to and from the disk are made, the command processor loads the appropriate microcode into the format controller, then relinquishes control. When the data transfer is complete, the command processor again takes control. One other important function that the command processor performs is managing the interface to the disk drives. The command processor contains an I/O port structure similar to many single-chip microcomputers in that the ports may be configured as input or output pins. Depending on the mode of operation selected by the Specify command, the command processor will use the bidirectional I/O lines for different functions.

### Command Register

This register is a write only register. It is selected when the A<sub>0</sub> input is high and the CS input is low. There are two kinds of commands: disk commands and auxiliary commands. Each command format is shown in figure 3.

An auxiliary command is accepted at any time and is immediately executed, while a disk command is ignored if the on-chip processor is busy processing another disk command. A valid disk command causes the processor to begin execution using the parameters previously loaded into the data buffer. Disk commands and the parameters needed are described in the Microprocessor Interface section.

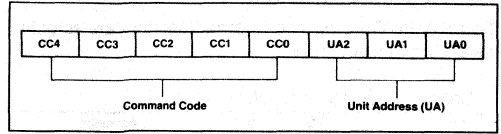
### Command Codes

CC4-CC0					
0	0	0	0	X	(Auxiliary Command)
0	0	0	1	X	Sense int. status (Note 1)
0	0	1	0	X	Specify (Note 1)
0	0	1	1	X	Sense unit status
0	1	0	0	X	Detect error (Note 1)
0	1	0	1	[B]	Recalibrate
0	1	1	0	[B]	Seek
0	1	1	1	[S]	Format
1	0	0	0	[S]	Verify ID
1	0	0	1	[S]	Read ID
1	0	1	0	X	Read diagnostic
1	0	1	1	X	Read data
1	1	0	0	X	Check
1	1	0	1	X	Scan
1	1	1	0	X	Verify data
1	1	1	1	X	Write data

**Note:**

- (1) The UA field is 000.
- [B] Indicates buffered mode when set.
- [S] Indicates skewed mode when set.

**Figure 3. Disk Command Byte**



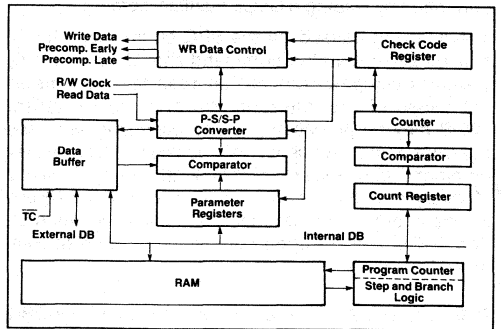
### Format Controller

The format controller is built with logic that enables it to execute instructions at very high speed: one instruction per single clock cycle. The major functions it performs are:

- Serial-to-parallel and parallel-to-serial data conversion
- CRC and ECC generation and checking
- MFM data decoding and encoding
- Write precompensation
- Address mark detection and generation
- ID field search in soft-sector format
- DMA data transfer control during read/write operations.

The major blocks in the format controller are the sequencer and the serial/parallel data handler. The sequencer consists of a writable control store (32 words by 16 bits), a program counter, branch logic, and the parameter register. The serial/parallel logic consists of a parallel-to-serial converter for disk write operations, a serial-to-parallel converter for disk read operations, precompensation logic for writing MFM data, comparator logic that locates sync fields, address marks, and ID fields. There is also comparator logic that is used during Verify Data commands. See figure 4.

**Figure 4. Block Diagram of the Format Controller**



Microprocessor Interface

Read/Write Control. The internal registers are selected as shown in truth table 6.

Table 6. Register Selection Table

CS	A <sub>0</sub>	RD	WR	Selection
0	0	0	1	Data buffer register (Note 1)
0	0	1	0	Data buffer register (Note 1)
0	1	0	1	Status register
0	1	1	0	Command register
0	X	1	1	Don't care
1	X	X	X	Don't care
0	X	0	0	Inhibited

Note: (1) Preset parameters and result status information are written and read from the result status register in the HDC through this data buffer register.

Interrupt. The interrupt request line is activated or inactivated according to the following equation:

INT = CEH + CEL + SRQ • SRQM

This means that if either of the command end bits is set or if the sense interrupt status request bit is set (and the SRQM mask is not set), then an interrupt will be generated. The command end bits, CEH and CEL, are set by command termination.

The SRQ bit is set when an equipment check condition or a state change of the ready signal from the disk drives is detected. It is also set when a seek operation is completed. Under these conditions the INT line is activated unless the SRQM mask is set.

Both of the CEH and CEL bits are cleared by a disk command, but both bits may be cleared before the next disk command by issuing a CLCE auxiliary command.

The interrupt caused by the SRQ bit indicates that a sense interrupt status command should be issued by the host microprocessor so that it can determine the exact cause of the interrupt. However, the μPD7261A/7261B may be processing a disk command when the interrupt occurs. Since it is not possible to issue a disk command while the μPD7261A/7261B is busy, an HSRQ auxiliary command can be issued to set the SRQM (sense interrupt request mask) and mask the interrupt. The SRQM is reset upon completion of the disk command in progress.

DMA Control. When true, the DREQ pin and the DRQ (data request) bit of the status register indicate a request for data transfer between the disk controller and external memory. These are activated during execution of the following disk commands:

HDC ← memory: Format, Verify ID, Scan, Verify Data, Write Data

HDC → memory: Read ID, Read Diagnostic, Read Data

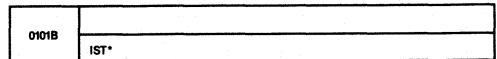
Data being read from a disk or external memory is temporarily stored in the data buffer (8 bytes maximum), and is transferred to external memory or a disk, respectively.

Data transfers are terminated externally by a reset signal or by a read or a write data operation coinciding with an active terminal count (TC) signal. They are also terminated internally when an abnormal condition is detected or all the data specified by the sector count parameter (SCNT) has been transferred.

Data transfers are accomplished by RD or WR signals to the μPD7261A/7261B when DREQ is active. During read operations, DREQ goes active when the FIFO contains three or more bytes. If the FIFO contains three bytes and an RD pulse is issued, DREQ goes low within tRRQ1. DREQ will stay active on the final sector until the final byte is extracted. In this case, DREQ goes low within tRRQ2. During write operations DREQ is asserted as soon as a Write Data command is accepted. DREQ remains high until the FIFO contains six bytes, at which time it goes low within tWAI. DREQ corresponds to FIFO almost-full and FIFO almost-empty as implemented in the μPD7261A/7261B. This has been done so that a fast DMA controller may actually overrun the FIFO by one or two bytes without harm.

Commands

Recalibrate



The read/write heads of the specified drive are retracted to the cylinder 0 position. IST is available as a result byte only if polling mode is disabled. See Specify.

Hard-Sector. An RTZ (Return to Zero) signal is asserted on the bit-6 line with the TAG-3 bit being set. Then the CEH bit of the status register is set indicating a normal termination of the command.

After this command is given, the HDC checks the seek end, unit ready, and fault lines of the drive continually until an active signal is detected on these lines. Then the SRQ bit of the status register is set indicating that a sense interrupt status command should be performed. Each bit of the IST (interrupt status) byte is set according to the result, in anticipation of the sense interrupt status command.

**Soft-Sector.** There are four different ways to implement the Recalibrate command when the ST506 interface mode has been specified. Both polling and nonpolling modes of operation are provided, with both normal or buffered Recalibrate commands available in either mode.

**Normal Mode with Polling.** The CEH bit of EST is set to 1 immediately after the Recalibrate command is issued (a recalibrate command may now be issued to another drive). The HDC now begins generating step pulses at the specified rate. The PCN for the drive is cleared and the TRK0 signal is checked while stepping pulses are sent to one or more drives. When TRK0 is asserted, the SEN (seek end) bit of the IST (interrupt status) byte is set and the SRQ bit of the status register is set. This causes an interrupt and requests that a sense interrupt status command be issued. If 1023 pulses have been sent and TRK0 is not asserted, then the SRQ bit is again set, but with the SER (seek error) and EQC (equipment check) bits of the IST byte set. The ready signal of each drive is checked before each step pulse is sent, and the Recalibrate command is terminated if the drive enters a not-ready state, whereby the NR bit of the IST byte is set to 1.

**Normal Mode with Polling Disabled.** Operation is similar to that in "Normal Mode with Polling", but the CEH and CEL bits of the status register are not set until either the SEN (seek end) or the SER (seek error) condition occurs. The SRQ bit is not set when polling is disabled, and the IST byte is now available as a result byte when the Recalibrate command is terminated (see "Pretest Parameters and Result Status Bytes"). It is not possible to overlap Recalibrate operations in this mode.

**Buffered Mode with Polling.** This mode operates in a manner similar to that described as "Normal Mode with Polling", but with the following differences:

- 1) 1023 step pulses are sent at a high rate of speed (approximately 50 μs between pulses)
- 2) After the required number of pulses are sent, the CEH bit is set, and then additional Recalibrate or Seek commands will be accepted for other drives
- 3) The SRQ bit is set when the drive asserts SKC, which causes the SEN bit of the IST byte to be set
- 4) If SEN is not set within the time it takes to send 1023 "normal" pulses (i.e., when in normal stepping mode), then SER and EQC of the IST byte are set.

**Buffered Mode with Polling Disabled.** 1023 stepping pulses are immediately sent after the Recalibrate command is issued. CEH and/or CEL is set when SEN or SER occurs. SEN is set when TRK0 from the addressed drive is asserted. SER is set if TRK0 is not asserted within the time required to send 1023 "normal" pulses. The Recalibrate command will be terminated abnormally if a not-ready condition occurs prior to SEN being

set. The SRQ bit of the status register is not set. The IST byte (interrupt status) is available as a result byte when either CEH or CEL is set.

## Seek

0110B	PCNH	PCNL
	IST*	

PCNH = Physical Cylinder Number, High Byte  
PCNL = Physical Cylinder Number, Low Byte

The read/write heads of the specified drive are moved to the cylinder specified by PCNH and PCNL. IST is available as a result byte only if polling mode is disabled. See Specify.

**Hard-Sector.** The contents of PCNH and PCNL are asserted on the BIT0 through BIT9 output lines of the SMD interface with the TAG1 control line being set. (The most significant six bits of PCNH are not used.) The CEH bit of the status register is then set, and the command is terminated normally.

The HDC then checks the seek end, unit ready and fault lines of the drive continually until an active signal is detected on these lines. The SRQ bit of the status register is then set requesting that a Sense Interrupt Status command be performed. Each bit of the IST (interrupt status) byte is set appropriately in anticipation of the Sense Interrupt Status command.

**Soft-Sector (Normal Stepping, Polling Enabled).** In this mode, the CEH bit of the status register is set to 1 as soon as the Seek command is issued. This allows a Seek or Recalibrate command to be issued to another drive. The HDC now sends stepping pulses at the specified rate and monitors the ready signal. Should the drive enter a not-ready state, the SER bit of the IST byte is set and the SRQ bit of the status register is set, causing an interrupt and requesting a Sense Interrupt Status command. When the drive asserts the seek complete (SKC) signal, the SEN bit of the IST byte is set and the SRQ bit of the status register is set, again requesting service.

**Soft-Sector (Normal Stepping, Polling Disabled).** Stepping pulses to the drive begin as soon as the Seek command is accepted. The ready signal is checked prior to each step pulse. If the drive enters a not-ready state the seek command is terminated abnormally (CEL = 1), and SER of the IST byte is set. If the seek operation is successful, the seek command will be terminated normally (CEH = 1) when the drive asserts SKC (seek complete). The SEN (seek end) bit of the IST byte is set and the IST (interrupt status) byte is available as a result byte. The Sense Interrupt Status command is not allowed (SRQ is not set), nor can seek operations be overlapped in this mode.



**Soft-Sector (Buffered Stepping, Polling Enabled).** As soon as the Seek command is accepted by the HDC, high-speed stepping pulses are generated. As soon as the required number of pulses are sent, CEH is set to 1, indicating a normal termination. Another Seek command in the same mode may now be issued. The drive is now controlling its own head positioner and asserts SKC when the target cylinder is reached.) If the drive has not asserted SKC (seek complete) within the time it takes to send the required number of pulses in normal stepping mode, or if the drive enters a not-ready state, then the SER bit of the IST byte and the SRQ bit of the status register are set. Otherwise, the SEN bit of the IST byte is set, along with SRQ of the status register.

**Soft-Sector (Buffered Stepping, Polling Disabled).** In this mode, the appropriate number of high-speed stepping pulses are sent as soon as the Seek command is issued. If the drive enters a not-ready state, or if SKC (seek complete) is not asserted within the time it takes to send the required number of pulses in normal stepping mode, then the Seek command is terminated normally (generating an interrupt). The IST byte is available as a result byte and the appropriate bit is set; i.e., SER and EQC or NR (not ready). If the seek operation is successful, the Seek command is terminated normally (CEH = 1) and the SEN bit of the IST byte is set. The IST byte is available as a result byte. The Sense Interrupt Status command is not allowed (SRQ is not set), nor can seek operations be overlapped in this mode.

### Format

0111S	PHN	(PSN)	SCNT	DPAT	GPL1	(GPL3)
	EST	SCNT				

PHN = Physical Head Number  
 PSN = Physical Sector Number  
 SCNT = Sector Count  
 DPAT = Data Pattern  
 GPL1 = Gap Length 1  
 GPL3 = Gap Length 3  
 EST = Error Status

This command is used to write the desired ID and data format on the disk.

(1) When using hard-sector drives, this command will begin format writing at the sector specified by PHN and PSN, which are loaded during command phase.

When soft-sector drives are specified, this command will begin format-writing at the sector immediately following the index pulse on the track specified by PHN.

In either case, data transmitted from the local memory by DMA operation is written into the ID field, and the data field is filled with the data constant specified by DPAT until DTL (data length) is zero. DTL is established during the specify command with DTLH and DTTL. The sector count, SCNT, is decremented by one at the end of the Format operation on each sector. The following

bytes are required by the HDC for each sector: (FLAG LCNH, LCNL, LHN, and LSN. FLAG is omitted on soft sector drives. These bytes are transferred by DMA.

The format operation produces the various gaps with length as specified by GPL1, GPL2 (See Specify), and GPL3 (For soft-sector only.)

**Note:**

GPL3 may not exceed decimal value of 44.

(2) The above operation is repeated until SCNT is equal to zero. The execution of the command is terminated normally, when the content of SCNT is equal to zero and the second index pulse has occurred.

(3) When using a hard-sector drive, it is possible to write the ID field displaced from the normal position by 6 bytes by setting the skew bit of the command byte (S = 1). This is useful when defective media prevent writing in the normal area of the sector.

(4) Items 4, 5, and 8 of the Read Data and item 4 of the Write Data command are identical for this command. Refer to these items (which appear later in this section) for remaining format operation details.

### Verify ID

1000S	PHN	(PSN)	SCNT
	EST	SCNT	

PHN = Physical Head Number  
 PSN = Physical Sector Number  
 SCNT = Sector Count  
 EST = Error Status

ID bytes of specified sectors are read and compared with the data that are accessed from local memory via DMA control. The first sector that is verified is specified by PHN and PSN when a hard-sector disk is used. For soft-sector disks, only PHN is given and the Verify ID command begins comparisons with the first physical sector on the track.

Byte comparisons continue as long as successful until the sector count is zero or a CRC error is found.

When using a hard-sector drive, it is possible to have the HDC verify a skewed ID field by setting the skew bit of the command byte. Refer to the Format section, given earlier, for details.

### Read ID

1001S	PHN	PSN	SCNT
	EST	SCNT	

PHN = Physical Head Number  
 PSN = Physical Sector Number  
 SCNT = Sector Count  
 EST = Error Status

ID bytes of specified sectors are read and transferred to local memory by DMA.

Hard-sector disks: Beginning with the sector specified by PHN and PSN, the ID bytes of each sector are read until an error is found or the SCNT has reached zero.

It is also possible to perform the above operation with skewed ID fields by setting the skew bit of the command byte. This will allow reading ID fields that have been shifted by 64 bytes by the Skewed Format command.

Soft-sector disks: This command will begin checking ID fields immediately following the index pulse and will continue until one valid ID field is read, or until the second index pulse is detected or SCNT = 0, whichever occurs first.

## Read Diagnostic

1010X	PHN PSN
	EST

PHN = Physical Head Number  
 PSN = Physical Sector Number  
 EST = Error Status

This command is implemented only for hard-sector disks. The desired physical sector is specified, and the data field will be read even if the ID bytes of that sector contain a CRC error. Only one sector at a time may be read by this command.

## Read Data

1011X	PHN (FLAG) LCNH LCNL LHN LSN SCNT
	EST PHN (FLAG) LCNH LCNL LHN LSN SCNT

PHN = Physical Head Number  
 FLAG = Flag Byte, Hard-Sector ID Field Only  
 LCNH = Logical Cylinder Number, High Byte  
 LCNL = Logical Cylinder Number, Low Byte  
 LHN = Logical Head Number  
 LSN = Logical Sector Number  
 SCNT = Sector Number  
 EST = Error Status

This command is used to read and transfer data via DMA from the disk to the local memory.

(1) The HDC reads data from the specified sector which is determined by the following preset parameters: FLAG (for hard-sector only), LCNH, LCNL, LHN, and LSN. The drive is selected by UA (unit address) in the command byte. The HDC then transfers the read data to the local memory via DMA operation.

(2) After reading each sector, the HDC updates the SCNT and LSN to point to the next sector, and repeats the above described operation until SCNT is equal to zero. During the above read operations, if LSN is equal to ESN, the HDC updates LSN, and continues the read operations after relocating the head (track) specified by LHN.

(3) The HDC abnormally terminates the execution of this command if SCNT is not equal to zero when the HDC reads out the data from the last sector (LSN = ESN and LHN = ETN). The ENC (end of cylinder) bit of EST (error status) is set to one in this situation.

(4) The HDC will terminate this command if a fault signal is detected while reading data. The HDC will set the EQC (equipment check) of the EST (error status) byte when this occurs.

(5) The HDC will terminate this command abnormally if the ready signal from the drive is not active or becomes not active while a Read Data command is being performed. The NR (not ready) bit of the EST (error status) register will be set to one in this case.

(6) The HDC will end this command abnormally if it cannot find an AM (address mark) (soft-sector mode) or a SYNC byte (hard-sector mode) of the ID field before four index pulses occur. Under these conditions, the RRQ (reset request) bit of the STR (status register) will be set. In order to perform further disk commands the HDC will have to be reset because the format controller is hung up looking for an AM or SYNC byte.

(7) ECC mode: If the HDC detects an ECC error during a read operation, it will execute the following operations: First, the HDC decides whether or not the error is correctable by checking the syndrome of the error pattern. If the error is correctable, the HDC terminates the command in the normal mode after setting the DER (data error) bit of EST register to one. The host system can input the error address and the error pattern information by issuing the Detect Error command. If it is not a correctable error, the HDC will terminate the command in the abnormal mode after setting the DER bit of the EST register to one.

CRC mode: If the HDC detects a CRC error on a sector during the read operation, the HDC will terminate the command in the abnormal mode after setting the DER bit of the EST register to one.

(8) If the HDC detects an overrun condition during a Read Data operation, the OVR (overrun) bit of the EST register is set. (An overrun condition occurs when the internal data FIFO is full, another data byte has been received from the disk drive, and a DMA service does not occur.) The command is then terminated in the abnormal mode.

(9) If the HDC cannot find the desired sector within the occurrence of three index pulses, the ND (no data) bit of the EST register is set to one and the command is terminated in the abnormal mode.

(10) If TC (terminal count) occurs during a Read Data command the DMA transfers to the local memory will stop. However, the HDC does continue the read operation until the end of the sector, if SCNT = 1.

5

If SCNT is 2 or more, DMA transfers restart when SCNT is updated to the next sector, and will continue until SCNT is zero.

(1) If the Read Data command has been successfully completed, the result status will be set indicating such, and the result status bytes will be updated according to the number of sectors that have been read. The logical disk parameters—LSN, LHN, and LCN—are incremented as follows:

LSN is incremented at the end of each sector until the value of ESN is reached. LSN is then set to 0 and LHN is incremented. If LHN reaches the value of ETN, then LHN is cleared and LCN is incremented.

In other words, if a Read or Write operation is terminated normally, the various parameters will point to the next logical sector.

If the command is terminated in the abnormal mode, the result status bytes will indicate on which sector, cylinder, and head the error occurred.

(12) If the HDC cannot detect the address mark (soft-sector) or SYNC bytes (hard-sector) immediately following the VFO sync in the data field, the HDC will set the MAM (missing address mark) bit of the EST register to one, and will terminate the command in the abnormal mode.

**Check**

1100X	PHN	(FLAG)	LCNH	LCNL	LHN	LSN	SCNT
	EST	PHN	(FLAG)	LCNH	LCNL	LHN	LSN

- PHN = Physical Head Number
- FLAG = Flag Byte, Hard-Sector ID Field Only
- LCNH = Logical Cylinder Number, High Byte
- LCNL = Logical Cylinder Number, Low Byte
- LHN = Logical Head Number
- LSN = Logical Sector Number
- SCNT = Sector Number
- EST = Error Status

This command is used to confirm that the data previously written to the medium by the Write Data command contains the correct CRC or ECC.

(1) The HDC reads the data in the sector specified by FLAG (hard-sector only), LCNH, LCNL, LHN, and LSN. The Check command differs from the Read Data command in that no DMA transfers occur.

With the exception of the ECC mode, the Check command is the same as the Read Data command. Please refer to items 2, 3, 4, 5, 6, 7, 8, 11, and 12 of Read Data command for details.

(2) If in the ECC mode, the HDC detects only ECC errors and does not execute any error correction operation even if the ECC errors are correctable. No data transfers have been made, and there is no data to correct.

**Scan**

1101X	PHN	(FLAG)	LCNH	LCNL	LHN	LSN	SCNT
	EST	PHN	(FLAG)	LCNH	LCNL	LHN	LSN

- PHN = Physical Head Number
- FLAG = Flag Byte, Hard-Sector ID Field Only
- LCNH = Logical Cylinder Number, High Byte
- LCNL = Logical Cylinder Number, Low Byte
- LHN = Logical Head Number
- LSN = Logical Sector Number
- SCNT = Sector Number
- EST = Error Status

(1) In executing the Scan command, the HDC reads the data from the sector specified by the preset parameters of the command phase. The HDC then compares this data with the data transmitted from the local memory (The purpose of this command is to locate a sector that contains the same data as the local memory.)

This command will terminate successfully if the data from the disk and the data from the local memory are the same. If they are not, the HDC updates SCNT and LSN, and executes the abovementioned operation again.

If the HDC cannot locate a sector that satisfies the scan conditions, the NCI bit of the STR will be set. The HDC tries to compare data until the end of the cylinder has been reached, or until SCNT is zero.

(2) If the value of the LSN (logical sector number) is equal to that of ESN (ending sector number) after updating LSN, the HDC updates the contents of LHN (increasing by 1) and that of LSN (LSN=0), and repeats the operation described in item 1 after selecting the next head.

(3) After comparing the data transferred from the host CPU with the data in the specified sectors, the result bytes (FLAG, which is only for hard-sector disks, LCNH, LCNL, LHN, and LSN) will be set equal to the sector location that satisfies the Scan command.

(4) The descriptions in 4, 5, 6, 8, and 9 of Read Data command, and items 3 and 4 of Verify Data command are identical for this command. Refer to these descriptions for additional details.

**Verify Data**

1100X	PHN	(FLAG)	LCNH	LCNL	LHN	LSN	SCNT
	EST	PHN	(FLAG)	LCNH	LCNL	LHN	LSN

- PHN = Physical Head Number
- FLAG = Flag Byte, Hard-Sector ID Field Only
- LCNH = Logical Cylinder Number, High Byte
- LCNL = Logical Cylinder Number, Low Byte
- LHN = Logical Head Number
- LSN = Logical Sector Number
- SCNT = Sector Number
- EST = Error Status

This command is used to verify data on the disk.

(1) The HDC reads the data from the specified sector, and compares the data transmitted from the local memory via DMA with the data from the disk.

The sector is specified by FLAG (hard-sector only), LCNH, LCNL, LHN, and LSN, and the drive is selected by UA. If the data transmitted from the local memory is the same as that read from the sector, the HDC updates the contents of LSN and SCNT, and continues the abovementioned operation. After updating SCNT, if the value of SCNT is equal to zero, the HDC ends the execution of the command in the normal mode. If the value of LSN is equal to that of ESN after updating LSN, the HDC updates the contents of LHN and LSN, and the HDC continues the verify data operation after selecting the head (track) specified by LHN.

If the data transmitted from the local memory is not the same as that read from the sector, the HDC ends the execution of the command in the abnormal mode after setting the NCI (not coincident) bit of STR to one.

(2) If, after verifying the data on the last sector, the contents of SCNT are not equal to zero, the HDC terminates execution of the command abnormally after setting the ENC (end of cylinder) bit of the EST register to one.

(3) After verifying the data read from a sector, the HDC checks the CRC bytes (CRC mode) or the ECC bytes (ECC mode).

If the HDC detects a CRC or an ECC error on a sector, the HDC terminates execution of the command abnormally after setting the DER bit of the EST register to a one.

(4) After detecting an active  $\overline{TC}$  signal ( $\overline{TC} = 0$ ), the HDC executes the above operation by comparing the read data from the disk drive with the data 00 instead of the data from the main system until the end of the sector.

In the case of SCNT greater than one, when SCNT is updated, DMA transfers restart and disk data is compared against host data until SCNT is zero.

(5) After verification of the data on all the sectors, FLAG (hard-sector only), LCNH, LCNL, LHN, and LSN are set to the values of FLAG, LCNH, LCNL, LHN, and LSN of the last verified sector.

(6) The descriptions in items 4, 5, 6, 8, 9, and 12 of the Read Data command are valid in this command. Please refer to these items for additional detail.

## Write Data

TTTTX	PHN (FLAG)	LCNH	LCNL	LHN	LSN	SCNT	
	EST	PHN	(FLAG)	LCNH	LCNL	LHN	LSN

PHN = Physical Head Number  
 FLAG = Flag Byte, Hard-Sector ID Field Only  
 LCNH = Logical Cylinder Number, High Byte  
 LCNL = Logical Cylinder Number, Low Byte  
 LHN = Logical Head Number  
 LSN = Logical Sector Number  
 SCNT = Sector Number  
 EST = Error Status

(1) This command is used to write data into the data field of the sectors specified by FLAG (hard disks only), LCNH, LCNL, LHN, and LSN, and to write CRC bytes or ECC bytes according to each internally specified mode (CRC or ECC). The data is written to the disk via DMA transfer from the local memory.

(2) After writing data on a sector, the HDC updates the contents of SCNT and LSN, and repeats the above described Write Data operation until SCNT is equal to zero.

During the above Write Data operations, if LSN is equal to ESN, the HDC updates LHN and LSN, and continues the Write Data operations after selecting the new head (track) specified by LHN.

As described above, the HDC has the capability of multi-sector and multi-track write operations.

(3) The HDC abnormally terminates the execution of this command if the SCNT is not equal to zero when the HDC writes the data to the last sector (LSN = ESN and LHN = ETN). The ENC (end of cylinder) bit of EST (error status) register is set to one in this situation.

(4) If the write protected signal is active (high) at the beginning of the execution of this command, the HDC ends the execution of this command in the abnormal mode after setting the NWR (not writable) bit of the EST register to one.

(5) After detecting an active  $\overline{TC}$  signal ( $\overline{TC} = 0$ ), the HDC writes the data 00 to the sector, instead of the data from the host system.

In the case of SCNT of two or more, when SCNT is updated, the DMA transfers will restart and writing of host data will continue until SCNT = 0.

(6) In the ST506-type mode, the HDC will set the reduced write current output bit to a one when the cylinder number becomes greater than that specified by RWCH and RWCL. These parameters are loaded during execution of the Specify command.

The descriptions in items 4, 5, 6, 8, 9, and 11 of the Read Data command are applicable here also. Refer to these items for further detail.

**Sense Interrupt Status**

0001X	
	IST

IST = Interrupt Status

(1) The HDC transfers the new disk status to the host CPU at the end of a Seek or Recalibrate operation or the new disk status resulting from a change of state of the ready signal, which may occur at any time.

(2) If the Seek or Recalibrate command in progress is completed when this command is issued or if there has been no change of state of the ready signal from the drive, this command will be terminated abnormally.

**Specify**

0010X	MODE	DTLH	DTLL	ETN	ESN	GPL2	(MGPL1) [RWCH]	[RWCL]

- MODE = Mode Byte; Selects Operation Mode
- DTLH = Data Length, High Byte
- DTLL = Data Length, Low Byte
- ETN = Ending Track Number
- ESN = Ending Sector Number
- GPL2 = Gap Length 2
- MGPL1 = Gap Length 1 (used in SMD mode only; Controls Read Gate Timing)
- RWCH = Reduced Write Current (Cylinder No.), High Byte
- RWCL = Reduced Write Current (Cylinder No.), Low Byte

The Specify command is used to set the operational mode of the HDC by presetting various parameters. Parameters such as MODE (figure 5, table 7), DTLH (figure 6), DTLL, ETN, ESN, GPL2, MGPL1/RWCH, and RWCL may be programmed into the HDC. This allows for a high degree of versatility. Data record length is programmable from 128 to 4095 bytes in soft-sector mode and 256 to 4095 bytes in hard-sector mode.

**Figure 5. Mode Byte**

0	ECC	CRCS	SSEC	DSL/ STP3	DSE/ STP2	SOM/ STP1	SOP/ STP0
---	-----	------	------	--------------	--------------	--------------	--------------

**Figure 6. DTLH Byte**

1	CRC	PAD	POL	DTL11	DTL10	DTL9	DTL8
---	-----	-----	-----	-------	-------	------	------

CRC = Initial Value of Polynomial Counter, Either All Zeros or All Ones  
 PAD = Selects ID/Data pad of 00H if 0  
       = Selects ID/Data pad of 4EH if 1  
 POL = Polling Mode if 0  
       = Nonpolling Mode if 1

**Table 7. Mode Byte Bits**

Bit Name	Specified Mode
ECC	1 ECC is appended in data field: $(x^{21}+1)(x^{11}+x^2+1)$
	0 CRC is appended in data field
CRCS	1 Generator polynomial: $(x^{16}+1)$
	0 Generator polynomial: $(x^{16}+x^{12}+x^5+1)$
SSEC	1 Soft-sector disk (floppy-like interface), MFM data
	0 Hard-sector disk (SMD interface), NRZ data
<b>SSEC = 0</b> <span style="float: right;"><b>SSEC = 1</b></span>	
DSL	Data strobe late STP3 (Note 1)
DSE	Data strobe early STP2 (Note 1)
SOM	Servo offset minus STP1 (Note 1)
SOP	Servo offset plus STP0 (Note 1)

Note:

- (1) Stepping rate for ST506 mode =  $(16-STP) \times 2110 \times t_{CY}$   
 Assuming a 10 MHz processor clock:  $F_H = 2.11 \text{ ms} \dots O_H = 33.76 \text{ ms}$

**Sense Unit Status**

**Soft-Sector Mode**

0011X	
	UST

**SMD Mode**

0011X	1	2	5
	UST	DS	DT

The Sense Unit Status (SUS) command is used to transfer the Unit Status (UST) to the host. In the case of SMD mode the SUS command may also be used to transfer the Detail Status (DS) and Device Type (DT) by using the appropriate preset parameter value as shown above. No preset parameters are used in the soft-sector mode, although one is required in the SMD mode. Values other than 1, 2, or 5 do not produce valid results.

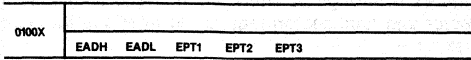
After result bytes are placed in FIFO, HDC generates a FAULT CLEAR when in SMD mode.

The DS and DT bytes are defined by the type of drives used. The UST is shown in table 8.

**Table 8. Unit Status Byte**

Bit	Interface Type		
	No.	SMD	ST506
D <sub>7</sub>	Unit selected		0
D <sub>6</sub>	Seek end		0
D <sub>5</sub>	Write protected		0
D <sub>4</sub>	0	Drive selected	
D <sub>3</sub>	Unit ready	Seek complete	
D <sub>2</sub>	On cylinder	Track 000	
D <sub>1</sub>	Seek error	Ready	
D <sub>0</sub>	Fault	Write fault	

## Detect Error



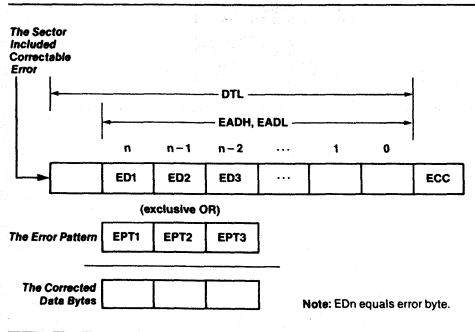
ADH = Error Address, High Byte  
 ADL = Error Address, Low Byte  
 PT1 = Error Pattern, Byte 1  
 PT2 = Error Pattern, Byte 2  
 PT3 = Error Pattern, Byte 3

This command is used to transfer the error pattern and error address to the host CPU, when correctable errors have occurred during the execution of a Read Data command with the ECC mode enabled.

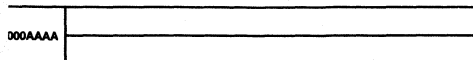
The error address (EADH and EADL) is calculated from the last data byte of the sector that contained a correctable error which was indicated by the status bit of the previous Read Data command with the ECC mode enabled. The error pattern is used for correcting the error data at the location where the error occurred. After receiving the error address and the error pattern, the host CPU can correct the error data by performing an exclusive-OR of the error pattern and the error data. See figure 7.

The result bytes are available to the host CPU within 10μs.

## Figure 7. Error Correction



## Auxiliary Command



There are no preset parameters or result bytes associated with this command. The definitions of the 4 LSBs (AAA) are given in figure 8 and table 9. The auxiliary command is accepted at any time and is immediately executed. The auxiliary command may be used to recover from certain types of error conditions, or to mask and clear interrupts.

Figure 8. Auxiliary Command

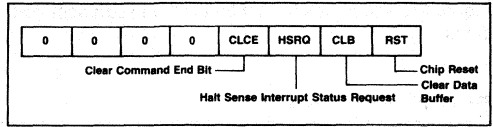


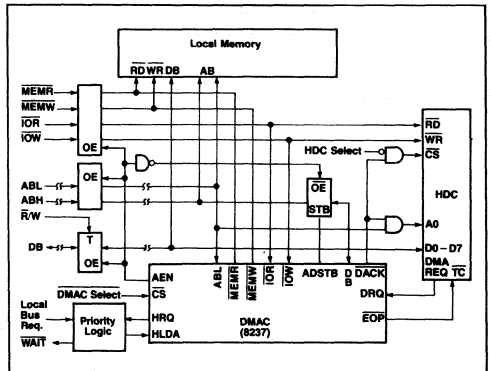
Table 9. Auxiliary Command Bits

Bit Name	Operation
CLCE	Clears the CE bits of the status register, inactivating the interrupt request output caused by Command End condition. This is used when no disk commands are going to be issued and it is desired to clear the interrupt.
HSRQ	Deactivates the interrupt request output caused by Sense Interrupt Status Request condition until a Command End occurs. However, this command has no effect on the SRQ bit of the status register.
CLB	Clears the data buffer.
RST	This has the same effect as a reset signal on the Reset input. This function is used whenever the RRQ bit in the status register is set (indicating the format controller is hung up), or when a software reset is needed.

## System Example

Figure 9 shows an example of a local bus system.

Figure 9. Local Bus System



**Track Format**

Figure 10 shows track format for hard- and soft-sectored disks.

**System Example Timing Diagrams**

Figures 11 through 22 show the interface timing (soft sector and hard-sector) required to interface the hard disk drive.

**Figure 10. Track Format**

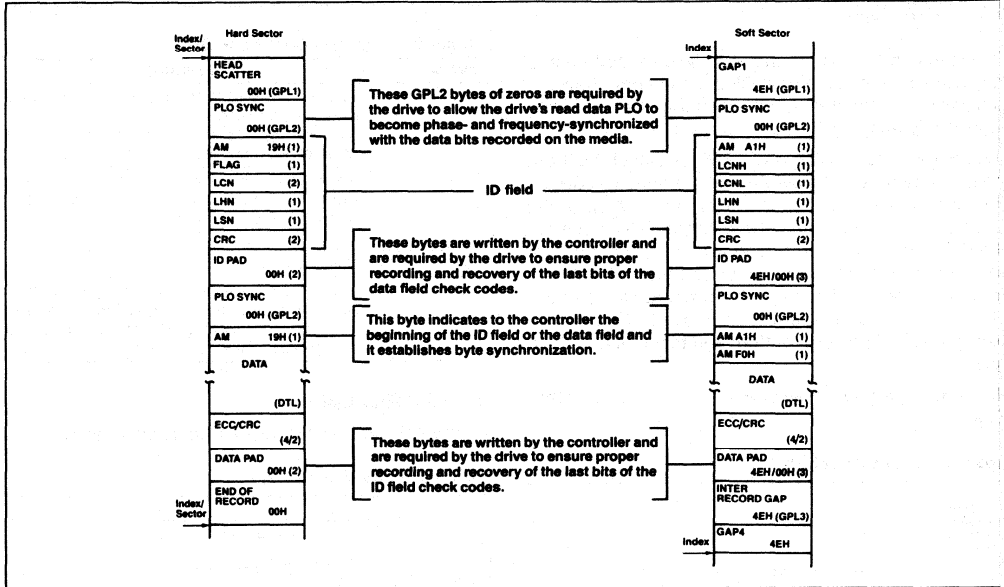


Figure 11. "Unit Selection" and "State Sense" Timing (Hard Sector)

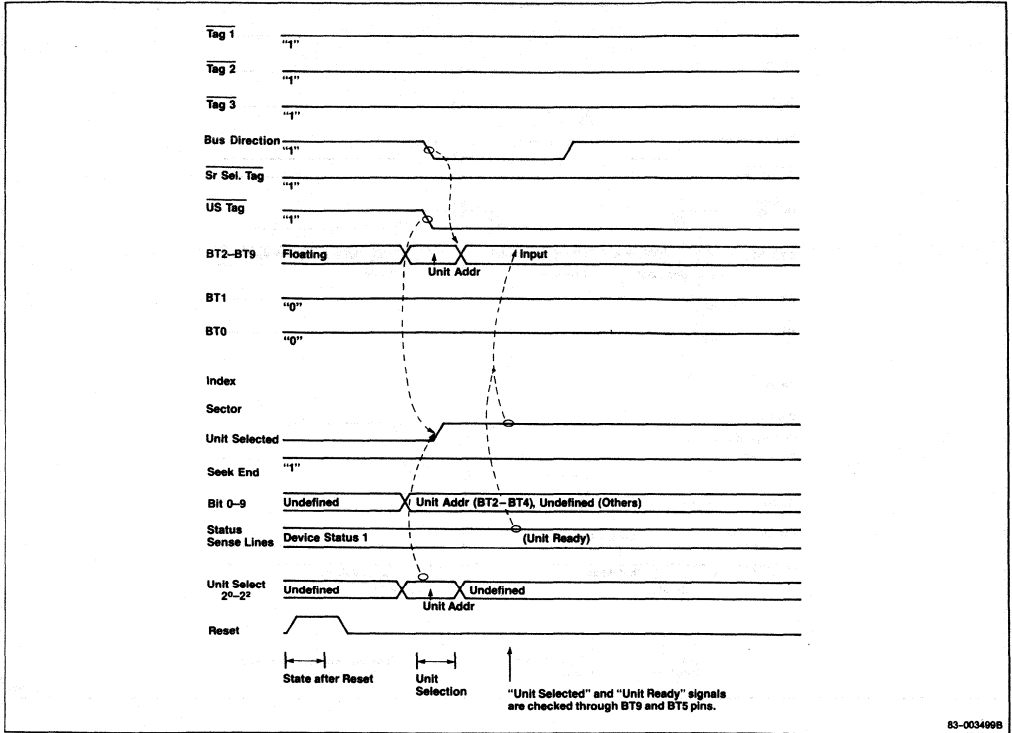




Figure 12. Return to Zero Timing (Hard Sector)

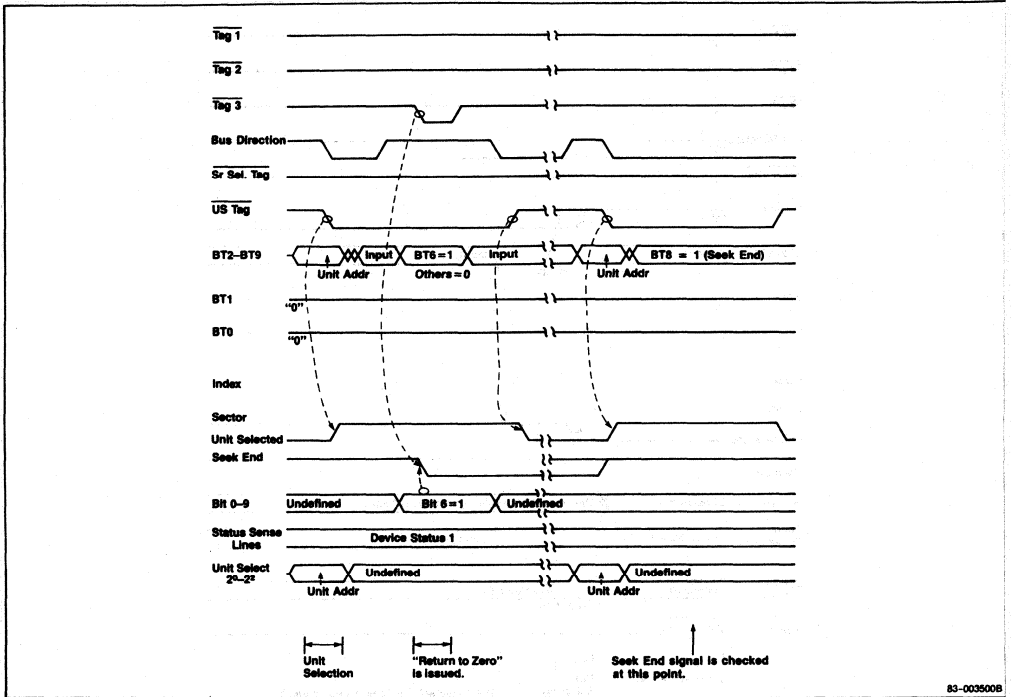


Figure 13. "Seek" Timing (Hard Sector)

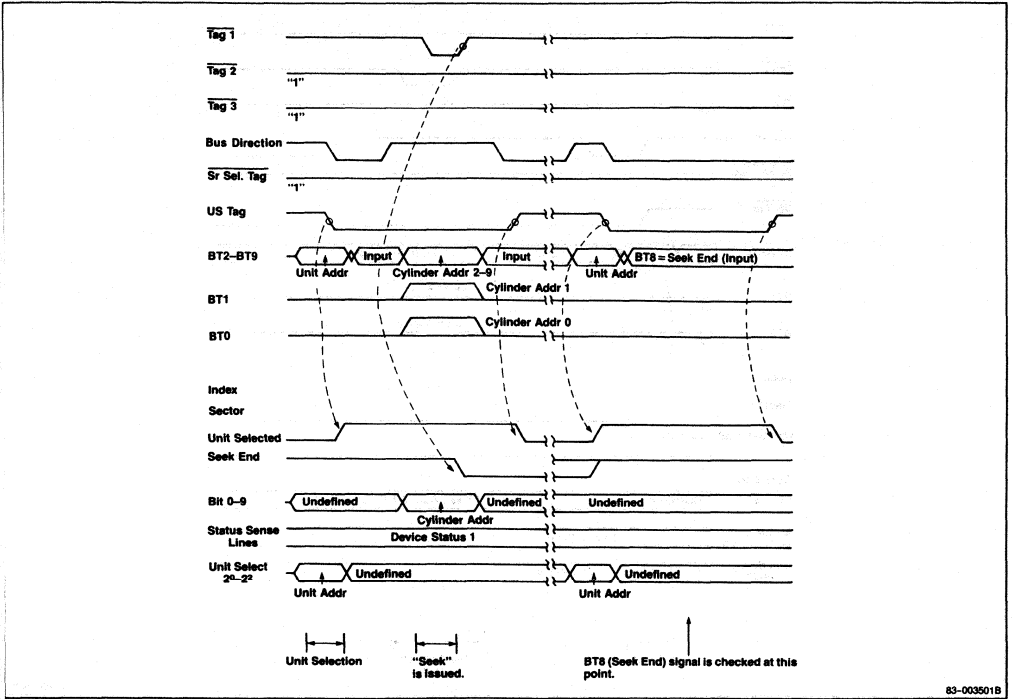


Figure 14. "Head Select" Timing (Hard Sector)

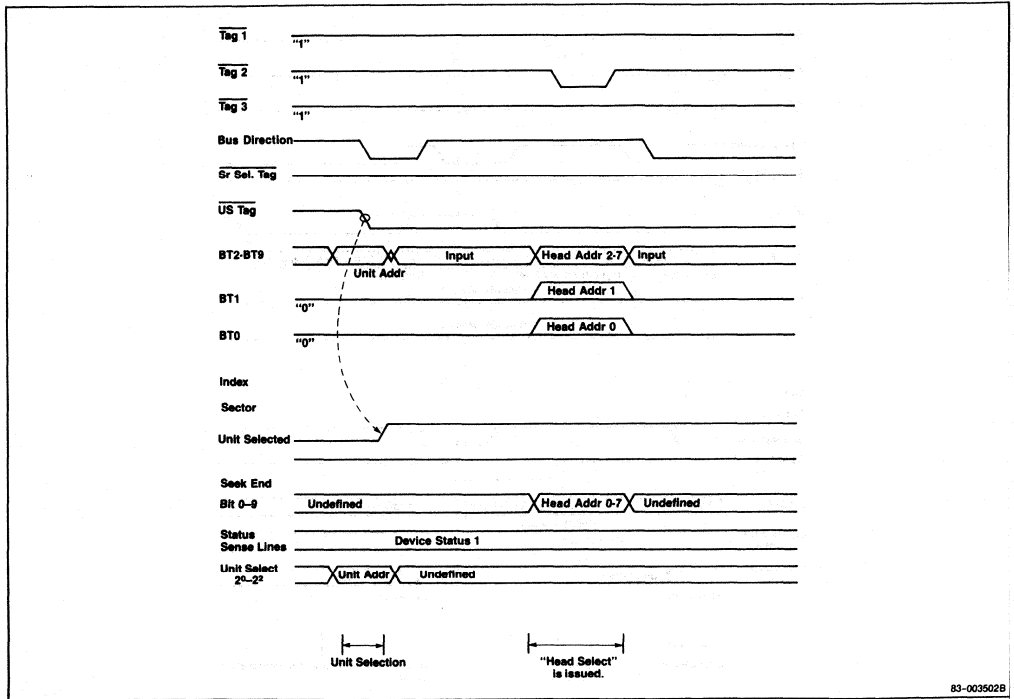


Figure 15. "Unit Status Sense" Timing (Hard Sector)

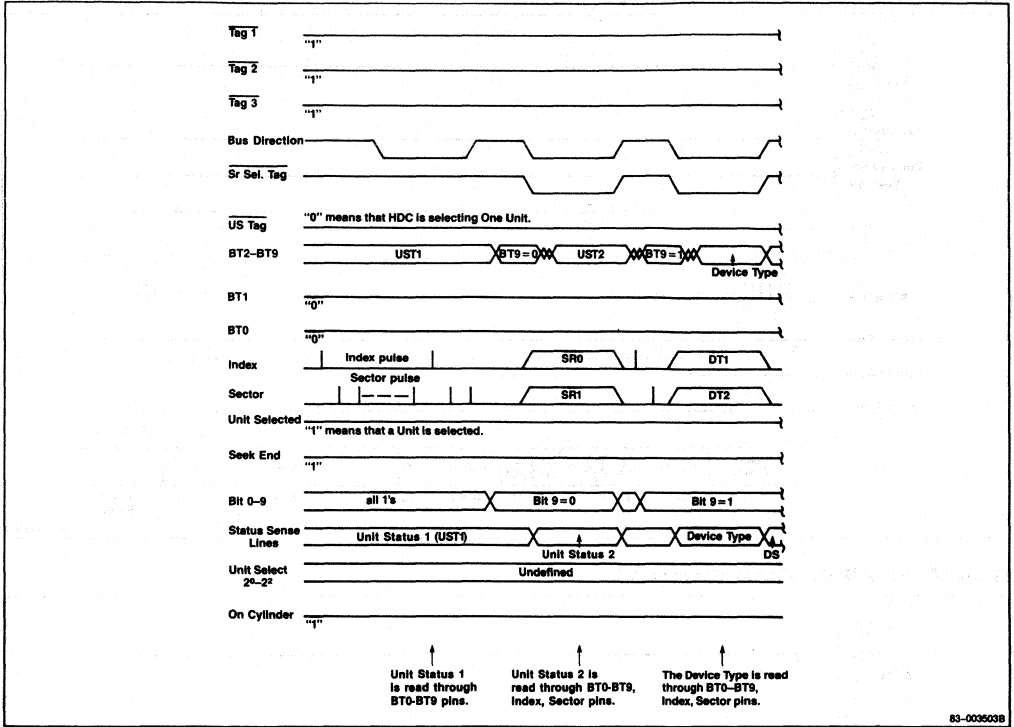


Figure 16. "Data Read" Timing (Hard Sector)

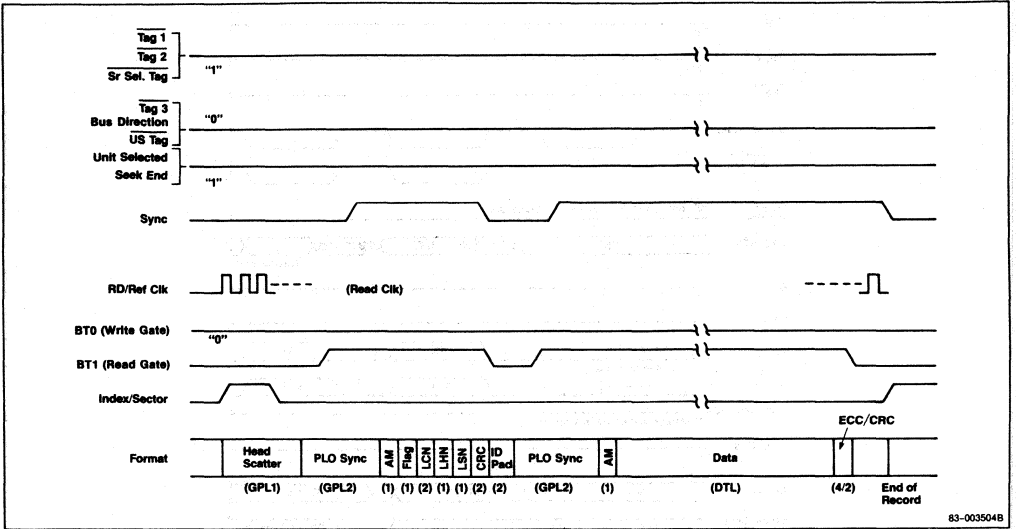
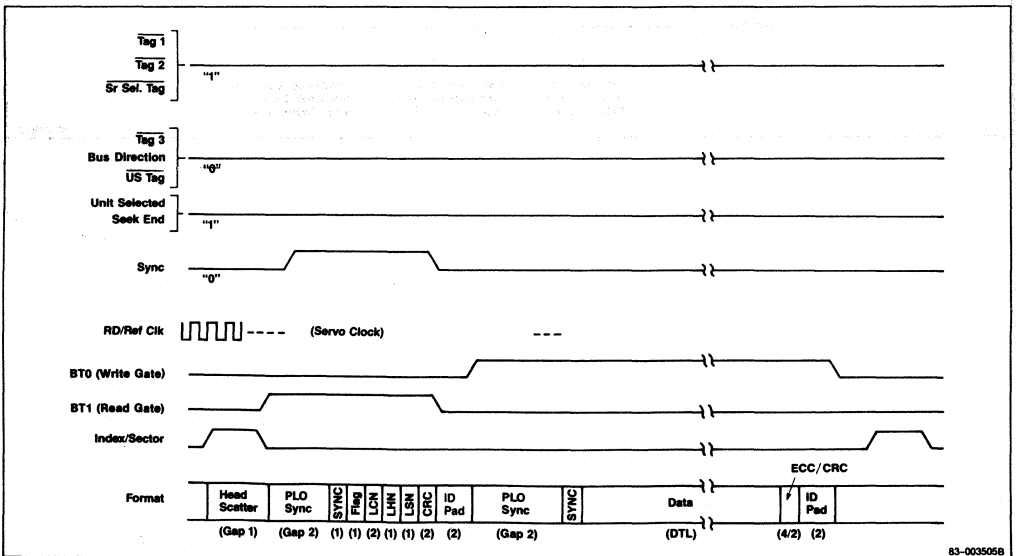
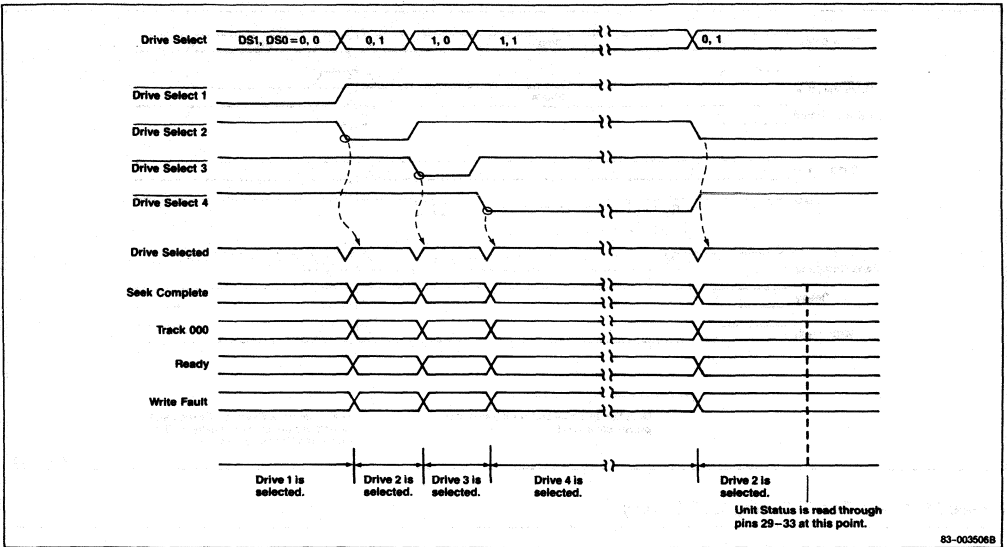


Figure 17. "Data Write" Timing (Hard Sector)



**Figure 18. "Drive Select" and "Unit Status Sense" Timing (Soft Sector)**



**Figure 19. "Normal Seek" Timing (Soft Sector)**

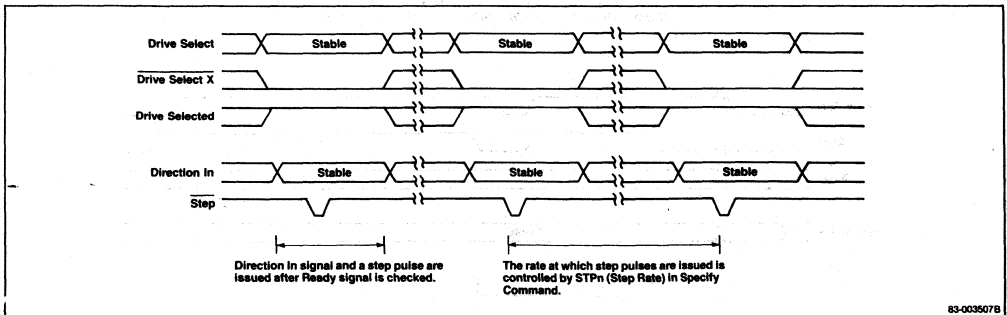


Figure 20. "Buffered Seek" Timing (Soft Sector)

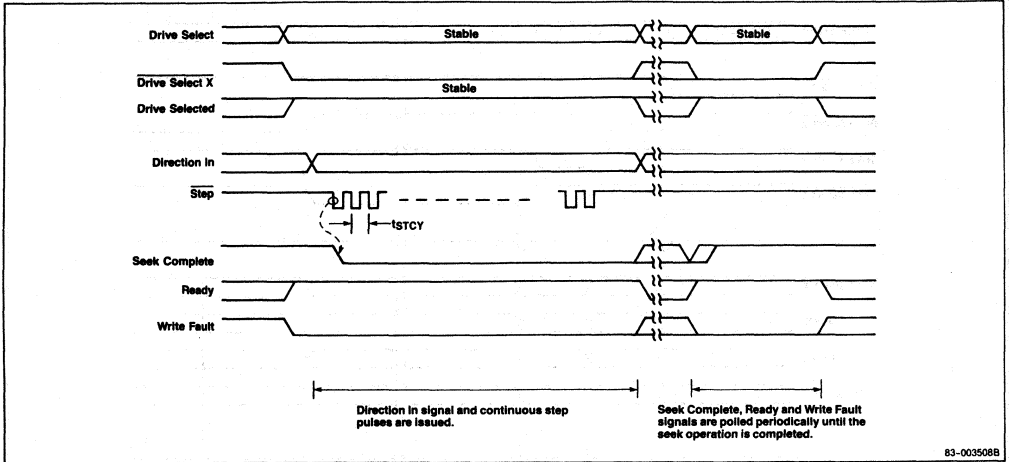


Figure 21. "Data Read" Timing (Soft Sector)

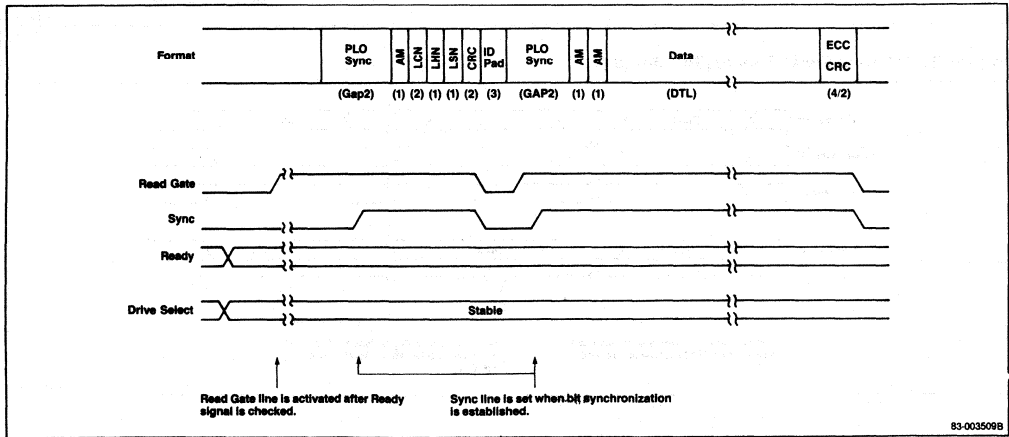
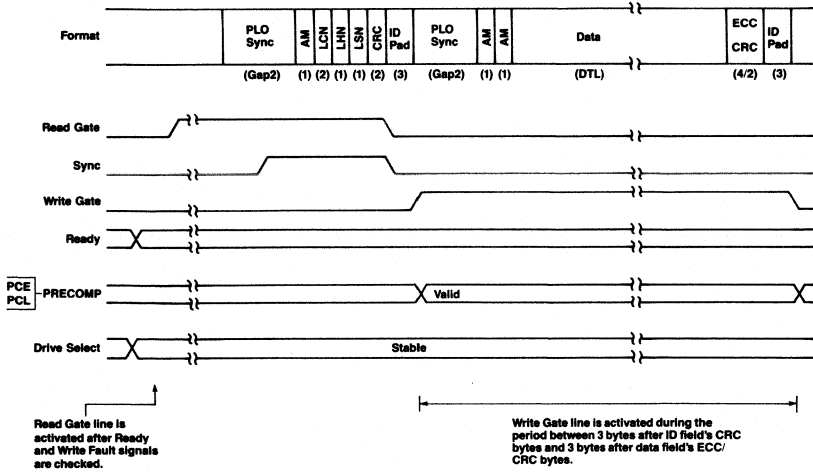


Figure 22. "Data Write" Timing (Soft Sector)



83-003510B





## Description

The μPD9306A hard-disk interface (HDI) chips are unique CMOS single-chip support devices intended for use with the μPD7261A hard-disk controller. The μPD9306A includes a high-performance, digital phase-locked loop (DPLL), write precompensation logic, and μPD7261A CLK and R/W CLK generation. The μPD9306A requires only two inexpensive passive delay lines and a crystal for the self-contained oscillator. It provides a simple but effective solution to the design of support circuitry for typical hard-disk controllers utilizing the ST506 type interface. Due to its fast acquisition time, the μPD9306A can actually provide increased storage by allowing for a size reduction in the sync field areas. The HDI also significantly reduces board area requirements and overall design time. The schematic examples included in this data sheet can be used to reduce the ST506 interface design time to a few hours.

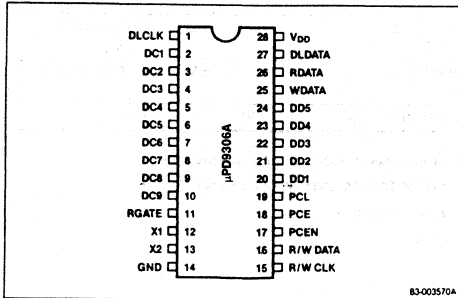
## Features

- Unique digital phase-locked loop (no adjustments)
- Precompensation logic
- 5-MHz MFM data rate
- Internal crystal oscillator
- CMOS technology
- Single +5V power supply

## Ordering Information

Part Number	Package Type
μPD9306AC	28-pin plastic DIP

## Pin Configuration



## Pin Identification

No.	Symbol	Function
1	DLCLK	Delay line clock output
2-10	DC1-DC9	Delay clock inputs
11	RGATE	Read gate input
12	X1	Crystal clock input
13	X2	Crystal clock output
14	GND	Ground
15	R / W CLK	Read / write clock output
16	R / W DATA	Read / write data input / output
17	PCEN	Precompensation enable input
18	PCE	Precompensation early input
19	PCL	Precompensation late input
20-24	DD1-DD5	Delayed data inputs
25	WDATA	Write data output
26	RDATA	Read data input
27	DLDATA	Delay line data output
28	V <sub>DD</sub>	+5 V power supply

## Pin Functions

### DC1-DC9 (Delayed Clock)

These nine inputs receive clock signals delayed relative to DLCLK. The delays for pins DC1-DC9 are 10 ns to 90 ns in 10 ns increments.

### DD1-DD5 (Delayed Data)

These five inputs receive the input data signals, delayed relative to DLDATA. The delays for pins DD1-DD5 are 40, 60, 80, 90, and 100 ns respectively. As an option, DD1 and DD2 may be connected to the 30-ns and 70-ns taps, respectively, of delay line 1. Comparative performance data is shown in table 1.

### DLDATA (Delay Line Data)

This output supplies the external delay line with processed read data from the disk or processed write data from the host.

### DLCLK (Delay Line Clock)

This pin is used for the output clock of the on-chip oscillator and to supply clocks for both the delay line and the μPD7261A.

**RGATE (Read Gate)**

When this input is active, the digital phase-locked loop (DPLL) circuit generates a read/write clock that is synchronized to the phase of the read data from the disk.

**R/W CLK (Read/Write Clock)**

When RGATE is active, the DPLL selects one clock input from DLCLK or DC1-DC9. The clock input is synchronized with the read data at the R/W DATA pin and output via R/W CLK. When RGATE is inactive, the DPLL outputs the previously selected clock.

**R/W DATA (Read/Write Data)**

This pin outputs read data that has been synchronized with R/W CLK when RGATE is high. This pin inputs write data when RGATE is low.

**RDATA (Read Data)**

Input for read data from the hard-disk drive.

**WDATA (Write Data)**

Output for write data to the hard-disk drive. Precompensation is according to PCE, PCL, and PCEN states.

**PCEN (Precompensation Enable)**

Write precompensation is performed when this input signal is active.

**PCE (Precompensation Early)**

When PCE and PCEN are active, write data is advanced in phase from its nominal position and output on the WDATA pin. External delay lines determine the amount of time advance.

**PCL (Precompensation Late)**

When PCL and PCEN are active, write data is delayed in phase from its nominal position and output on the WDATA pin. External delay lines determine the amount of time delay.

**X1, X2 (Crystal)**

These two pins connect the crystal to the on-chip oscillator and clock generator.

**V<sub>DD</sub> (Power Supply)**

+5 V power supply input.

**GND (Ground)**

Ground.

**Absolute Maximum Ratings**

T<sub>A</sub> = 25°C

Power supply voltage, V <sub>DD</sub>	-0.5 to +7.0 V (Note 1)
Input voltage, V <sub>I</sub>	-0.5 to +7.0 V (Note 1)
Output current, I <sub>O</sub>	10 mA
Operating temperature, T <sub>OPR</sub>	0°C to +70°C
Storage temperature, T <sub>STG</sub>	-65°C to +150°C

Note:

(1) With respect to ground.

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**DC Characteristics**

T<sub>A</sub> = 0°C to +70°C, V<sub>DD</sub> = +5.0 V ± 10% unless otherwise specified

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage high	V <sub>IH1</sub>	2.0		V <sub>DD</sub> +0.5	V	
	V <sub>IH2</sub>	0.7V <sub>DD</sub>		V <sub>DD</sub> +0.5	V	
Input voltage low	V <sub>IL1</sub>	-0.5		+0.8	V	(Note 3)
	V <sub>IL2</sub>	0		0.3V <sub>DD</sub>	V	(Note 4)
Output voltage high	V <sub>OH1</sub>	V <sub>DD</sub> -0.4			V	I <sub>OH</sub> = -1.0 mA (Note 1)
	V <sub>OH2</sub>	V <sub>DD</sub> -0.4			V	I <sub>OH</sub> = -2.0 mA (Note 2)
Output voltage low	V <sub>OL1</sub>			+0.4	V	I <sub>OL</sub> = 3.2 mA (Note 1)
	V <sub>OL2</sub>			+0.4	V	I <sub>OL</sub> = 6.4 mA (Note 2)
Input leakage current	I <sub>LI</sub>			±10	μA	0 V < V <sub>I</sub> < V <sub>DC</sub>
Output leakage current	I <sub>OL</sub>			±10	μA	0 V < V <sub>O</sub> < V <sub>DD</sub>
Supply current	I <sub>DD</sub>		10	30	mA	

Note:

(1) All pins except DLCLK, DLDATA and R/W CLK.

(2) DLCLK, DLDATA, and R/W CLK pins only.

(3) all inputs except X1.

(4) X1 input.

## Capacitance

T<sub>A</sub> = 25°C, f<sub>C</sub> = 1MHz

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	C <sub>I</sub>		10		pF	(Note 1)
Output capacitance	C <sub>O</sub>		15		pF	(Note 1)
I/O capacitance	C <sub>I/O</sub>		15		pF	(Note 1)

### Note:

(1) All unmeasured pins returned to ground.

## AC Characteristics

T<sub>A</sub> = 0°C to +70°C, V<sub>DD</sub> = +5.0V ± 10% (Note 1)

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
DLCLK, DLDATA rise time	t <sub>DLR</sub>		20		ns	(Note 2)
DLCLK, DLDATA fall time	t <sub>DLF</sub>		20		ns	(Note 2)
DLCLK cycle time	t <sub>CYDLK</sub>		100		ns	
DLCLK high level width	t <sub>WDLKH</sub>	40	50	60	ns	
DLCLK low level width	t <sub>WDLKL</sub>	40	50	60	ns	
DLDATA high level width	t <sub>WDLH</sub>	55	70	100	ns	
DC1-DC9, DD1-DD5 rise time	t <sub>DR</sub>		30		ns	
DC1-DC9, DD1-DD5 fall time	t <sub>DF</sub>		30		ns	
DC1-DC9 cycle time	t <sub>CYDC</sub>		100		ns	
DC1-DC9 high level width	t <sub>WDCH</sub>	40	50	60	ns	
DC1-DC9 low level width	t <sub>WDCL</sub>	40	50	60	ns	
DD1-DD5 high level width	t <sub>WDH</sub>	55	70	100	ns	
R/W CLK rise time	t <sub>RWR</sub>		10		ns	
R/W CLK fall time	t <sub>RWF</sub>		10		ns	
R/W CLK cycle time	t <sub>CYRW</sub>	83	100		ns	
R/W CLK high level width	t <sub>WRWH</sub>	30			ns	
R/W CLK low level width	t <sub>WRWL</sub>	30			ns	

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Crystal frequency	f <sub>XTAL</sub>		10	10.5	MHz	
DLCLK ↑ to DC9 ↓ delay time	t <sub>DDC1</sub>	85	90	95	ns	(Note 3)
DCn ↑ to DCn+1 ↓ delay time (n = 1, 2, ..., 8)	t <sub>DDC2</sub>	8	10	12	ns	(Note 3)
DLDATA ↑ to DD5 ↑ delay time	t <sub>DDD1</sub>	95	100	105	ns	(Note 3)
DDn ↑ to DDn+1 ↑ delay time (n = 2, 3, 4)	t <sub>DDD2</sub>	8	10	12	ns	(Note 3)
R/W CLK ↑ to R/W DATA delay time	t <sub>DRW</sub>	10	20	45	ns	RGATE = 1

### Note:

(1) C<sub>LOAD</sub> = 30 pF

(2) When delay line is driven

(3) Delay line specs used:

Delay time step = 10 ± 2 ns; total delay time = 100 ± 5 ns

**Table 1. μPD9306A Performance**

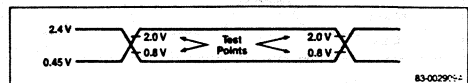
Connection (Note 1)	Bit Jitter Margin	Speed Variation Tolerance
DD1 to 40-ns delay line tap and DD2 to 60-ns delay line tap	± 30 ns	± 2% (Note 2)
DD1 to 30-ns delay line tap and DD2 to 70-ns delay line tap	± 35 ns	± 1.5% (Note 2)

### Note:

(1) Performance depends on precision of externally connected delay line.

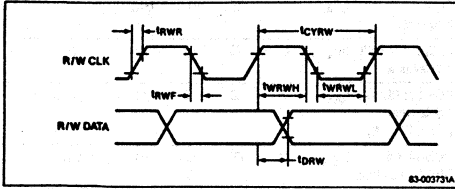
(2) Modern Winchester drives seldom exceed 0.5% speed variation.

## AC Test Points

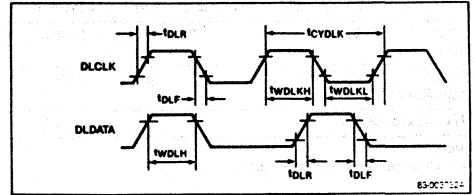


### Timing Waveforms

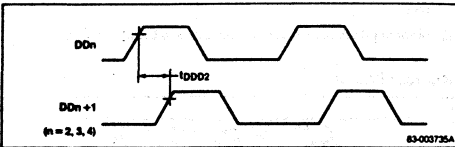
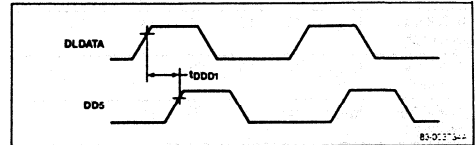
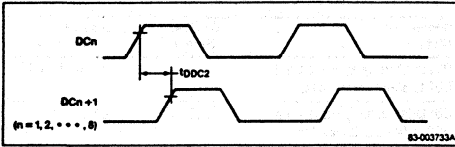
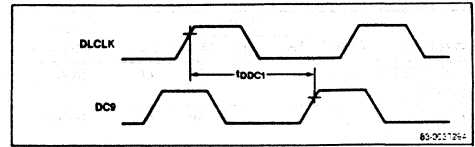
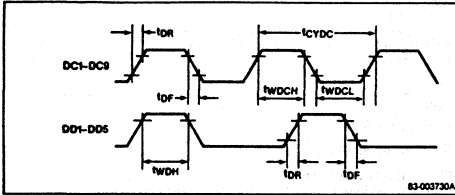
#### Controller Interface Timing (Read Operation Only)



#### Delay Line Inputs



#### Delay Line Timing Requirements

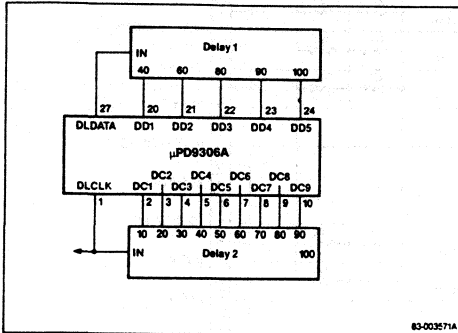


## Functional Description

### System Configuration

The schematic diagram in figure 1 illustrates the use of the μPD9306 A in conjunction with active delay lines.

**Figure 1. System Configuration with Delay Lines**



### Precompensation Circuit

Write precompensation is a technique that reduces the bit jitter present in read data, thereby increasing reliability. It is typically used only on the inner cylinders. When data is written to the disk, pulse crowding takes place on the higher-numbered inner cylinders where the same amount of data is compressed into less space than on the outer cylinders. A high percentage of the bit jitter present in the read data is due to magnetic effects causing flux transitions to occur displaced from their nominal position. These effects are predictable based on the pattern of data being recorded. Precompensation reduces bit jitter by writing the data slightly before or slightly after the nominal pulse transition time in a direction opposite to the expected jitter.

Various manufacturers of many ST-506 style Winchester disk drives use delay values of 10–12 ns. The μPD7261A generates the two precompensation control signals, precompensation early (PCE) and precompensation late (PCL), to direct the write data through one of three delay pathways. There is circuitry within the μPD9306A allowing it to operate with PCE/L and R/W DATA skewed from each other by as much as 50 ns. This eliminates the need for synchronizing the precompensation and write data signals externally.

The μPD9306 A utilizes the data path delay line for both the precompensation and the phase-comparator circuit. When RGATE is low, data appearing on the R/W DATA line is written to the drive. The write data is passed through delay line 1, and the 80-, 90- and 100-ns taps are used for the early, nominal and late signals.

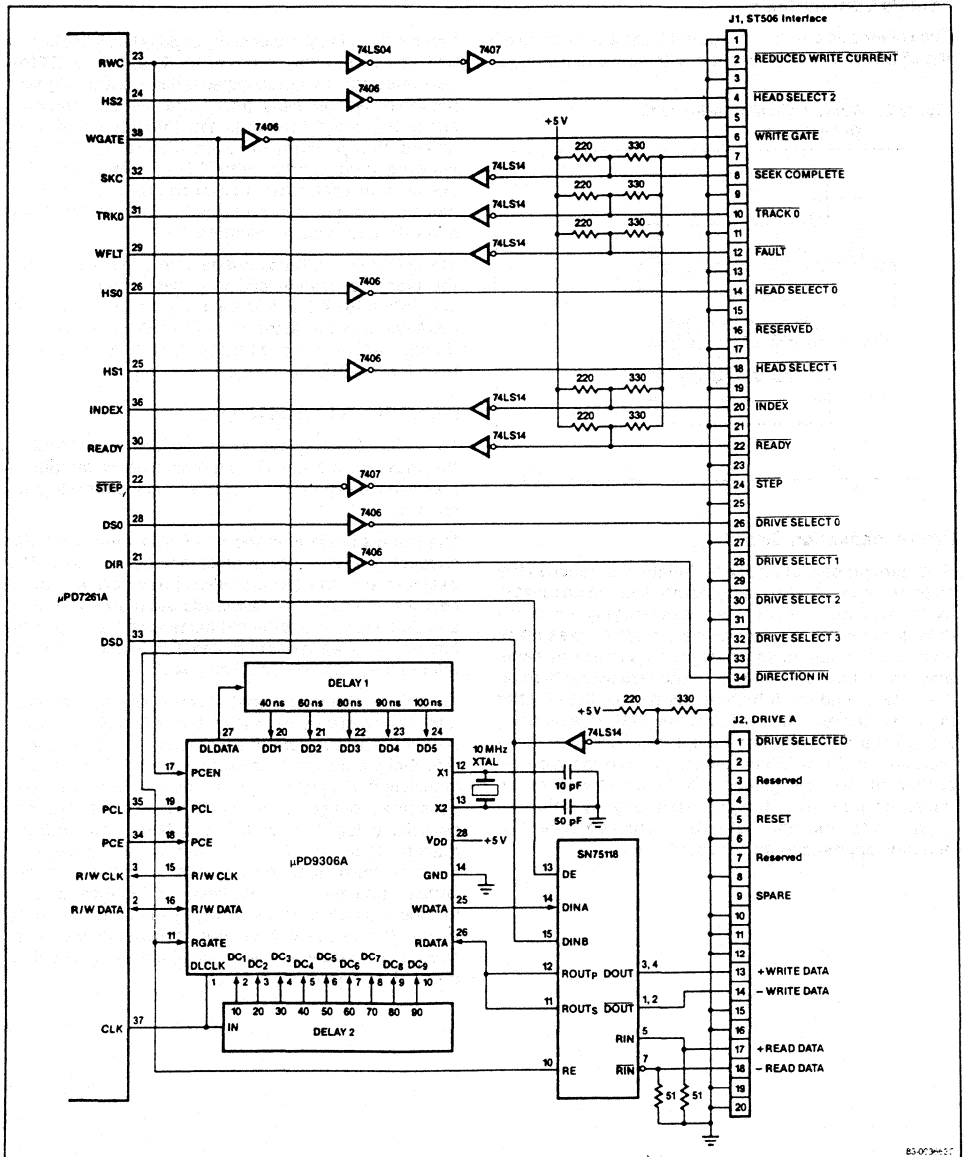
### Digital Phase-Locked Loop

The μPD9306 A employs unique circuitry to accomplish the phase-locked loop (PLL) function, which simplifies the overall design and provides very low error rate data recovery.

The raw read data from the ST-506 Winchester is MFM encoded and consists of clock and data pulses. The data format on the disk is broken into sectors with each sector containing sync fields, address marks, ID fields, and data fields. The different fields each have specific functions. For an in-depth explanation of each of these, refer to the μPD7261A user's manual.

Typically, a Winchester controller will have a data separator that recovers data from the MFM data stream. Within the data separator are several functional blocks that include a sync field detector, phase-locked loop (consisting of a phase comparator, error amplifier, low-pass filter, voltage-controlled oscillator, and pulse synchronizing logic), reference oscillator, and address mark detector. The μPD9306 A eliminates the need for many of these functional blocks. It acquires "lock" within 4 bit times in a sync field, yet it is incapable of locking to a harmonic, as analog PLL circuits are prone to do. The μPD9306 A is also immune to the high-frequency bursts that may occur during the write splice areas of the disk.

Figure 2. ST 506 Interface Proposal



The  $\mu$ PD9306 A simulates the function of an analog VCO by using a digital phase-shift network. One of the external delay lines is used to generate ten phase-shifted reference clocks. These clocks have a frequency of 10 MHz and are phase shifted in equal degree increments. The total delay line time is 100 ns, which is the same as the period of the clock, providing a complete 360° phase shift. The  $\mu$ PD9306 A synthesizes the VCO signal by internally selecting one of the phase-shifted clock signals. The rate at which the clock is phase-shifted in one direction or the other corresponds to an increase or decrease in the resulting frequency.

The internal phase comparator uses the data delay line (Delay 1) to divide the data window into ten slices. Depending on where the sampling edge of the recovery clock falls within the data window, a proprietary algorithm changes the phase of the recovery clock. The  $\mu$ PD9306 A has the same jitter rejection abilities that you would expect from a well-designed analog PLL. It can accept disk data with jitter in excess of plus or minus 30 ns. As an option, delay line 1 taps DD1 and DD2 may be connected to the 30-ns and the 70-ns tap respectively. Due to this option, the  $\mu$ PD9306 A performance is affected by its immunity to bit jitter and its tolerance to speed variation as shown in table 1.





### Description

The μPD7262 is a highly-integrated, single-chip controller for ESDI Winchester Disks. While conforming to the complete revision E of the ESDI specification, this device executes 22 high-level commands that provide flexibility and ease of usage. The μPD7262 is based on the proven μPD7261A/μPD7260 architecture but adapted to the special requirements of this disk interface. It eliminates numerous ICs and gives complete access to all of the features implemented by the ESDI disk drive manufacturers.

### Features

- Controls ESDI serial mode disks
- Controls up to seven disk drives
- Programmable soft and hard sector formats
- 18-MHz data transfer rate
- Multi-sector, -track, and -cylinder transfer capability
- Implied seek and parallel seek capability
- 22 high level disk commands and 4 auxiliary commands
- CRC error detection
- ECC error detection and correction
- Single +5 volt supply
- NMOS 40-pin DIP

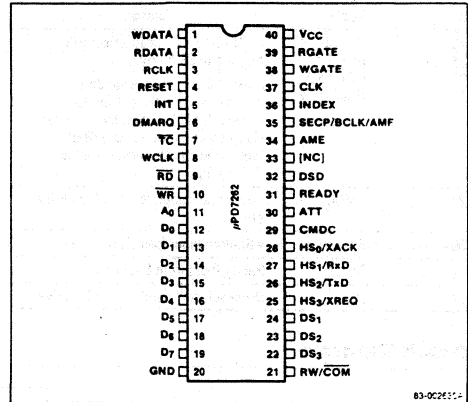
### Disk Commands

Check	Read Diagnostic
Chip Reset	Read ID
Clear Command End Bit	Recalibrate
Clear Data FIFO	Scan
Detect Error	Send
Format Sector	Send Extended
Format Track	Sense Seek Status
Get Internal Information	Sense Unit Status
Group Assign	Specify1
Logical Seek	Specify2
Mask SRQ Interrupt	Verify Data
Physical Seek	Verify ID
Read Data	Write Data

### Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD7262D-18	40-pin ceramic DIP	18 MHz
μPD7262D-12	40-pin ceramic DIP	12 MHz

### Pin Configuration



### Pin Identification

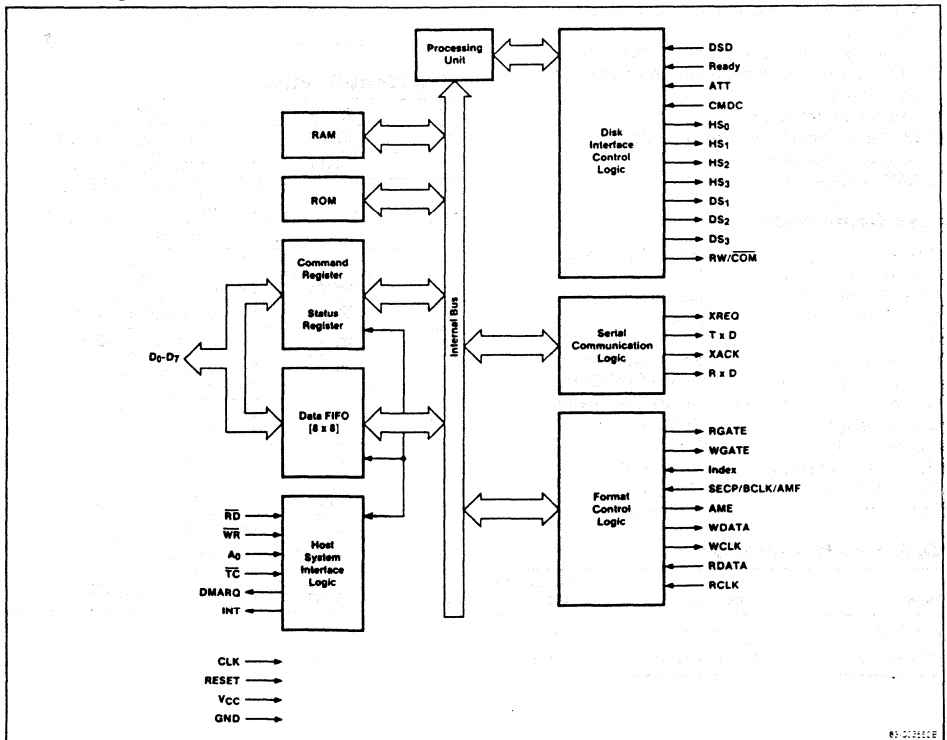
No.	Symbol	Function
1	WDATA	NRZ write data output to ESDI drive
2	RDATA	NRZ read data input from ESDI drive
3	RCLK	Read/reference clock input from ESDI drive
4	RESET	System reset input
5	INT	Interrupt request output
6	DMARQ	DMA request output
7	TC	Terminal count input from DMAC
8	WCLK	Write clock output to ESDI drive
9	RD	Read control input signal from host computer
10	WR	Write control input signal from host computer
11	A <sub>0</sub>	Address select input from host computer
12-19	D <sub>0</sub> -D <sub>7</sub>	Data bus from host computer
20	GND	System ground
21	RW/COM	This output specifies the status of pins 25-28
22-24	DS <sub>2</sub> -DS <sub>1</sub>	Drive select outputs to ESDI drive

**Pin Identification (cont)**

No.	Symbol	Function
25	HS <sub>3</sub> /XREQ	If RW/COM = 1: head select (HS) outputs to ESDI drive.
26	HS <sub>2</sub> /TxD	If RW/COM = 0 for serial data transfer to ESDI drive: transmit request (XREQ) output, transmit data (TxD) output, receive data (RxD) input, and transfer acknowledge (XACK) input.
27	HS <sub>1</sub> /RxD	
28	HS <sub>0</sub> /XACK	
29	CMDC	
30	ATT	Attention input from ESDI drive
31	READY	Ready input from ESDI drive
32	DSD	Drive selected input from ESDI drive

No.	Symbol	Function
33	NC	Not connected; leave open
34	AME	Address mark enable output from ESDI drive
35	SECP/BCLK/AMF	Sector pulse or byte clock or address mark found; input from ESDI drive (mutually exclusive)
36	INDEX	Index detected input from ESDI drive
37	CLK	System clock input to μPD7262
38	WGATE	Write gate output to ESDI drive
39	RGATE	Read gate output to ESDI drive
40	VCC	+5 V (Typical) input

**Block Diagram**



### Absolute Maximum Ratings (Ta = 25 °C)

Item	Sym.	Condition	Rating	Unit
Supply voltage	V <sub>CC</sub>		- 0.5 ~ + 7.0	V
Input voltage	V <sub>I</sub>		- 0.5 ~ + 7.0	V
Output voltage	V <sub>O</sub>		- 0.5 ~ + 7.0	V
Operating temp.	T <sub>OPT</sub>		- 10 ~ + 70	°C
Storage temp.	T <sub>STG</sub>		- 65 ~ + 150	°C

### DC Characteristics (Ta = - 10 ~ + 70 °C, V<sub>CC</sub> = + 5.0 V ± 10%)

Item	Sym.	Condition	Min.	Typ.	Max.	Unit
Input voltage low	V <sub>IL1</sub>	all except CLK, RCLK	- 0.5		+ 0.8	V
	V <sub>IL2</sub>	CLK, RCLK	- 0.5		+ 0.6	V
Input voltage high	V <sub>IH1</sub>	all except CLK, RCLK	+ 2.2		V <sub>CC</sub> + 0.5	V
	V <sub>IH2</sub>	CLK, RCLK	+ 3.3		V <sub>CC</sub> + 0.5	V
Output voltage	V <sub>OL</sub>	I <sub>ol</sub> = + 2.0 mA			+ 0.4	V
Output voltage high	V <sub>OH1</sub>	I <sub>oh</sub> = - 100 μA all except pins 21 - 33	+ 2.4			V
	V <sub>OH2</sub>	I <sub>oh</sub> = - 50 μA pins 21 - 33	+ 2.4			V
Input leakage current	I <sub>LI1</sub>	V <sub>i</sub> = 0.45 V to V <sub>CC</sub> all except pins 21 - 33			± 10	μA
	I <sub>LI2</sub>	V <sub>i</sub> = 0.45 V to V <sub>CC</sub> pins 21 - 33			- 700	μA
Output leakage current	I <sub>LO</sub>	V <sub>o</sub> = 0.45 V to V <sub>CC</sub>			± 30	μA
Supply current	I <sub>CC</sub>			250	350	mA

### Capacitance (Ta = 25 °C, V<sub>CC</sub> = GND = 0V)

Item	Sym.	Condition	Min.	Typ.	Max.	Unit
Input Capacitance	C <sub>i</sub>	f = 1 MHz all unmeasured pins tied to GND			15	pF
Output Capacitance	C <sub>o</sub>				15	pF
Input/Output Capacitance	C <sub>io</sub>				20	pF

**AC Characteristics** (Ta = - 10 ~ + 70 °C, VCC = + 5.0 V ± 10%)

Host Interface

Item	Sym.	Condition	μPD7262-18		μPD7262-12		Unit
			Min.	Max.	Min.	Max.	
CLK cycle time	tCY		55	10000	83	10000	ns
CLK high width	tCH		20		30		ns
CLK low width	tCL		20		30		ns
CLK rise time	tCR			10		10	ns
CLK fall time	tCF			10		10	ns
AO setup to $\overline{RD} \downarrow$	tAR		0		0		ns
AO hold from $\overline{RD} \uparrow$	tRA		0		0		ns
$\overline{RD}$ pulse width	tRR		100		100		ns
Data delay from AO	tAD			85		85	ns
Data delay from $\overline{RD} \downarrow$	tRD			85		85	ns
Data float delay from $\overline{RD} \uparrow$	tRDF		0	75	0	75	ns
AO setup to $\overline{WR} \downarrow$	tAW		0		0		ns
AO hold from $\overline{WR} \uparrow$	tWA		0		0		ns
$\overline{WR}$ pulse width	tWW		100		100		ns
Data setup to $\overline{WR} \uparrow$	tDW		55		55		ns
Data hold from $\overline{WR} \uparrow$	tWD		5		5		ns
Recovery time from $\overline{RD} \uparrow$ , $\overline{WR} \uparrow$	tRV		70		70		ns
INT delay from $\overline{WR} \uparrow$	tWI			200		200	ns
DMARQ delay from $\overline{WR} \uparrow$	tWRQ			125		125	ns
DMARQ delay from $\overline{RD} \uparrow$	tRRQ1	*		160		160	ns
DMARQ delay from $\overline{RD} \downarrow$	tRRQ2	**		100		120	ns
$\overline{TC}$ pulse width	tTT		80		80		ns
$\overline{TC} \uparrow$ delay from $\overline{RD} \downarrow$ or $\overline{WR} \downarrow$	tRWT		80		80		ns
$\overline{RD} \uparrow$ or $\overline{WR} \uparrow$ delay from $\overline{TC} \downarrow$	tTWR		80		80		ns
RESET pulse width	tRES		100		100		tCY

\* : during read operation

\*\* : after read operation

**AC Characteristics** (Ta = -10 ~ +70 °C, VCC = +5.0 V ± 10%)  
Drive Interface

Item	Sym.	Condition	μPD7262-18		μPD7262-12		Unit
			Min.	Max.	Min.	Max.	
RCLK cycle time	tRCY		55	10000	83	10000	ns
RCLK high width	tRCH		0.40		0.40		tRCY
RCLK low width	tRCL		0.40		0.40		tRCY
RCLK rise time	tRCR			8		8	ns
RCLK fall time	tRCF			8		8	ns
Data setup to RCLK ↑	tRDRC		15		15		ns
Data hold from RCLK ↑	tRDRD		15		15		ns
WCLK cycle time	tWCY		0.95	1.05	0.95	1.05	tRCY
WCLK high width	tWCH		0.25	0.75	0.25	0.75	tRCY
WCLK low width	tWCL		0.25	0.75	0.25	0.75	tRCY
WCLK rise time	tWCR			10		10	ns
WCLK fall time	tWCF			10		10	ns
Data setup to WCLK ↑	tWDWC		0.20		0.20		tRCY
Data hold from WCLK ↑	tWCWD		0.20		0.20		tRCY
RGATE ↑ delay from AMF ↑	tAFRG			300		300	nS
AME ↓ delay from AMF ↓	tFAFE		0		0		ns
SECP pulse width	tSTPF		400		400		ns
INDEX pulse width	tDXF			400		400	ns

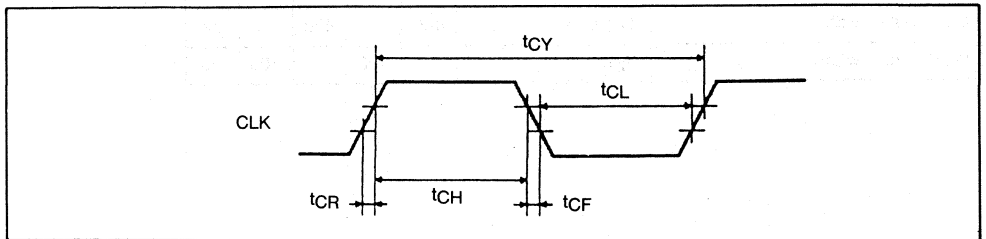
**AC Characteristics** (Ta = - 10 ~ + 70 °C, VCC = + 5.0 V ± 10%)

Drive Interface

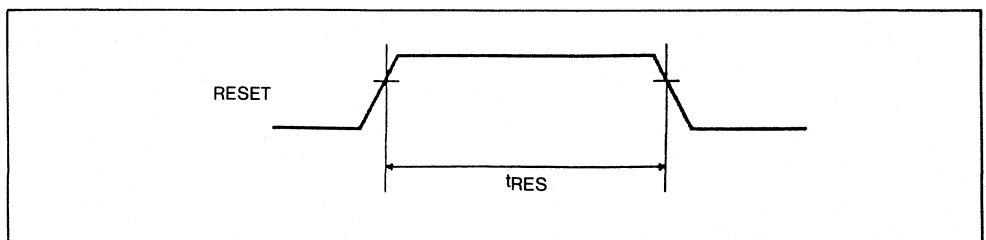
Item	Sym.	Condition	Min.	Max.	Unit
WGATE setup to AME ↑	tWGAE	*	440		nS
WGATE hold from AME ↓	tAEWG	*	440		nS
WGATE mask time	tWGM	*	8		tRCY
Communication setup to RW/COM ↓	tCC		830		nS
HS0-1 float delay from RW/COM ↓	tCHF			50	
HS0-3 output delay from RW/COM ↓	tRWH			50	
TxD setup to XREQ	tTDXR		50		nS
TxD hold from XREQ	tXRTD	1 bit	0	2tCY+400	nS
XACK delay from XREQ	tXRXAT	send	0		nS
XREQ delay from XACK	tXAXRT		50	19tCY+300	nS
RxD setup to XACK	tRDXA		50		nS
RxD hold from XACK	tXARD	1 bit	0		nS
XACK delay from XREQ	tXRXAR	receive	0		nS
XREQ delay from XACK	tXAXRR		50	19tCY+300	nS

\*: during format write operation

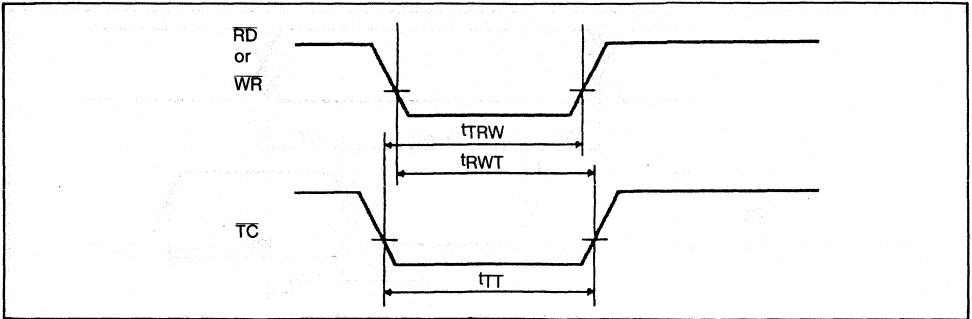
**CLK pulse timing**



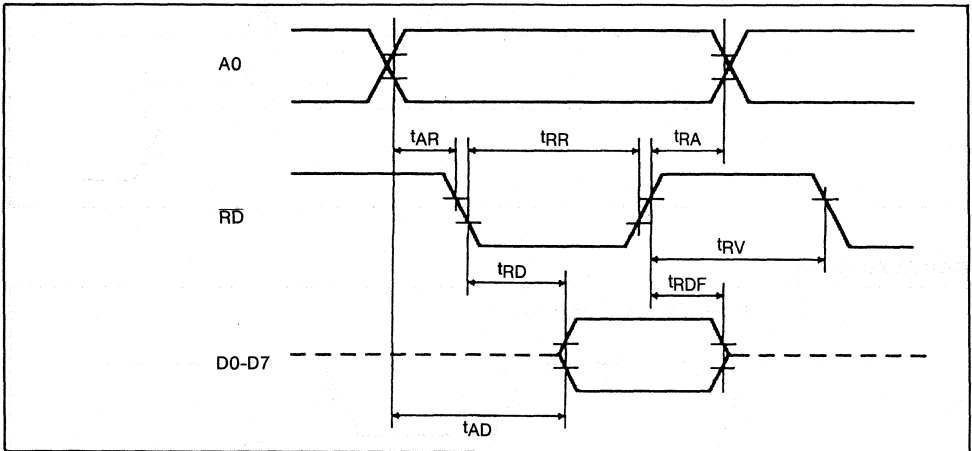
**RESET pulse timing**



### TC pulse timing

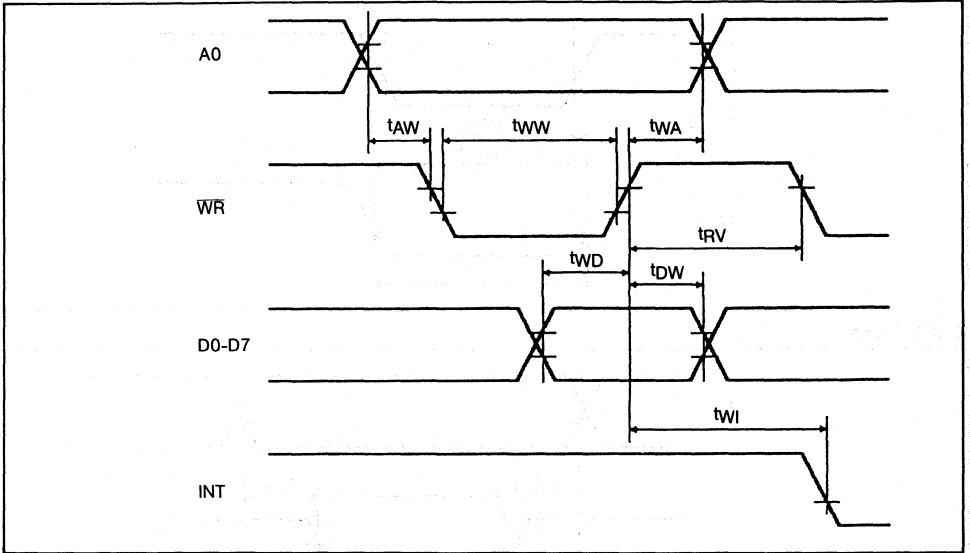


### READ timing

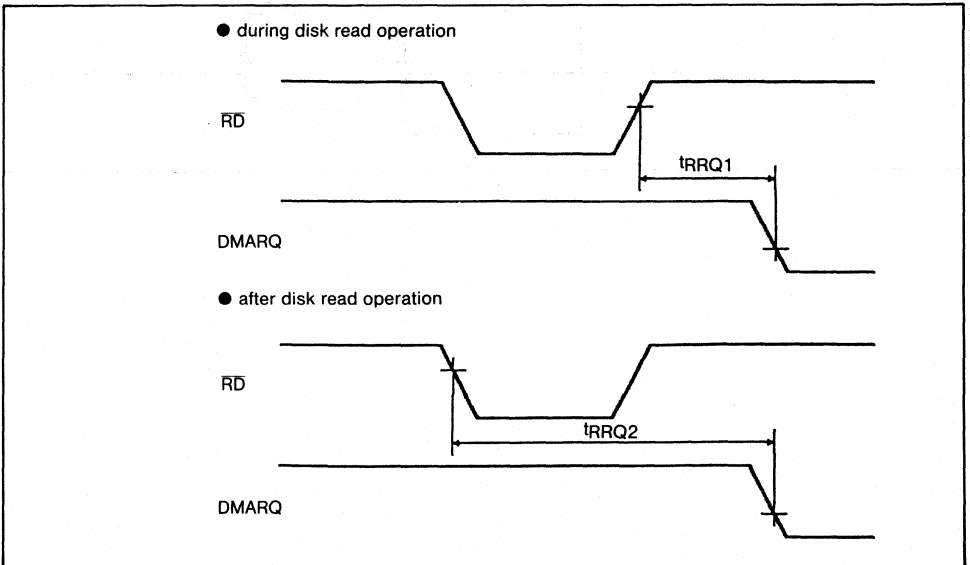




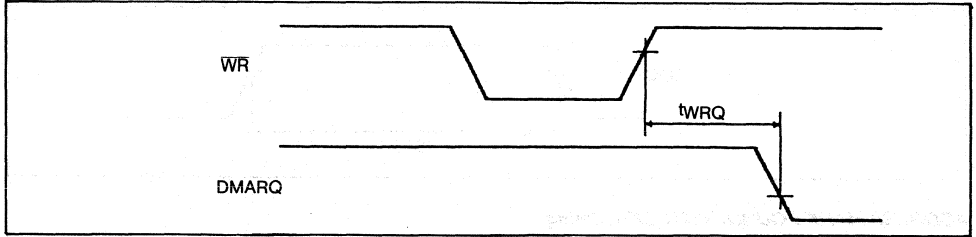
WRITE timing



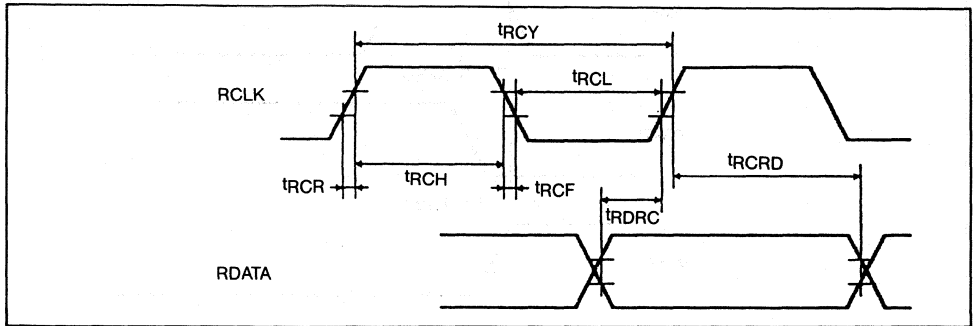
DAM READ timing



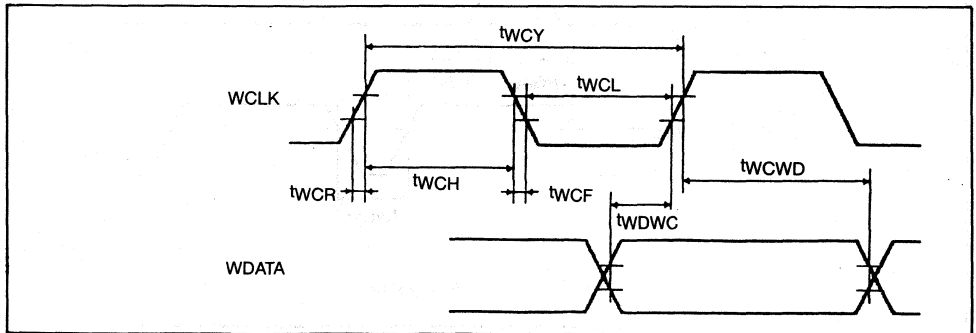
### DMA WRITE timing



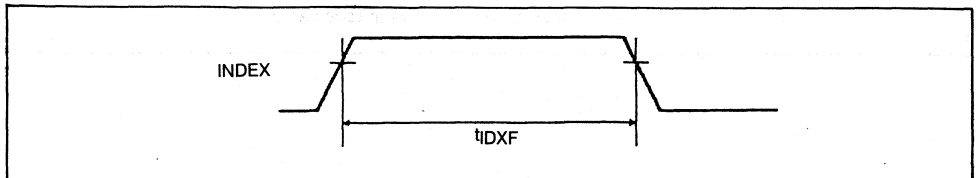
### DATA READ timing



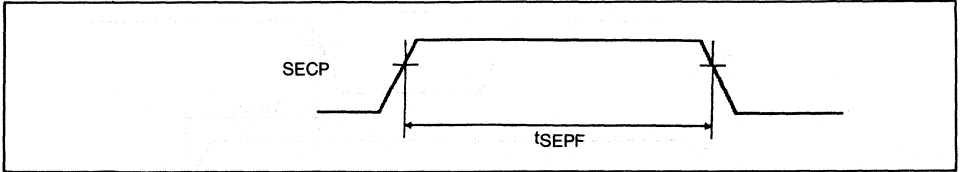
### DATA WRITE timing



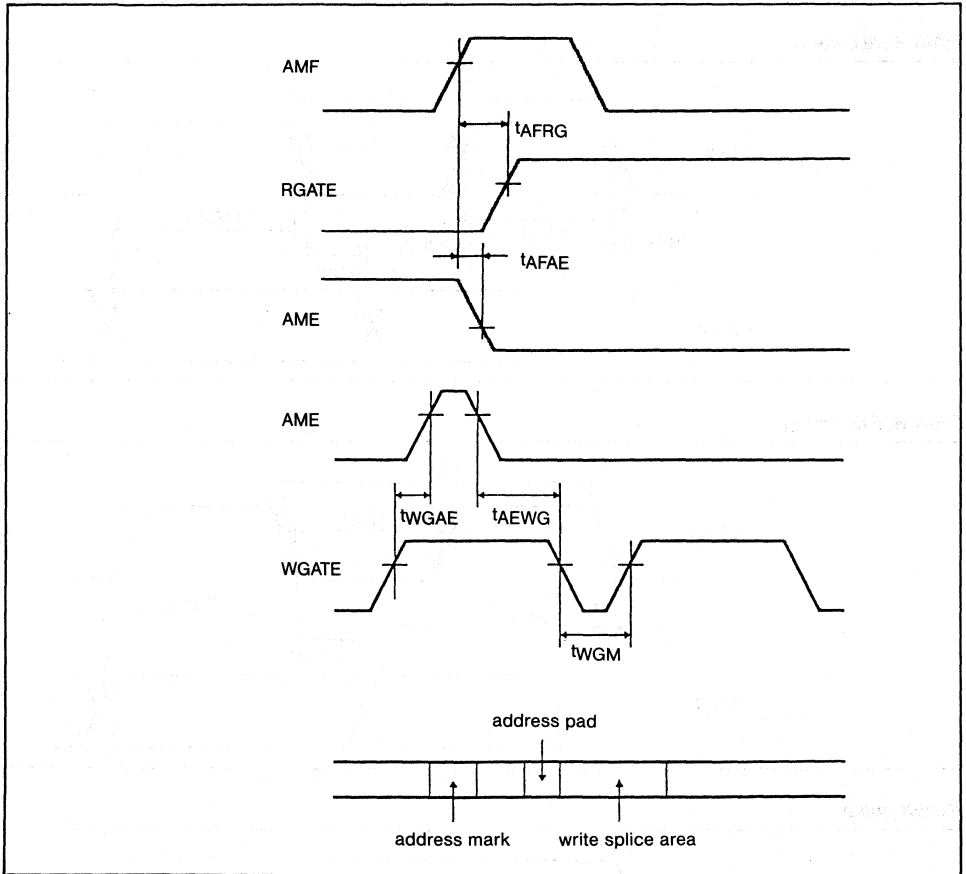
### INDEX pulse



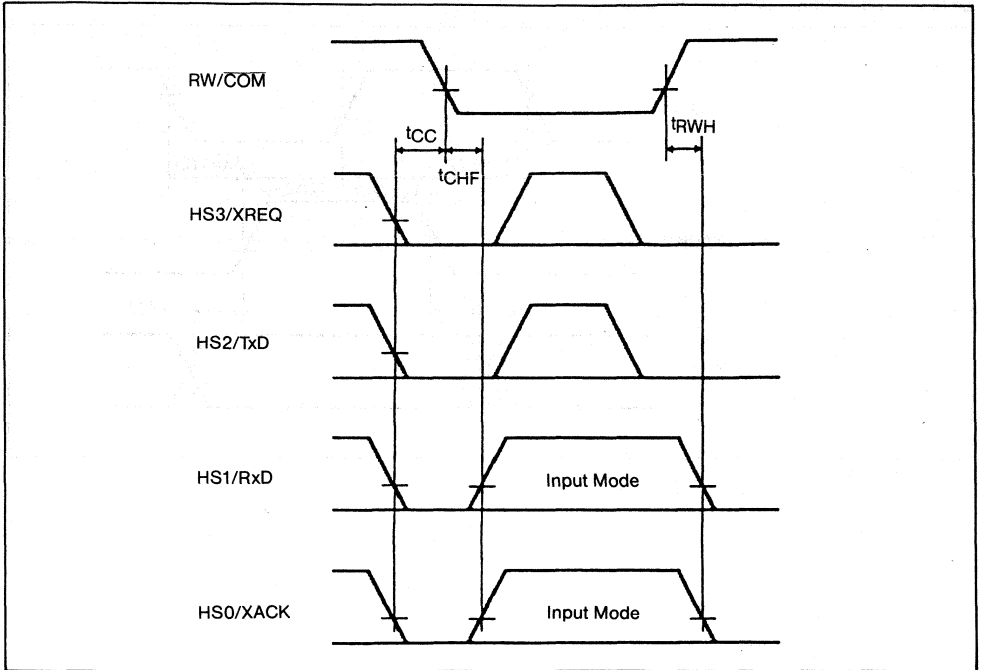
**SECTOR pulse**



**ADDRESS MARK FOUND/READ GATE timing**

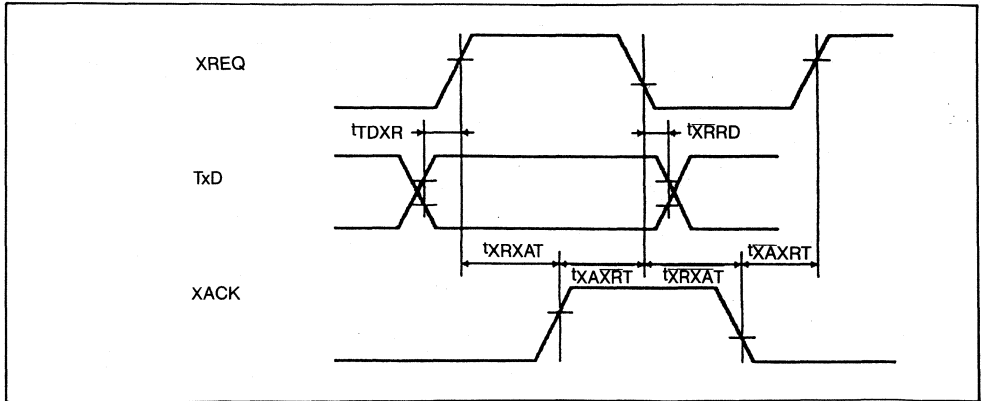


### READ/WRITE Communicate timing

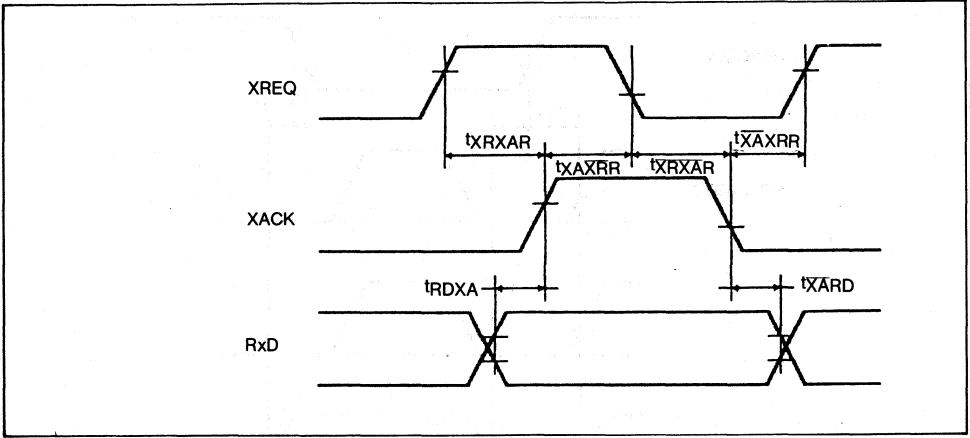


5

### ONE BIT TRANSFER timing - to drive



ONE BIT TRANSFER timing – from drive



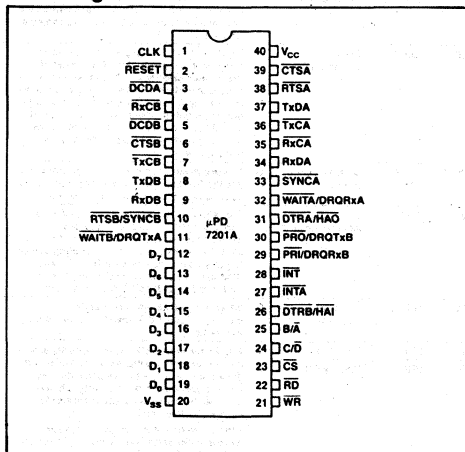
### Description

The μPD7201A is a dual-channel multifunction peripheral communication controller designed to satisfy a wide variety of serial data communication requirements in computer systems. Its basic function is a serial-to-parallel, parallel-to-serial converter/controller and within that role it is configurable by systems software so its "personality" can be optimized for a given serial data communications application.

The μPD7201A is capable of handling asynchronous and synchronous byte-oriented protocols such as IBM Bisync, and synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device can also be used to support virtually any other serial protocol for applications other than data communications.

The μPD7201A can generate and check CRC codes in any synchronous mode and can be programmed to check data integrity in various modes. The device 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.

### Pin Configuration



### Features

- Two fully independent duplex serial channels
- Four independent DMA channels for send/received data for both serial inputs/outputs
- Programmable interrupt vectors and interrupt priorities
- Modem controls signals
- Variable software programmable data rate, up to 1.1M baud at 5MHz clock
- Double buffered transmitter data and quadruply buffered received data
- Programmable CRC algorithm
- Selection of Interrupt, DMA or Polling mode of operation
- Asynchronous operation
  - Character length: 5, 6, 7, or 8 bits
  - Stop bits: 1, 1½, 2
  - Transmission speed: x1, x16, x32, or x64 clock frequency
  - Parity: odd, even, or disable
  - Break generation and detection
  - Interrupt on parity, overrun, or framing errors
- Monosync, bisync, and external sync operations
  - Software selectable sync characters
  - Automatic sync insertion
  - CRC generation and checking
  - I-field residue handling
- HDLC and SDLC operations
  - Abort sequence generation and detection
  - Automatic zero insertion and detection
  - Address field recognition
  - CRC generation and checking
- N-channel MOS technology
- Single +5V power supply; interface to most microprocessors including 8080, 8085, 8086, and others.
- Single-phase TTL clock
- Available in plastic and ceramic dual-in-line packages

# μPD7201A

## Pin Identification

Pin			
No.	Symbol	Name	I/O
1	CLK	System Clock	I
2	RESET	Reset	I
3, 5	DCDA, DCDB	Data Carrier Detect	I
4, 35	RxCA, RxCB	Receiver Clocks	I
6, 39	CTSA, CTSB	Clear to Send	I
7, 36	TxCa, TxCb	Transmitter Clocks	I
8, 37	TxDa, TxDb	Transmit Data	O (Active High)
9, 34	RxDa, RxDb	Receive Data	I (Active High)
10, 33	SYNCA, SYNCB	Synchronization	I/O
10, 38	RTSA, RTSB	Request to Send	O
11, 29, 30, 32	DROTxA, DROTxB, DRQRxA, DRQRxB	DMA Request	O
11, 32	WAITA, WAITB	Wait	O

**Description**

The μPD7201A uses a standard TTL clock.

A low RESET disables both receivers and transmitters, forces TxDA and TxDB marking, forces the modem controls high and disables all interrupts. The control registers must be rewritten after the μPD7201A is reset and before data is transmitted or received. RESET must be active for a minimum of one complete CLK cycle. (Active Low)

These signals are similar to the CTS inputs except they can be used as receiver enables. (Active Low)

The receiver clocks may be 1, 16, 32, or 64 times the data rate in asynchronous modes. Receive data is sampled on the rising edge of RxC.

When programmed as auto enables, a low on these inputs enables the respective transmitter. If not programmed as auto enables, these inputs may be programmed as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The μPD7201A detects pulses on these inputs and interrupts the processor on both logic level transitions. The Schmitt-trigger inputs do not guarantee a specified noise-level margin. (Active Low)

In asynchronous modes, the transmitter clocks may be 1, 16, 32, or 64 times the data rate. The multiplier for the transmitter and the receiver must be the same. Both TxC and RxC inputs are Schmitt-trigger buffered for relaxed rise- and fall-time requirements (no noise margin is specified). TxD changes on the falling edge of TxC.

(Active High)

(Active High)

These pins can act either as inputs or outputs. In the asynchronous receive mode they are inputs similar to CTS and DCD. In this mode the transitions on these lines affect the state of the Sync/Hunt status bits in read register 0. In the external sync mode these lines also act as inputs. When external synchronization is achieved, SYNC must be driven low on the second rising edge of RxC after that rising edge of RxC on which the last bit of the sync character was received. In other words, after the sync pattern is detected, the external logic must wait for two full receive clock cycles to activate the SYNC input. Once SYNC is forced low it is wise to keep it low until the processor informs the external sync logic that synchronization has been lost or a new message is about to start. Character assembly begins on the rising edge of RxC that immediately precedes the falling edge of SYNC in the external sync mode. (Active Low)

In the internal synchronization mode (mono-sync and bi-sync) these pins act as outputs that are active during the part of the receive clock (RxC) cycle in which sync characters are recognized. The sync condition is not latched.

When the RTS bit is set, the RTS output goes low. When the RTS bit is reset in the asynchronous mode, the output goes high after the transmitter is empty. In asynchronous modes, the RTS pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs. (Active Low)

These signals are generated by the receiver or transmitter of channel A or channel B. These signals can be connected to most DMA controllers directly and are used for handshaking during DMA transfer. (Active High)

Wait lines for both channels that synchronize the processor to the μPD7201A data rate. The reset state is open drain.

Pin			
No.	Symbol	Name	I/O
12-19	D <sub>0</sub> -D <sub>7</sub>	System Data Bus	I/O
20	V <sub>SS</sub>	Ground	
21	WR	Write	I
22	RD	Read	I
23	CS	Chip Select	I
24	CD	Control or Data Select	I
25	B/A	Channel A or B Select	I
26	HAI <sup>⊙</sup>	DMA Acknowledge	I
26, 31	DTRA, DTRB	Data Terminal Ready	O
27	INTA	Interrupt Acknowledge	I
28	INT	Interrupt Request	O
29	PRI	Priority In	I
30	PRO	Priority Out	O
31	HAO <sup>⊙</sup>	DMA Acknowledge	O

**Description**

The system data bus transfers data and commands between the processor and the μPD7201A. D<sub>0</sub> is the least significant bit. (Tristate)

Ground.

The WR signal is used to control the transfer of either command or data from the processor or the memory to the μPD7201A. (Active Low)

If RD is active, a memory or I/O read operation is in progress. RD is used with CD, B/A, and CS to transfer data from the μPD7201A to the processor or the memory. (Active Low)

A low level at this input enables the μPD7201A to accept command or data inputs from the processor during a write cycle, or to transmit data to the processor during a read cycle. (Active Low)

This input defines the type of information transfer performed between the processor and the μPD7201A. A high at this input during a processor write to or read from the μPD7201A causes the information on the data bus to be interpreted as a command for the channel selected by B/A. A low at CD means that the information on the data bus is data. (High selects Control)

This input defines which channel is accessed during a data transfer between the processor and the μPD7201A. (High selects Channel B)

Typically, the HLDA signal driven from the processor is input to the HAI terminal of the highest priority μPD7201A, and the HAO output of that μPD7201A is daisy-chained to the HAI input of the lower priority μPD7201A and propagated downstream. HAI and HAO signals provide acknowledgement for the highest priority outstanding DMA request. (Active Low)

These outputs follow the state programmed into the DTR bit. They can also be programmed as general-purpose outputs. (Active Low)

This signal is generated by the processor and is sent to all peripheral devices. It serves to acknowledge the interrupt and to allow the highest priority interrupting device to put an 8-bit vector on the bus. INT and INTA are compatible with the fully nested option of the μPD8259A-S. (Active Low)

When the μPD7201A is requesting an interrupt, it pulls INT low. (Open Collector, Active Low)

PRI is used with PRO to form a priority daisy-chain when there is more than one interrupt-driven device. A low on this line indicates no other device of higher priority is being serviced by a processor interrupt service routine. (Active Low)

PRO is low only if PRI is low and the processor is not servicing an interrupt from the μPD7201A. Thus, this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its processor interrupt service routine. (Active Low)

Typically, the HLDA signal driven from the processor is input to the HAI terminal of the highest priority μPD7201A, and the HAO output of that μPD7201A is daisy-chained to the HAI input of the lower priority μPD7201A and propagated downstream. HAI and HAO signals provide acknowledgement for the highest priority outstanding DMA request. (Active Low)

Note: ⊙ The HAI/HAO functions are only available in the DMA internal priority mode.

## Programming the MPSC<sup>2</sup>

The software operation of the MPSC<sup>2</sup> is very straightforward. Its consistent register organization and high-level command structure help minimize the number of operations required to implement complex protocol designs. Programming is further simplified by the MPSC<sup>2</sup>'s extensive interrupt and status reporting capabilities. This section is divided into two parts.

## The MPSC<sup>2</sup> Registers

The MPSC<sup>2</sup> interfaces to the system software with a number of control and status registers associated with each channel. Commonly used commands and status bits are accessed directly through control and status registers 0. Other functions are accessed indirectly with a register pointer to minimize the address space that must be dedicated to the MPSC<sup>2</sup>.

### Control Register

Control Register	Function
0	Frequently used commands and register pointer control
1	Interrupt control
2	Processor/bus interface control
3	Receiver control
4	Mode control
5	Transmitter control
6	Sync/address character
7	Sync character

### Status Register

Status Register	Function
0	Buffer and "external/status" status
1	Received character error and special condition status
2 (Channel B only)	Interrupt vector
3	Tx byte count register, low byte
4	Tx byte count register, high byte

All control and status registers except CR2 are separately maintained for each channel. Control and status registers 2 are linked with the overall operation of the MPSC<sup>2</sup> and have different meanings when addressed through different channels.

When initializing the MPSC<sup>2</sup> control register 2A (and 2B if desired) should be programmed first to establish the MPSC<sup>2</sup> processor/bus interface mode. Each channel may then be programmed to be used separately, beginning with control register 4 to set the protocol mode for that channel. The remaining registers may then be programmed in any order.

### Control Register 0

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CRC Control Command		Command			Register Pointer		

### Register pointer (D<sub>0</sub> – D<sub>2</sub>)

The register pointer specifies which register number is accessed at the next control register write or status register read. After a hardware or software reset the register pointer is set to zero. Therefore, the first control byte goes to control register 0. When the register pointer is set to a value other than zero the next control or status (C/D = 1) access is to the specified register, after which the pointer is reset to zero. Other commands can be freely combined in control register 0 by setting the register pointer.

### Commands (D<sub>3</sub> – D<sub>5</sub>)

Commands commonly used during the operation of the MPSC<sup>2</sup> are grouped in control register 0. They include the following:

**Null (000):** This command has no effect and is used only to set the register pointer or issue a CRC command.

**Send abort (001):** When operating in the SDLC mode this command causes the MPSC<sup>2</sup> to transmit the SDLC abort code, issuing 8 to 13 consecutive ones. Any data currently in the transmitter or the transmitter buffer is destroyed. After sending the abort the transmitter reverts to the idle phase (flags). When using the Tx byte count mode enable (D<sub>6</sub> of CR1), and an underrun condition occurs, the μPD7201A will automatically issue the send abort command.

**Reset external status interrupts (010):** When the external/status change flag is set, the condition of bits D<sub>3</sub> – D<sub>7</sub> of status register 0 are latched to allow the capture of the short pulses that may occur. The reset external/status interrupts command reenables the latches so that new interrupts may be sensed.

**Channel reset (011):** This command has the same effect on a single channel as an external reset at pin 2. A channel reset command to channel A resets the internal interrupt prioritization logic. This does not occur when a channel reset command is issued to channel B. All control registers associated with the channel to be reset must be reinitialized. After a channel reset, wait at least four system clock cycles before writing new commands or controls to that channel.

**Enable interrupt on next character (100):** When operating the MPSC<sup>2</sup> in an interrupt on first received character mode this command may be issued at any time. This command must be issued at the end of a message to reenables the interrupt logic for the next received character (the first character of the next message).

### Reset pending transmitter interrupt/DMA request

**(101):** A pending transmitter buffer becoming empty interrupt or DMA request can be reset without sending another character by issuing this command (typically at the end of a message). A new transmitter buffer becoming empty interrupt or DMA request is not made until another character has been loaded and transferred to the transmitter shift register or when, if operating in the synchronous or SDLC modes, the first CRC character has been sent.

**Error Reset (110):** This command resets a special receive condition interrupt. It also reenables the parity and overrun error latches that allow errors at the end of a message to be checked.

5



**End of interrupt (I11) (channel A only):** Once an interrupt request has been issued by the MPSC<sup>2</sup> all lower priority internal and external interrupts in the daisychain are held off to permit the current interrupt to be serviced while allowing higher priority interrupts to occur. At some point in the interrupt service routine (generally at the end), the end of interrupt command must be issued to channel A to reenable the daisychain and allow any pending lower priority internal interrupt requests to occur. The EOI command must be sent to channel A for interrupts that occurred on either channel.

**CRC Control Commands (D<sub>6</sub> – D<sub>7</sub>)**

The following commands control the operation of the CRC generator/checker logic.

**Null (00):** This command has no effect and is used when issuing other commands or setting the register pointer.

**Reset receiver CRC checker (01):** This command resets the CRC checker to zero when the channel is in a synchronous mode and resets to all ones when in an SDLC mode.

**Reset transmitter CRC generator (10):** This command resets the CRC generator to zero when the channel is in a synchronous mode and resets to all ones when in an SDLC mode.

**Reset idle/CRC latch (11):** This command resets the idle/CRC latch so that when a transmitter underrun condition occurs (that is, the transmitter has no more characters to send), the transmitter enters the CRC phase of operation and begins to send the 16-bit CRC character calculated up to that point. The latch is then set so that if the underrun condition persists, idle characters are sent following the CRC. After a hardware or software reset the latch is in the set state. This latch is automatically reset after the first character has been loaded into the Tx buffer in the SDLC mode.

**Control Register 1**

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
Wait Function Enable	Tx Byte Count Mode Enable	Wait on Receiver Transmitter	Receiver Interrupt Mode	Condition Affects Vector	Transmitter Interrupt Enable	Ext/Status INT Enable	

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
Low Byte							

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
High Byte							

**External/status interrupt enable (D<sub>0</sub>)**

When this bit is set to one the MPSC<sup>2</sup> issues an interrupt whenever any of the following conditions occur:

- Transition of the DCD input pin
- Transition of the CTS input pin
- Transition of the SYNC input pin
- Entering or leaving synchronous hunt phase, break detection or termination
- SDLC abort detection or termination
- Idle/CRC latch becoming set (CRC being sent)
- After ending flag is sent in the SDLC mode

**Transmitter interrupt enable (D<sub>1</sub>)**

When this bit is set to one the MPSC<sup>2</sup> issues an interrupt when:

- 1) The character currently in the transmitter buffer is transferred to the shift register (transmitter buffer becoming empty) or,
- 2) The transmitter enters the idle phase and begins transmitting sync or flag characters.
- 3) The Tx byte mode enable bit is set (CR1 – D6 = 1). The 7201A will automatically issue a Tx interrupt or DMA request when the transmitter becomes enabled (CR5 – D3 = 1).

**Condition affects vector (D<sub>2</sub>) (programmed in channel B for both channels)**

When this bit is set to zero the fixed vector programmed in CR2B during MPSC<sup>2</sup> initialization is returned in an interrupt acknowledge sequence. When this bit is set to one the vector is modified to reflect the condition that caused the interrupt.

**Receiver interrupt mode (D<sub>3</sub> – D<sub>4</sub>)**

This field controls how the MPSC<sup>2</sup>'s interrupt/DMA logic handles the character received condition.

**Receiver interrupts/DMA request disabled (00)**

The MPSC<sup>2</sup> does not issue an interrupt or a DMA request when a character has been received.

**Interrupt/DMA on first received character only (01)**

In this mode the MPSC<sup>2</sup> issues an interrupt only for the first character received after an enable interrupt/DMA on first character command (CR0) has been given. If the channel is in a DMA mode, a DMA request is issued for each character received including the first. This mode generally is used whenever the MPSC<sup>2</sup> is in a DMA or block transfer mode. This will signal the processor that the beginning of an incoming message has been received.

**Interrupt (and issue a DMA request) on all received characters (10)**

In this mode an interrupt (and DMA request if the DMA mode is selected) is issued whenever there is a character present in the receiver buffer. A parity error is considered a special receive condition.

**Interrupt (and issue a DMA request) on all received characters (11)**

This mode is the same as the one above except that a parity error is not considered a special receive condition. The following are considered special receive conditions:

- Receiver overrun factor
- Asynchronous framing error
- Parity error (if specified)
- SDLC end of message (final flag received)

**Wait on receiver/transmitter (D<sub>5</sub>)**

If the wait function is enabled for block mode transfers, setting this bit to zero causes the MPSC<sup>2</sup> to issue a wait (WAIT output goes low) when the processor attempts to write a character to the transmitter while the transmitter buffer is full. Setting this bit to one causes the MPSC<sup>2</sup> to issue a wait when the processor attempts to read a character from the receiver while the receiver buffer is empty.

### Tx byte count mode enable (D<sub>6</sub>)

Each channel has a 16-bit Tx byte count register used for automatic transmit termination. When this bit is set to one the next two consecutive command cycle writes will be to the byte count register. The first byte is loaded into the lower 8 bits and the second to the upper 8 bits of the byte count register. The byte count register holds the number of transfers to be performed by the transmitter. A byte counter is incremented each time a transfer is performed until the value of the byte counter is equal to the value in the byte count register. When equal, interrupts or DMA requests will be stopped until the byte count enable bit is issued and a new byte count is loaded into the byte count register. If a transmit underrun occurs in the SDLC mode, and the byte count is not equal to the byte count register, the abort sequence will be sent automatically.

Also, when using the Tx byte count mode, a transmit interrupt or DMA request will automatically become active after issuing the Tx enable command to CR5.

The Tx byte count mode can be cleared by either a channel reset command or a hardware reset.

### Wait function enable (D<sub>7</sub>)

Setting this bit to one enables the wait function which is described in CR1.

### Control Register 2 (Channel A)

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
Pin 10 SYNCB/RTSB	Rx INT Mask	Interrupt Vector Mode			Priority	DMA Mode Select	

### DMA mode select (D<sub>0</sub> – D<sub>1</sub>)

Setting this field determines whether channel A or B is used in a DMA mode (i.e., data transfers are performed by a DMA controller) or in a non-DMA mode where transfers are performed by the processor in either a polled, interrupt, or block transfer mode. The functions of some MPSC<sup>2</sup> pins are also controlled by this field.

### DMA Mode Selection

Channel		Pin Function							
D <sub>1</sub>	D <sub>0</sub>	A	B	11	26	29	30	31	32
0	0	Non-DMA	Non-DMA	WAITB	DTRB	PRI	PRO	DTRA	WAITA
0	1	DMA	Non-DMA	DRQTxA	HAI	PRI	PRO	HAO	DRQRxA
1	0	DMA	DMA	DRQTxA	HAI	DRQRxB	DRQTxB	HAO	DRQRxA
1	1	DMA	DMA	DRQTxA	DTRB	DRQRxB	DRQTxB	DTRA	DRQRxA

### Priority (D<sub>2</sub>)

This bit selects the relative priorities of the various interrupt and DMA conditions according to the application requirements.

### DMA/Interrupt Priorities

D <sub>2</sub>	Mode	Channel A	Channel B	DMA Priority Relation	Interrupt Priority Relation
0	INT	INT	—	—	SRxA, RxA > TxA > SRxB, RxB > TxB > ExTA > ExTB
1	INT	INT	—	—	SRxA, RxA > SRxB, RxB > TxA > TxB > ExTA > ExTB
0	DMA	INT	—	RxA > TxA	SRxA, RxA > SRxB, RxB > TxB > ExTA > ExTB
1	DMA	INT	—	RxA > TxA	SRxA, RxA > SRxB, RxB > TxB > ExTA > ExTB
0	DMA	DMA	—	RxA > TxA > RxB > TxB	SRxA, RxA > SRxB, RxB > ExTA > ExTB
1	DMA	DMA	—	RxA > RxB > TxA > TxB	SRxA, RxA > SRxB, RxB > ExTA > ExTB

### Interrupt vector mode (D<sub>3</sub> – D<sub>5</sub>)

This field determines how the MPSC<sup>2</sup> responds to an interrupt acknowledge sequence from the processor.

### Interrupt Acknowledge Sequence Response

D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	Mode	Status Register 2B and Interrupt Vector Bits Affected When Condition Affects Vector Is Enabled
0	0	0	Nonvectored	D <sub>4</sub> D <sub>3</sub> D <sub>2</sub>
0	0	1	Nonvectored	D <sub>4</sub> D <sub>3</sub> D <sub>2</sub>
0	1	0	Nonvectored	D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>
0	1	1	Illegal	—
1	0	0	8085 Master	D <sub>4</sub> D <sub>3</sub> D <sub>2</sub>
1	0	1	8085 Slave	D <sub>4</sub> D <sub>3</sub> D <sub>2</sub>
1	1	0	8086	D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>
1	1	1	8085/8259A Slave	D <sub>4</sub> D <sub>3</sub> D <sub>2</sub>

### Rx INT mask (D<sub>6</sub>)

This option is generally used in the DMA modes. Enabling this bit inhibits the interrupt from occurring when the interrupt/DMA Request On First Received Character mode is selected. In other words, only a DMA request will be generated when the first character is received.

### Pin 10 SYNCB/RTSB select (D<sub>7</sub>)

Programming a zero into this bit selects RTSB as the function of pin 10. A one selects SYNCB as the function.

### Control Register 2 (Channel B)

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
Interrupt Vector							

### Interrupt vector (D<sub>0</sub> – D<sub>7</sub>)

When the MPSC<sup>2</sup> is used in the vectored interrupt mode the contents of this register is placed on the bus during the appropriate portion of the interrupt acknowledge sequence. Its value is modified if status affects vector is enabled. The value of SR2B can be read at any time. This feature is particularly useful in determining the cause of an interrupt when using the MPSC<sup>2</sup> in a nonvectored interrupt mode.

### Control Register 3

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
Number of Received Bits per Character	Auto Enables	Enter Hunt Phase	Receiver CRC Enable	Address Search Mode	Sync Character Load Inhibit	Receiver Enable	

### Receiver enable (D<sub>0</sub>)

After the channel has been completely initialized, setting this bit to one allows the receiver to begin operation. This bit may be set to zero at any time to disable the receiver.

### Sync character load inhibit (D<sub>1</sub>)

In the character synchronous modes, this bit inhibits the transfer of sync characters to the receiver buffer thus performing a "sync-stripping" operation. When using the MPSC<sup>2</sup>'s CRC checking ability this feature should be used only to strip leading sync characters preceding a message since the load inhibit does not exclude sync characters embedded in the message from the CRC calculation. Synchronous protocols using other types of block checking such as checksum or LRC are free to strip embedded sync characters with this bit.

**Address search mode (D<sub>2</sub>)**

In the SDLC mode, setting this bit places the MPSC<sup>2</sup> in an address search mode. Character assembly does not begin until the 8-bit character (secondary address field) following the starting flag of a message matches either the address programmed into CR6 or the global address 11111111.

**Receiver CRC enable (D<sub>3</sub>)**

This bit enables and disables (1 = enable) the CRC checker in the COP mode allowing characters from the CRC calculation to be selectively included or excluded. The MPSC<sup>2</sup> features a one-character delay between the receiver shift register and the CRC checker so that the enabling or disabling takes effect with the last character transferred from the shift register to the receiver buffer. Therefore, there is one full character time in which to read the character and decide whether or not it should be included in the CRC calculation. In the SDLC mode, there is no 8-bit delay.

**Enter hunt phase (D<sub>4</sub>)**

Although the MPSC<sup>2</sup> receiver automatically enters the sync hunt phase after a reset, there are times when reentry may be desired, such as when it has been determined that synchronization has been lost or, in an SDLC mode, to ignore the current incoming message. Writing a one into this bit at any time after initialization causes the MPSC<sup>2</sup> to reenter the hunt phase.

**Auto enables (D<sub>5</sub>)**

Setting this bit to one causes the  $\overline{DCD}$  and  $\overline{CTS}$  inputs to act as enable inputs to the receiver and transmitter, respectively.

**Number of received bits per character (D<sub>6</sub> – D<sub>7</sub>)**

This field specifies the number of data bits assembled to make each character. The value may be changed on the fly while a character is being assembled and, if the change is made before the new number of bits has been reached it affects that character. Otherwise the new specifications take effect on the next character received.

**Received Bits per Character**

D <sub>7</sub>	D <sub>6</sub>	Bits per Character
0	0	5
0	1	7
1	0	6
1	1	8

**Control Register 4**

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
Clock Rate		Sync Mode		Number of Stop Bits per Sync Mode	Parity Even/Odd	Parity Enable	

**Parity enable (D<sub>0</sub>)**

Setting this bit to one adds an extra data bit containing parity information to each transmitted character. Each received character is expected to contain this extra bit and the receiver parity checker is enabled.

**Parity even/odd (D<sub>1</sub>)**

Programming a zero into this bit when parity is enabled causes the transmitted parity bit to take on the value required for odd parity. The received character is checked for odd parity. Conversely, a one in this bit signifies even parity generation and checking.

**Number of stop bits per sync mode (D<sub>2</sub> – D<sub>3</sub>)**

This field specifies whether the channel is used in a synchronous (SDLC) or an asynchronous mode. In an asynchronous mode this field also specifies the number of bit times used as the stop bit length by the transmitter. The receiver always checks for one stop bit.

**Stop Bits**

D <sub>2</sub>	D <sub>3</sub>	Mode
0	0	Synchronous modes
0	1	Asynchronous 1-bit time (1 stop bit)
1	0	Asynchronous 1½ bit times (1½ stop bits)
1	1	Asynchronous 2-bit times (2 stop bits)

**Sync mode (D<sub>4</sub> – D<sub>5</sub>)**

When the stop bits/sync mode field is programmed for synchronous modes (D<sub>2</sub>, D<sub>3</sub> = 00), this field specifies the particular synchronous format to be used. This field is ignored in an asynchronous mode.

**Synchronous Formats**

Sync Mode 1	Sync Mode 2	Mode
D <sub>4</sub>	D <sub>5</sub>	
0	0	8-bit internal synchronization character (monosync)
0	1	16-bit internal synchronization character (bisync)
1	0	SDLC
1	1	External synchronization (SYNC pin becomes an input)

**Clock rate (D<sub>6</sub> – D<sub>7</sub>)**

This field specifies the relationship between the transmitter and receiver clock inputs (TxC, RxC) and the actual data rates at TxD and RxD. When operating in a synchronous mode a 1x clock rate must be specified. In asynchronous modes any of the rates may be specified, however, with a 1x clock rate the receiver cannot determine the center of the start bit. In this mode, the sampling (rising) edge of RxC must be externally synchronized with the data.

**Clock Rates**

Clock Rate 1	Clock Rate 2	Clock Rate
D <sub>7</sub>	D <sub>6</sub>	
0	0	Clock Rate = 1x Data Rate
0	1	Clock Rate = 16x Data Rate
1	0	Clock Rate = 32x Data Rate
1	1	Clock Rate = 64x Data Rate

**Control Register 5**

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
DTR	Number of Transmitted Bits per Character	Send Break	Transmitter Enable	CRC Polynomial Select	RTS	Transmitter CRC Enable	

**Transmitter CRC enable (D<sub>0</sub>)**

A one or a zero enables or disables respectively, the CRC generator calculation. The enable or disable does not take effect until the next character is transferred from the transmitter buffer to the shift register, thus allowing specific characters to be included or excluded from the CRC calculation. By setting or resetting this bit just before loading

the next character, it and subsequent characters are included or excluded from the calculation. If this bit is zero when the transmitter becomes empty the MPSC<sup>2</sup> goes to the idle phase regardless of the state of the idle/CRC latch.

### RTS (D<sub>1</sub>)

In synchronous and SDLC modes setting this bit to one causes the RTS pin to go low while a zero causes it to go high. In an asynchronous mode setting this bit to zero does not cause RTS to go high until the transmitter is completely empty. This feature facilitates programming the MPSC<sup>2</sup> for use with asynchronous modems.

### CRC polynomial select (D<sub>2</sub>)

This bit selects the polynomial used by the transmitter and receiver for CRC generation and checking. A one selects the CRC-16 polynomial ( $x^{16} + x^{15} + x^2 + 1$ ). A zero selects the CRC-CCITT polynomial ( $x^{16} + x^{12} + x^5 + 1$ ). In an SDLC mode CRC-CCITT must be selected. Either polynomial may be used in other synchronous modes.

### Transmitter enable (D<sub>3</sub>)

After a reset the transmitted data output (TxD) is held high (marking) and the transmitter is disabled until this bit is set. In an asynchronous mode TxD remains high until data is loaded for transmission.

In synchronous and SDLC modes the MPSC<sup>2</sup> automatically enters the idle phase and sends the programmed sync or flag characters.

When the transmitter is disabled in an asynchronous mode any character currently being sent is completed before TxD returns to the marking state.

If the transmitter is disabled during the data phase in a synchronous mode the current character is sent. TxD then goes high (marking). In an SDLC mode the current character is sent, but the marking line following is zero-inserted. That is, the line goes low for one bit time out of every five.

The transmitter should never be disabled during the SDLC data phase unless a reset is to follow immediately. In either case, any character in the buffer register is held.

Disabling the transmitter during the CRC phase causes the remainder of the CRC character to be bit-substituted with the sync (or flag). The total number of bits transmitted is correct and TxD goes high after they are sent.

If the transmitter is disabled during the idle phase the remainder of the sync (flag) character is sent. TxD then goes high.

### Send break (D<sub>4</sub>)

Setting this bit to one immediately forces the transmitter output (TxD) low (spacing). This function overrides the normal transmitter output and destroys any data being transmitted although the transmitter is still in operation. Resetting this bit releases the transmitter output.

### Transmitted bits per character (D<sub>5</sub> - D<sub>0</sub>)

This field controls the number of data bits transmitted in each character. The number of bits per character may be changed by rewriting this field just before the first character is loaded to use the new specification.

### Transmitted Bits per Character

Transmitted Bits per Character 1	Transmitted Bits per Character	Bits per Character
D <sub>5</sub>	D <sub>4</sub>	
0	0	5 or less (see below)
0	1	7
1	0	6
1	1	8

Normally each character is sent to the MPSC<sup>2</sup> right-justified and the unused bits are ignored. However, when sending five bits or less the data should be formatted as shown below to inform the MPSC<sup>2</sup> of the precise number of bits to be sent.

### Transmitted Bits per Character for 5 Characters or Less

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Number of Bits per Character
1	1	1	1	0	0	0	D <sub>0</sub>	1
1	1	1	0	0	0	D <sub>1</sub>	D <sub>0</sub>	2
1	1	0	0	0	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	3
1	0	0	0	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	4
0	0	0	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	5

### DTR (data terminal ready) (D<sub>7</sub>)

When this bit is one the DTR output is low (active). Conversely, when this bit is zero DTR is high.

### Control Register 6

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
Sync Byte 1							

### Sync byte 1 (D<sub>0</sub> - D<sub>7</sub>)

Sync byte 1 is used in the following modes:

- Monosync 8-bit sync character transmitted during the idle phase
- Bisync Least significant (first) 8 bits of the 16-bit transmit and receive sync character
- External Sync Sync character transmitted during the idle phase
- SDLC Secondary address value matched to secondary address field of the SDLC frame when the MPSC<sup>2</sup> is in the address search mode

### Control Register 7

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
Sync Byte 2							

### Sync byte 2 (D<sub>0</sub> - D<sub>7</sub>)

Sync byte 2 is used in the following modes:

- Monosync 8-bit sync character matched by the receiver
- Bisync Most significant (second) 8 bits of the 16-bit transmit and receive sync characters
- SDLC The flag character, 01111110, must be programmed into control register 7 for flag matching by the MPSC<sup>2</sup> receiver

**Status Register 0**

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
Break/Abort	Idle/CRC	CTS	Sync Status	DCD	Transmitter Buffer Empty	Interrupt Pending	Received Character Available

**Received character available (D<sub>0</sub>)**

When this bit is set it indicates that one or more characters in the receiver buffer is available for the processor to read. Once all the available characters have been read the MPSC<sup>2</sup> resets this bit until a new character is received.

**Interrupt pending (D<sub>1</sub> — channel A only)**

The interrupt pending bit is used with the interrupt vector register (status register 2) to make it easier to determine the MPSC<sup>2</sup>'s interrupt status, particularly in a nonvectored interrupt mode where the processor must poll each device to determine the interrupt source. In this mode interrupt pending is set when status register 2B is read, the PFI input is active (low), and the MPSC<sup>2</sup> is requesting interrupt service.

The status registers of both channels need not be analyzed to determine if an interrupt is pending. If the status affects vector is enabled and the interrupt pending is set the vector read from SR2 contains valid condition information.

In a vectored interrupt mode interrupt pending is set during the interrupt acknowledge cycle (on the leading edge of the second INTA pulse) when the MPSC<sup>2</sup> is the highest priority device requesting interrupt service (PFI is active). In either mode if there are no other pending interrupt requests interrupt pending is reset when the end of the interrupt command is issued.

**Transmitter buffer empty (D<sub>2</sub>)**

This bit is set whenever the transmitter buffer is empty except during the transmission of CRC. (The MPSC<sup>2</sup> uses the buffer to facilitate this function.) After a reset the buffer is considered empty and transmit buffer empty is set.

**External/status flags (D<sub>3</sub>–D<sub>7</sub>)**

The following status bits reflect the state of the various conditions that cause an external/status interrupt. The MPSC<sup>2</sup> latches all external/status bits whenever a change occurs that would cause an external/status interrupt (regardless of whether this interrupt is enabled). This allows transient status changes on these lines to be captured with relaxed software timing requirements.

When the MPSC<sup>2</sup> is operated in an interrupt-driven mode for external/status interrupts, status register 0 should be read when this interrupt occurs and a reset external/status interrupt command issued to reenables the interrupt and the latches. To poll these bits without interrupts, the reset external/status interrupt command can be issued to first update the status to reflect the current values.

**DCD (D<sub>3</sub>):** This bit reflects the inverted state of the DCD input. When DCD is low the DCD status bit is high. Any transition on this bit causes an external/status interrupt request.

**Sync status (D<sub>4</sub>):** The meaning of this bit depends on the operating mode of the MPSC<sup>2</sup>.

*Asynchronous mode:* Sync status reflects the inverted state of the SYNC input. When SYNC is low, sync status is high. Any transition on this bit causes an external/status interrupt request.

*External synchronization mode:* Sync status operates in the same manner as an asynchronous mode. The MPSC<sup>2</sup>'s receiver synchronization logic is also tied to the sync status bit in an external synchronization mode and a low-to-high transition (SYNC input going low) informs the receiver that synchronization has been achieved and character assembly begins.

A low-to-high transition on the SYNC input indicates that synchronization has been lost and is reflected both in the sync status becoming zero and the generation of an external/status interrupt. The receiver remains in the receive data phase until the enter hunt phase bit in control register 3 is set.

*Monosync, bisync, SDLC modes:* In these modes, sync status indicates whether the MPSC<sup>2</sup> receiver is in the sync hunt or receive data phase of operation. A zero indicates that the MPSC<sup>2</sup> is in the receive data phase and a one indicates that the MPSC<sup>2</sup> is in the sync hunt phase (as after a reset or a setting of the enter sync hunt phase bit). As in the other modes a transition on this bit causes an external/status interrupt to be issued. It should be noted that entering a sync hunt phase after either a reset or when programmed causes an external/status interrupt request which may be cleared immediately with a reset external/status interrupt command.

**CTS (D<sub>5</sub>):** This bit reflects the inverted state of the CTS input. When CTS is low, the CTS status bit is high. Any transition on this bit causes an external/status interrupt request.

**Idle/CRC (D<sub>6</sub>) (Tx underrun/EOM):** This bit indicates the state of the idle/CRC latch used in the synchronous and SDLC modes. After a hardware reset this bit is set to one, indicating that the transmitter is completely empty. When the MPSC<sup>2</sup> enters idle phase it automatically transmits sync or flag characters.

In the SDLC mode the MPSC<sup>2</sup> automatically resets this latch after the first byte of a frame is written to the Tx buffer. When the transmitter is completely empty, the MPSC<sup>2</sup> sends the 16-bit CRC character and sets the latch again. An external/status interrupt is issued when the latch is set, indicating that CRC is being sent. No interrupt is issued when the latch is reset.

**Break/abort (D<sub>7</sub>):** In the asynchronous mode this bit indicates the detection of a break sequence (a null character plus framing error that occurs when the RxD input is held low, spacing, for more than one character time). Break/abort is reset when RxD returns high (marking).

In the SDLC mode, Break/abort indicates the detection of an abort sequence when seven or more ones are received in sequence. It is reset when a zero is received.

Any transition of the break/abort bit causes an external/status interrupt.

**Status Register 1**

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
End of SDLC Frame	CRC Framing Error	Overrun Error	Parity Error	SDLC Residue Code			All Sent

### All sent (D<sub>0</sub>)

This bit is set when the transmitter is empty and reset when a character is present in the transmitter buffer or shift register. This feature simplifies the modem control software routines. In the bit synchronous mode, this bit will be set when the ending flag pattern is sent.

### SDLC residue code (D<sub>1</sub> - D<sub>3</sub>)

Since the data portion of an SDLC message can consist of any number of bits and not necessarily an integral number of characters, the MPSC<sup>2</sup> features special logic to determine and report when the end of frame flag has been received, the boundary between the data field and the CRC character in the last few data characters that were just read. When the end of frame condition is indicated, that is, status register 1 D<sub>7</sub> = 1 and special receive condition interrupt (if enabled), the last bits of the CRC character are in the receiver buffer. The residue code for the frame is valid in the status register 1 byte associated with that data character. (SR1 tracks the received data in its own buffer.)

The meaning of the residue code depends upon the number of bits per character specified for the receiver. The previous character refers to the last character read before the end of frame, and so forth.

### Residue Codes

8 Bits per Character				
D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	Previous Character	2nd Previous Character
1	0	0	C C C C C C C C	C C C C C D D D
0	1	0	C C C C C C C C	C C C C D D D D
1	1	0	C C C C C C C C	C C D D D D D D
0	0	1	C C C C C C C C	C D D D D D D D
1	0	1	C C C C C C C C	C D D D D D D D
0	1	1	C C C C C C C C	D D D D D D D D (no residue)
1	1	1	C C C C C C C D	D D D D D D D D
0	0	0	C C C C C C D D	D D D D D D D D

7 Bits per Character				
D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	Previous Character	2nd Previous Character
1	0	0	C C C C C C C	C C C C C D D
0	1	0	C C C C C C C	C C C C D D D
1	1	0	C C C C C C C	C C C D D D D
0	0	1	C C C C C C C	C D D D D D D
1	0	1	C C C C C C C	C D D D D D D
0	1	1	C C C C C C C	D D D D D D D (no residue)
0	0	0	C C C C C C D	D D D D D D D

6 Bits per Character				
D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	Previous Character	2nd Previous Character
1	0	0	C C C C C C	C C C C C D
0	1	0	C C C C C C	C C C C D D
1	1	0	C C C C C C	C C C D D D
0	0	1	C C C C C C	C C D D D D
1	0	1	C C C C C C	C D D D D D
0	0	0	C C C C C C	D D D D D D (no residue)

5 Bits per Character				
D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	2nd Previous Character	3rd Previous Character
1	0	0	C C C C C	D D D D D (no residue)
0	1	0	C C C C D	D D D D D
1	1	0	C C C D D	D D D D D
0	0	1	C C D D D	D D D D D
0	0	0	C D D D D	D D D D D

### Special receive condition flags

The status bits described below — parity error (if parity as a special receive condition is enabled), receiver overrun error, CRC/framing error, and end of SDLC frame — all represent special receive conditions.

When any of these conditions occur and interrupts are enabled, the MPSC<sup>2</sup> issues an interrupt request. In addition, if a condition affects vector mode is enabled, the vector generated (and the contents of SR2B for nonvectored interrupts) is different from that of a received character available condition. Thus, it is not necessary to analyze SR1 with each character to determine if an error has occurred.

As a further convenience, the parity error and receiver overrun error flags are latched. That is, once one of these errors occurs, the flag remains set for all subsequent characters until reset by the error reset command. With this facility SR1 need only be read at the end of a message to determine if either of these errors occurred anywhere in the message. The other flags are not latched and follow each character available in the receiver buffer.

**Parity error (D<sub>4</sub>):** This bit is set and latched when parity is enabled and the received parity bit does not match the sense (odd or even) calculated from the data bits.

**Receiver overrun error (D<sub>5</sub>):** This error occurs and is latched when the receiver buffer already contains three characters and a fourth character is completely received, overwriting the last character in the buffer.

**CRC/framing error (D<sub>6</sub>):** In the asynchronous mode a framing error is flagged (but not latched) when no stop bit is detected at the end of a character (i.e., RxD is low one bit time after the center of the last data or parity bit). When this condition occurs, the MPSC<sup>2</sup> waits an additional one-half bit time before sampling again so that the framing error is not interpreted as a new start bit.

In the synchronous and SDLC modes this bit indicates the result of the comparison between the current CRC result and the appropriate check value and is usually set to one since a message rarely indicates a correct CRC result until correctly completed with the CRC check character. Note that a CRC error does not result in a special receive condition interrupt.

**End of SDLC frame (EOF) (D<sub>7</sub>):** This status bit is used only in the bit synchronous mode to indicate that the end of frame flag has been received and that the CRC error flag and residue code are valid. This flag can be reset at any time by issuing an error reset command. The MPSC<sup>2</sup> also automatically resets this bit when the first character of the next message frame is sent.

### Status Register 2B

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
Interrupt Vector							

### Interrupt vector (D<sub>0</sub> - D<sub>7</sub> — channel B only)

Reading status register 2B returns the interrupt vector that is programmed into control register 2B. If a condition affects vector mode is enabled the value of the vector is modified as shown in the following table.

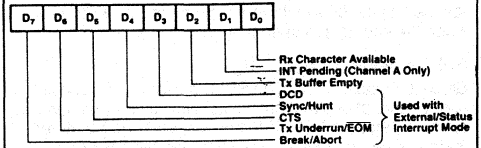
**Condition Affects Vector Modifications**

Interrupt Pending (SP0, D <sub>1</sub> Channel A)	8085 Modes			Condition
	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	
0	1	1	1	No interrupt pending
1	0	0	0	Channel B transmitter buffer empty
1	0	0	1	Channel B external/status change
1	0	1	0	Channel B received character available
1	0	1	1	Channel B special receive condition
1	1	0	0	Channel A transmitter buffer empty
1	1	0	1	Channel A external/status change
1	1	1	0	Channel A received character available
1	1	1	1	Channel A special receive condition

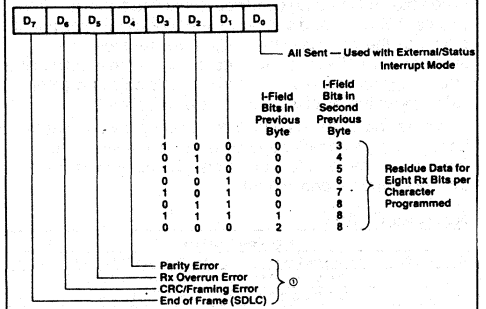
As can be seen code 111 can mean either channel A special receive condition or no interrupt pending. They can be easily distinguished by examining the interrupt pending bit (D<sub>1</sub>) of status register 0, channel A. In a nonvectored interrupt mode the vector register must be read first for the interrupt pending to be valid.

**Read Register Bit Functions**

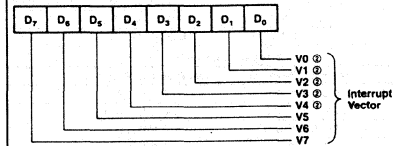
**Read Register 0**



**Read Register 1<sup>⓪</sup>**



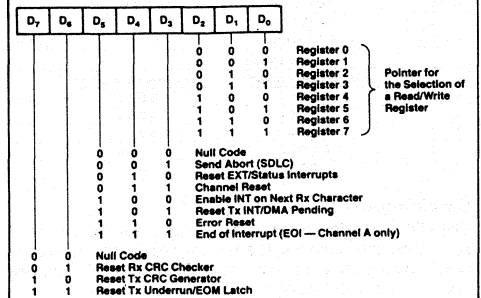
**Read Register 2**



Notes: ⓪ Used with special receive condition mode.  
Ⓛ Variable if Status Affects Vector is programmed.

**Write Register Bit Functions**

**Write Register 0**

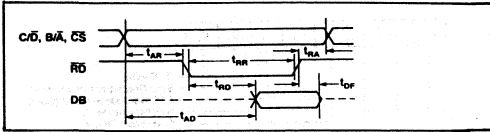




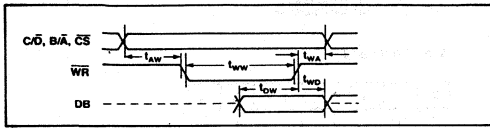


### Timing Waveforms

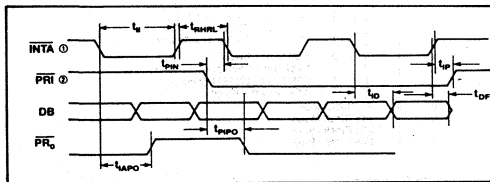
#### Read Cycle



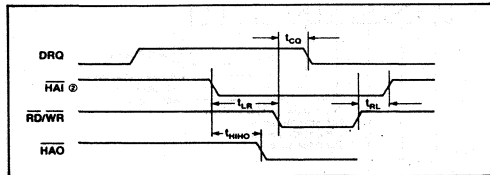
#### Write Cycle



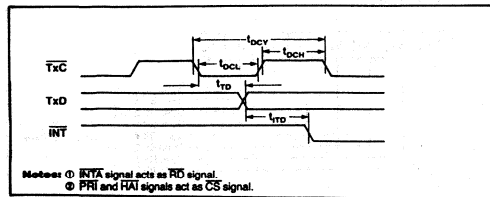
#### INTA Cycle



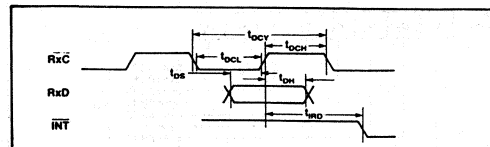
#### DMA Cycle



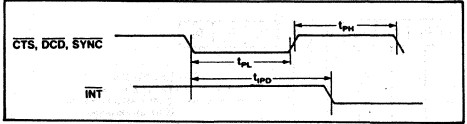
#### Transmit Data Cycle



#### Receive Data Cycle



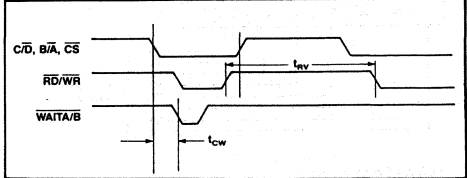
#### Other Timing



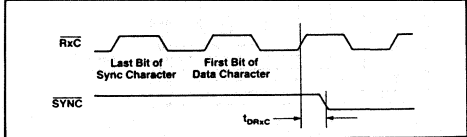
#### Clock



#### Read/Write Cycle (Software Block Transfer Mode)



#### Sync Pulse Generation (External Sync Mode)



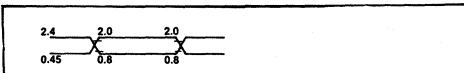
## AC Characteristics

$T_a = 0^\circ\text{C to } +70^\circ\text{C}; V_{CC} = +5V \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Clock Cycle	$t_{CY}$	200	4000	ns		
Clock High Width	$t_{CH}$	70	2000	ns		
Clock Low Width	$t_{CL}$	70	2000	ns		
Clock Rise Time	$t_r$	0	30	ns		
Clock Fall Time	$t_f$	0	30	ns		
Address Setup to RD	$t_{AR}$	0	ns			
Address Hold from RD	$t_{RA}$	0	ns			
RD Pulse Width	$t_{RR}$	200	ns			
Data Output Delay from Address	$t_{AD}$		200	ns		
Data Output Delay from RD	$t_{RD}$		200	ns		
Data Float Delay from RD	$t_{DF}$	10	100	ns		
Address Setup to WR	$t_{AW}$	0	ns			
Address Hold from WR	$t_{WA}$	0	ns			
WR Pulse Width	$t_{WR}$	200	ns			
Data Setup to WR	$t_{DW}$	130	ns			
Data Hold from WR	$t_{WD}$	0	ns			
PRO Delay from PRI	$t_{PPO}$		100	ns		
PRO Delay from INTA	$t_{APO}$		200	ns		
PRI Setup to INTA	$t_{PH}$	0	ns			
PRI Hold from INTA	$t_{PI}$	20	ns			
INTA Pulse Width	$t_{PI}$	200	ns			
Data Output Delay from INTA	$t_{ID}$		200	ns		
Data Float Delay from INTA	$t_{IF}$	10	100	ns		
Request Hold from RD/WR	$t_{CR}$		150	ns		
HAI Setup to RD/WR	$t_{LR}$	300	ns			
HAI Hold from RD/WR	$t_{RL}$	0	ns			
HAI Delay from HAI	$t_{SHO}$		100	ns		
Data Clock Cycle	$t_{DCY}$	400	ns	RxC, TxC		
Data Clock High Width	$t_{DCH}$	180	ns	RxC, TxC		
Data Clock Low Width	$t_{DCL}$	180	ns	RxC, TxC		
Tx Data Delay from TxC	$t_{TD}$		300 1000	ns x1 Mode x16, 32, 64		
Rx Data Setup to RxC	$t_{DS}$	0	ns			
Rx Data Hold from RxC	$t_{DH}$	140	ns			
INT Delay Time from Tx Data	$t_{TD}$		4-6	$t_{CY}$		
INT Delay Time from RxC	$t_{RD}$		7-11	$t_{CY}$		
CTS, DCD, SYNC High Pulse Width	$t_{PH}$	200	ns			
CTS, DCD, SYNC Low Pulse Width	$t_{PL}$	200	ns			
External INT from CTS, DCD, SYNC	$t_{PI}$		500	ns		
Recovery Time Between Controls	$t_{RY}$	300	ns			
WAIT Delay Time from Address	$t_{CW}$		120	ns		
SYNC Setup to RxC	$t_{DRC}$		100	ns		

Notes: 1: RESET must be active for a minimum of one complete CLK cycle.  
2: In all modes system clock rate must be 4.5 times data rate.

## AC Waveform Measurement Points



## μPD7201A Target Specifications

### Absolute Maximum Ratings

$T_a = 25^\circ\text{C}$

Power Supply, $V_{CC}$	-0.5V to +7.0V
Input Voltages, $V_I$	-0.5V to +7.0V
Output Voltages, $V_O$	-0.5V to +7.0V
Operating Temperature, $T_{OPT}$	$0^\circ\text{C to } +70^\circ\text{C}$
Storage Temperature, $T_{STG}$	$-65^\circ\text{C to } +125^\circ\text{C}$

\*COMMENT: Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## DC Characteristics

$T_a = 0^\circ\text{C to } +70^\circ\text{C}; V_{CC} = +5V \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input Low Voltage	$V_{IL}$	-0.5		+0.8	V	
Input High Voltage	$V_{IH}$	+2.0		$V_{CC} + 0.5$	V	
Output Low Voltage	$V_{OL}$			+0.45	V	$I_{OL} = +2.0\text{mA}$
Output High Voltage	$V_{OH}$			+2.4	V	$I_{OH} = 200\mu\text{A}$
Input Leakage Current	$I_{IL}$			$\pm 10$	$\mu\text{A}$	$V_{IN} = V_{CC}$ to 0V
Output Leakage Current	$I_{OL}$			$\pm 10$	$\mu\text{A}$	$V_{OUT} = V_{CC}$ to 0V
$V_{CC}$ Supply Current	$I_{CC}$			230	mA	

## Capacitance

$T_a = 25^\circ\text{C}; V_{CC} = \text{GND} = 0\text{V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input Capacitance	$C_{IN}$			10	pF	$f_c = 1\text{MHz}$
Output Capacitance	$C_{OUT}$			15	pF	Unmeasured pins returned to GND.
I/O Capacitance	$C_{I/O}$			20	pF	



### Introduction

The μPD72001 Advanced Multiprotocol Serial Controller (AMPSC) is a high-performance, single-chip communication controller applicable to a wide range of serial data communication purposes.

The AMPSC, located between a host system and data communication equipment, basically provides parallel-to-serial or serial-to-parallel data conversion according to a predetermined format.

### Features

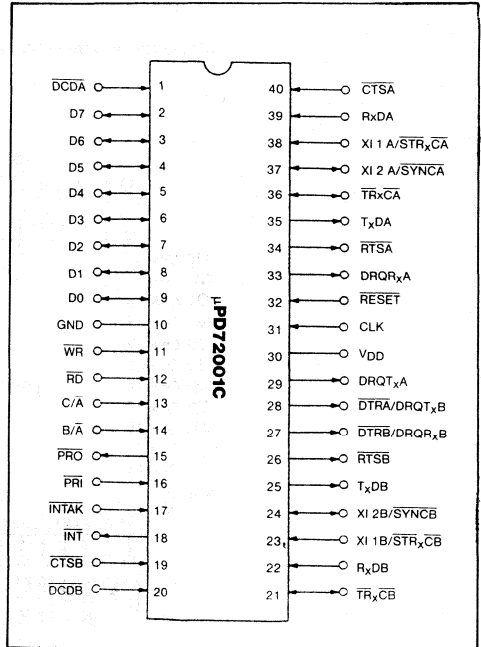
- LSI to high-performance data communications
- Multiprotocol operations:
  - Asynchronous
  - Character-oriented protocol (COP)
  - Bit-oriented protocol (BOP)
- Two full-duplex channels
- Baud rate: DC to 1.6 Mbps (−11 : 2.1 Mbps)
- Transmitter: Double buffer
- Receiver: Quadruple buffer
- Interrupt control capability
- DMA request signal: Transmit and Receive request/channel
- Overrun error detection
- Data formats: NRZ, NRZI and FM
- Local self test capability
- On-chip crystal oscillator
- On-chip DPLL circuit
- On-chip baud rate generator
- General-purpose input/output pins: 4 pins per channel
- Standby function
- System clock rate: 8MHz max. (−11 : 11 MHz)
- CMOS technology
- 40-pin DIP
- Single +5 V power supply
- Asynchronous operation
  - Character bit length: 5, 6, 7, or 8 bits
  - Stop bit length: 1, 1.5 or 2 bits
  - Clock rate: x1, x16, x32, or x64
  - Parity generation/checking
  - Framing error detection
  - Break generation/detection
- Character-oriented protocol (COP) operation
  - Mono-sync, Bi-sync, or External sync mode
  - Character bit length: 5, 6, 7, or 8 bit
  - Character synchronization: Internal or external
  - SYNC character bit length: 6 or 8 bits
  - BCS generation/checking (CR-16, CRC-CCITT)
  - Parity generation/checking
  - Automatic SYNC character transmission/checking/deletion
- Bit-oriented protocol (BOP) operation
  - HDLC, SDLC, or SDLC Loop mode

- Flag transmission/detection
- Zero insertion/suppression
- Address field detection
- FCS generation/check (CCITT-1)
- Short frame detection
- Automatic abort transmission/detection
- Idle detection
- Go-Ahead detection
- Transmit data count control (16-bit transmit length register/counter)

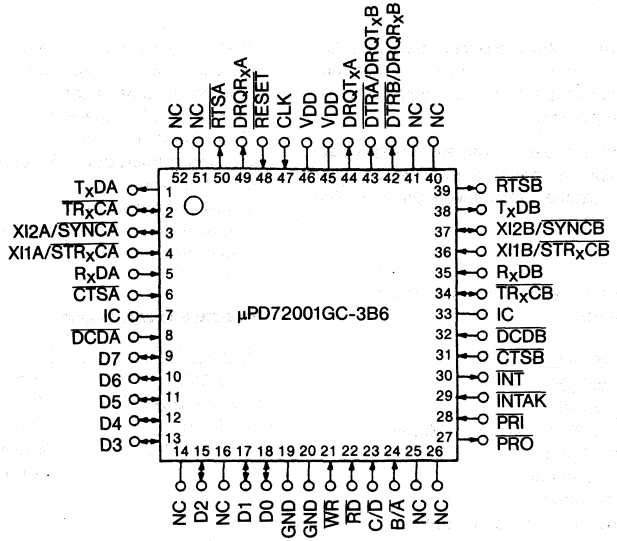
### Ordering Information

Part Number	Package Type
μPD72001C	40-pin plastic DIP (600mil)
μPD72001GC-3B6	52-pin plastic flat (bent lead)
μPD72001L	52-pin PLCC
μPD72001C-11	40-pin plastic DIP (600mil) max. freq. 11 MHz

### Pin Configuration (Top View)

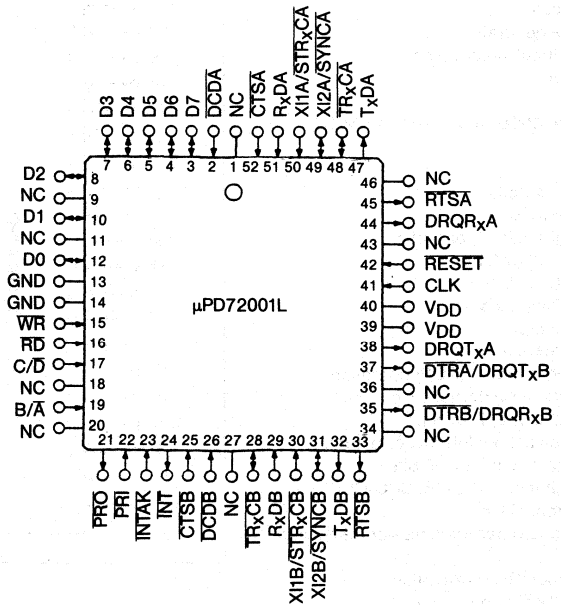


## μPD72001

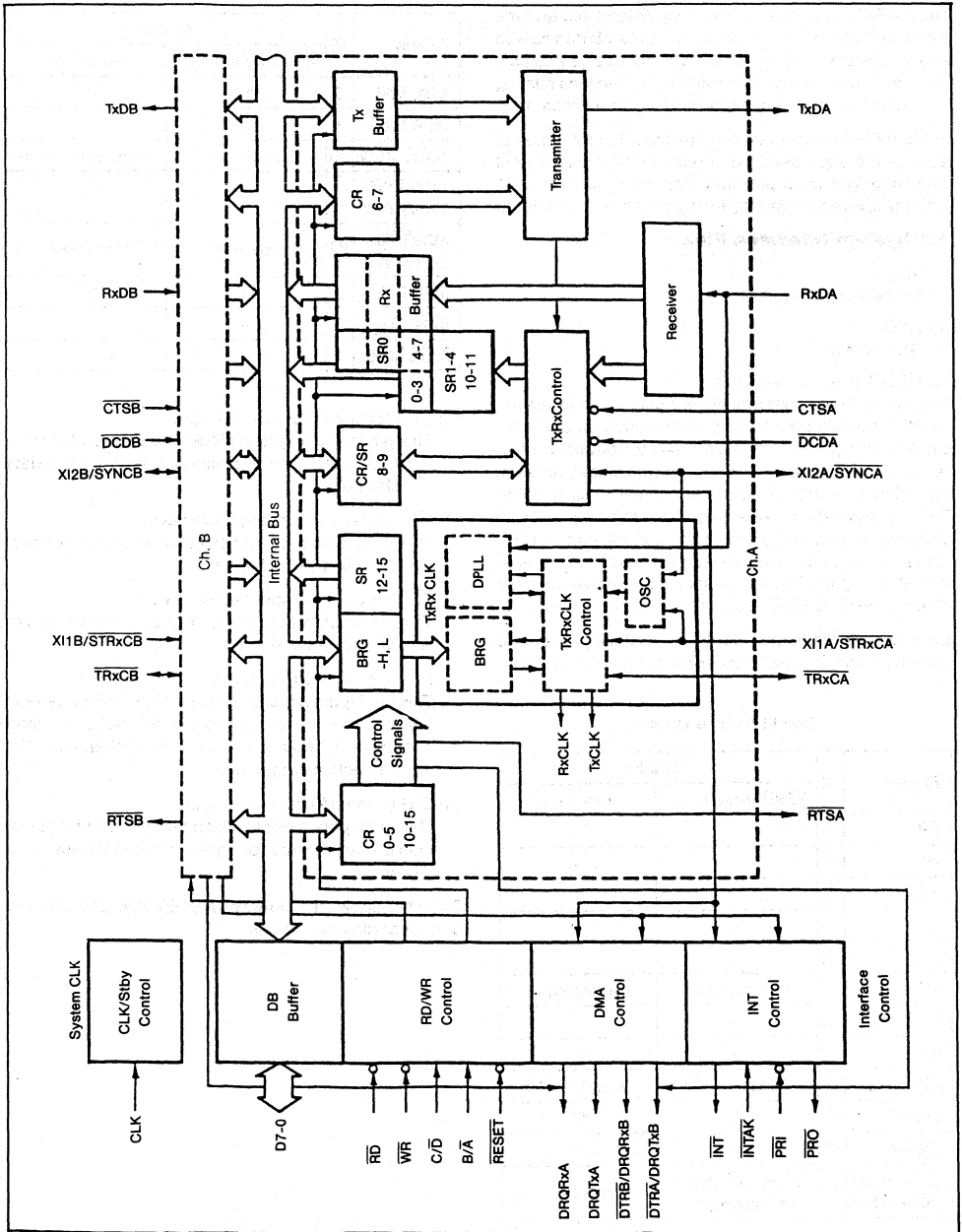


NC: Non-Connection

IC: Internally Connection (Leave unconnected)



## Block Diagram



### 1. Pin Function

The AMPSC functions are basically divided into two: the system interface function which controls interfacing with a host system, and the data transmit/receive function. This section describes the AMPSC pin functions relating to system interface and send/receive control separately.

In the following descriptions, the input/output status of each pin is expressed, as a rule, by "H" (voltage level satisfying  $V_{IH}$  for input, and  $VOH$  for output) and "L" (voltage level satisfying  $V_{IL}$  for input, and  $VOL$  for output).

#### 1.1 System Interface Pins

(1) VDD  
Power supply input pin.

(2) GND  
Ground pin.

(3) RESET (Reset) --- Input  
This pin inputs an external Reset signal which resets the AMPSC. Applying a "L" signal continuously for 2 or more clock cycles ( $2t_{cy}$ ) to this pin resets the device (system reset). A system reset causes the transmitter, receiver, and interrupt and DMA functions to be disabled, and the Tx/D and general-purpose output pins to be set to "H". It also resets (to zeros) all bits of the control register (CR). Therefore, once a system reset is executed, the CR must be set up again. Table 1-1 shows the pin status after channel reset (CR0: D5, D4, D3 = 0, 1, 1).

Upon system reset, the AMPSC automatically enters the standby mode, in which power consumption is reduced.

Table 1-1 Pin Status after Reset

Pin name	I/O	Pin status	
		RESET (system reset)	Channel reset
WR	I	-	-
RD	I	-	-
B/A	I	-	-
C/D	I	-	-
D7-D0	I/O	-	-
INT	O	High impedance	High impedance
INTAK	I	-	-
PRI	I	-	-
PRO	O	Depends on PRI	Depends on PRI
DRQTxA	O	"L"	"L"
DRQRxA	O	"L"	"L"
DTRA/DRQTxB DTRE/DRQRxB	O	Enters DTR function and becomes "H"	Holds status

Pin name	I/O	Pin status	
		RESET (system reset)	Channel reset
TxDA, TxDB	O	"H"	"H"
RxDA, RxDB	I	-	-
TRxCA, TRxCB	I/O	Input state	Holds current status
X11A/STRxCA X11B/STRxCB	I	-	-
X12A/SYNCA X12B/SYNCB	I/O	Input state	Input state
RTSA, RTSB	O	"H"	"H"
CTSA, CTSB	I	-	-
DCDA, DCDB	I	-	-

- (4) CLK (System Clock) --- Input  
This pin inputs a system clock. The system clock rate must be more than five times as high as the data transfer rate.
- (5) WR (Write) --- Input (active Low)  
This pin inputs a control signal used to write control words or Tx data to the device.
- (6) RD (Read) --- Input (active Low)  
This pin inputs a control signal used to read status or Rx data out of the device.
- (7) B/A (Channel B/Channel A) --- Input  
This pin inputs a channel select signal which selects the channel to be accessed for write or read operation. "L" input to this pin selects channel A; "H" input selects channel B.
- (8) C/D (Control/Data) --- Input  
This pin inputs a signal which determines the type of data on the data bus during read or write access to the device.

Table 1-2 shows the status of WR, RD, B/A, and C/D and corresponding operations.

**Table 1-2 AMPSC Control Signals Versus Operations**

WR	RD	B/A	C/D	Operation	
L	H	L	L	Channel A	Writes transmit data to Tx buffer
		H	L	Channel B	
H	L	L	L	Channel A	Reads receive data from Rx buffer
		H	L	Channel B	
L	H	L	H	Channel A	Writes to control register
		H	H	Channel B	
H	L	L	H	Channel A	Reads status register
		H	H	Channel B	
H	H	X	X	High-impedance state	
L	L	X	X	Use inhibited	

X: Don't Care

(9) D7-D0 (Data Bus) --- Input/Output

These pins constitute a three-state, eight-bit, bidirectional data bus. This bus is connected to the host processor's data bus to transfer control words, status, and transmit/receive data.

(10) INT (Interrupt) --- Output (open drain)

This pin outputs the interrupt request signal. This pin is set to active "L" if an interrupt condition occurred within the AMPSC. Being an open-drain output, this pin requires a pull-up resistor.

(11) INTAK (Interrupt Acknowledge) --- Input (active Low)

This pin inputs an acknowledge signal returned in response to an interrupt request. This pin is used to select the Vector mode (CR2A: D7 = 1). When selecting the Non-Vector mode (CR2A: D7 = 0), this pin must be pulled up to "H"

(12) PRI (Priority Input) --- Input (active Low)

This pin inputs a signal which controls interrupt request generation and interrupt vector output. This pin serves to interrupt generation control during general operation, and to interrupt vector output control during the INTAK sequence. The handling of this pin depends on the interrupt mode.

(a) Vectored mode (D7 of CR2A = 1)

The PRI pin usually controls the generation of interrupts. If Type A-3 or Type B-2 (D5, D4, D3 of CR2A = 0, 1, 0 or 1, 0, 0) interrupt vector output mode is selected, the PRI pin can be set to either "H" or "L". If any other interrupt vector output mode is selected, the PRI pin must be set to "L" to enable interrupt request.

During the INTAK sequence, input of "L" level to the PRI pin causes the device to enable interrupt

vector output in any interrupt vector output mode. If the PRI pin is set to "H", interrupt vector output is disabled.

(b) Non-Vectored mode (D7 of CR2A = 0)

Since this mode has no INTAK sequence, the PRI pin serves only for interrupt generation control. If a vector output mode other than Type A-3 and Type B-2 is selected, input of "L" level to the PRI pin enables interrupt generation. If the pin is set to "H", interrupt generation is disabled.

When a daisy chain is configured for interrupt, "L" input to this pin indicates that the interrupt from a device with higher priority is not serviced or the device is not in request for interrupt service. Only the AMPSC with "L" applied to this pin can request interrupt.

(13) PRO (Priority Output) --- Output (active Low)

This pin is used to configure an interrupt daisy chain. It controls interrupt requests from a device with a lower order of priority. It is usually used with the PRI pin, and is set to the following status depending on the PRI status:

If PRI = "H", PRO = "H".

If PRI = "L", PRO = "H" provided an interrupt request is present, and PRO = "L" provided no interrupt request is present.

(14) DRQTxA (DMA Request TxA) --- Output (active High)

This is a DMA request output to the DMA controller. This pin is set to "H" when the Tx buffer in transmitter channel A is emptied; The conditions under which this pin is set to "H" differ depending on the status of D2 of CR1:

CR1A: D2=0: DRQTxA is set to "H" when the Tx buffer is empty after the first Tx data was written into the buffer. If remains at "L" when the buffer is empty by re-set operation

CR1A: D2=1: DRQTxA is set to "H" when the Tx buffer is empty.

The DRQTxA is reset when transmit data is written into channel A.

(15) DRQRxA (DMA Request RxA) --- Output (active High)

This is a DMA request output to the DMA controller. This pin is set to "H" when the receiver on channel A enters the Rx Character Available state. It is reset when received data is read out of channel A.



(16)  $\overline{DTRA}$ /DRQTxB (Data Terminal Ready A/DMA Request TxB) --- Output  
 This pin serves the following two functions depending on the status of bits D1 and D0 of CR2A.

(a) When CR2A: D1, D0 = 0, 0 or 0, 1  
 This pin functions as  $\overline{DTRA}$ , which is a general-purpose output usable for modem control or other purposes. The  $\overline{DTRA}$  pin status is as follows:

When CR5A: D7 = 0:  $\overline{DTRA}$  = "H"  
 When CR5A: D7 = 1:  $\overline{DTRA}$  = "L"

(b) When CR2A: D1, D0 = 0, 1  
 This pin functions as a DRQTxB output. Its function is identical to that of the DRQTxA pin, with the exception that the former is used on channel B.

(17)  $\overline{DTRB}$ /DRQRxB (Data Terminal Ready B/DMA request RxB) --- Output  
 This pin serves the following two functions depending on the status of bits D1 and D0 of DR2A.

(a) When D1, D0 of CR2A = 0, 0 or 0, 1  
 This pin serves as the  $\overline{DTRB}$  output. Its function is identical to that of the  $\overline{DTRA}$  pin, with the exception that the former is used on channel B.

(b) When D1, D0 of CR2A = 1, 0  
 This pin serves as the DRQRxB output. Its function is identical to that of the DRQRxA pin, with the exception that the former is used on channel B.

(18)  $\overline{CTS_A}$  (Clear to Send A),  $\overline{CTS_B}$  (Clear to Send B) --- Inputs  
 These pins are general-purpose inputs usable for modem control or other purposes. A status change on these pins affects E/S bit latch operation. If E/S INT is enabled (D0 of CR1 set to 1), an E/S interrupt occurs. When the Auto Enable mode is selected (D5 of CR3 set to 1), these pins can be used with the Tx Enable bit (D3 of CR5) to control the transmitter status. This is shown in Table 1-3.

Table 1-3 Auto Enable Mode and CTS Pin

CTS pin	Tx Enable bit	Transmitter status
L	1	Enable
H	1	Disable
H or L	0	Disable

(19)  $\overline{DCDA}$  (Data Carrier Detect A) --- Input  
 $\overline{DCDB}$  (Data Carrier Detect B) --- Input  
 These are general-purpose input pins usable for modem control or other purposes. A status change on these pins affects E/S bit latch operation. If E/S INT is enabled (D0 of CR1 set to 1), an E/S interrupt occurs.

If the Auto Enable mode is selected (D5 of CR3 set to 1), these pins can be used with Rx Enable bit (D0 of CR3) to control the receiver status. This is shown in Table 1-4.

Table 1-4 Auto Enable Mode and DCD Pin

$\overline{DCD}$ pin	Rx Enable bit	Receiver status
L	1	Enable
H	1	Disable
H or L	0	Disable

(20) RTS<sub>A</sub> (Request to Send A) --- Output  
 RTS<sub>B</sub> (Request to Send B) --- Output  
 These are general-purpose output pins usable for modem control or other purposes. The status of these pins depends on the operation protocol setting and Auto Enable bit status, which is summarized in Table 1-5.

**Table 1-5 Auto Enable Bit and RTS Pin**

Protocol	function	Auto Enable bit	RTS bit (CR5: D1)	RTS pin status
Asynchronous	0	0	0	H
			1	L
	1	0	0	Remains "L" until All Sent = 1 and then becomes "H"
			1	L
COP/BOP	0 or 1	0	0	H
			1	L

**1.2 Pins Relating to Transmit/Receive Operations**

- (1) TxDA (Transmit Data A), TxDB (Transmit Data B) --- Outputs  
These are Tx data output pins.
- (2) RxDA (Receive Data A), RxDB (Receive Data B) --- Inputs  
These are Rx data input pins.
- (3) XI1A/STRxC $\overline{A}$  (Crystal Input 1A/Source of Transmit Receive Clock A)  
XI1B/STRxC $\overline{B}$  (Crystal Input 1B/Source --- Inputs of Transmit Receive Clock B)  
The function of these pins depends on the status of D7 of CR15:
  - (a) When D7 of CR15 is 0  
These pins function as STRxC, which inputs a transmit/receive clock or a source clock for the onchip baud rate generator (BRG) or digital phase locked loop (DPLL).
  - (b) When D7 of CR15 is 1  
These pins function as XI1, which, along with pin XI2, connects with a crystal resonator for TxRx CLK source oscillation.
- (4) XI2A/SYNCA (Crystal Input 2A/Synchronization A)  
XI2B/SYNCB (Crystal Input 2B/Synchronization B) --- Inputs  
The function of these pins depends on the status of D7 of CR15:
  - (a) When D7 of CR15 is 0  
These pins function as SYNC, which provides the functions as shown in Table 1-6 depending on the setting of CR4.
  - (b) When D7 of CR15 is 1  
These pins function as XI2, which, along with pin XI1, connects with a crystal resonator for TxRx CLK source oscillation.

**Table 1-6 SYNC Pin functions versus CR4 Value (D7 of CR15 = 0)**

Operation Protocol	Sync detection method	SYNC pin function	CR4						Function	
			D7	D6	D5	D4	D3	D2		
Async		Input	X		X			0 1 1	1 0 1	Functions as a general-purpose input. Status changes at this pin ("H" to "L" or "L" to "H") affect Sync/Hunt bit (D4 of SR1) latch operation, and cause an E/S interrupt.
COP	Internal sync	Output	X		0 0	0 1		0 0	0 0	If a SYNC character is detected in the received data, this pin is set to "L" for 1 RxC cycle period.
	External sync	Input	0 0	0 1	0 1					Accepts character synchronization signal. If this pin is set from "H" to "L", the device exits the Hunt phase and establishes character synchronization. Character synchronization is maintained while the SYNC input is at "L". Receive character assembling is initiated at the rising edge of the receive clock which precedes the falling edge of the SYNC input.
BOP		No function	X	1	0					SYNC pin does not function.

X: Don't Care

- (5) TRxCA (Transmit Receive Clock A) --- Input, Output  
TRxCB (Transmit Receive Clock B) --- Input, Output  
 (a) If D2 of CR15 = 0

These pins function as transmit/receive clock inputs. They input an external transmit/receive clock.

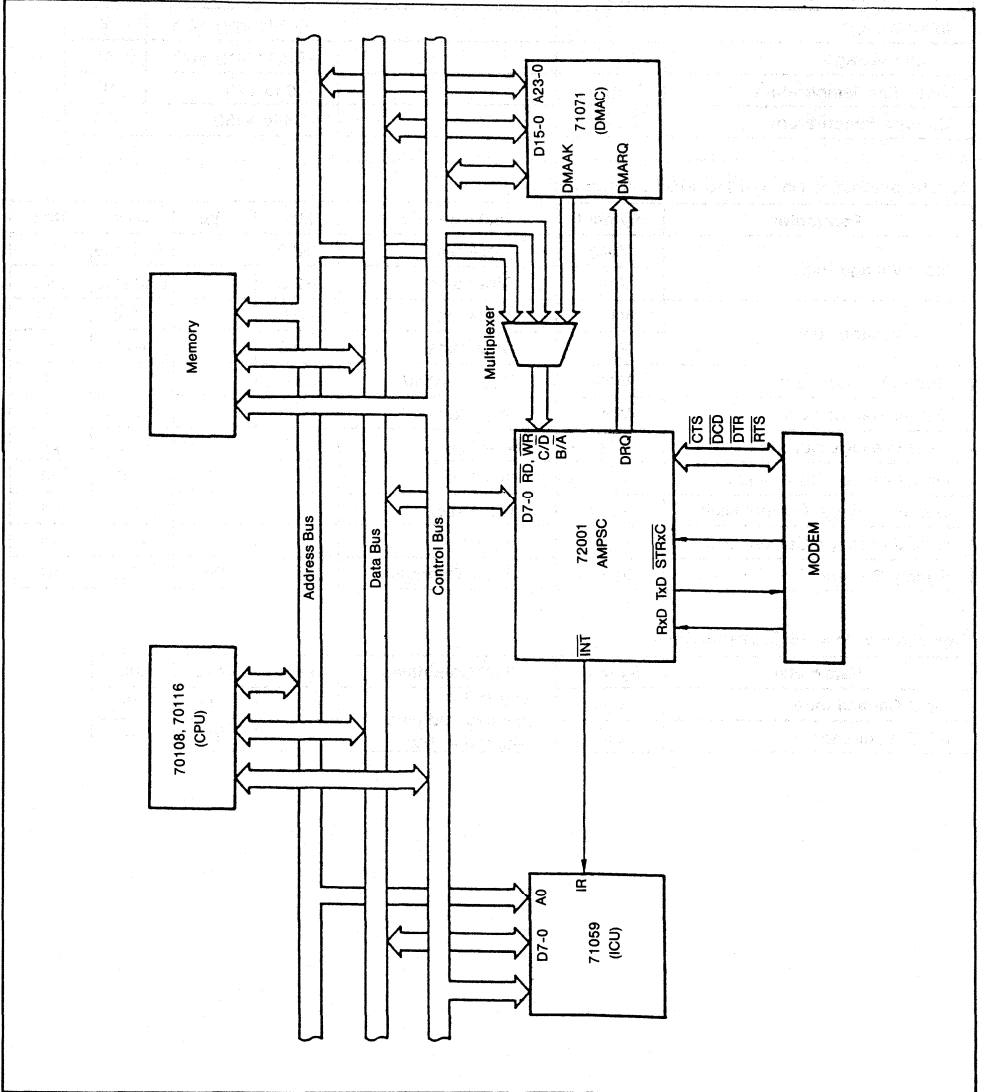
(Exception): If at least one of the conditions, (D6, D5 Of CR15 = 0, 1) or (D4, D3 of CR15 = 0, 1) is satisfied, these pins function as inputs even if D2 of CR15 set to 1.

- (b) When bit D2 of CR15 = 1

These pins function as outputs. The output clock source is selectable from a crystal oscillator, BRG, DPLL, and transmit clock depending on the setting of D1 and D0 of CR15. However, when the condition given in the (Exception) of the above item (a) is fulfilled, these pins unconditionally function as inputs, in which case the setting of D2, D1, and D0 of CR15 is invalid.

## 2. System Configuration Example

The following figure shows a system configuration example in which the μPD72001 is interfaced with a modem, with both channels placed in the DMA mode.



**3. Electrical Specification**

Absolute Maximum Ratings (Ta = 25°C)

Parameter	Symbol	Test Conditions	Ratings	Unit
Power Supply Voltage	VDD		-0.5 to +7.0	V
Input Voltage	VI		-0.5 to VDD +0.5	V
Output Voltage	VO		-0.5 to VDD +0.5	V
Operating Temperature	Topt		-10 to +70	°C
Storage Temperature	Tstg		-65 to +150	°C

DC Characteristics (Ta = -10 to +70°C, VDD = 5V±10%)

Parameter	Symbol	Test Conditions	Min.	Typ	Max	Unit
Input Voltage High	VIHC	Clock pin	+3.3		VDD+0.5	V
	VIH	Other pins	+2.2		VDD+0.5	V
Input Voltage Low	VILC	Clock pin	-0.5		+0.6	V
	VIL	Other pins	-0.5		+0.8	V
Output Voltage High	VOH	IOH = -400μA	0.7VDD			V
Output Voltage Low	VOL	IOL = 2.0 mA			+0.45	V
Input Leakage Current High	ILIH	VI = VDD			+10	μA
Input Leakage Current Low	ILIL	VI = 0 V			-10	μA
Output Leakage Current High	ILOH	VO = VDD			+10	μA
Output Leakage Current Low	ILOL	VO = 0 V			-10	μA
Supply Current	IDD	8MHz Operation		20	40	mA

Capacitance (Ta = 25°C, VDD = 0V)

Parameter	Symbol	Test Conditions	Min.	Max	Unit
Input Capacitance	CIN	fC = 1MHz Unmeasured pins returned to 0V.		10	pF
I/O Capacitance	CIO			20	pF

AC Characteristics (Ta = -10 to +70°C, VCC = +5V ± 10%)  
System Interface

Parameter	Symbol	Test Conditions	Standard value		Unit
			Min.	Max	
Clock Cycle	t <sub>CYK</sub>		125	2000	ns
Clock high-level width	t <sub>WKH</sub>		50	1000	ns
Clock low-level width	t <sub>WKL</sub>		50	1000	ns
Clock rise time	t <sub>KR</sub>	1.5V to 3.0V		10	ns
Clock fall time	t <sub>KF</sub>	3.0V to 1.5V		10	ns
Address setup time to $\overline{RD} \downarrow$	t <sub>SAR</sub>		0		ns
Address hold time from $\overline{RD} \uparrow$	t <sub>HRA</sub>		0		ns
$\overline{RD}$ pulse width	t <sub>WRL</sub>		150		ns
Data output delay time from address	t <sub>DAD</sub>			120	ns
Data output delay time from $\overline{RD}$	t <sub>DRD</sub>			120	ns
Data float delay time from $\overline{RD}$	t <sub>FRD</sub>		10	85	ns
Address setup time to $\overline{WR} \downarrow$	t <sub>SAW</sub>		0		ns
Address hold time from $\overline{WR} \uparrow$	t <sub>HWA</sub>		0		ns
$\overline{WR}$ pulse width	t <sub>WWL</sub>		150		ns
Data setup time to $\overline{WR} \uparrow$	t <sub>SDW</sub>		120		ns
Data hold time from $\overline{WR} \uparrow$	t <sub>HWD</sub>		0		ns
$\overline{RD}/\overline{WR}$ recovery time	t <sub>RV</sub>		160		ns

5

Serial control

Parameter	Symbol	Test Conditions	Standard value		Unit
			Min	Max	
Transmit/Receive data cycle	t <sub>CYD</sub>		5		t <sub>CYK</sub>
$\overline{STRx\overline{C}}$ , $\overline{TRx\overline{C}}$ input clock cycle	t <sub>CYC</sub>		125		ns
$\overline{STRx\overline{C}}$ , $\overline{TRx\overline{C}}$ input clock pulse width	High	t <sub>WCH</sub>	50		ns
	Low	t <sub>WCL</sub>	50		ns
Tx $\overline{D}$ delay time from $\overline{STRx\overline{C}}$ , $\overline{TRx\overline{C}} \downarrow$	t <sub>DTCTD1</sub>	x 1 mode		100	ns
	t <sub>DTCTD2</sub>	x 16, 32, 64 mode		300	ns
RxD setup time to $\overline{STRx\overline{C}}$ , $\overline{TRx\overline{C}} \uparrow$	t <sub>SRDRC</sub>		0		ns
RxD hold time from $\overline{STRx\overline{C}}$ , $\overline{TRx\overline{C}} \uparrow$	t <sub>HRCRD</sub>		140		ns
$\overline{INT}$ delay time from Tx $\overline{D}$	t <sub>TDIQ</sub>	Tx INT mode	4	6	t <sub>CYK</sub>
DRQ <sub>Tx</sub> delay time from Tx $\overline{D}$	t <sub>TDDDQ</sub>	Tx DMA mode	4	6	t <sub>CYK</sub>
$\overline{INT}$ delay time from Rx $\overline{C}^* \uparrow$	t <sub>DRCIQ</sub>	Rx INT mode	7	11	t <sub>CYK</sub>
DRQ <sub>Rx</sub> delay time from Rx $\overline{C}^* \uparrow$	t <sub>DRCDQ</sub>	Rx DMA mode	7	11	t <sub>CYK</sub>
DRQ <sub>Rx</sub> delay time from $\overline{RD} \downarrow$	t <sub>DRDQ</sub>			120	ns
DRQ <sub>Tx</sub> delay time from $\overline{WR} \uparrow$	t <sub>DWDQ</sub>			120	ns

\*:  $\overline{STRx\overline{C}}$  or  $\overline{TRx\overline{C}}$ , which is used for the receive clock.

Interrupt control

Parameter	Symbol	Test Conditions	Standard value		Unit
			Min	Max	
INTAK low-level width	tWIAL		150		ns
PRO delay time from PRI	tDPIPO			50	ns
PRI setup time to INTAK ↓	tSPIIA	When vector output is permitted	0		ns
PRI hold time from INTAK ↑	tHIAPI	When vector output is permitted	20		ns
Data output delay time from INTAK	tDIAD			120	ns
Data float delay time from INTAK	tFIAD		10	85	ns

Modem control

Parameter	Symbol	Test Conditions	Standard value		Unit
			Min	Max	
CTS, DCD, SYNC pulse width	High	tWMH	2		tCYK
	Low	tWML	2		tCYK
INT delay time from CTS, DCD, SYNC	tDMIQ			2	tCYK
SYNC delay time from STRxC, TRxC ↑	tDTRKSY	COP external synchronization	0	2	tCYK

Crystal oscillation and reset

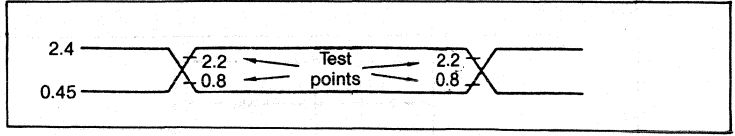
Parameter	Symbol	Test Conditions	Standard value		Unit
			Min	Max	
XI1 Input cycle time	tCYX		125	1000	ns
RESET pulse width	tWRSL		2		tCYK

Note: At all modes, system clock cycle must be more than five times of data rate.

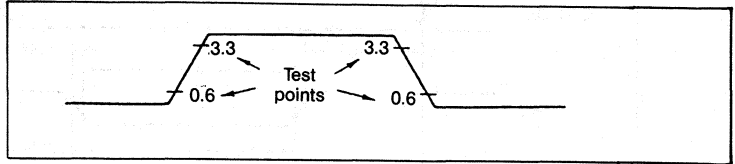
Difference of 72001-11:

Parameter	Symbol	Test Conditions	Standard value		Unit
			Min.	Max	
Clock Cycle	tCYCK		91	2000	ns
Clock high-level width	tWKH		40	1000	ns
Clock low-level width	tWKL		40	1000	ns

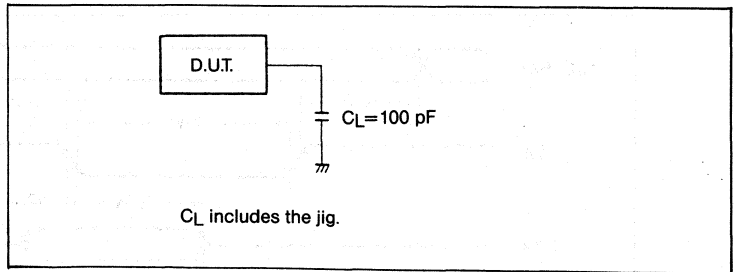
AC test I/O wave (except clock)



AC test clock input wave



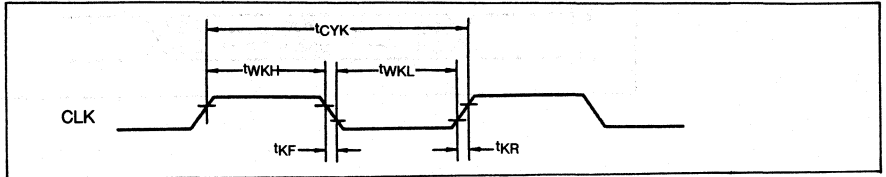
Load circuit for AC test



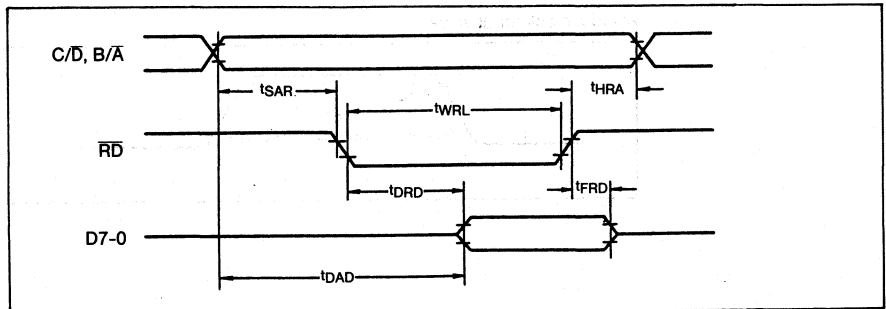


### Timing Waveform

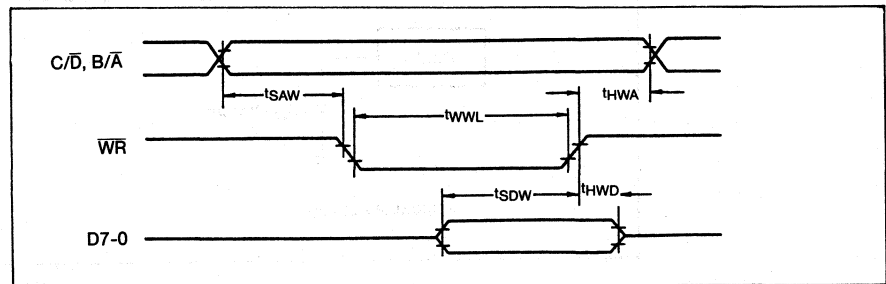
#### Clock timing



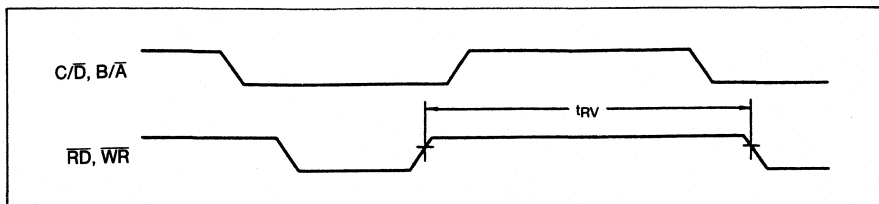
#### Read cycle timing



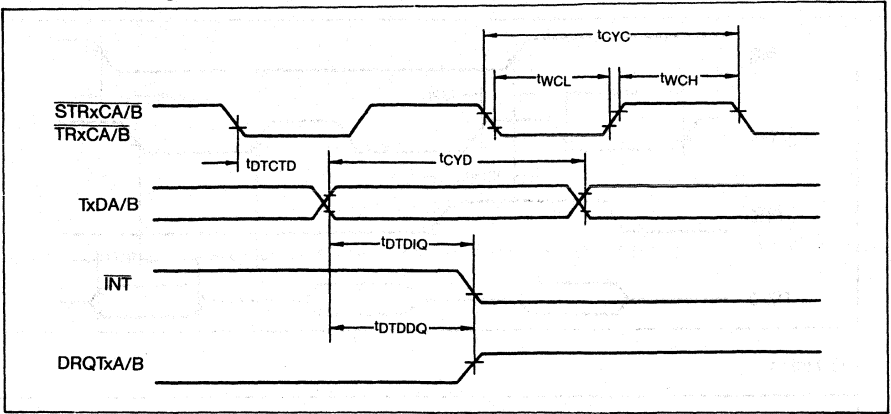
#### Write cycle timing



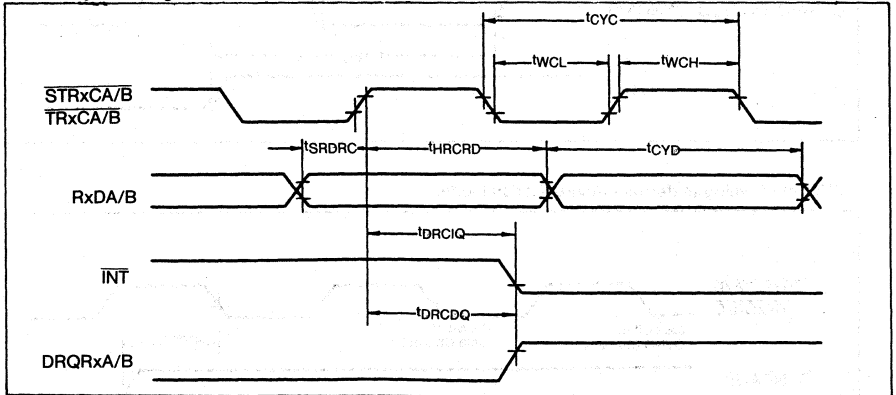
#### Read/Write cycle timing (data transfer by software)



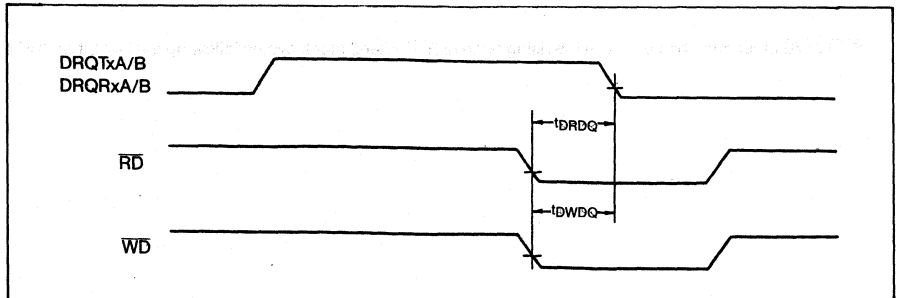
### Transmit cycle timing



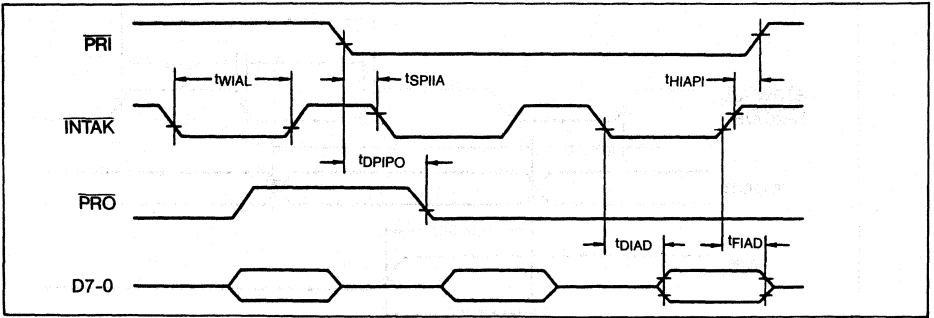
### Receive cycle timing



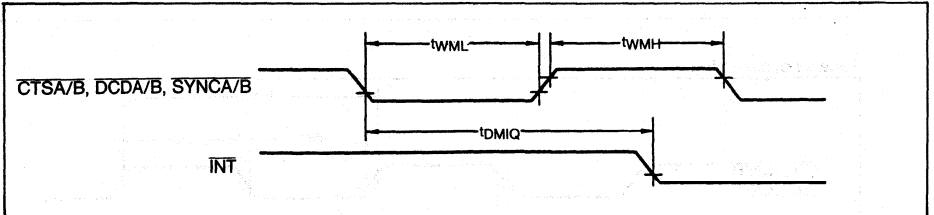
### DMA request reset timing



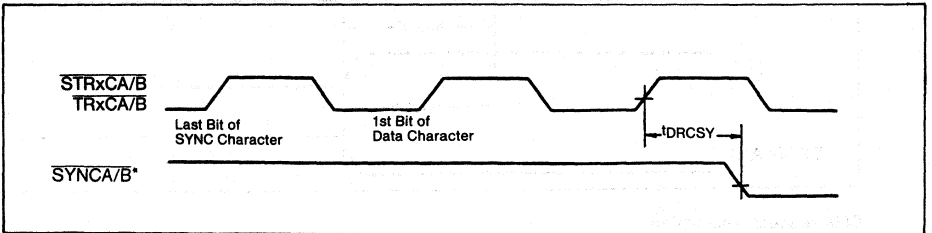
### INTAK cycle timing



### E/S timing

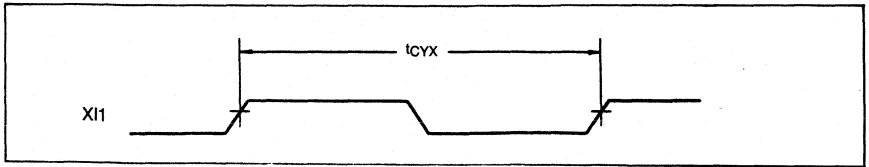


### SYNC input timing (external synchronization mode)

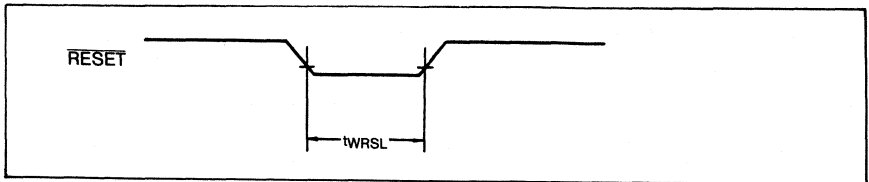


\*: SYNCA/B input must be set to 0 at the rising edge of RxC after 2 clock cycles following the last bit of SNC character.

XI1 input timing



RESET pulse





## Description

The μPD72105 provides local area network (LAN) communications implementing the OMNINET® I and II protocols in a single CMOS 48-pin DIP. The device can transmit data at a rate of up to 4 Mb/s using RS-422 bus transmitters and receivers. The controller responds to 17 OMNINET commands using the on-chip CPU.

The chip also contains a DMA controller with four independent channels for use with an 8- or 16-bit data bus, and can address a 16M-byte address space. The transmit section contains a 12-byte FIFO and the receiver contains a 20-byte FIFO to accommodate the high data rate. The OMNINET controller provides network diagnostics capability as well as CRC generation and checking using a 16- or 32-bit CRC for data reliability.

The μPD72105 provides a single chip solution to LAN implementation. The excellent memory addressing and data handling capability of this controller can significantly reduce the overhead on the system CPU.

OMNINET is a registered trademark of Corvus Systems.

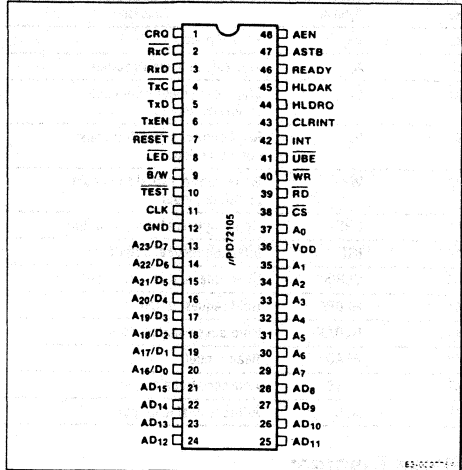
## Features

- Fully implements OMNINET I and II protocols
- Data rates up to 4 Mbps
- 17 OMNINET commands
- On-chip CPU
- On-chip DMAC with four independent channels
- 8- or 16-bit data bus
- 16M-byte (2<sup>24</sup>) address space for dual-ported local or global memory
- 12-byte transmitter FIFO
- 20-byte receiver FIFO
- 16- or 32-bit CRC
- On-chip 40-MHz DPLL
- Network diagnostics
- 8-MHz system clock input, independent of serial clock
- CMOS technology

## Ordering Information

Part No.	Package Type
μPD72105C	48-pin plastic DIP
μPD72105L	52-pin PLCC

## Pin Configuration



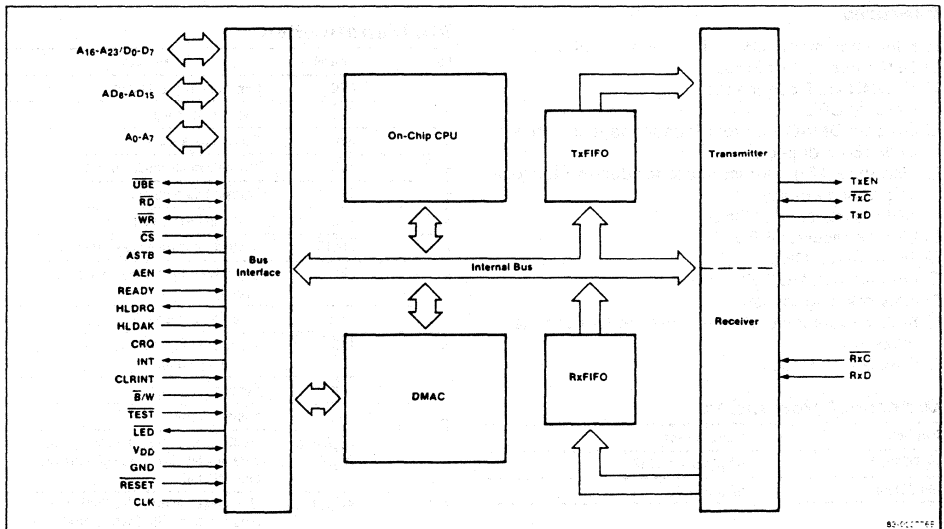
## Pin Identification

No.	Symbol	Function
1	CRQ	Command request input
2	RxC	Receive clock input
3	RxD	Receive data input
4	TxC	Transmit clock input/output
5	TxD	Transmit data output
6	TxEN	Transmit enable output
7	RESET	System reset from host computer
8	LED	LED drive output, general purpose output
9	B/W	Byte/word mode select input
10	TEST	Test input, must be held high for normal operation
11	CLK	System clock input
12	GND	System ground
13-20	A <sub>23</sub> /D <sub>7</sub> to A <sub>16</sub> /D <sub>0</sub>	Multiplexed address bits 16-23 and data bus bits 0-7. These signals are bidirectional.
21-28	AD <sub>15</sub> /AD <sub>8</sub>	Multiplexed address bits 8-15 and data bus bits 8-15. These signals are bidirectional.
29-35	A <sub>7</sub> -A <sub>1</sub>	Address bits 1 to 7; bit 1 is an input/output, bits 2-7 are output only.

### Pin Identification (cont)

No.	Symbol	Function
36	V <sub>DD</sub>	+5 V (typical)
37	A <sub>0</sub>	Address bit 0, input/output
38	$\overline{CS}$	Chip select input from host computer; input
39	$\overline{RD}$	Read control signal from host computer; input
40	$\overline{WR}$	Write control signal from host computer; input
41	$\overline{UBE}$	Upper byte enable input/output
42	INT	Interrupt request output
43	CLRINT	Clear interrupt request input
44	HLDRO	Hold request output
45	HLDKA	Hold acknowledge input
46	READY	Ready input
47	ASTB	Address strobe output
48	AEN	Address enable output

### Block Diagram



89-001776F

## INTERNAL CONFIGURATION

The μPD72105 has a transmitter and a receiver that transmits/ receives serial data as well as TxFIFO and RxFIFO buffers that respectively hold parallel data for the transmitter and receiver. In addition, it has a system interface to transfer data to and from the host system, a DMA controller to control data transfer, and an internal controller to control operation of each block.

### Internal Controller

The internal controller interprets the commands sent from the host processor and controls the serial section and DMA controller. It also generated results which are to be transmitted to the host system after command execution has been completed.

### System Interface

This is hardware for interfacing with the host system to process the I/O access and DMA transfer initiated by the host system.

### DMA controller

The DMA controller processes the information (such as commands, data, and results) transferred between the host system and μPD72105 on a memory basis, using DMA. The transfer is performed in units of 4-byte blocks. Each time the transmitter or receiver processes the 4-byte data, the DMA controller issues a HLDRQ signal.

### RxFIFO

This is a 20-byte receive buffer.

### TxFIFO

This is a 12-byte transmit buffer.

### Receiver

The receiver receives packets via the RxD pin and stores them in RxFIFO.

### Transmitter

The transmitter transmits the contents of TxFIFO via the TxD pin.

5

## INTERFACING WITH HOST SYSTEM

The host system can access three registers in the μPD72105: the control, status, and address FIFO registers. The control and status registers are used to control the μPD72105 without using the CRQ, INT, and CLRINT pins. The address FIFO is used to provide command addresses to the μPD72105.

### Register Selection (x: Don't Care)

CS	WR	RD	A1	A0	Operation
0	0	1	0	0	Writes to the control register
	1	0			Reads from the status register
	0	1	0	1	Specifies the address FIFO
	0	1	1	1	Writes to the address FIFO
	All others				Use prohibited
1	x			Not selected	

### Control Register

D7	D6	D5	D4	D3	D2	D1	D0	(X: Don't Care)
X	X	X	MIRQ	CIRQ	0	0	CMDRQ	

### CMDRQ (Command Request)

This bit has the same functions as the CRQ pin. The host system can issue a command request to the μPD72105 by setting this bit to 1. This bit is automatically reset when the command is executed.



**CIRQ (Clear Interrupt Request):**

This bit has the same functions as the CLRINT pin. The host system can clear the INT signal of the μPD72105 by setting this bit to 1. When the INT signal is cleared, this bit is automatically reset.

**Status Register**

D7	D6	D5	D4	D3	D2	D1	D0	
—	—	—	IRQ	—	—	WNR	—	(—: Undefined)

**WNE (Write Not Ready):**

This bit indicates that the μPD72105 is not ready when the host system is preparing to write a command address to the address FIFO. The host system must wait for this bit to become 0 before writing the next byte.

**IRQ (Interrupt Request):**

This bit indicates the status of the INT pin. The host system can detect an interrupt from the μPD72105 by polling this bit.

**Address FIFO**

This is a register that receives the memory address to which the host system has written a command and then issues a command request. The μPD72105 then receives a command from the memory area indicated by the address in the address FIFO and executes it.

When the host system issues a command request without writing an address in the address FIFO, the μPD72105 fetches a command from the memory area indicated by the default address (see NEW DEFLT ADDR command) and executes it.

**COMMANDS**

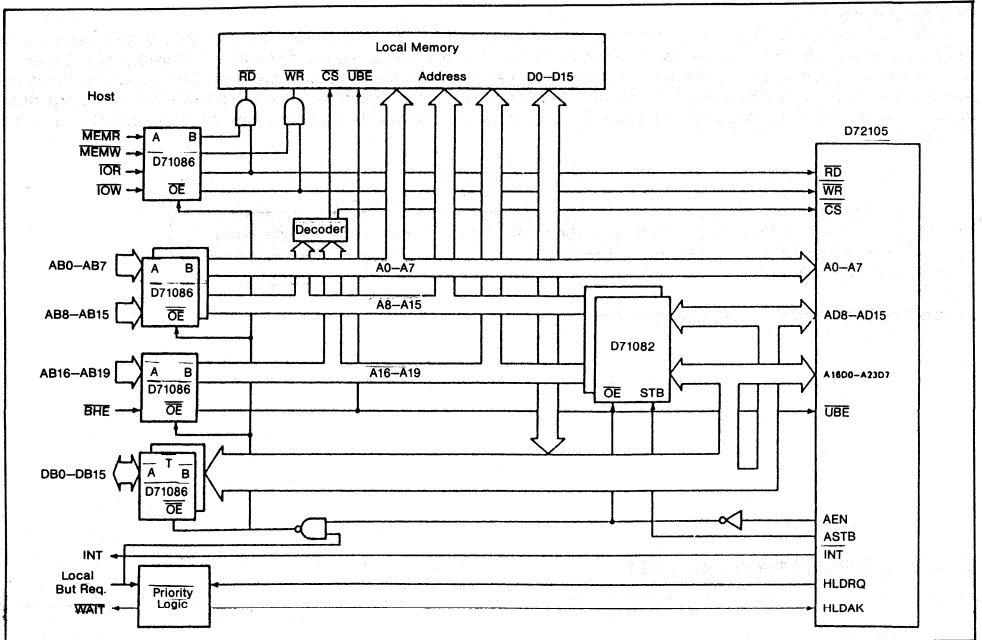
**Types of Commands**

The μPD72105 is provided with the following commands:

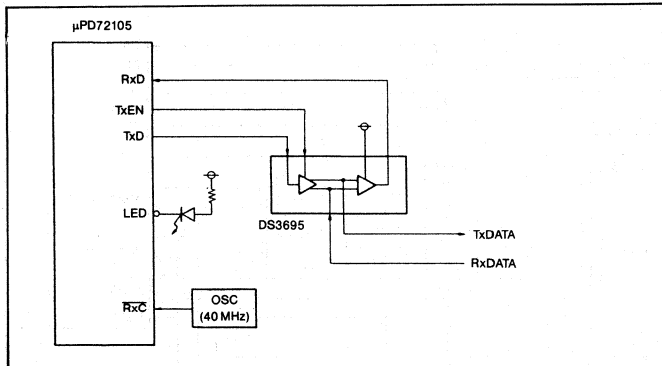
Command name	Description
INIT (Initialize)	Initializes the μPD72105.
SEND (Send)	Transmits a message to the other nodes through the network.
SETUP RCV (Setup Receive)	Activates a socket for receive.
END RCV (End Receive)	Inactivates the set-up socket.
WAIT RCV (Wait receive)	Directs a particular socket to await reception for a specific period of time.
RCV LIST (Receive List)	Sets up a list of the receive buffers.
ECHO (Echo)	Transmits an ECHO packet to the other transporter on the network.
LOOP BACK (Loop Back)	Transfers data stored in a particular buffer in the node an other buffer via the μPD72105.
INIT MONIT (Initialize Monitor)	Starts the network monitor.
MONIT OFF (Monitor OFF)	Suspends transfer during network monitoring.
MONIT ON (Monitor ON)	Resumes suspended transfer during network monitoring.
SET PARM (Set Parameters)	Sets the internal parameters of the μPD72105.
GET PARM (Get Parameters)	Reads the internal parameters of μPD72105.
NEW CHAIN (New Chain)	Specifies a command address to execute the command stored there.
NEW DEFLT ADDR (New Default Address)	Sets the default value of a command address to execute the command stored there.
CLR STAT (clear Statistics)	Clears the Tx / Rx statistics of the μPD72105.
GET STAT (Get statistics)	Reads the Tx / Rx statistics of the μPD72105.

Note: INIT MONIT, MONIT OFF and MONIT ON commands don't start operation without setting the μPD72105 in a state. The setting way will be informed to the licensees of corvus systems Inc.

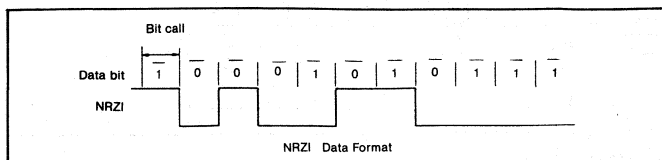




System Configuration Example II



Example for serial interface



**Packet Format**

		Packet Information				
		User Message				
I	Message Packet	Message Header	User Control x (0-255)	User Data x (0-2047)	CRC x 2	Trailing Flags
II		Message Header	User Control x (0-255)	User Data x (0-2048)	CRC x 4	Trailing Flags

I	Destination Address	Source Address	Validation "A 5"	Destination Socket	Retry	Parity	User Data Length	User Control Length
II	Destination Address	Source Address	Validation "5 A"	Destination Socket	Retry	Parity	User Data Length	User Control Length

I	Leading Flags	Destination Address	Source Address	Validation "A 5"	Socket "00"	(no use) x 5	CRC x 2	Trailing Flags
II	Leading Flags	Destination Address	Source Address	Validation "5 A"	Socket "00"	Retry	CRC x 4	Trailing Flags

I	Leading Flags	Destination Address "50"	ACK/NAK Code	Validation "A 5"	(no use)	CRC x 2	Trailing Flags	
II	Leading Flags	Destination Address	Source Address	Validation "5 A"	Socket "01"	ACK/NAK Code	CRC x 4	Trailing Flags

I	Leading Flags	Destination Address "60"	Source Address	Validation "A 5"	(no use) x 6	Retry	CRC x 2	Trailing Flags
II	Leading Flags	Destination Address	Source Address	Validation "5 A"	Socket "02"	Retry	CRC x 4	Trailing Flags

I = OMNINET I protocol, II = OMNINET II protocol

**AC/DC Target spec.****ABSOLUTE MAXIMUM RATINGS (Ta = 25°C)**

Parameter	Symbol	Test Conditions	Ratings	Units
Power Supply Voltage	VDD		-0.5 ~ +7.0	V
Input Voltage	VI		-0.5 ~ VDD +0.3	V
Output Voltage	VO		-0.5 ~ VDD + 0.3	V
Operating Temperature	Topt		-40 ~ +85	°C
Storage Temperature	Tstg		-40 ~ +125	°C

**DC CHARACTERISTICS (Ta = -40 ~ +85°C, VDD = 5V ± 10%)**

Parameter	Symbol	Test Conditions	MIN	TYP	MAX	Units
Input Low Voltage	VILC	CLK Pin	-0.5		+0.8	V
	VILR	RXC when using on-chip DPLL	-0.5		+0.4	V
	VIL	Others	-0.5		+0.8	V
Input High Voltage	VIHC	CLK Pin	+3.3		VDD+0.5	V
	VIHR	RXC when using on-chip DPLL	+2.4		VDD+0.5	V
	VIH	Others	+2.2		VDD+0.5	V
Output Voltage Low	VOL	IOL = 2.5 mA			+0.4	V
Output Voltage High	VOH	IOL = -400uA		0.7*VDD		V
Supply Current	IDDI	When operating		50		mA
Input Leakage Current	ILI	0V ≤ VIN ≤ VDD			± 10	uA
Output leakage Current	ILO	0V ≤ VOUT ≤ VDD			± 10	uA

## AC CHARACTERISTIC (Ta = -40 ~ +85 °C, VDD = 5V ± 10%)

### BUS-MASTER MODE

Parameter	Symbol	Test Conditions	MIN	MAX	Unit
CLK Cycle Time	TCYK		125	1000	nS
CLK Active Low	TKKL		50		nS
CLK Active High	TKKH		50		nS
CLK Rise Time	TKR	1.5-3.0V		10	nS
CLK Fall Time	TKF	3.0-1.5V		10	nS
HLDRQ <sup>†</sup> Delay Time from CLK <sup>↓</sup>	TDHQH			100	nS
HLDRQ <sup>↓</sup> Delay Time from CLK <sup>†</sup>	TDHQL			100	nS
HLDAK Setup Time to CLK <sup>†</sup>	TSHA		35		nS
AEN <sup>†</sup> Delay Time from CLK <sup>↓</sup>	TDAEH			100	nS
AEN <sup>↓</sup> Delay Time from CLK <sup>†</sup>	TDAEL			100	nS
ASTB <sup>†</sup> Delay Time from CLK <sup>†</sup>	TDSTH			70	nS
ASTB Pulse Width	TSTSTH		TKKH-15		nS
ASTB <sup>↓</sup> Delay Time from CLK <sup>†</sup>	TDSTL			100	nS
ADR/ <u>UBE</u> / <u>RD</u> / <u>WR</u> Delay Time from CLK <sup>†</sup>	TDA			100	nS
ADR/ <u>UBE</u> / <u>RD</u> / <u>WR</u> Float Delay from CLK <sup>↓</sup>	TFA			70	nS
ADR Setup Time to ASTB <sup>↓</sup>	TSAST		TKKH-35		nS
ADR Hold Time from ASTB	THSTA		TKKL-20		nS
<u>RD</u> <sup>↓</sup> Delay Time from ADR Float	TDAR		TKKH-30		nS
<u>RD</u> <sup>↓</sup> Delay Time from CLK <sup>↓</sup>	TDRL			70	nS
<u>RD</u> Pulse Width	TRRL2		1.5TCYK -50		nS
<u>RD</u> <sup>†</sup> Delay Time from CLK <sup>†</sup>	TDRH			100	nS
DATA Setup Time to <u>RD</u> <sup>†</sup>	TSDR		70		nS
DATA Hold Time from <u>RD</u> <sup>†</sup>	THRD		0		nS
<u>WR</u> <sup>†</sup> Delay Time from CLK <sup>↓</sup>	TDWL			70	nS
<u>WR</u> Pulse Width	TWWL2		1.5TCYK -50		nS
<u>WR</u> <sup>†</sup> Delay Time from CLK <sup>†</sup>	TDWH			80	nS
READY Setup Time to CLK <sup>†</sup>	TSRY		35		nS
READY Hold Time from CLK <sup>†</sup>	THRY		20		nS

Parameter	Symbol	Test Conditions	MIN	MAX	Units
CLRINT Pulse Width	TCLCLH		100		nS
INT↑ Delay Time from CLK↑	TDIH			100	nS
INT↓ Delay Time from CLRINT↑	TDIL			100	nS
LED↓ Delay Time from CLK↑	TDLL			100	nS
LED↑ Delay Time from CLK↑	TDLH			100	nS
CRQ Pulse Width	TCRCRH		100		nS

**BUS-SLAVE MODE**

Parameter	Symbol	Test Conditions	MIN	MAX	Units
WR Pulse Width	TWWL		100		nS
CS Hold Time from WR↑	THWCS		0		nS
ADR/UBE/CS Setup Time to WR↓	TSAW		0		nS
ADR/UBE Hold Time from WR	THWA		0		nS
DATA Setup Time to WR↑	TSDW		100		nS
DATA Hold Time from WR↑	THWD		0		nS
RD Pulse Width	TRRL		150		nS
ADR/CS Setup Time to RD↓	TSAR		35		nS
ADR/CS Hold Time from RD↑	THRA		0		nS
DATA Delay Time from RD↓	TDRD			120	nS
DATA Float Delay from RD↑	TFRD		10	100	nS
RESET Pulse Width	TRSTL		7TGYK		nS
VDD Setup Time to RESET↑	TSVDD		1000		nS
1st WR/RD from RESET↑	TSYWR		2TCYK		nS
Recovery Time from WR/RD	TRVWR		200		nS
High Setup Time to HLDK↑	TSWR		-20		nS
High Hold Time from AEN↓	THWR		100		nS

\* 1 Input levels for AC test are  
 2.4V (as "1") and  
 0.4V (as "0").

Test points are  
 2.2V (as "1<sub>n</sub>") and  
 0.8V (as "0").

\* 2 Fix B/W pin to "1" or "0"

## Serial Part

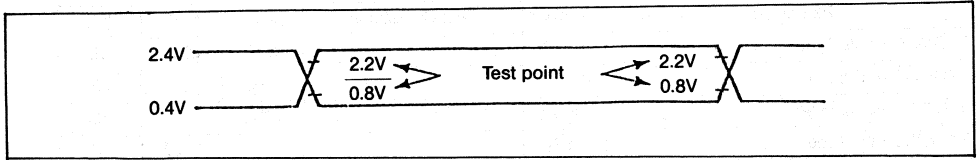
Parameter	Symbol	Test Conditions	MIN	MAX	Units
TXC/RXC Cycle Time	TCYS		250	2000	nS
TXC/RXC Active Low	TSSL		110		nS
TXC/RXC Active High	TSSH		110		nS
TXC/RXC Rise Time	TSR			20	nS
TXC/RXC Fall Time	TSF			12	nS
TXD Delay Time from $\overline{\text{TCX}}\downarrow$	TDTXD			100	nS
TXEN <sup>†</sup> Delay Time from $\overline{\text{TCX}}\downarrow$	TDTEH			100	nS
TXEN <sup>†</sup> Delay Time from $\overline{\text{TCX}}\downarrow$	TDTEL			100	nS
RXD Setup Time to $\overline{\text{RXC}}\uparrow$	TSRXD		50		nS
RXD Hold Time from $\overline{\text{RXC}}\uparrow$	THRXD		70		nS
$\overline{\text{RXC}}$ Cycle Time *3	TCYH	When using on-chip DPLL	25	200	nS
RXC Active Low	THHL	When using on-chip DPLL	5		nS
RXC Active High	THHH	When using on-chip DPLL	5		nS
RXC Rise Time	THR	When using on-chip DPLL		5	nS
RXC Fall Time	THF	When using on-chip DPLL		5	nS
TX, RX DATA Cycle Time	TCYD	When using on-chip DPLL	250	2000	nS
RXD Setup Timing	TRX	When using on-chip DPLL			nS
TXC Active Low	TTTL	When using on-chip DPLL			nS
TXC Active High	TTTH	When using on-chip DPLL			nS
TXD Change Delay from $\overline{\text{TCX}}\downarrow$	TDTX1	When using on-chip DPLL		50	nS
TXD Change Delay from $\overline{\text{TCX}}\downarrow$	TDTX2	When using on-chip DPLL			nS

\* 3 DPLLed clock cycle time should be from 250 ns to 2000 ns.

This is a target spec. and can be changed during development.

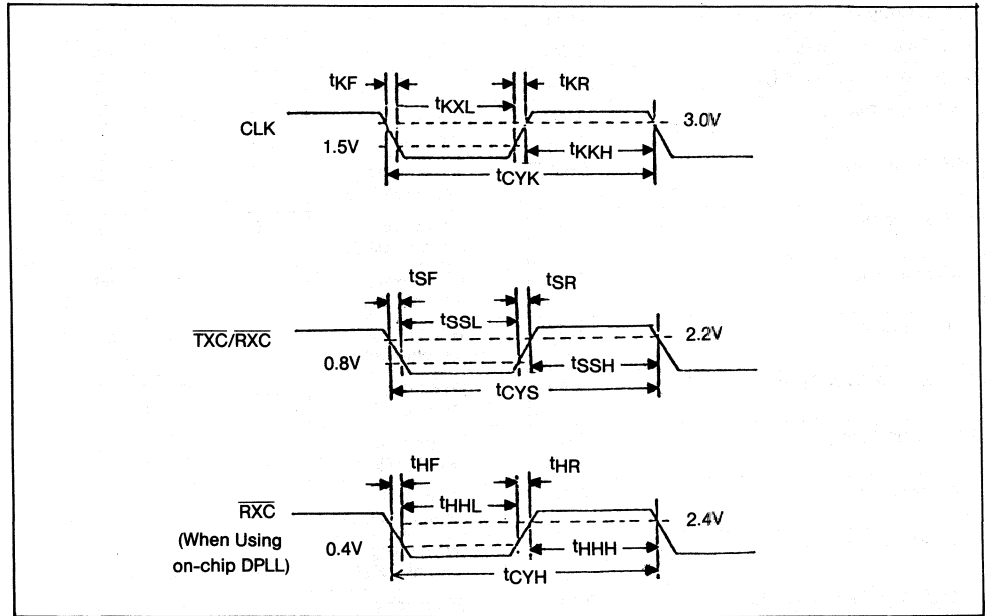


AC Timing Test Points



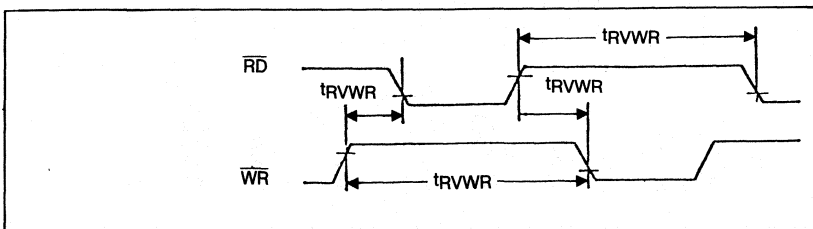
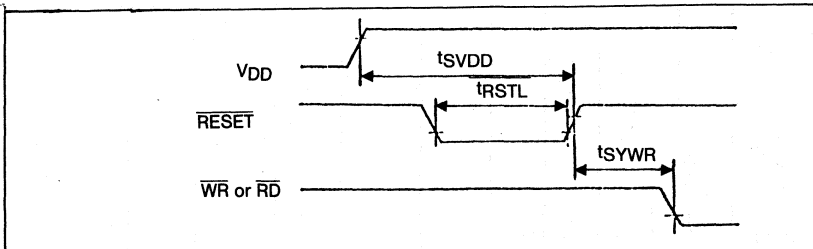
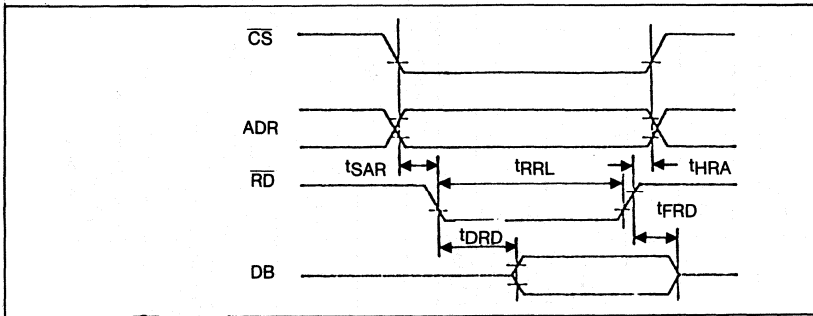
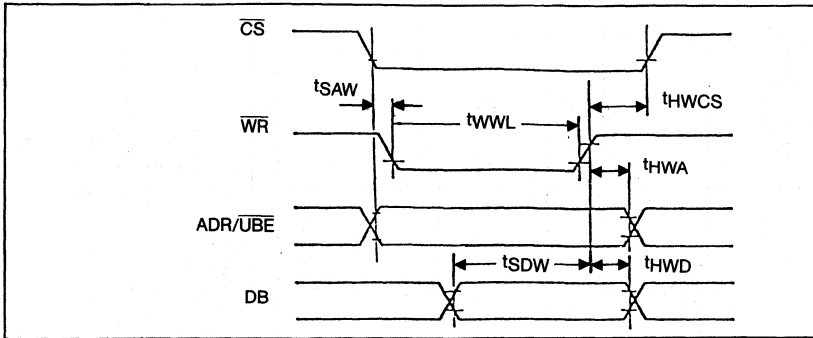
Timing Waveform

CLK Waveform

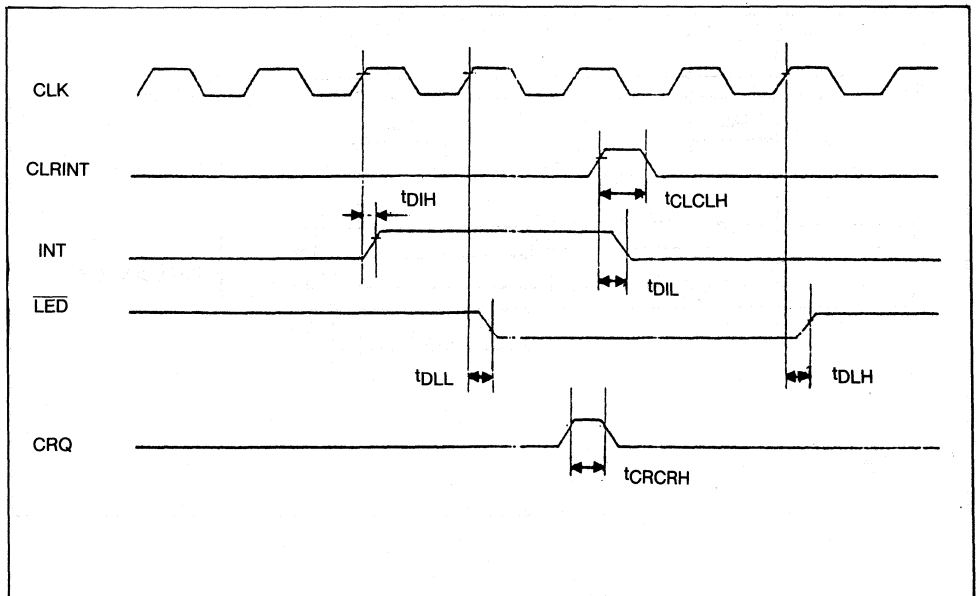
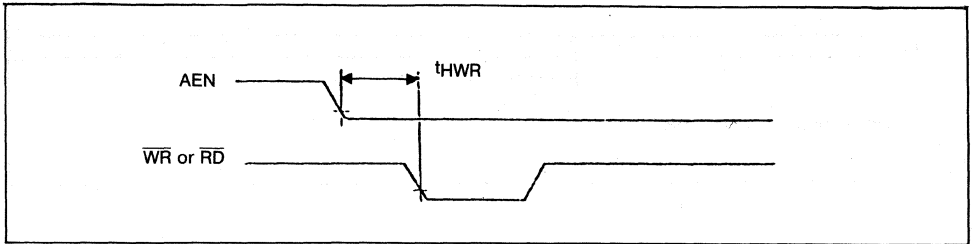
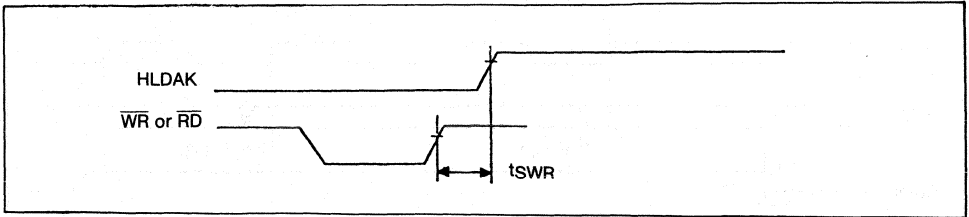


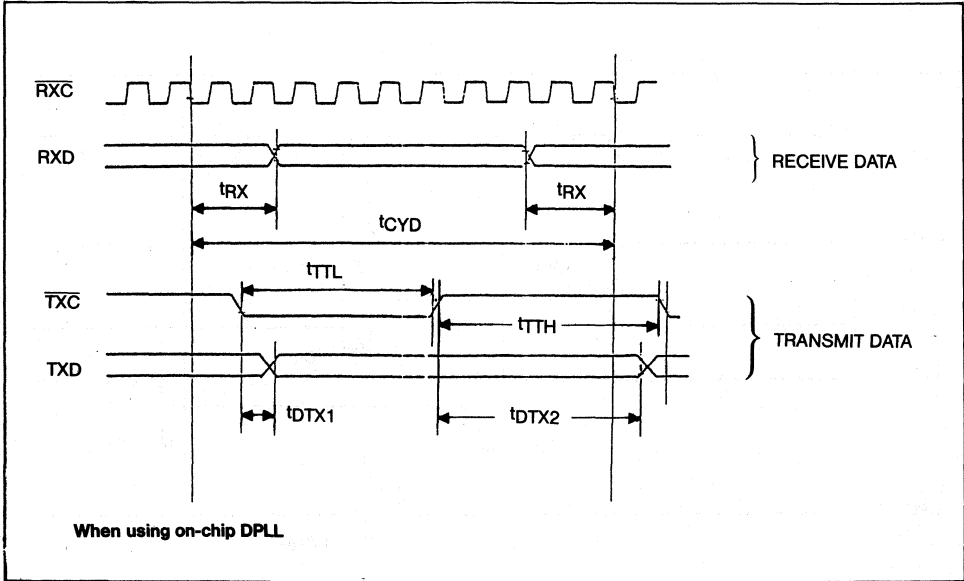
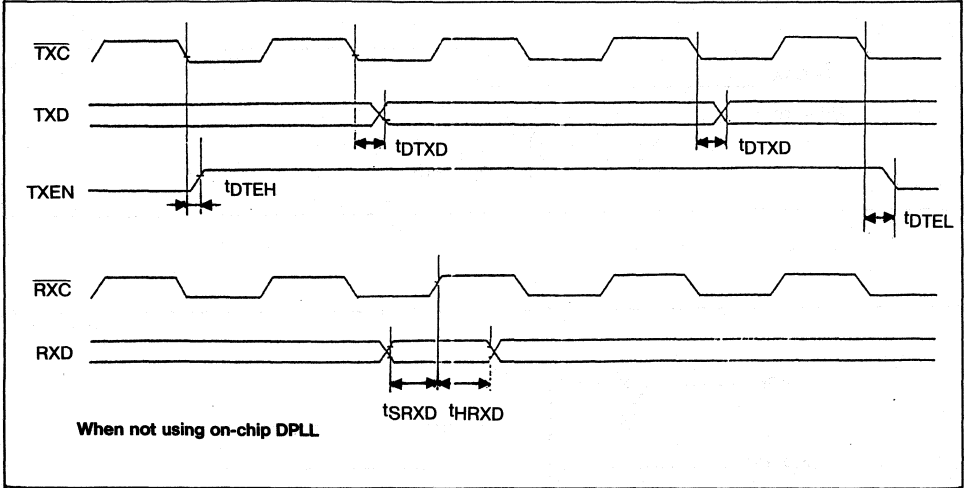


### Bus Slave mode (1)



## Bus Slave mode (2)





## Description

The μPD7210 is an intelligent, general purpose interface bus (GPIB) controller designed to meet all of the functional requirements for talker, listener, and controller (TLC) as specified by IEEE Standard 488-1978. Connected between a processor bus and the GPIB, the controller provides high-level management of the GPIB to unburden the processor and to simplify both hardware and software design. The μPD7210 is fully compatible with most processor architectures and requires only the addition of bus driver/receiver components to implement any type of GPIB.

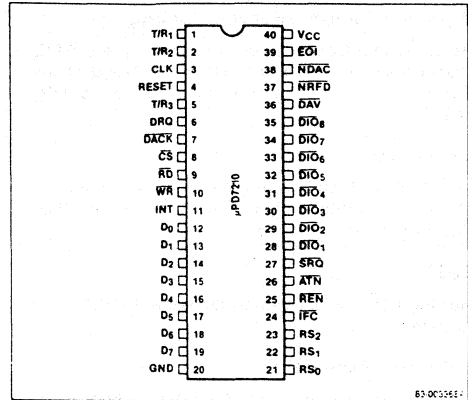
## Features

- All-functional interface capability meeting IEEE Standard 488-1978
  - SH1 (source handshake)
  - AH1 (acceptor handshake)
  - L3 or LE3 (listener or extended listener)
  - T5 or TE5 (talker or extended talker)
  - SR1 (service request)
  - RL1 (remote local)
  - PP1 or PP2 (parallel poll, remote or local configuration)
  - DC1 (device clear)
  - DT1 (device trigger)
  - C1-C5 (controller, all functions)
- Programmable data transfer rate
- 16 MPU accessible registers: 8 read and 8 write
- 2 address registers
  - Detection of MTA, MLA, MSA (my talk/my listen/my secondary addresses)
  - 2 device addresses
- EOS message automatic detection
- Command (IEEE Standard 488-1978) automatic processing and undefined command read capability
- DMA capability
- Programmable bus transceiver I/O specification (works with T.I./Motorola/Intel)
- 1-MHz to 8-MHz clock range
- TTL-compatible
- NMOS
- +5 V single power supply
- 8080/85/86-compatible

## Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD7210C	40-pin plastic DIP	8 MHz

## Pin Configuration



## Pin Identification

No.	Symbol	Function
1, 2, 5	T/R <sub>1</sub> -T/R <sub>3</sub>	Transmit/receive control outputs
3	CLK	Clock input
4	RESET	Reset input
6	D/RQ	DMA request output
7	DACK	DMA acknowledge input
8	CS	Chip select input
9	RD	Read input
10	WR	Write input
11	INT	Interrupt request output
12-19	D <sub>0</sub> -D <sub>7</sub>	Bidirectional data bus
20	GND	Ground
21-23	RS <sub>0</sub> -RS <sub>2</sub>	Register select input
24	IFC	Interface clear I/O
25	REN	Remote enable I/O
26	ATN	Attention control line I/O
27	SRQ	Service request I/O
28-35	DIO <sub>1</sub> -DIO <sub>8</sub>	8-bit bidirectional data bus
36	DAV	Data valid I/O
37	NRFD	Ready for data I/O
38	NDAC	Data accepted I/O
39	EOI	End or identify I/O
40	V <sub>CC</sub>	+5 V power supply

**Pin Functions****T/R<sub>1</sub>-T/R<sub>3</sub> [Transmit/Receive Control]**

This is the input/output control signal for the GPIB transceivers. The values of TRM1 and TRM0 of the address mode register determine the functions of T/R<sub>2</sub> and T/R<sub>3</sub>.

**CLK [Clock]**

This 1-MHz to 8-MHz reference clock generates the state change prohibit times T<sub>1</sub>, T<sub>6</sub>, T<sub>7</sub>, and T<sub>9</sub> specified in IEEE Standard 488-1978.

**RESET**

When high, the RESET signal places the μPD7210 in an idle state.

**DRQ [DMA Request]**

DRQ becomes low on input of the DMA acknowledge signal DACK.

**DACK [DMA Acknowledge]**

This signal connects the computer system data bus to the data register of the μPD7210.

**CS [Chip Select]**

The chip select input enables access to the register selected by the read or write operation (RS<sub>0</sub>-RS<sub>2</sub>).

**RD [Read]**

The read input places the contents of the read register specified by RS<sub>0</sub>-RS<sub>2</sub> on the computer bus (D<sub>0</sub>-D<sub>7</sub>).

**WR [Write]**

This input writes data on D<sub>0</sub>-D<sub>7</sub> into the write register specified by RS<sub>0</sub>-RS<sub>2</sub>.

**INT, INT [Interrupt Request]**

This output is active high/low. It becomes active due to any one of 13 internal interrupt factors (unmasked). Its active state is software configurable, and it is active high on chip reset.

**D<sub>0</sub>-D<sub>7</sub> [Data Bus]**

The 8-bit bidirectional data bus interfaces to the computer system.

**GND [Ground]**

This is the ground.

**RS<sub>0</sub>-RS<sub>2</sub> [Register Select]**

These lines select one of eight read (write) registers during a read (write) operation.

**IFC [Interface Clear]**

This bidirectional control line is used for clearing the interface functions.

**REN [Remote Enable]**

This bidirectional control line is used to select remote or local control of the devices.

**ATN [Attention]**

This bidirectional control line indicates whether data on the DIO lines is an interface message or a device-dependent message.

**SRQ [Service Request]**

This bidirectional control line is used to request service from the controller.

**DIO<sub>1</sub>-DIO<sub>8</sub> [Data Input/Output]**

This 8-bit bidirectional bus transfers messages on the GPIB.

**DAV [Data Valid]**

This handshake line indicates that data on the DIO line is valid.

**NRFD [Ready for Data]**

This handshake line indicates that the device is ready for data.

**NDAC [Data Accepted]**

This handshake line indicates the completion of message reception.

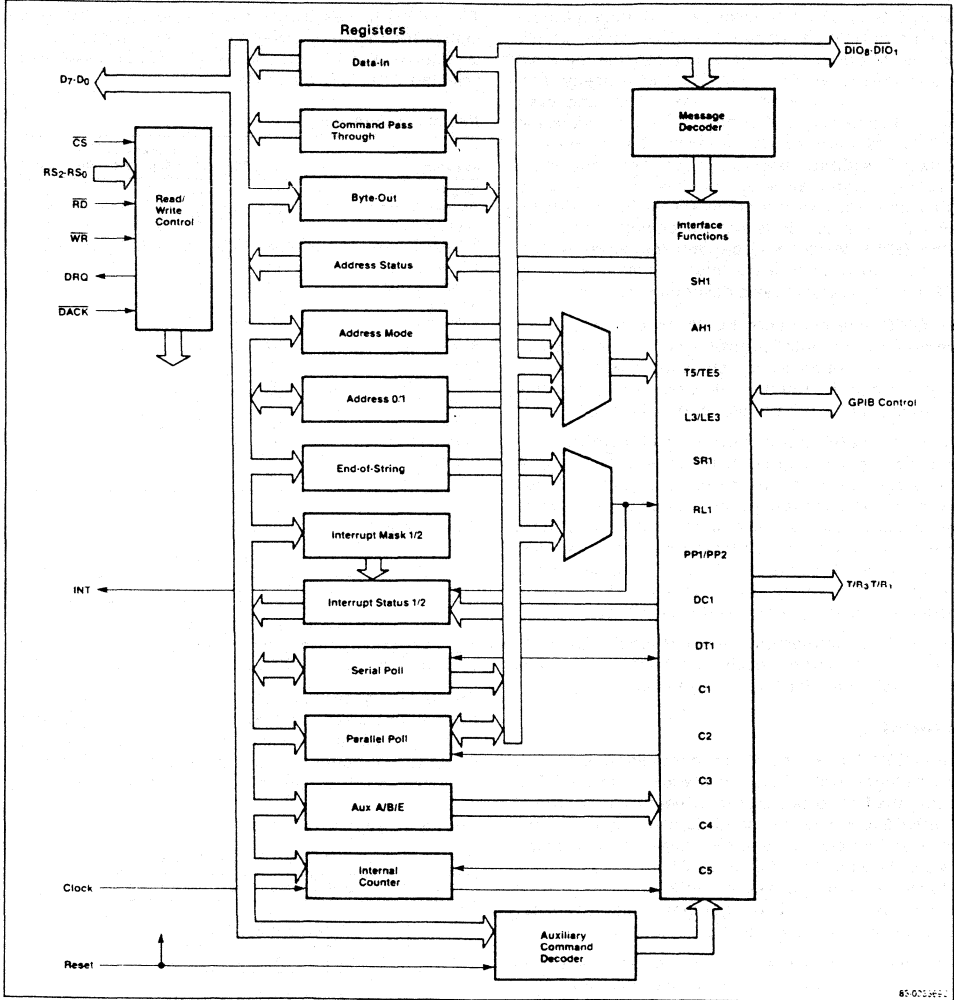
**EOI [End or Identify]**

This control line is used to indicate the end of a multiple byte transfer sequence or to execute parallel polling in conjunction with ATN.

**VCC [Power Supply]**

+5 V power supply.

## Block Diagram





**History**

The IEEE Standard 488 describes a "Standard Digital Interface for Programmable Instrumentation" which, since its introduction in 1975, has become the most popular means of interconnecting instruments and controllers in laboratory, automatic test, and even industrial applications. Refined over several years, the 488-1978 Standard, also known as the General Purpose Interface Bus (GPIB), is a highly sophisticated standard providing a high degree of flexibility to meet virtually all instrumentation requirements. The μPD7210 implements all of the functions that are required to interface to the GPIB. While it is beyond the scope of this document to provide a complete explanation of the IEEE 488 Standard, a basic description follows:

The GPIB interconnects up to 15 devices over a common set of data control lines. Three types of devices are defined by the standard: talker, listener, and controller, although some devices may combine functions such as talker/listener or talker/controller.

Data on the GPIB is transferred in a bit-parallel, byte-serial fashion over eight data I/O lines ( $\overline{DI}_0$ - $\overline{DI}_7$ ). A three-wire handshake is used to ensure synchronization of transmission and reception. In order to permit more than one device to receive data at the same time, these control lines are "open collector" so that the slowest device controls the data rate. A number of other control lines perform a variety of functions such as device addressing, interrupt generation, and so forth.

The μPD7210 implements all functional aspects of talker, listener, and controller functions as defined by the 488-1978 Standard on a single chip.

**General**

The μPD7210 is an intelligent controller designed to provide high-level protocol management of the GPIB, freeing the host processor for other tasks. Control of the μPD7210 is accomplished via 16 internal registers. Data may be transferred either under program control or via DMA using the μPD7210's DMA control facilities to further reduce processor overhead. The processor interface of the μPD7210 is general in nature and may be readily interfaced to most processor lines.

In addition to providing all control and data lines necessary for a complete GPIB implementation, the μPD7210 also provides a unique set of bus transceiver controls permitting a variety of transceiver configurations for maximum flexibility.

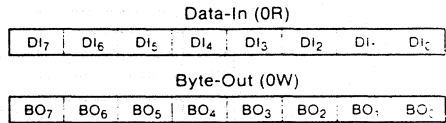
**Internal Registers**

The μPD7210 has eight read registers (0R-7R) and eight write registers (0W-7W). The register number is selected via the RS<sub>2</sub>, RS<sub>1</sub>, and RS<sub>0</sub> lines; read or write is selected via WR, RD, and CS.

**Register Addressing**

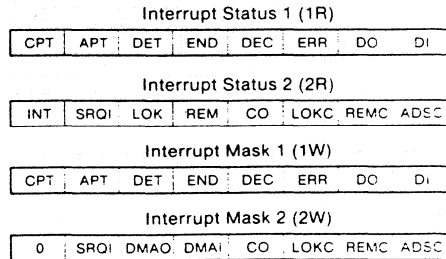
Register	Addressing						
	RS <sub>2</sub>	RS <sub>1</sub>	RS <sub>0</sub>	WR	RD	CS	
Data-In	0R	0	0	0	1	0	0
Interrupt Status 1	1R	0	0	1	1	0	0
Interrupt Status 2	2R	0	1	0	1	0	0
Serial Poll Status	3R	0	1	1	1	0	0
Address Status	4R	1	0	0	1	0	0
Command Pass Through	5R	1	0	1	1	0	0
Address 0	6R	1	1	0	1	0	0
Address 1	7R	1	1	1	1	0	0
Byte Out	0W	0	0	0	0	1	0
Interrupt Mask 1	1W	0	0	1	0	1	0
Interrupt Mask 2	2W	0	1	0	0	1	0
Serial Poll Mode	3W	0	1	1	0	1	0
Address Mode	4W	1	0	0	0	1	0
Auxiliary Mode	5W	1	0	1	0	1	0
Address 0/1	6W	1	1	0	0	1	0
End of String	7W	1	1	1	0	1	0

**Data Registers**



The data registers are used for data and command transfers between the GPIB and the microcomputer system. The Data-In register holds data sent from the GPIB to the computer; the Byte-Out register holds information written into it for transfer to the GPIB.

**Interrupt Registers**



The interrupt registers are composed of interrupt status bits, interrupt mask bits, and some other non-interrupt related bits.

There are 13 factors that can generate an interrupt from the μPD7210, each with its own status bit and mask bit.

The interrupt status bits are always set to 1 if the interrupt condition is met. The interrupt mask bits decide whether or not the INT bit and the interrupt pin will be active for that condition.

### Interrupt Status Bits

INT	OR of all unmasked interrupt status bits
CPT	Command pass through
APT	Address pass through
DET	Device trigger
END	End (END or EOS message received)
DEC	Device clear
ERR	Error
DO	Data out
DI	Data in
SROI	Service request input
LOKC	Lockout change
REMC	Remote change
ADSC	Address status change
CO	Command output

### Noninterrupt Related Bits

LOK	Lockout
REM	Remote/local
DMAO	Enable/disable DMA out
DMAI	Enable/disable DMA in

### Serial Poll Registers

#### Serial Poll Status (3R)

S <sub>8</sub>	PEND	S <sub>6</sub>	S <sub>5</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>
----------------	------	----------------	----------------	----------------	----------------	----------------	----------------

#### Serial Poll Mode (3W)

S <sub>8</sub>	rsv	S <sub>6</sub>	S <sub>5</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>
----------------	-----	----------------	----------------	----------------	----------------	----------------	----------------

The serial poll mode register holds the STB (status byte: S<sub>8</sub>, S<sub>6</sub>-S<sub>1</sub>) sent over the GPIB and the local message rsv (request service). The serial poll mode register may be read through the serial poll status register. The PEND is set by rsv = 1 and cleared by NPRS · rsv = 1 (NPRS means negative poll response state).

### Address Mode/Address Status Registers

#### Address Status (4R)

CIC	ATN	SPMS	LPAS	TPAS	LA	TA	MJMN
-----	-----	------	------	------	----	----	------

#### Address Mode (4W)

I <sub>on</sub>	I <sub>on</sub>	TRM1	TRM0	0	0	AMD1	AMD0
-----------------	-----------------	------	------	---	---	------	------

The address mode register selects the address mode of the device and also sets the mode for the transceiver control lines, T/R<sub>3</sub> and T/R<sub>2</sub>.

The functions of T/R<sub>2</sub> (pin 2) and T/R<sub>3</sub> (pin 5) are determined by the TRM1, TRM0 values of the address mode register.

#### Function of T/R<sub>2</sub> and T/R<sub>3</sub>

T/R <sub>2</sub>	T/R <sub>3</sub>	TRM1	TRM0
EOIOE	TRIG	0	0
CIC	TRIG	0	1
CIC	EOIOE	1	0
CIC	PE	1	1

$$EOIOE = TACS + SPAS + CIC \cdot CSBS$$

This denotes the input/output of the EO terminal.

When 1: output

When 0: input

$$CIC = CIDS + CADS$$

This denotes whether or not the controller interface function is active.

When 1:  $\overline{ATN}$  = output,  $\overline{SRQ}$  = input

When 0:  $\overline{ATN}$  = input,  $\overline{SRQ}$  = output

$$PE = CIC + PPAS$$

This indicates the type of bus driver connected to the DIO<sub>8</sub> to DIO<sub>1</sub>, and DAV lines.

When 1: three-state

When 0: open-collector

TRIG: When DTAS state is initiated or when a trigger auxiliary command is issued, a high pulse is generated.

Upon reset, TRM0 and TRM1 become 0 (TRM0 = TRM1 = 0) and a local message port is provided so that T/R<sub>2</sub> and T/R<sub>3</sub> both become low.

**Address Modes**

$t_{on}$	$t_{off}$	ADMI	ADMO	Address Mode	Contents of Address 0 Register	Contents of Address 1 Register
1	0	0	0	Talk only mode	Address identification not necessary (No controller on the GPIB)	
0	1	0	0	Listen only mode	Not used	
0	0	0	1	Address mode 1 (Note 1)	Major talk address or major listen address	Minor talk address or minor listen address
0	0	1	0	Address mode 2 (Note 2)	Primary address (talk or listen)	Secondary address (talk or listen)
0	0	1	1	Address mode 3 (Note 3)	Primary address (major talk or major listen)	Primary address (minor talk or minor listen)

**Note:**

- (1) Either MTA or MLA reception is indicated by coincidence of either address with the received address, interface function T or L.
- (2) Address register 0 = primary; address register 1 = secondary; interface function TE or LE.
- (3) CPU must read secondary address via Command Pass Through register interface function (TE or LE).
- (4) Combinations other than those indicated are prohibited.

**Address Status Bits**

ATN	Data transfer cycle (device in CSBS)
LPAS	Listener primary addressed state
TPAS	Talker primary addressed state
CIC	Controller active
LA	Listener addressed
TA	Talker addressed
MJMN	Sets minor T/L address, reset = major T/L address
SPMS	Serial poll mode state

**Address Registers**

Address 0 (6R)

X	DT0	DL0	AD5-0	AD4-0	AD3-0	AD2-0	AD-1
---	-----	-----	-------	-------	-------	-------	------

Address 1 (7R)

EOI	DT1	DL1	AD5-1	AD4-1	AD3-1	AD2-1	AD1-1
-----	-----	-----	-------	-------	-------	-------	-------

Address 0/1 (6W)

ARS	DT	DL	AD <sub>5</sub>	AD <sub>4</sub>	AD <sub>3</sub>	AD <sub>2</sub>	AD <sub>1</sub>
-----	----	----	-----------------	-----------------	-----------------	-----------------	-----------------

The μPD7210 is able to detect automatically two types of addresses that are held in address registers 0 and 1. The addressing modes are outlined below.

Address settings are made by writing into the address 0/1 register. The function of each bit is described below.

**Address 0/1 Register Bit Selections**

ARS	Selects either address register 0 or 1
DT	Permits or prohibits address to be detected as Talk
DL	Permits or prohibits address to be detected as Listen
AD <sub>5</sub> -AD <sub>1</sub>	Device address value
EOI	Holds the value of EOI line when data is received

**Command Pass Through Register [5R]**

CPT <sub>7</sub>	CPT <sub>6</sub>	CPT <sub>5</sub>	CPT <sub>4</sub>	CPT <sub>3</sub>	CPT <sub>2</sub>	CPT <sub>1</sub>	CPT <sub>0</sub>
------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------

The CPT register is used such that the CPU may read the DIO lines in the cases of undefined command, secondary address, or parallel poll response.

**End-of-String Register [7W]**

EC <sub>7</sub>	EC <sub>6</sub>	EC <sub>5</sub>	EC <sub>4</sub>	EC <sub>3</sub>	EC <sub>2</sub>	EC <sub>1</sub>	EC <sub>0</sub>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

This register holds either a 7- or 8-bit EOS message byte used in the GPIB system to detect the end of a data block. Auxiliary register A controls the specific use of this register.

**Auxiliary Mode Register [5W]**

CNT <sub>2</sub>	CNT <sub>1</sub>	CNT <sub>0</sub>	COM <sub>4</sub>	COM <sub>3</sub>	COM <sub>2</sub>	COM <sub>1</sub>	COM <sub>0</sub>
------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------

This is a multipurpose register. A write to this register generates one of the following operations according to the values of the CNT bits.

### Auxiliary Mode Operations

CNT				COM				Operation
2	1	0	4	3	2	1	0	
0	0	0	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	Issues an auxiliary command specified by C <sub>4</sub> to C <sub>0</sub> .
0	0	1	0	F <sub>3</sub>	F <sub>2</sub>	F <sub>1</sub>	F <sub>0</sub>	The reference clock frequency is specified and T <sub>1</sub> , T <sub>6</sub> , T <sub>7</sub> , and T <sub>9</sub> are determined as a result.
0	1	1	U	S	P <sub>3</sub>	P <sub>2</sub>	P <sub>1</sub>	Makes a write operation to the parallel poll register.
1	0	0	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Makes a write operation to the auxiliary A register.
1	0	1	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	Makes a write operation to the auxiliary B register.
1	1	0	0	0	0	E <sub>1</sub>	E <sub>0</sub>	Makes a write operation to the auxiliary E register.

## Commands and Other Registers

### Auxiliary Commands

0	0	0	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>
---	---	---	----------------	----------------	----------------	----------------	----------------

### Auxiliary Command Descriptions

Command					Auxiliary Command	Description
C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>		
0	0	0	0	0	iepon	Immediate execute pon; generate local pon message.
0	0	0	1	0	crst	Chip reset (same as external reset)
0	0	0	1	1	rrfd	Release RFD
0	0	1	0	0	trig	Trigger
0	0	1	0	1	rtl	Return to local message generation
0	0	1	1	0	seoi	Send EOI message
0	0	1	1	1	nvid	Nonvalid (OSA reception); release DAC holdoff
0	1	1	1	1	vid	Valid (MSA Reception, CPT, DEC, DET); release DAC holdoff
0	X	0	0	1	sppf	Set/reset parallel poll flag
1	0	0	0	0	gts	Go to standby
1	0	0	0	1	tca	Take control asynchronously

### Auxiliary Command Descriptions (cont)

Command					Auxiliary Command	Description
C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>		
1	0	0	1	0	tcs	Take control synchronously
1	1	0	1	0	tcse	Take control synchronously on end
1	0	0	1	1	ltn	Listen
1	1	0	1	1	ltnc	Listen with continuous mode
1	1	1	0	0	lun	Local unlisten
1	1	1	0	1	epp	Execute parallel poll
1	X	1	1	0	sifc	Set/reset IFC
1	X	1	1	1	sren	Set/reset REN
1	0	1	0	0	dsc	Disable system control

### Internal Counter

0	0	1	0	F <sub>3</sub>	F <sub>2</sub>	F <sub>1</sub>	F <sub>0</sub>
---	---	---	---	----------------	----------------	----------------	----------------

The internal counter generates the state change prohibit times (T<sub>1</sub>, T<sub>6</sub>, T<sub>7</sub>, T<sub>9</sub>) specified in IEEE Standard 488-1978 with reference to the clock frequency.

### Auxiliary A Register

1	0	0	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
---	---	---	----------------	----------------	----------------	----------------	----------------

Of the five bits that may be specified as part of the access word, two bits control the GPIB data receiving modes of the μPD7210 and three bits control how the end-of-string (EOS) message is used.

### Data Receiving Modes

A <sub>1</sub>	A <sub>0</sub>	Data Receiving Mode
0	0	Normal handshake mode
0	1	RFD holdoff on all data modes
1	0	RFD holdoff on end mode
1	1	Continuous mode

### EOS Message

Bit Name	Function	
A <sub>2</sub>	0 Prohibit	Permits (prohibits) the setting of the END bit by reception of the EOS message
	1 Permit	
A <sub>3</sub>	0 Prohibit	Permits (prohibits) automatic transmission of END message simultaneously with the transmission of EOS message TACS.
	1 Permit	
A <sub>4</sub>	0 7-bit EOS	Makes the 8 bits (7 bits) of the EOS register the valid EOS message
	1 8-bit EOS	

**Auxiliary B Register**

1	0	1	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
---	---	---	----------------	----------------	----------------	----------------	----------------

The auxiliary B register is much like the A register in that it controls the special operating features of the device.

**Special Features**

Bit Name		Function
B <sub>0</sub>	1 Permit	Permits (prohibits) the detection of an undefined command. In other words, it permits (prohibits) the setting of the CPT bit on receipt of an undefined command.
	0 Prohibit	
B <sub>1</sub>	1 Permit	Permits (prohibits) the transmission of the END message when in serial poll active state (SPAS).
	0 Prohibit	
B <sub>2</sub>	1 T <sub>1</sub> (high-speed)	T <sub>1</sub> (high speed) as T <sub>1</sub> in source handshake function after transmission of second byte following data transmission.
	0 T <sub>1</sub> (low-speed)	Sets T <sub>1</sub> (low speed) as T <sub>1</sub> in all cases.
B <sub>3</sub>	1 $\overline{\text{INT}}$	Specifies the active level of the $\overline{\text{INT}}$ pin.
	0 INT	
B <sub>4</sub>	1 ist = SRQS	SRQS indicates the value of the ist level local message (the value of the parallel poll flag is ignored). SRQS = 1 . . . ist = 1 SRQS = 0 . . . ist = 0
	0 ist = Parallel Poll Flag	The value of the parallel poll flag is taken as the ist local message.

**Auxiliary E Register**

1	1	0	0	0	0	0	E <sub>1</sub>	E <sub>0</sub>
---	---	---	---	---	---	---	----------------	----------------

This register controls the Data Acceptance modes of the μPD7210.

**Data Acceptance Modes**

Bit Name		Function
E <sub>0</sub>	1 Enable	DAC holdoff by initialization of DCAS
	0 Disable	
E <sub>1</sub>	1 Enable	DAC holdoff by initialization of DTAS
	0 Disable	

**Parallel Poll Register**

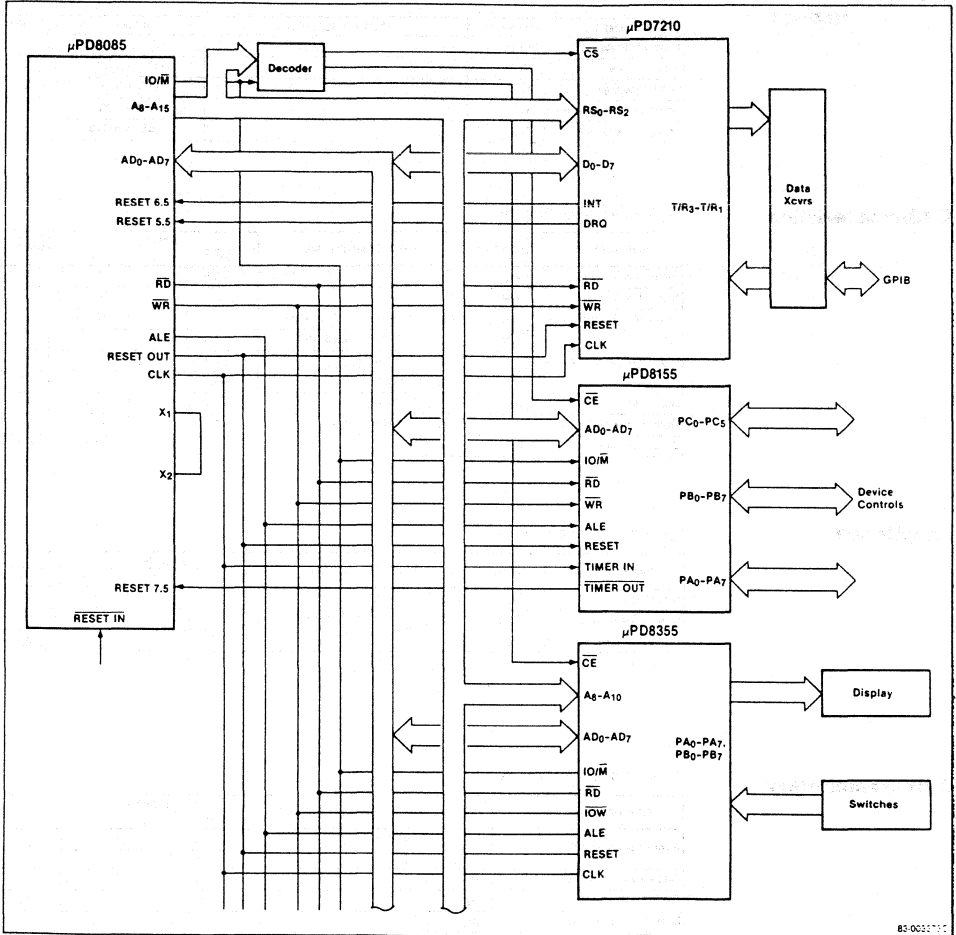
0	1	1	U	S	P <sub>3</sub>	P <sub>2</sub>	P <sub>1</sub>
---	---	---	---	---	----------------	----------------	----------------

The parallel poll register defines the parallel poll response of the μPD7210.

**Parallel Poll Response**

Bit Name		Function
U	1	No response to parallel poll
	0	Response to parallel poll
S	1	In phase
	0	Reverse phase
P <sub>3</sub> -P <sub>1</sub>	000-111	Status bit output line DIO <sub>7</sub> to DIO <sub>6</sub>

## Minimum 8085 System with μPD7210



**Absolute Maximum Ratings**

T<sub>a</sub> = 25°C

PARAMETER	SYMBOL	TEST CONDITIONS	RATINGS	UNITS
Supply Voltage	V <sub>CC</sub>		-0.5 ~ +7.0	V
Input Voltage	V <sub>I</sub>		-0.5 ~ +7.0	V
Output Voltage	V <sub>O</sub>		-0.5 ~ +7.0	V
Operating Temperature	T <sub>opt</sub>		0 ~ +70	°C
Storage Temperature	T <sub>stg</sub>		-65 ~ +125	°C

**DC Characteristics**

T<sub>a</sub> = 0 ~ +70°C, V<sub>CC</sub> = 5V ± 10%

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
Input Low Voltage	V <sub>IL</sub>		-0.5		+0.8	V
Input High Voltage	V <sub>IH</sub>		+2.0		V <sub>CC</sub> +0.5	V
Low Level Output Voltage	V <sub>OL</sub>	I <sub>OL</sub> = 2mA (4mA : T/R1 Pin)			+0.45	V
High Level Output Voltage	VOH1	I <sub>OH</sub> = -400uA, Except INT	+2.4			V
	VOH2	I <sub>OH</sub> = -400uA I <sub>OH</sub> = -50uA	+2.4 +3.5			V V
		INT Pin				
Input Leakage Current	I <sub>IL</sub>	I <sub>IN</sub> = 0V ~ V <sub>CC</sub>	-10		+10	uA
Output Leakage Current	I <sub>OL</sub>	I <sub>OUT</sub> = 0.45V ~ V <sub>CC</sub>	-10		+10	uA
Supply Current	I <sub>CC</sub>				+180	mA

**Capacitance**

T<sub>a</sub> = 25°C, V<sub>CC</sub> = GND = 0V

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
Input Capacitance	C <sub>IN</sub>	f = 1 MHz			10	pF
Output Capacitance	C <sub>OUT</sub>	All Pins Except Pin Under Test Tied to AC Ground			15	pF
I/O Capacitance	C <sub>I/O</sub>				20	pF

**AC Characteristics**

T<sub>a</sub> = 0 ~ +70°C, V<sub>CC</sub> = 5V ± 10%

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
Address Setup to $\overline{RD}$	t <sub>AR</sub>	RS2.0	85			ns
		CS	0			
Address Hold from $\overline{RD}$	t <sub>RA</sub>		0			ns
$\overline{RD}$ Pulse Width	t <sub>RR</sub>		170			ns
Data Delay from Address	t <sub>AD</sub>				250	ns
Data Delay from $\overline{RD} \downarrow$	t <sub>RD</sub>				150	ns
Output Float Delay from $\overline{RD} \uparrow$	t <sub>DF</sub>		0		80	ns
$\overline{RD}$ Recovery Time	t <sub>RV</sub>		250			ns

Address Setup to $\overline{WR}$	t <sub>AW</sub>		0			ns
Address Hold from $\overline{WR}$	t <sub>WA</sub>		0			ns
$\overline{WR}$ Pulse Width	t <sub>WW</sub>		170			ns
Data Setup to $\overline{WR}$	t <sub>DW</sub>		150			ns
Data Hold from $\overline{WR}$	t <sub>WD</sub>		0			ns
$\overline{WR}$ Recovery Time	t <sub>RV</sub>		250			ns

DMAREQ $\downarrow$ Delay from DMAACK	t <sub>AKRO</sub>				130	ns
Data Delay from DMAACK	t <sub>AKD</sub>				200	ns

T<sub>a</sub> = 0 ~ +70°C, V<sub>CC</sub> = 5V ± 10%

## AC Characteristics

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
E <sub>O1</sub> ↓ → D <sub>O</sub>	!EOD1	PPSS → PPAS, ATN = True			250	ns
E <sub>O1</sub> ↓ → T/R1 ↑	!EOT11	PPSS → PPAS, ATN = True			155	ns
E <sub>O1</sub> ↑ → T/R1 ↓	!EOT12	PPAS → PPSS, ATN = False			200	ns
ATN ↓ → NDAC ↓	!ATND	AIDS → ANRS, LIDS			155	ns
ATN ↓ → T/R1 ↓	!ATT1	TACS + SPAS → TADS, CIDS			155	ns
ATN ↓ → T/R2 ↓	!ATT2	TACS + SPAS → TADS, CIDS			200	ns
DAV ↓ → DMAREQ ↑	!DVRQ	ACRS → ACDS, LACS			600	ns
DAV ↓ → NRFD ↓	!DVNR1	ACRS → ACDS			350	ns
DAV ↓ → NDAC ↑	!DVND1	ACRS → ACDS → AWNS			650	ns
DAV ↑ → NDAC ↓	!DVND2	AWNS → ANRS			350	ns
DAV ↑ → NRFD ↑	!DVNR2	AWNS → ANRS → ACRS			350	ns
RD ↓ → NRFD ↑	!RNR	ANRS → ACRS LACS, DI reg. selected			500	ns
NDAC ↑ → DMAREQ ↑	!NDRQ	STRS → SWNS → SGNS, TACS			400	ns
NDAC ↑ → DAV ↑	!NDV	STRS → SWNS → SGNS			350	ns
WR ↑ → D <sub>O</sub>	!WDI	SGNS → SDYS, BO reg. selected			250	ns
NRFD ↑ → DAV ↓	!NRDV	SDYS → STRS, T <sub>1</sub> = True			350	ns
WR ↑ → DAV ↓	!WDV	SGNS → SDYS → STRS BO reg. selected, RFD = True Nf = f <sub>c</sub> = 8 MHz, T <sub>1</sub> (High Speed)			830 *tSYNC	ns
TRIG Pulse Width	!TRIG		50			ns

## Extended Temperature Range

T<sub>a</sub> = 25°C

PARAMETER	SYMBOL	TEST CONDITIONS	RATINGS	UNITS
Supply Voltage	V <sub>CC</sub>		-0.5 ~ +7.0	V
Input Voltage	V <sub>I</sub>		-0.5 ~ +7.0	V
Output Voltage	V <sub>O</sub>		-0.5 ~ +7.0	V
Operating Temperature	T <sub>opt</sub>		-40 ~ +85	°C
Storage Temperature	T <sub>stg</sub>		-65 ~ +125	°C

T<sub>a</sub> = -40°C...+85°C, V<sub>CC</sub> = +5V±10%

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
Input Low Voltage	V <sub>IL</sub>		-0.5		+0.6	V
Input High Voltage	V <sub>IH</sub>		+2.2		V <sub>CC</sub> +0.5	V
Low Level Output Voltage	V <sub>OL</sub>	I <sub>OL</sub> = 2mA (4mA: T/R1 Pin)			+0.45	V
High Level Output Voltage	V <sub>OH1</sub>	I <sub>OH</sub> = -400uA, Except INT	+2.3			V
		I <sub>OH</sub> = -400uA I <sub>OH</sub> = -50uA	INT Pin	+2.3		
				+3.4		
Input Leakage Current	I <sub>IL</sub>	I <sub>IN</sub> = 0V ~ V <sub>CC</sub>	-10		+10	uA
Output Leakage Current	I <sub>OL</sub>	I <sub>OUT</sub> = 0.45V ~ V <sub>CC</sub>	-10		+10	uA
Supply Current	I <sub>CC</sub>				+220	mA

T<sub>a</sub> = 25°C, V<sub>CC</sub> = GND = 0V

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
Input Capacitance	C <sub>IN</sub>	f = 1 MHz			10	pF
Output Capacitance	C <sub>OUT</sub>	All Pins Except Pin Under Test Tied to AC Ground			15	pF
I/O Capacitance	C <sub>I/O</sub>				20	pF

## Absolute Maximum Ratings

## DC Characteristics

## Capacitance

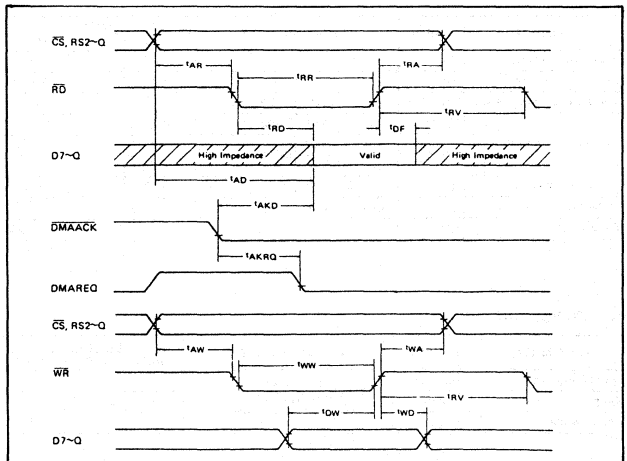


AC Characteristics

T<sub>a</sub> = -40° C ± 85° C, V<sub>CC</sub> = + 5 V ± 10%

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
Address Setup to RD	tAR	RS2-Q	85			ns
		CS	+ 20			
Address Hold from RD	tRA		0			ns
RD Pulse Width	tRR		170			ns
Data Delay from Address	tAD					ns
Data delay from RD ↓	tRD				275	ns
Output Float Delay from RD ↓	tDF		0		150	ns
RD Recovery Time	tRV		250		80	ns
DMAREQ ↓ Delay from DMAACK	tAKRQ					ns
Data Delay from DMAACK	tAKD				160	ns
Address Setup to WR	tAW		0		240	ns
Address Hold from WR	tWA		+ 20			ns
WR Pulse Width	tWW		170			ns
Data Setup to WR	tDW		150			ns
Data Hold from WR	tWD		+ 20			ns
WR Recovery Time	tRV		250			ns
EOI ↓ - DIO	tEODI	PPSS - PPAS, ATN = True			300	ns
EOI ↓ - T/R ↓	tEOT11	PPSS - PPAS, ATN = True			202	ns
EOI ↓ - T/R1 ↓	tEOT12	PPAS - PPSS, ATN = False			260	ns
ATN ↓ - NDAC ↓	tATND	AIDS - ANRS, LIDS			202	ns
ATN ↓ - T/R1 ↓	tATT1	TACS + SPAS - TADS, CIDS			202	ns
ATN ↓ - T/R2 ↓	tATT2	TACS + SPAS - TADS, CIDS			260	ns
DAV ↓ - DMAREQ ↓	tDVRQ	ACRS - ACDS, LACS			720	ns
DAV ↓ - NRFD ↓	tDVNR1	ACRS - ACDS			420	ns
DAV ↓ - NDAC ↓	tDVND1	ACRS - ACDS - AWNS			780	ns
DAV ↓ - NDAC ↓	tDVND2	AWNS - ANRS			420	ns
DAV ↓ - NRFD ↓	tDVNR2	AWNS - ANRS - ACRS			420	ns
RD ↓ - NRFD ↓	tRNR	ANRS - ACRS LACS, DI reg. selected			600	ns
NDAC ↓ - DMAREQ ↓	tNDRQ	STRS - SWNS - SGNS, TACS			480	ns
NDAC ↓ - DAV ↓	tNDDV	STRS - SWNS - SGNS			420	ns
WR ↓ - DIO	tWDI	SGNS - SDYS, Bo reg. selected			300	ns
NRFD ↓ - DAV ↓	tNRDV	SDYS - STRS, T <sub>1</sub> = True			420	ns
WR ↓ - DAV ↓	tWDV	SGNS - SDYS - STRS BO reg. selected, RFD=True NF=f <sub>c</sub> =8 MHz, T <sub>1</sub> (High Speed)			860 +1SYNC	ns
TRIG Pulse Width	tTRIG		45			ns

Timing Waveforms



## Description

The μPD7220A high-performance graphics display controller (GDC) is an intelligent microprocessor peripheral designed to be the heart of a high-performance raster scan computer graphics and character display system. Positioned between the video display memory and the microprocessor bus, the GDC performs the tasks needed to generate the raster display and manage the display memory. Processor software overhead is minimized by the GDC's sophisticated instruction set, graphics figure drawing, and DMA transfer capabilities. The display memory supported by the GDC can be configured in any number of formats and sizes up to 256K 16-bit words. The display can be zoomed and panned, while partitioned screen areas can be independently scrolled. With its light pen input and multiple controller capability, the GDC is ideal for advanced computer graphics applications.

For a more detailed description of the GDC's operation, please refer to the 7220/7220A product description and users manual.

## System Considerations

The GDC is designed to work with a general purpose microprocessor to implement a high-performance computer graphics system. Through the division of labor established by the GDC's design, each of the system components is used to the maximum extent through a six-level hierarchy of simultaneous tasks. At the lowest level, the GDC generates the basic video raster timing, including sync and blanking signals. Partitioning areas on the screen and zooming are also accomplished at this level. At the next level, video display memory is modified during the figure drawing operations and data moves. Third, display memory addresses are calculated pixel by pixel as drawing progresses. Outside the GDC at the next level, preliminary calculations are done to prepare drawing parameters. At the fifth level, the picture must be represented as a list of graphics figures drawable by the GDC. Finally, this representation must be manipulated, stored, and communicated. By handling the first three levels, the GDC takes care of the high-speed and repetitive tasks required to implement a graphics system.

## Features

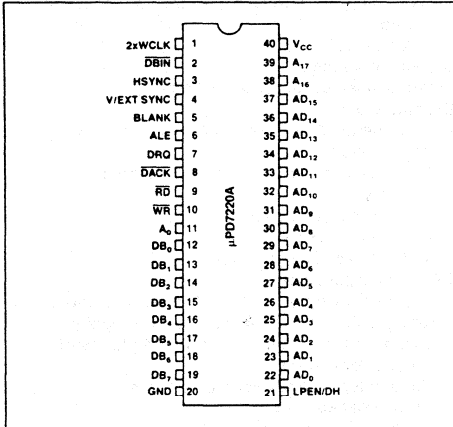
- Microprocessor interface
  - DMA transfers with 8257- or 8237-type controllers
  - FIFO command buffering
- Display memory interface
  - Up to 256K words of 16-bits
  - Read-modify-write (RMW) display memory cycles as fast as 500 ns
  - Dynamic RAM refresh cycles for nonaccessed memory
- Light pen input
- Drawing hold input
- External video synchronization mode
- Graphic mode
  - Four megabit, bit-mapped display memory
- Character mode
  - 8K character code and attributes display memory
- Mixed graphics and character mode
  - 64K if all characters
  - 1 megapixel if all graphics
- Graphics capabilities
  - Figure drawing of lines, arc/circles, rectangles, and graphics characters in 500 ns per pixel
  - Display 1024-by-1024 pixels with 4 planes of color or grayscale
  - Two independently scrollable areas
- Character capabilities
  - Auto cursor advanced
  - Four independently scrollable areas
  - Programmable cursor height
  - Characters per row: up to 256
  - Character rows per screen: up to 100
- Video display format
  - Zoom magnification factors of 1 to 16
  - Panning
  - Command-settable video raster parameters
- NMOS technology
- Single +5 V power supply
- DMA capability
  - Bytes or word transfers
  - 4 clock periods per byte transferred
- On-chip pull-up resistor for VSYNC/EXT, HSYNC and DACK, and a pull-down resistor for LPEN/DH

## μPD7220A

### Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD7220AD	40-pin ceramic DIP	6 MHz
μPD7220AD-1	40-pin ceramic DIP	7 MHz
μPD7220AD-2	40-pin ceramic DIP	8 MHz

### Pin Configuration



### Character Mode Pin Utilization

Pin		
No.	Symbol	Function
35-37	AD <sub>13</sub> -AD <sub>15</sub>	Line counter bits 0 to 2 outputs
38	AD <sub>16</sub>	Line counter bit 3 output
39	AD <sub>17</sub>	Cursor output and line counter bit 4

### Mixed Mode Pin Utilization

Pin		
No.	Symbol	Function
35-37	AD <sub>13</sub> -AD <sub>15</sub>	Address and data bits 13 to 15
38	A <sub>16</sub>	Attribute blink and clear line counter output
39	A <sub>17</sub>	Cursor and bit-map area flag output

### Pin Identification

Pin		
No.	Symbol	Function
1	2xWCLK	Clock input
2	DBIN	Display memory read input flag
3	HSYNC	Horizontal video sync output
4	V/EXT SYNC	Vertical video sync output or external VSYNC input
5	BLANK	CRT blanking output
6	ALE	Address latch enable output
7	DRQ	DMA request output
8	DACK	DMA acknowledge input
9	RD	Read strobe input for microprocessor interface
10	WR	Write strobe input for microprocessor interface
11	A <sub>0</sub>	Address select input for microprocessor interface
12-19	DB <sub>0</sub> -DB <sub>7</sub>	Bidirectional data bus to host microprocessor
20	GND	Ground
21	LPEN/DH	Light pen detect input drawing hold input
22-34	AD <sub>0</sub> -AD <sub>12</sub>	Address data lines to display memory
35-37	AD <sub>13</sub> -AD <sub>15</sub>	Utilization varies with mode of operation
38	A <sub>16</sub>	Utilization varies with mode of operation
39	A <sub>17</sub>	Utilization varies with mode of operation
40	V <sub>CC</sub>	+5 V ±10% power supply

### Graphics Mode Pin Utilization

Pin		
No.	Symbol	Function
35-37	AD <sub>13</sub> -AD <sub>15</sub>	Address and data bits 13 to 15
38	A <sub>16</sub>	Address bit 16 output
39	A <sub>17</sub>	Address bit 17 output

## Pin Functions

### 2xWCLK [Clock Input]

2xWCLK is the clock input.

### DBIN [Data Bus Input Enable]

The DBIN output indicates the time the GDC will accept data read from display RAM during read-modify-write (RMW) cycles.

### HSYNC [Horizontal Sync]

The HSYNC output indicates the time the CRT's beam is to start its retrace back to the left side of the screen.

### V/EXT SYNC [Vertical SYNC Output/External Sync Input]

The GDC can be programmed to output a vertical sync signal at the start of the return of the CRT's beam from the lower right of the screen to the upper left during vertical retrace. The GDC may also be programmed to accept an external sync input when used in slave mode.

### BLANK [Blank]

BLANK is output during inactive display times (horizontal and vertical retrace) of the CRT and during a read-modify-write memory cycle when in flash mode.

### ALE [Address Latch Enable]

The falling edge of ALE indicates the first clock cycle of a display memory cycle and the availability of the memory address on pins AD<sub>0</sub>-AD<sub>17</sub>. ALE and external logic can generate display memory control signals.

### A<sub>0</sub> [Address Bit 0]

A<sub>0</sub> is the address select input for the microprocessor interface.

### A<sub>1</sub> [Address Bit 1]

The A<sub>1</sub> input selects registers when reading or writing to the GDC.

### DACK [DMA Acknowledge]

DACK is the DMA acknowledge Input handshake line that directly interfaces to the μPD8257 or μPD8237 DMA controller.

### DRQ [DMA Request]

DRQ is the DMA request output handshake line that directly interfaces to the μPD8257 or μPD8237 DMA controller.

### $\overline{RD}$ [Read Strobe]

The host CPU clears the  $\overline{RD}$  input to 0 when reading the internal status and FIFO registers.

### $\overline{WR}$ [Write Strobe]

The host CPU clears  $\overline{WR}$  to 0 when writing to the internal command and parameter registers.

### DB<sub>0</sub>-DB<sub>7</sub> [Data Bus]

DB<sub>0</sub>-DB<sub>7</sub>, the 8-bit, three-state bidirectional data bus transfers data to and from the host CPU via the system bus.

### LPEN/DH [Light Pen/Drawing Hold]

The LPEN/DH input can be programmed as either a light pen input or drawing hold input. The drawing hold input halts all read-modify-write operations.

### AD<sub>0</sub>-AD<sub>17</sub> [Address and Data Lines]

AD<sub>0</sub>-AD<sub>17</sub> are address and data lines to display memory. AD<sub>13</sub>-AD<sub>17</sub> functions vary with the mode of operation of the GDC. The μPD7220/7220A Graphics Display Controller User's Manual describes these functions and modes of operation.

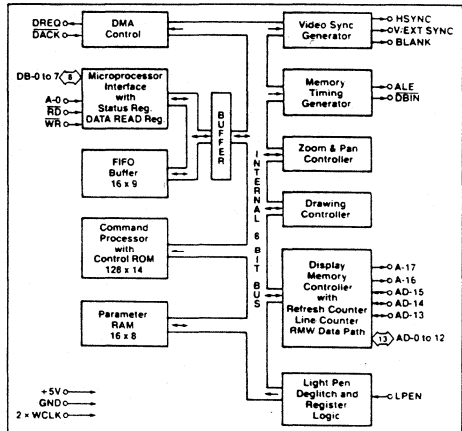
### VCC [Power Supply]

VCC is the +5 V power supply input.

### GND [Ground]

GND is ground potential.

## Block Diagram



## **HGDC Components**

### **Microprocessor Bus Interface**

Control of the GDC by the system microprocessor is achieved through an 8-bit bidirectional interface. The status register is readable at any time. Access to the FIFO buffer is coordinated through flags in the status register and operates independently of the various internal GDC operations, due to the separate data bus connecting the interface and the FIFO buffer.

### **Command Processor**

The contents of the FIFO are interpreted by the command processor. The command bytes are decoded, and the succeeding parameters are distributed to their proper destinations within the GDC. The command processor yields to the bus interface when both access the FIFO simultaneously.

### **DMA Control**

The DMA control circuitry in the GDC coordinates transfers over the microprocessor interface when using an external DMA controller. The DMA Request and Acknowledge handshake lines directly interface with a μPD8257 or μPD8237 DMA controller, so that display data can be moved between the microprocessor memory and the display memory.

### **Parameter RAM**

The 16-byte RAM stores parameters that are used repetitively during the display and drawing processes. In character mode, this RAM holds four sets of partitioned display area parameters; in graphics mode, the drawing pattern and graphics character take the place of two of the sets of parameters.

### **Video Sync Generator**

Based on the clock input, the sync logic generates the raster timing signals for almost interlaced, non-interlaced, or "repeat field" interlaced video format. The generator is programmed during the idle period following a reset. In video sync slave mode, it coordinates timing between multiple GDC's.

### **Memory Timing Generator**

The memory timing circuitry provides two memory cycle types: a two-clock period refresh cycle and the read-modify-write (RMW) cycle, which takes four clock periods. The memory control signals needed to drive the display memory devices are easily generated from the GDC's ALE and DBIN outputs.

### **Zoom & Pan Controller**

Based on the programmable zoom display factor and the display area entries in the parameter RAM, the zoom and pan controller determines when to advance to the next memory address for display refresh and when to go on to the next display area. A horizontal zoom is produced by slowing down the display refresh rate while maintaining the video sync rates. Vertical zoom is accomplished by repeatedly accessing each line a number of times equal to the horizontal repeat. Once the line count for a display area is exhausted, the controller accesses the starting address and line count of the next display area from the parameter RAM. The system microprocessor, by modifying a display area starting address, can pan in any direction, independently of the other display areas.

### **Drawing Controller**

The drawing processor contains the logic necessary to calculate the addresses and positions of the pixels of the various graphics figures. Given a starting point and the appropriate drawing parameters, the drawing controller needs no further assistance to complete the figure drawing.

## Display Memory Controller

The display memory controller's tasks are numerous. Its primary purpose is to multiplex the address and data information in and out of the display memory. It also contains the 16-bit logic unit used to modify the display memory contents during RMW cycles, the character mode line counter, and the refresh counter for dynamic RAMs. The memory controller apportions the video field time between the various types of cycles.

## Light Pen Deglitcher/Drawing Hold

Only if two rising edges on the light pen input occur at the same point during successive video fields are the pulses accepted as a valid light pen detection. A status bit indicates to the system microprocessor that the light pen register contains a valid address. If this input is held high for a period greater than four  $2xWCLK$  cycles, drawing execution is halted when bit 7 of  $P5$  of the SYNC command is set.

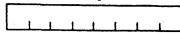
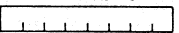
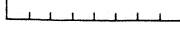
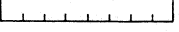
## Programmer's View of GDC

The GDC occupies two addresses on the system microprocessor bus through which the GDC's status register and FIFO are accessed. Commands and parameters are written into the GDC's FIFO and are differentiated based on address bit  $A_0$ . The status address line.

Commands to the GDC take the form of a command byte followed by a series of parameter bytes as needed for specifying the details of the command. The command processor decodes the commands, unpacks the parameters, loads them into the appropriate registers within the GDC, and initiates the required operations.

The commands available the GDC can be organized into five categories as described in the following section.

## GDC Microprocessor Bus Interface Registers

A0	READ	WRITE
0	Status Register 	Parameter Into FIFO 
1	FIFO Read 	Command Into FIFO 

## GDC Commands Summary

### Video Control Commands

1. RESET1 Resets the GDC to its idle state. Resynchronizes video timing. Blanks the display.
2. RESET2 Resets the HGDC to its idle state. Does not resynchronize video timing. Blanks the display.
3. RESET3 Resets the HGDC to its idle state. Does not resynchronize video timing. Does not blank the display
4. SYNC Specifies the video display format.
5. VSYNC Selects master or slave video synchronization mode.
6. CCHAR Specifies the cursor and character row heights.

### Display Control Commands

1. START Ends idle mode and unblanks the display.
2. BLANK1 Controls the blanking and unblanking of the display, along with video resynchronization.
3. BLANK2 Controls the blanking and unblanking of the display. Does not blank the display.
4. ZOOM Specifies zoom factors for the display and graphics characters writing.
5. CURS Sets the position of the cursor in display memory.
6. PRAM Defines starting addresses and lengths of the display areas and specifies the eight bytes for the graphics character.
7. PITCH Specifies the width of the X dimension of display memory.

### Drawing Control Commands

1. WDAT Writes data words or bytes into display memory.
2. MASK Sets the mask register contents.
3. FIGS Specifies the parameters for the drawing controller.
4. FIGD Draws the figure as specified above.
5. GCHRD Draws the graphics character into display memory.

### Data Read Commands

1. RDAT Reads data words or bytes from display memory.
2. CURD Reads the cursor position.
3. LPRD Reads the light pen address.

### DMA Control Commands

1. DMAR Requests a DMA read transfer.
2. DMAW Requests a DMA write transfer.

### Status Register Flags

#### SR-7: Light Pen Detect

When this bit is set to 1, the light pen address (LAD) register contains a deglitched value that the system microprocessor may read. This flag is reset after the 3-byte LAD is moved into the FIFO in response to the light pen read command.

#### SR-6: Horizontal Blank Active/Vertical Blank Active

A 1 value for this flag signifies that horizontal retrace blanking or vertical retrace blanking is currently underway dependent on the status of the VH bit in SYNC or the RESETx parameter 6.

#### SR-5: Vertical Sync

Vertical retrace sync occurs while this flag is a 1. The vertical sync flag coordinates display format modifying commands to the blanked interval surrounding vertical sync. This eliminates display disturbances.

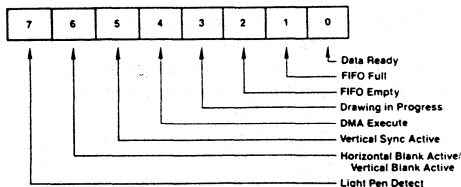
#### SR-4: DMA Execute

This bit is a 1 during DMA data transfers.

#### SR-3: Drawing in Progress

While the GDC is drawing a graphics figure, this status bit is a 1.

#### Status Register (SR)



#### SR-2: FIFO Empty

This bit and the FIFO-full flag coordinate system microprocessor accesses with the GDC FIFO. When it is 1, the Empty flag ensures that all the commands and parameters previously sent to the GDC have been interpreted.

#### SR-1: FIFO Full

A 1 at this flag indicates a full FIFO in the GDC. A 0 ensures that there is room for at least one byte. This flag needs to be checked before each write into the HGDC.

#### SR-0: Data Ready

When this flag is a 1, it indicates that a byte is available to be read by the system microprocessor. This bit must be tested before each read operation. It drops to a 0 while the data is transferred from the FIFO into the microprocessor interface data register.

### FIFO Operation & Command Protocol

The first-in, first-out buffer (FIFO) in the GDC handles the command dialogue with the system microprocessor. This flow of information uses a half-duplex technique, in which the single 16-location FIFO is used for both directions of data movement, one direction at a time. The FIFO's direction is controlled by the system microprocessor through the GDC's command set. The host microprocessor coordinates these transfers by checking the appropriate status register bits.

The command protocol used by the GDC requires differentiation of the first byte of a command sequence from the succeeding bytes. The first byte contains the operation code and the remaining bytes carry parameters. Writing into the GDC causes the FIFO to store a flag value alongside the data byte to signify whether the byte was written into the command or the parameter address. The command processor in the GDC tests this bit as it interprets the entries in the FIFO.

The receipt of a command byte by the command processor marks the end of any previous operation. The number of parameter bytes supplied with a command is cut short by the receipt of the next command byte. A read operation from the GDC to the microprocessor can be terminated at any time by the next command.

The FIFO changes direction under the control of the system microprocessor. Commands written into the GDC always put the FIFO into write mode if it was not in it already. If it was in read mode, any read data in the FIFO at the time of the turnaround is lost. Commands which require an GDC response, such as RDAT, CURD and LPRD, put the FIFO into read mode after the command is interpreted by the GDC's command processor. Any commands and parameters behind the read-evoking command are discarded when the FIFO direction is reversed.

## Read-Modify-Write Cycle

Data transfers between the GDC and the display memory are accomplished using a read-modify-write (RMW) memory cycle. The four-clock period timing of the RMW cycle is used to 1. output the address, 2. read data from the memory, 3. modify the data, and 4. write the modified data back into the initially selected memory address. This type of memory cycle is used for all interactions with display memory including DMA transfers, except for the two-clock period display and RAM refresh cycles.

The operations performed during the modify portion of the RMW cycle merit additional explanation. The circuitry in the GDC uses three main elements: the Pattern register, the Mask register, and the 16-bit Logic unit. The Pattern register holds the data pattern to be moved into memory. It is loaded by the WDAT parameters or, during drawing, from the parameter RAM. The Mask register contents determine which bits of the read data will be modified. Based on the contents of these registers, the Logic unit performs the selected operations of REPLACE, COMPLEMENT, SET, or CLEAR on the data read from display memory.

The Pattern register contents are ANDed with the Mask register contents to enable the actual modification of the memory read data, on a bit-by-bit basis. For graphics drawing, one bit at a time from the Pattern register is combined with the Mask. When ANDed with the bit set to a 1 in the Mask register, the proper single pixel is modified by the Logic unit. For the next pixel in the figure, the next bit in the Pattern register is selected and the Mask register bit is moved to identify the pixel's location within the word. The Execution word address pointer register, EAD, is also adjusted as required to address the word containing the next pixel.

In character mode, all of the bits in the Pattern register are used in parallel to form the respective bits of the modify data word. Since the bits of the character code word are used in parallel, unlike the one-bit-at-a-time graphics drawing process, this facility allows any or all of the bits in a memory word to be modified in one RMW memory cycle. The Mask register must be loaded with ones in the positions where modification is to be permitted.

The Mask register can be loaded in either of two ways. In graphics mode, the CURS command contains a 4-bit dAD field to specify the dot address. The command processor converts this parameter into the 1-of-16 format used in the Mask register for figure drawing. A full 16-bits can be loaded into the Mask register using the MASK command. In addition to the character mode use mentioned above, the 16-bit MASK load is convenient in graphics mode when all of the pixels of a word are to be set to the same value.

The Logic unit combines the data read from display memory, the Pattern register, and the Mask register to generate the data to be written back into display memory. Any one of four operations can be selected: REPLACE, COMPLEMENT, CLEAR or SET. In each case, if the respective Mask bit is 0, that particular bit of the read data is returned to memory unmodified. If the Mask bit is 1, the modification is enabled. With the REPLACE operation, the Pattern register data simply takes the place of the read data for modification enabled bits. For the other three operations, a 0 in the modify data allows the read data bit to be returned to memory. A 1 value causes the specified operation to be performed in the bit positions with set Mask bits.

## Figure Drawing

The GDC draws graphics figures at the rate of one pixel per read-modify-write (RMW) display memory cycle. These cycles take four clock periods to complete. At a clock frequency of 8 MHz, this is equal to 500 ns. During the RMW cycle the GDC simultaneously calculates the address and position of the next pixel to be drawn.

The graphics figure drawing process depends on the display memory addressing structure. Groups of 16 horizontally adjacent pixels form the 16-bit words which are handled by the GDC. Display memory is organized as a linearly addressed space of these words. Addressing of individual pixels is handled by the GDC's internal RMW logic.

5



During the drawing process, the GDC finds the next pixel of the figure which is one of the eight nearest neighbors of the last pixel drawn. The GDC assigns each of these eight directions a number from 0 to 7, starting with straight down and proceeding counter-clockwise.

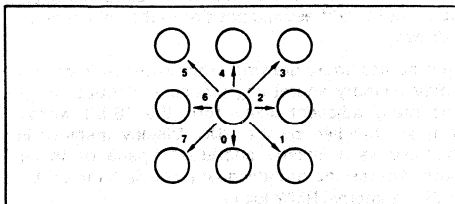
Figure drawing requires the proper manipulation of the address and the pixel bit position according to the drawing direction to determine the next pixel of the figure. To move to the word above or below the current one, it is necessary to subtract or add the number of words per line in display memory. This parameter is called the pitch. To move to the word to either side, the Execute word address cursor, EAD, must be incremented or decremented as the dot address pointer bit reaches the LSB or the MSB of the Mask register. To move to a pixel within the same word, it is necessary to rotate the dot address pointer to the right or left. The table below summarizes these operations for each direction.

Dir	Operations to Address the Next Pixel
000	EAD - P → EAD
001	EAD - P → EAD dAD (MSB) = 1: EAD - 1 → EAD dAD → LR
010	dAD (MSB) = 1: EAD - 1 → EAD dAD → LR
011	EAD - P → EAD dAD (MSB) = 1: EAD - 1 → EAD dAD → LR
100	EAD - P → EAD
101	EAD - P → EAD dAD (LSB) = 1: EAD - 1 → EAD dAD → RR
110	dAD (LSB) = 1: EAD - 1 → EAD dAD → RR
111	EAD - P → EAD dAD (LSB) = 1: EAD - 1 → EAD dAD → RR

**Note:**

P = Pitch, LR = Left Rotate, RR = Right Rotate, EAD = Execute Word Address, and dAD = Dot Address stored in the Mask register.

**Drawing Directions**



Whole word drawing is useful for filling areas in memory with a single value. By setting the Mask register to all 1s with the MASK command, both the LSB and MSB of the dAD will always be 1, so that the EAD value will be incremented or decremented for each cycle regardless of direction. One RMW cycle will be able to affect all 16 bits of the word for any drawing type. One bit in the Pattern register is used per RMW cycle to write all the bits of the word to the same value. The next Pattern bit is used for the word, etc.

For the various figures, the effect of the initial direction upon the resulting drawing is shown below:

Dir.	Line	Arc	Character	Slant Char	Rectangle	DMA
000						
001						
010						
011						
100						
101						
110						
111						

Note that during line drawing, the angle of the line may be anywhere within the shaded octant defined by the DIR value. Arc drawing starts in the direction initially specified by the DIR value and veers into an arc as drawing proceeds. An arc may be up to 45° in length. DMA transfers are done on word boundaries only, and follow the arrows indicated in the table to find successive word addresses. The slanted paths for DMA transfers indicate the GDC changing both the X and Y components of the word address when moving to the next word. It does not follow a 45° diagonal path by pixels.

## Drawing Parameters

In preparation for graphics figure drawing, the GDC's Drawing processor needs the figure type, direction and drawing parameters, the starting pixel address, and the pattern from the microprocessor. Once these are in place within the GDC, the Figure Draw command, FIGD, initiates the drawing operation. From that point on, the system microprocessor is not involved in the drawing process. The GDC Drawing controller coordinates the RMW circuitry and address registers to draw the specified figure pixel by pixel.

The algorithms used by the processor for figure drawing are designed to optimize its drawing speed. To this end, the specific details about the figure to be drawn are reduced by the microprocessor to form conducive to high-speed address calculations within the GDC. In this way the repetitive, pixel-by-pixel calculations can be done quickly, thereby minimizing the overall figure drawing time. The table below summarizes the parameters.

Drawing Type	DC	D	D2	D1	DM
Initial Value (1)	0	8	8	-1	-1
Line	$ \Delta x , 2 \Delta y , - \Delta x , 2( \Delta x  -  \Delta y ), 2 \Delta y , -$				
Arc (2)	$r \sin \theta$	$r-1$	$2(r-1)$	-1	$r \sin \theta$
Rectangle	3	A-1	B-1	-1	A-1
Area fill	B-1	A	A	-	-
Graphic character (3)	B-1	A	A	-	-
Read & write data	W-1	-	-	-	-
DMAW	D-1	C-1	-	-	-
DMAR	D-1	C-1	$(C-1)/2 \uparrow$	-	-

### Note:

All numbers are shown in base 10 for convenience. The HGDC accepts base 2 numbers (2's complement notation) where appropriate.

- (1) Initial values for the various parameters remain as each drawing process ends.
- (2) Circles are drawn with 8 arcs, each of which span  $45^\circ$ , so that  $\sin \theta = 1/\sqrt{2}$  and  $\sin \theta = 0$ .
- (3) Graphic characters are a special case of bit-map area filling in which B and A  $\leq 8$ . If A = 8 there is no need to load D and D2.

## Symbol Definitions

- 1 = All ONES value.
- = No parameter bytes sent to GDC for this parameter.
- $\Delta l$  = The larger at  $\Delta x$  or  $\Delta y$ .
- $\Delta D$  = The smaller at  $\Delta x$  or  $\Delta y$ .
- r = Radius of curvature, in pixels.
- $\phi$  = Angle from major axis to end of the arc.  
 $\phi \leq 45^\circ$ .
- $\theta$  = Angle from major axis to start of the arc.  
 $\theta \leq 45^\circ$ .
- $\uparrow$  = Round up to the next higher integer.
- $\downarrow$  = Round down to the next lower integer.
- A = Number of pixels in the initially specified direction.
- B = Number of pixels in the direction at right angles to the initially specified direction.
- W = Number of words to be accessed.
- C = Number of bytes to be transferred in the initially specified direction. (Two bytes per word if word transfer mode is selected.)
- D = Number of words to be accessed in the direction at right angles to the initially specified direction.
- DC = Drawing count parameter which is one less than the number of RMW cycles to be executed.
- DM = Dots masked from drawing during arc drawing.
- $\uparrow$  = Needed only for word reads.

## Graphics Character Drawing

Graphics characters can be drawn into display memory pixel by pixel. The up to 8-by-8 character display is loaded into the GDC's parameter RAM by the system microprocessor. Consequently, there are no limitations on the character set used. By varying the drawing parameters and drawing direction, numerous drawing options are available. In area fill applications, a character can be written into display memory as many times as desired without reloading the parameter RAM.

Once the parameter RAM has been loaded with up to eight graphics character bytes by the appropriate PRAM command, the GCHRD command can be used to draw the bytes into display memory starting at the cursor. The zoom magnification factor for writing, set by the ZOOM command, controls the size of the character written into the display memory in integer multiples of 1 through 16. The bit values in the PRAM are repeated horizontally and vertically the number of times specified by the zoom factor.

The movement of these PRAM bytes to the display memory is controlled by the parameters of the FIGS command.

Based on the specified height and width of the area to be drawn, the parameter RAM is scanned to fill the required area.

For an 8-by-8 graphics character, the first pixel drawn uses the LSB of RA-15, the second pixel uses bit 1 of RA-15, and so on, until the MSB of RA-15 is reached.

The GDC jumps to the corresponding bit in RA-14 to continue the drawing. The progression then advances toward the LSB of RA-14. This snaking sequence is continued for the other 6 PRAM bytes. This progression matches the sequence of display memory addresses calculated by the drawing processor as shown above. If the area is narrower than 8 pixels wide, the snaking will advance to the next PRAM byte before the MSB is reached. If the area is less than 8 lines high, fewer bytes in the parameter RAM will be scanned. If the area is larger than 8 by 8, the GDC will repeat the contents of the parameter RAM in two dimensions, as required to fill the area with the 8-by-8 mosaic. (Fractions of the 8-by-8 pattern will be used to fill areas which are not multiples of 8 by 8.)

**Parameter RAM Contents: RAM Address RA-0 to RA-15**

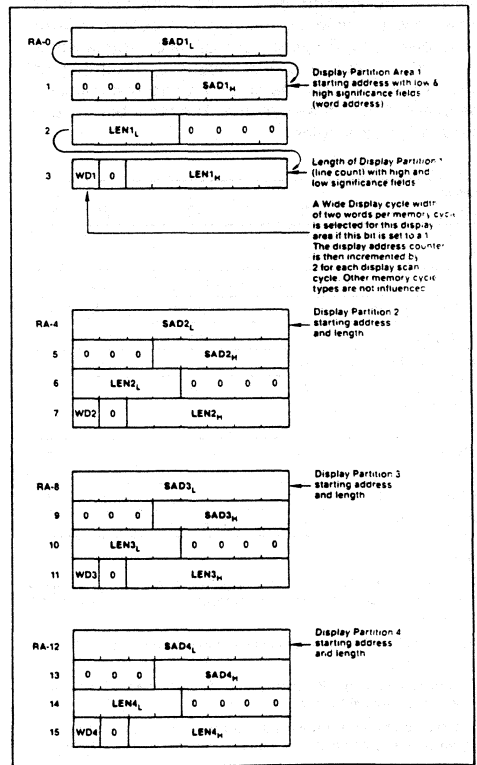
The parameters stored in the parameter RAM, PRAM, are available for the GDC to refer to repeatedly during figure drawing and raster-scanning. In each mode of operation the values in the PRAM are interpreted by the GDC in a predetermined fashion. The host microprocessor must load the appropriate parameters into the proper PRAM locations. PRAM loading command allows the host to write into any location of the PRAM and transfer as many bytes as desired. In this way any stored parameter byte or bytes may be changed without influencing the other bytes.

The PRAM stores two types of information. For specifying the details of the display area partitions, blocks of four bytes are used. The four parameters stored in each block include the starting address in display memory of each display area, and its length. In addition, there are two mode bits for each area which specify whether the area is a bit-mapped graphics area or a coded-character area, and whether a 16-bit or a 32-bit wide display cycle is to be used for that area.

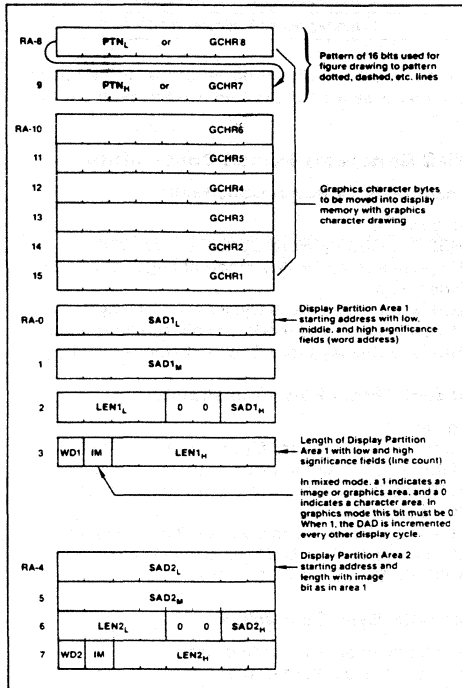
The other use for the PRAM contents is to supply the pattern for figure drawing when in a bit-mapped graphics area or mode. In these situations, PRAM bytes 8 through 16 are reserved for this patterning information. For line, arc, and rectangle drawing (linear figures) locations 8 and 9 are loaded into the Pattern register to allow the GDC to draw dotted, dashed, etc. lines. For area filling and graphics bit-mapped character drawing, locations 8 through 15 are referenced for the pattern or character to be drawn.

Details of the bit assignments are shown for the various modes of operation.

**Character Mode**



## Graphics and Mixed Graphics and Character Modes



## Command Bytes Summary

START	0 1 1 0 1 0 1 1
ZOOM	0 1 0 0 0 1 1 0
CURS	0 1 0 0 1 0 0 1
PRAM	0 1 1 1 SA
PITCH	0 1 0 0 0 1 1 1
WDAT	0 0 1 TYPE 0 MOD
MASK	0 1 0 0 1 0 1 0
FIGS	0 1 0 0 1 1 0 0
FIGD	0 1 1 0 1 1 0 0
GCHRD	0 1 1 0 1 0 0 0
RDAT	1 0 1 TYPE 0 MOD
CURD	1 1 1 0 0 0 0 0
LPRD	1 1 0 0 0 0 0 0
DMAR	1 0 1 TYPE 1 MOD
DMAW	0 0 1 TYPE 1 MOD

## Command Bytes Summary

RESET1	0 0 0 0 0 0 0 0
RESET2	0 0 0 0 0 0 0 1
RESET3	0 0 0 0 1 0 0 1
BLANK1	0 0 0 0 1 1 0 DE
BLANK2	0 0 0 0 0 1 0 DE
SYNC	0 0 0 0 1 1 1 DE
VSYNC	0 1 1 0 1 1 1 M
CCHAR	0 1 0 0 1 0 1 1

## Video Control Commands

### Reset

RESETX:	0 0 0 0 0 0 0 0	Blank the display, enter idle mode, and initialize within the HGDC — FIGO — Command Processor — Internal Counters
---------	-----------------	--

This command can be executed at any time and does not modify any of the parameters already loaded into the GDC.

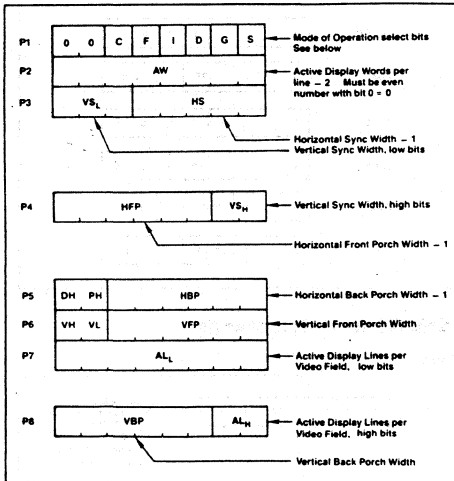
If followed by the parameter bytes, this command also sets the sync generator parameters as described below. Idle mode is exited with the START command.

- RESET1: Resync video timing in slave mode.
- RESET2: Blank the display and so not resync.
- RESET3: Unblank the display and do not resync.

In graphics mode, a word is a group of 16 pixels. In character mode, a word is one character code and its attributes, if any. The number of active words per line must be an even number from 2 to 256. An all-zero parameter value selects a count equal to  $2^n$  where  $n$  = number of bits in the parameter field for vertical parameters. All horizontal widths are counted in display words. All vertical intervals are counted in lines.

If the Drawing Hold (DH) is set to one, pin 21 (LPEN/DH) is used as the drawing hold control pin. When the input to LPEN/DH is held high for over four 2 x WCLK clocks, the drawing address output is temporarily held and the display address is output.

The GDC allows an even or odd number of lines per frame. Selection is via the VL flag, the seventh bit of the sixth parameter byte following a RESET or SYNC command. When VL is 0, an odd number of display lines is generated.



VL	Number of lines in interlaced mode
0	Odd, as in 7220
1	Even

When VH = 0, status operation is as in μPD7220.

VH	Blank Status Bit Definition
0	Status register bit 6 indicates horizontal blank
1	Status register bit 6 indicates vertical blank

PH is the most significant bit (9) of the display pitch parameter. Use the PITCH command to set the lower eight bits.

### SYNC Generator Period Constraints

#### Horizontal Back Porch Constraints

- In general:  
HBP  $\geq$  3 Display Word Cycles (6 clock cycles).
- If the Image bit or WD mode changes within one video field:  
HBP  $\geq$  5 Display Word Cycles (10 clock cycles).
- If interlaced, mixed mode, or split screen is used:  
HBP  $\geq$  5 Display Word Cycles (10 clock cycles).

#### Horizontal Front Porch Constraints

- In general:  
HFP  $\geq$  2 Display Word Cycles (4 clock cycles).
- If the GDC is used in video sync Slave mode:  
HFP  $\geq$  4 Display Word Cycles (8 clock cycles).
- If the Light Pen is used:  
HFP  $\geq$  6 Display Word Cycles (12 clock cycles).
- If interlaced mode, DMA, or ZOOM is used:  
HFP  $\geq$  3 Display Word Cycles (6 clock cycles).

#### Horizontal Sync Constraints

- If interlaced display mode is used:  
HS  $\geq$  5 Display Word Cycles (6 clock cycles).
- If DRAM Refresh is enabled:  
HS  $\geq$  2 Display Word Cycles (4 clock cycles).

#### Modes of Operation Bits

C	G	Display Mode
0	0	Mixed graphics and character
0	1	Graphics mode
1	0	Character mode
1	1	Invalid

I	S	Video Framing
0	0	Non-interlaced
0	1	Invalid
1	0	Interlaced repeat field for character displays
1	1	Interlaced

- Repeat Field Framing: 2 field sequence with 1/2 line offset between otherwise identical fields.
- Interlaced Framing: 2 field sequence with 1/2 line offset. Each field displays alternate lines.
- Non-Interlaced Framing: 1 field brings all the information to the screen.

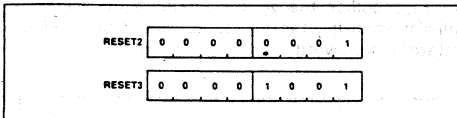
<b>0</b>	<b>Dynamic RAM Refresh Cycles Enable</b>
0	No refresh—static RAM
1	Refresh—dynamic RAM

Dynamic RAM refresh is important when high display zoom factors or DMA are used in such a way that not all of the rows in the RAMs are regularly accessed during display raster generation and for otherwise inactive display memory.

<b>F</b>	<b>Drawing Time Window</b>
0	Drawing during active display time and retrace blanking
1	Drawing only during retrace blanking

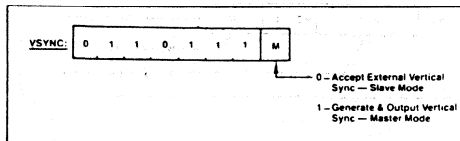
Access to display memory can be limited to retrace blanking intervals only, so that no disruptions of the image are seen on the screen.

Both commands allow a reset while preventing re-initialization of the internal sync generator by an external sync source (slave mode).



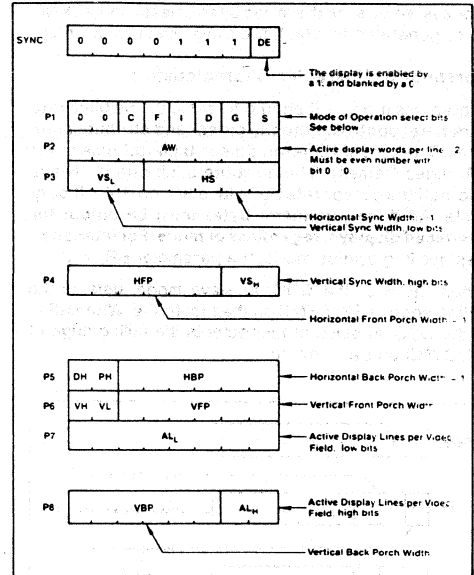
### Vertical Sync Mode

When using two or more GDCs to contribute to one image, one GDC is defined as the master sync generator, and the others operate as its slaves. The VSYNC pins of all GDCs are connected together.



### SYNC Format Specify

This command also loads parameters into the sync generator. The various parameter fields and bits are identical to those at the RESET command. The GDC is not reset nor does it enter idle mode.



### Slave Mode Operation

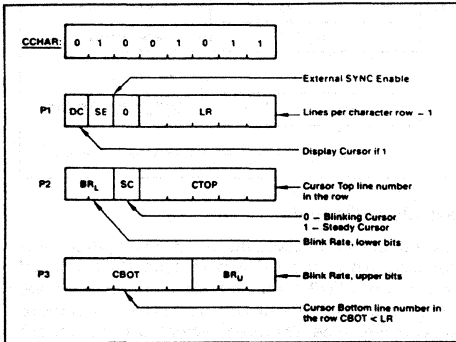
A few considerations should be observed when synchronizing two or more GDC's to generate overlaid video via the V/EXT SYNC pin. As mentioned above, the Horizontal Front Porch (HFP) must be four or more display cycles wide. This is equivalent to eight or more clock cycles. This gives the slave GDC's time to initialize their internal video sync generators to the proper point in the video field to match the incoming vertical sync pulse (VSYNC). This resetting of the generator occurs just after the end of the incoming VSYNC pulse, during the HFP interval. Enough time during HFP is required to allow the slave GDC to complete the operation before the start of the HSYNC interval.

Once the GDC's are initialized and set up as master and slaves, they must be given time to synchronize. It is a good idea to watch the VSYNC status bit of the master GDC and wait until after one or more VSYNC pulses have been generated before the display process is started. The START command will begin the active display of data and will end the video synchronization process, so be sure there has been at least one VSYNC pulse generated to which the slaves can synchronize.

**Cursor and Character Characteristics**

In graphics mode, LR should be set to 0. The blink rate parameter controls both the cursor and attribute blink rates. The cursor blink-on time = blink-off time = 2 x BR (video frames). The attribute blink rate is always one-half the cursor rate but with a 3/4-on-1/4-off duty cycle. All three parameter bytes must be output for interlaced displays, regardless of mode. For interlaced displays in graphics mode, the parameter BR<sub>L</sub> = 3.

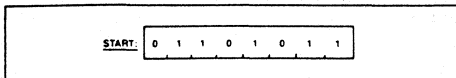
When SE = 0, the GDC, in slave mode, detects the falling edge of EX. SYNC on the first frame. When SE = 1, the GDC, in slave mode, detects the falling edge of EX. SYNC on every frame.



**Display Control Commands**

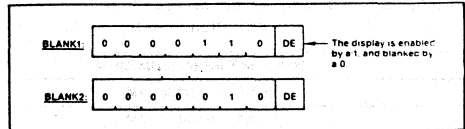
**Start Display and End Idle Mode**

The START command generates the video signals as specified by the RESETX or SYNC command.



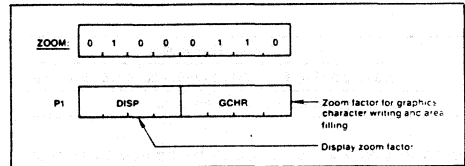
**Display Blanking Control**

BLANK2 does not cause the resyncing of an HGDC in slave mode. BLANK1 does cause the resyncing of an GDC in slave mode.



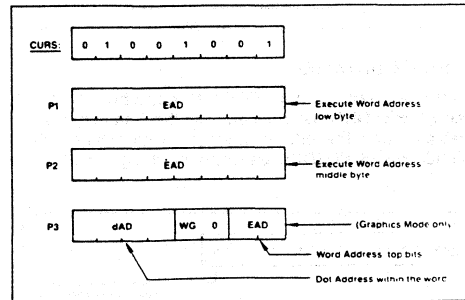
**Zoom Factors Specify**

Zoom magnification factors of 1 through 16 are available using codes 0 through 15, respectively.



**Cursor Position Specify**

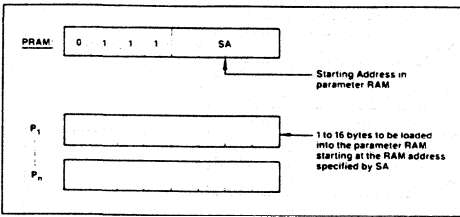
In character mode, the third parameter byte is not needed. The cursor is displayed for the word time in which the display scan address (DAD) equals the cursor address. In graphics mode, the cursor word address specifies the word containing the starting pixel of the drawing; the dot address value specifies the pixel within that word.



When the WG bit is set to one, any data following the WDAT command is written as is. When the WG bit is set to zero, the 7220A performs as the 7220 does: The pattern written is determined by the least significant bit of each parameter byte following the WDAT command. This bit is expanded into 16 identical bits which form the pattern.

### Parameter RAM Load

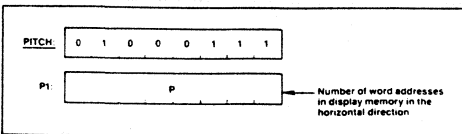
From the starting address, SA, any number of bytes may be loaded into the parameter RAM at incrementing addresses, up to location 15. The sequence of parameter bytes is determined by the next command byte entered into the FIFO. The parameter RAM stores 16 bytes of information in predefined locations which differ for graphics and character modes. See the parameter RAM discussion for bit assignments.



### Pitch Specification

This value is used during drawing by the drawing processor to find the word directly above or below the current word, and during display to find the start of the next line.

The Pitch parameter (width of display memory) is set by two different commands. In addition to the PITCH command, the RESET (or SYNC) command also sets the pitch value. The "active-words-per-line" parameter, which specifies the width of the raster-scan display, also sets the pitch of the display memory. Note that the AW value is two less than the display window width. The PITCH command must be used to set the proper memory width larger than the window width.



### Drawing Control Commands

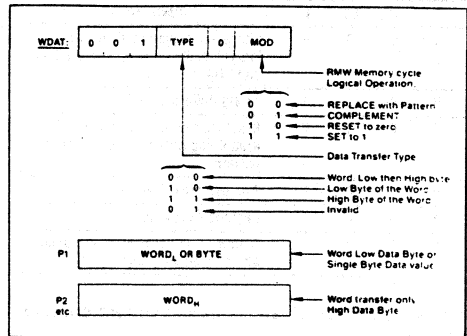
#### Write Data into Display Memory

Upon receiving a set of parameters (two bytes for a word transfer, one for a byte transfer), one RMW cycle into video memory is done at the address pointed to by the cursor EAD. The EAD pointer is advanced to the next word, according to the previously specified direction. More parameters can then be accepted.

For byte writes, the unspecified byte is treated as all zeros during the RMW memory cycle.

In graphics bit-map situations, only the LSB of the WDAT parameter bytes is used as the pattern in the RMW operations. Therefore it is possible to have only an all ones or all zeros pattern. If the WG bit of the third parameter of the CURS command is set to one, any byte following the WDAT command is written as is. In coded character applications all the bits of the WDAT parameters are used to establish the drawing pattern.

The WDAT command operates differently from the other commands which initiate RMW cycle activity. It requires parameters to set up the Pattern register while the other commands use the stored values in the parameter RAM. Like all of these commands, the WDAT command must be preceded by a FIGS command and its parameters. Only the first three parameters need be given following the FIGS opcode to set up the type of drawing, the DIR direction, and DC value. The DC parameter +1 will be the number of RMW cycles done by the GDC with the first set of WDAT parameters. Additional sets of WDAT parameters will see a DC value of 0 which will cause only one RMW cycle to be executed per set of parameters.

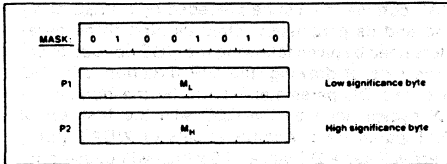




**Mask Register Load**

This command sets the value of the 16-bit Mask register of the figure drawing processor. The Mask register controls which bits can be modified in the display memory during a read-modify-write cycle.

The Mask register is loaded both by the MASK command and the third parameter byte of the CURS command. The MASK command accepts two parameter bytes to load a 16-bit value into the Mask register. All 16-bits can be individually one or zero, under program control. The CURS command, on the other hand, puts a 1-of-16 pattern into the Mask register based on the value of the Dot Address value, dAD. If normal single-pixel-at-a-time graphics figure drawing is desired, there is no need to do a MASK command at all since the CURS command will set up the proper pattern to address the proper pixels as drawing progresses. For coded character DMA, and screen setting and clearing operations using the WDAT command, the MASK command should be used after the CURS command if its third parameter byte has been output. The Mask register should be set to all ones for any "word-at-a-time" operation.

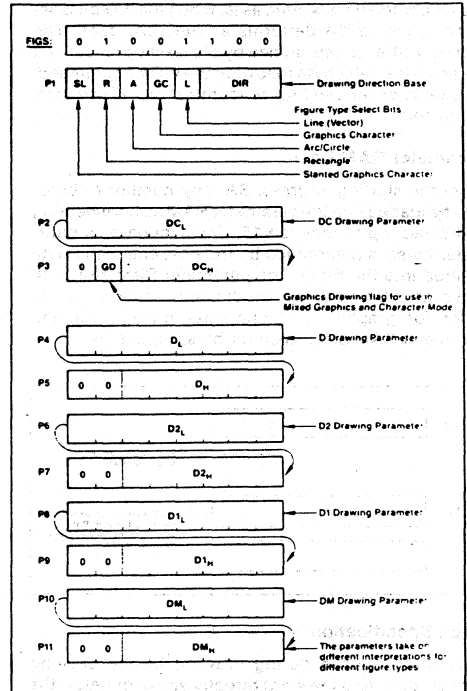


**Valid Figure Type Select Combinations**

SL	R	A	GC	L	Operation
0	0	0	0	0	Character display mode drawing, individual dot drawing, DMA, WDAT, and RDAT
0	0	0	0	1	Straight line drawing
0	0	0	1	0	Graphics character drawing and area filling with graphics character pattern
0	0	1	0	0	Arc and circle drawing
0	1	0	0	0	Rectangle drawing
1	0	0	1	0	Slanted graphics character drawing and slanted area filling

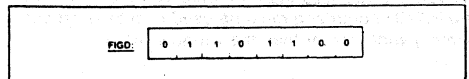
Only these bit combinations assure correct drawing operation.

**Figure Drawing Parameters Specify**



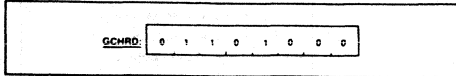
**Figure Draw Start**

On execution of this instruction, the GDC loads the parameters from the parameter RAM into the drawing processor and starts the drawing process at the pixel pointed to by the cursor, EAD, and the dot address, dAD.



## Graphics Character Draw and Area Filling Start

Based on parameters loaded with the FIGS command, this command initiates the drawing of the graphics character or area filling pattern stored in parameter RAM. Drawing begins at the address in display memory pointed to by the EAD and dAD values.

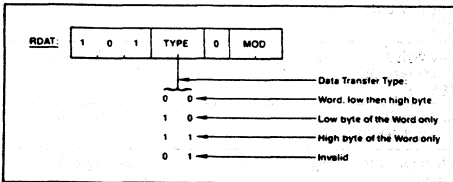


## Data Read Commands

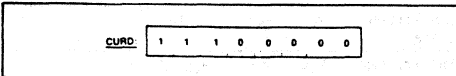
### Read Data from Display Memory

Using the DIR and DC parameters of the FIGS command to establish direction and transfer count, multiple RMW cycles can be executed without specification of the cursor address after the initial load (DC = number of words or bytes).

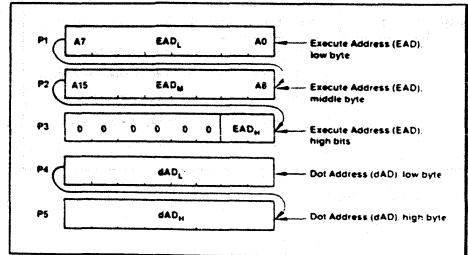
As this instruction begins to execute, the FIFO buffer direction is reversed so that the data read from display memory can pass to the microprocessor. Any commands or parameters in the FIFO at this time will be lost. A command byte sent to the GDC will immediately reverse the buffer direction back to write mode, and all RDAT information not yet read from the FIFO will be lost. MOD should be set to 00 if no modification to video buffer is desired.



### Cursor Address Read



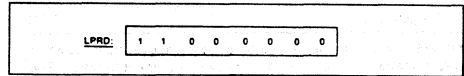
The following bytes are returned by the GDC through the FIFO:



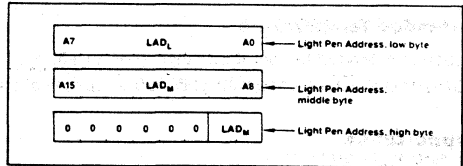
The execute address, EAD, points to the display memory word containing the pixel to be addressed.

The dot address, dAD, within the word is represented as a 1-of-16 code for graphics drawing operations.

### Light Pen Address Read



The following bytes are returned by the GDC through the FIFO:

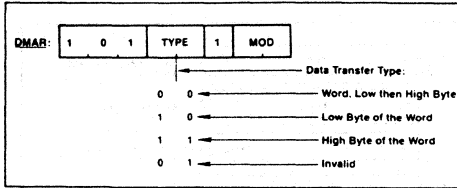


The light pen address, LAD, corresponds to the display word address, DAD, at which the light pen input signal is detected and deglitched.

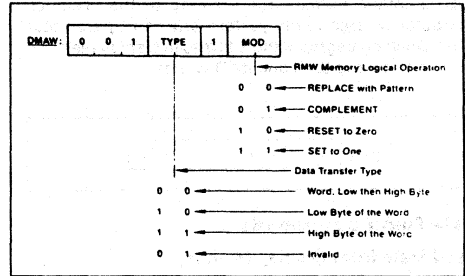
The light pen may be used in graphics, character, or mixed modes but only indicates the word address of light pen position.

### DMA Control Commands

#### DMA Read Request



#### DMA Write Request



### Absolute Maximum Ratings

Operating temperature*	0 to +70 °C
Storage temperature	-65 to +150 °C
Voltage on any pin with respect to ground	-0.5 to +7 V
Power dissipation	1.5 W

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### \*Extended Temperature

In case of extended temperature range (-40 to +85°C) μPD7220AD-2 can be used with μPD7220AD specification

### Capacitance

T<sub>A</sub> = 25 °C; V<sub>CC</sub> = GND = 0 V

Parameter	Symbol	Limits			Test Conditions
		Min	Typ	Max	
Input capacitance	C <sub>IN</sub>		10	pF	f <sub>c</sub> = 1 MHz
IO capacitance	C <sub>IO</sub>		20	pF	V <sub>i</sub> (unmeasured) = 0 V
Output capacitance	C <sub>OUT</sub>		20	pF	
Clock input capacitance	C <sub>φ</sub>		20	pF	

### DC Characteristics

T<sub>A</sub> = 0 to +70 °C; V<sub>CC</sub> = 5 V ± 10%; GND = 0 V

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input low voltage	V <sub>IL</sub>	-0.5		0.8	V	(Note 1)
Input high voltage	V <sub>IH</sub>	2.2		V <sub>CC</sub> + 0.5	V	(Notes 2, 3)
Output low voltage	V <sub>OL</sub>		0.45		V	I <sub>OL</sub> = 2.2 mA
Output high voltage	V <sub>OH</sub>	2.4			V	I <sub>OH</sub> = -400 μA
Input low leak current (except VSYNC, DACK)	I <sub>IL</sub>			-10	μA	V <sub>i</sub> = 0 V
Input low leak current (VSYNC, DACK)	I <sub>IL</sub>			-500	μA	V <sub>i</sub> = 0 V
Input high leak current (except LPEN/DH)	I <sub>IH</sub>			+10	μA	V <sub>i</sub> = V <sub>CC</sub>
Input high leak current (LPEN/DH)	I <sub>IH</sub>			+500	μA	V <sub>i</sub> = V <sub>CC</sub>
Output low leak current	I <sub>OL</sub>			-10	μA	V <sub>O</sub> = 0 V
Output high leak current	I <sub>OH</sub>			+10	μA	V <sub>O</sub> = V <sub>CC</sub>
Clock input low voltage	V <sub>CL</sub>	-0.5		0.6	V	
Clock input high voltage	V <sub>CH</sub>	3.5		V <sub>CC</sub> + 1.0	V	
V <sub>CC</sub> supply current	I <sub>CC</sub>			270	mA	

#### Note:

- (1) For 2xWCLK, V<sub>IL</sub> = -0.5 to +0.6 V.
- (2) For 2xWCLK, V<sub>IH</sub> = +3.9 V to V<sub>CC</sub> + 1.0 V.
- (3) For  $\overline{WR}$ , V<sub>IH</sub> = 2.5 V to V<sub>CC</sub> + 0.5 V.

## AC Characteristics

$T_A = 0 \text{ to } -70^\circ\text{C}$ ;  $V_{CC} = 5.0 \text{ V} \pm 10\%$ ;  $GND = 0 \text{ V}$

Parameter	Symbol	7220AD Limits		7220AD-1 Limits		7220AD-2 Limits		Unit	Test Conditions
		Min	Max	Min	Max	Min	Max		
<b>Read Cycle (GDC → CPU)</b>									
Address setup to $\overline{RD}\downarrow$	$t_{AR}$	0		0		0		ns	
Address hold from $\overline{RD}\downarrow$	$t_{RA}$	0		0		0		ns	
$\overline{RD}$ pulse width	$t_{RH1}$	$t_{RD1} - 20$	$t_{RCY} - 1/2 t_{CLK}$	$t_{RD1} - 20$	$t_{RCY} - 1/2 t_{CLK}$	$t_{RD1} - 20$	$t_{RCY} - 1/2 t_{CLK}$	ns	
Data delay from $\overline{RD}\downarrow$	$t_{RD1}$		75		65		55	ns	$C_L = 50 \text{ pF}$
Data floating from $\overline{RD}\downarrow$	$t_{DF}$	0	75	0	65	0	55	ns	
$\overline{RD}$ pulse cycle	$t_{RCY}$	$4 t_{CLK}$		$4 t_{CLK}$		$4 t_{CLK}$		ns	
<b>Write Cycle (GDC → CPU)</b>									
Address setup to $\overline{WR}\downarrow$	$t_{AW}$	0		0		0		ns	
Address hold from $\overline{WR}\downarrow$	$t_{WA}$	10		10		10		ns	
$\overline{WR}$ pulse width	$t_{WW}$	80	$t_{WCY} - t_{CLK}$	70	$t_{WCY} - t_{CLK}$	60	$t_{WCY} - t_{CLK}$	ns	
Data setup to $\overline{WR}\downarrow$	$t_{DW}$	65		55		45		ns	
Data hold from $\overline{WR}\downarrow$	$t_{WD}$	0		10		10		ns	
$\overline{WR}$ pulse cycle	$t_{WCY}$	$4 t_{CLK}$		$4 t_{CLK}$		$4 t_{CLK}$		ns	
<b>DMA Read Cycle (GDC → CPU)</b>									
$\overline{DACK}$ setup to $\overline{RD}\downarrow$	$t_{KR}$	0		0		0		ns	
$\overline{DACK}$ hold from $\overline{RD}\downarrow$	$t_{RK}$	0		0		0		ns	
$\overline{RD}$ pulse width	$t_{RR2}$	$t_{RD2} + 20$		$t_{RD2} + 20$		$t_{RD2} - 20$		ns	
Data delay from $\overline{RD}\downarrow$	$t_{RD2}$		$1.5 t_{CLK} - 80$		$1.5 t_{CLK} + 70$		$1.5 t_{CLK} - 60$	ns	$C_L = 50 \text{ pF}$
DREQ delay from $2 \times \overline{WCLK}\uparrow$	$t_{REQ}$		100		85		75	ns	$C_L = 50 \text{ pF}$
DREQ setup to $\overline{DACK}\downarrow$	$t_{DK}$	0		0		0		ns	
$\overline{DACK}$ high-level width	$t_{DK}$	$t_{CLK}$		$t_{CLK}$		$t_{CLK}$		ns	
$\overline{DACK}$ pulse cycle	$t_E$	$4 t_{CLK} (1)$		$4 t_{CLK} (1)$		$4 t_{CLK} (1)$		ns	
DREQ $\downarrow$ delay from $\overline{DACK}\downarrow$	$t_{KQ(R)}$		$t_{CLK} - 100$		$t_{CLK} - 90$		$t_{CLK} - 80$	ns	$C_L = 50 \text{ pF}$
$\overline{DACK}$ low-level width	$t_{LK}$	$2 t_{CLK}$		$2 t_{CLK}$		$2 t_{CLK}$		ns	
<b>DMA Write Cycle (GDC → CPU)</b>									
$\overline{DACK}$ setup to $\overline{WR}\downarrow$	$t_{KW}$	0		0		0		ns	
$\overline{DACK}$ hold from $\overline{WR}\downarrow$	$t_{WK}$	0		0		0		ns	

**Note:**

(1) For high-byte and low-byte transfers:  $t_E = 5 t_{CLK}$ .

### AC Characteristics (cont)

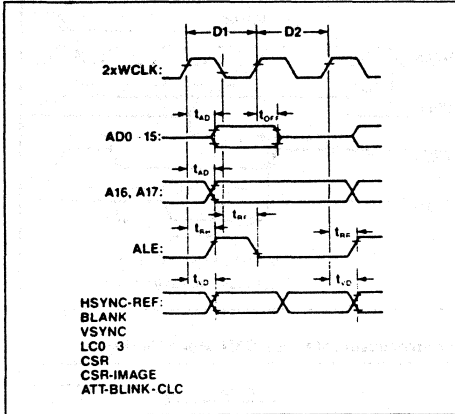
$T_A = 0 \text{ to } +70^\circ\text{C}$ ;  $V_{CC} = 5.0 \text{ V} \pm 10\%$ ;  $GND = 0 \text{ V}$

Parameter	Symbol	7220AD Limits		7220AD-1 Limits		7220AD-2 Limits		Unit	Test Conditions
		Min	Max	Min	Max	Min	Max		
<b>RMW Cycle (GDC → Display Memory)</b>									
Address/data display from $2xWCLK\uparrow$	$t_{AD}$	20	105	20	90	15	80	ns	$C_L = 50 \text{ pF}$
Address/data floating from $2xWCLK\uparrow$	$t_{OFF}$	20	105	20	90	15	80	ns	$C_L = 50 \text{ pF}$
Input data setup to $2xWCLK\downarrow$	$t_{DIS}$	0		0		0		ns	
Input data hold from $2xWCLK\downarrow$	$t_{DIH}$	$t_{DE}$		$t_{DE}$		$t_{DE}$		ns	
$\overline{DBIN}$ delay from $2xWCLK\downarrow$	$t_{DE}$	20	80	20	70	15	60	ns	$C_L = 50 \text{ pF}$
ALE $\uparrow$ delay from $2xWCLK\downarrow$	$t_{RR}$	20	80	20	70	15	60	ns	$C_L = 50 \text{ pF}$
ALE $\downarrow$ delay from $2xWCLK\downarrow$	$t_{RF}$	20	65	20	55	15	50	ns	$C_L = 50 \text{ pF}$
ALE high width	$t_{RW}$	$1/3 t_{CLK}$		$1/3 t_{CLK}$		$1/3 t_{CLK}$		ns	$C_L = 50 \text{ pF}$
ALE low width	$t_{RL}$	$1.5 t_{CLK} - 30$		$1.5 t_{CLK} - 30$		$1.5 t_{CLK} - 30$		ns	
Address setup to ALE $\downarrow$	$t_{AA}$	30		30		30			
<b>Display Cycle (GDC → Display Memory)</b>									
Video signal display from $2xWCLK\uparrow$	$t_{VD}$		90		80		70	ns	$C_L = 50 \text{ pF}$
<b>Input Cycle (GDC → Display Memory)</b>									
Input signal setup to $2xWCLK\uparrow$	$t_{PS}$	10		10		10		ns	
Input signal width	$t_{PW}$	$t_{CLK}$		$t_{CLK}$		$t_{CLK}$		ns	
<b>Clock (<math>2xWCLK</math>)</b>									
Clock rise time	$t_{CR}$		15		15		15	ns	
Clock fall time	$t_{CF}$		15		15		15	ns	
Clock high pulse width	$t_{CH}$	70		61		52		ns	
Clock low pulse width	$t_{CL}$	70		61		52		ns	
Clock cycle	$t_{CLK}$	165	10000	145	10000	125	10000	ns	

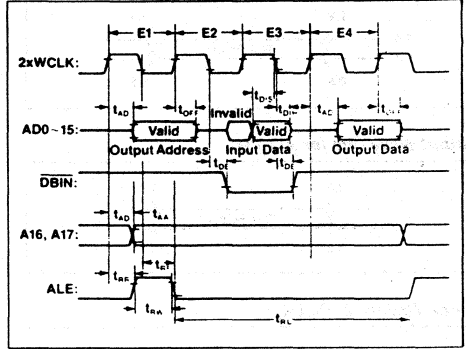


**Timing Waveforms (cont)**

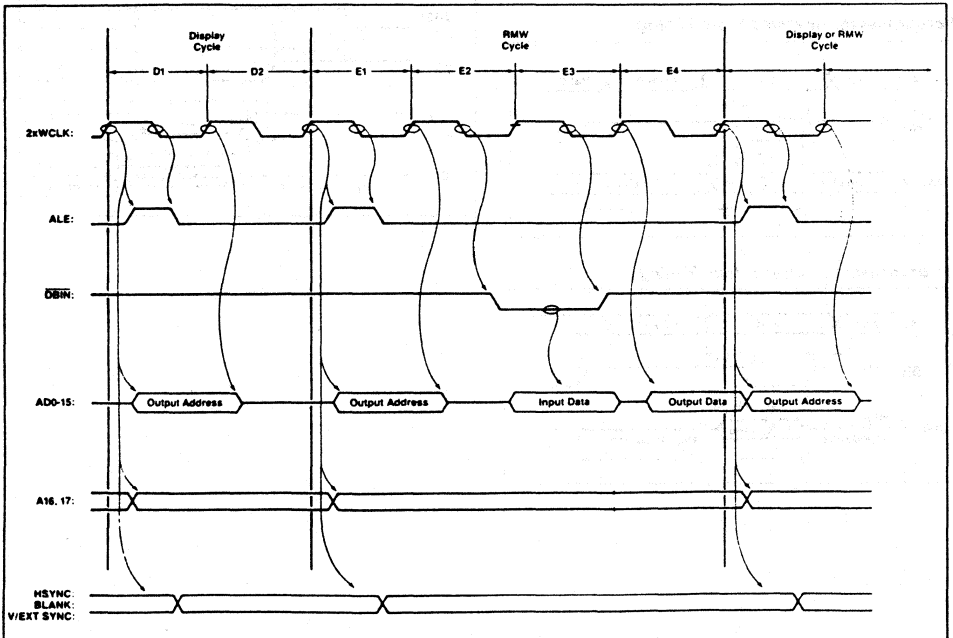
**Display Memory Display Cycle Timing**



**Display Memory RMW Timing**

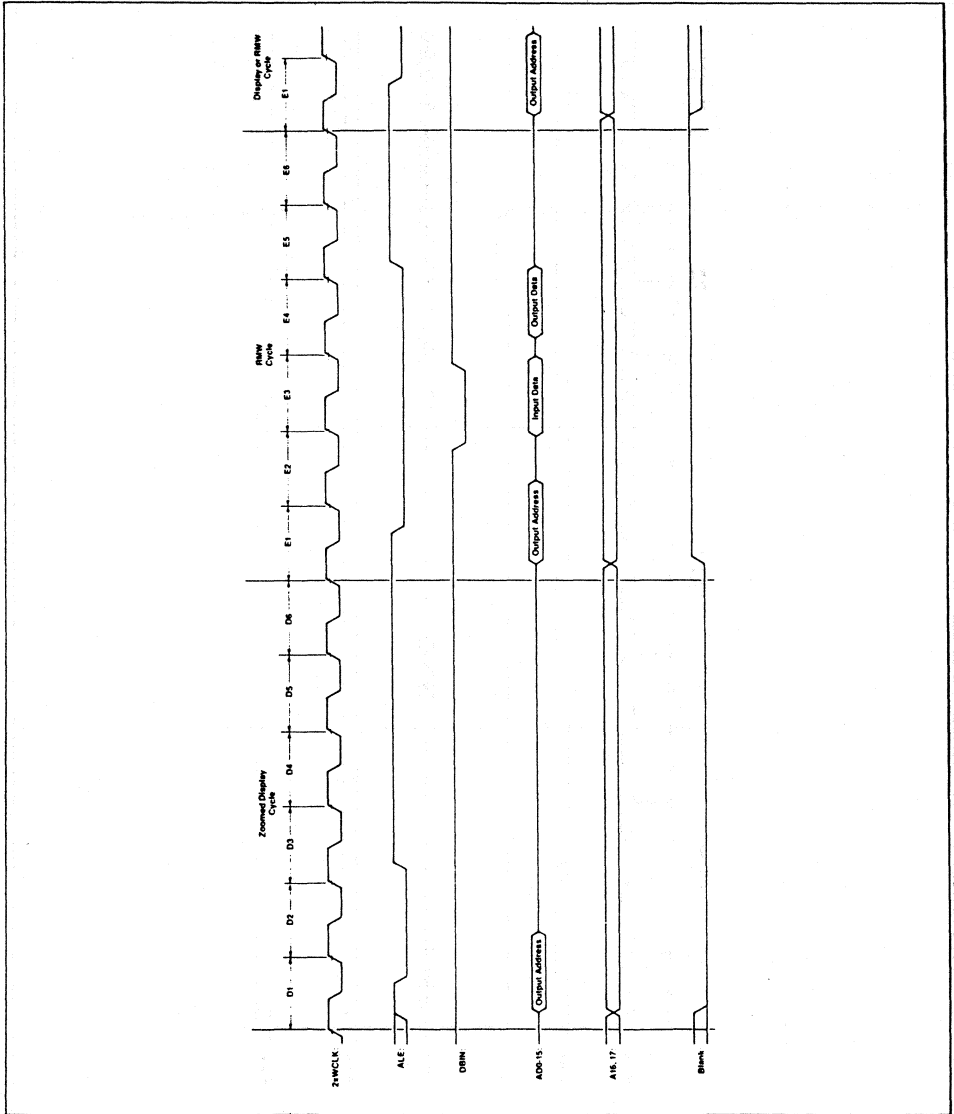


**Display and RMW Cycles (1x Zoom)**



## Timing Waveforms (cont)

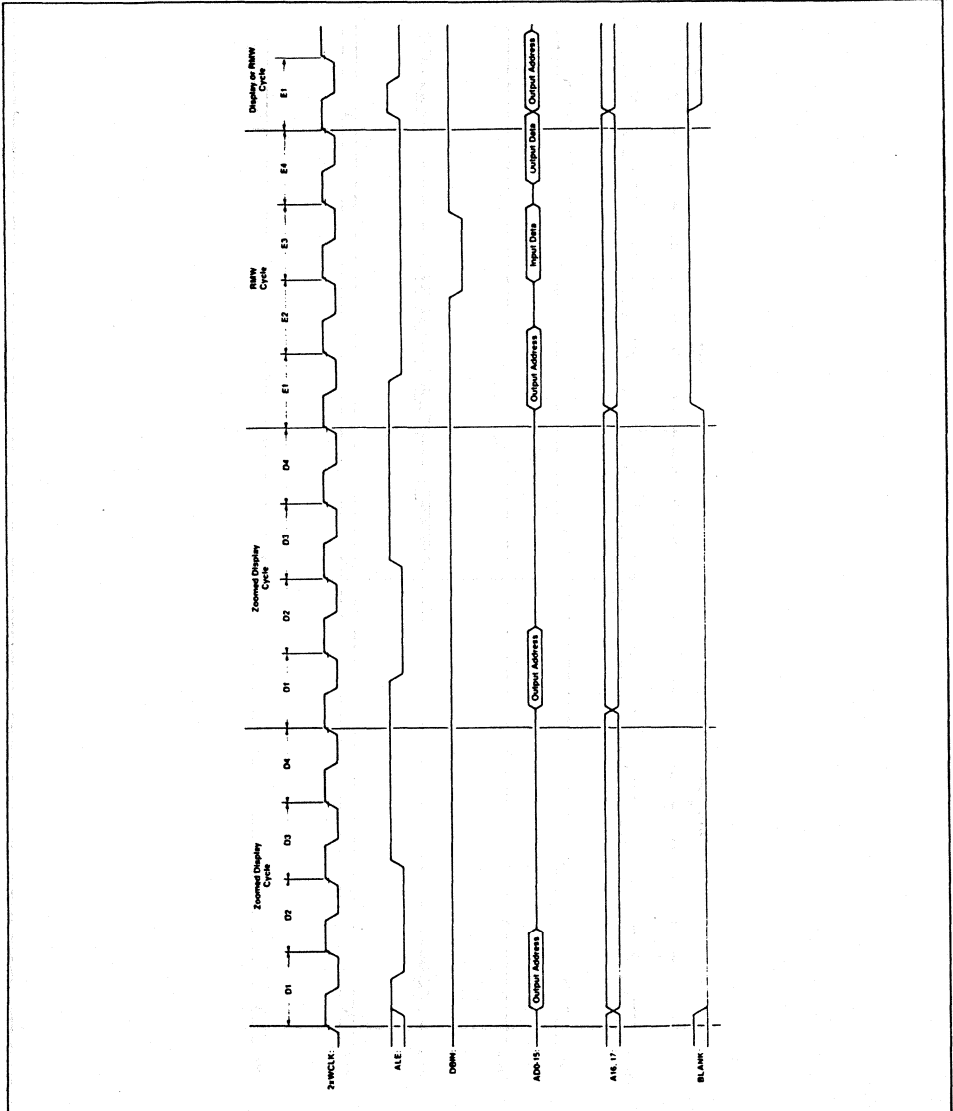
### Display and RMW Cycles (2x Zoom)





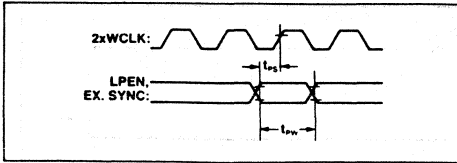
Timing Waveforms (cont)

Display and RMW Cycles (3x Zoom)

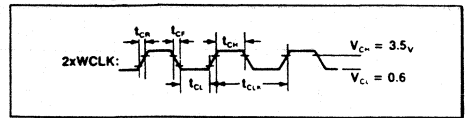


## Timing Waveforms (cont)

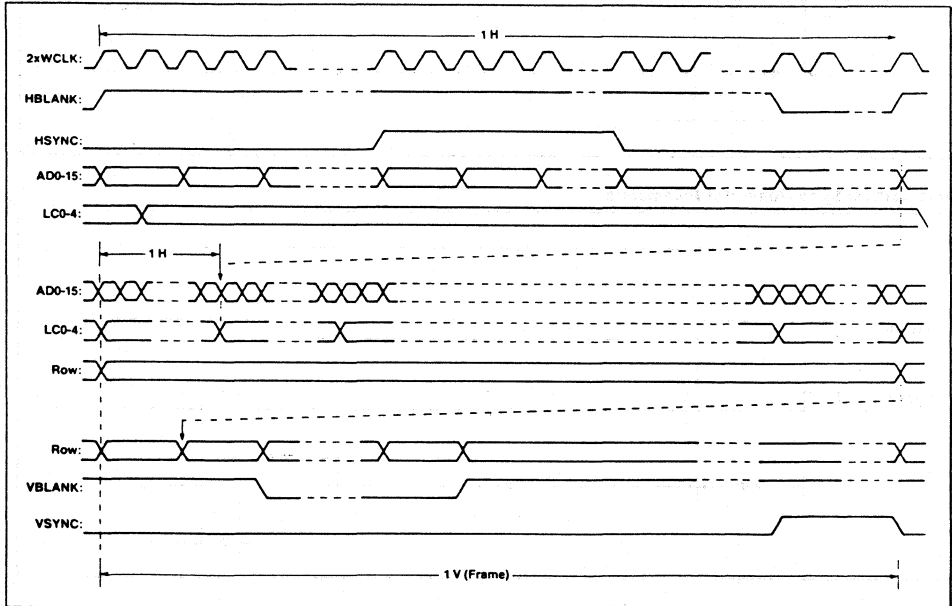
### Light Pen and External Sync Input Timing



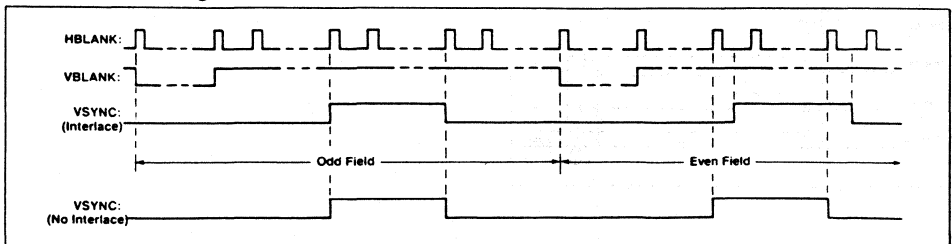
### Clock Timing (2xWCLK)



### Video Sync Signals Timing

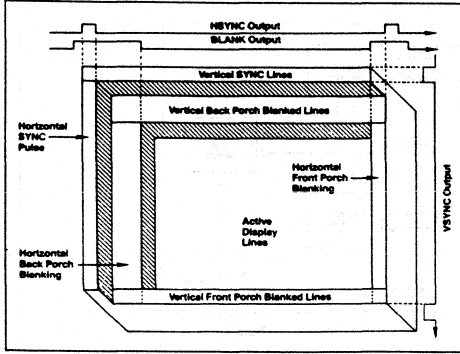


### Interlaced Video Timing

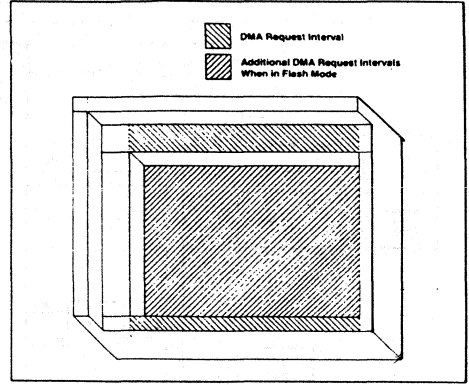




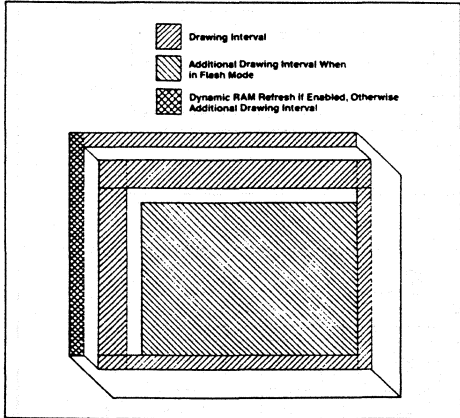
## Video Field Timing



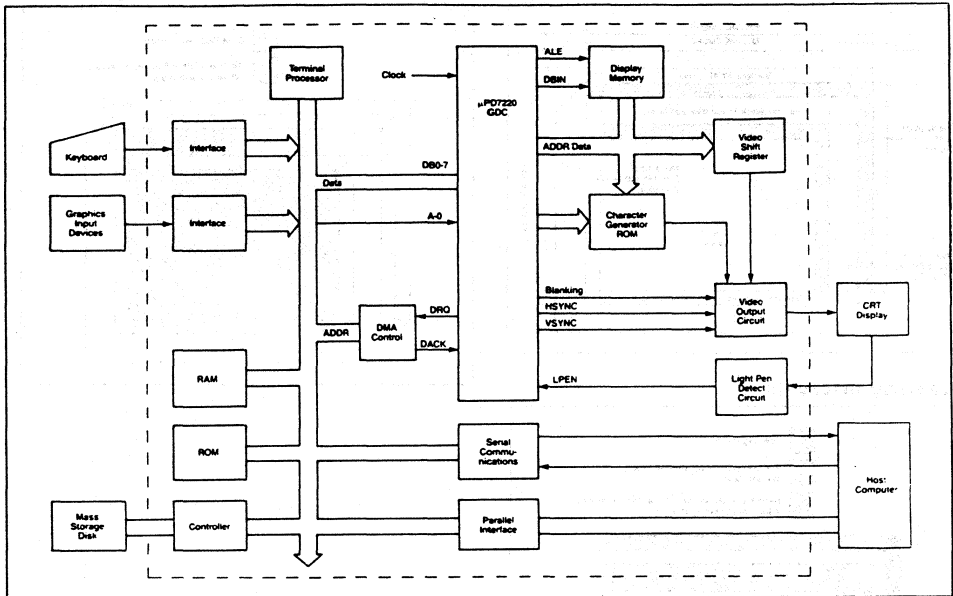
## DMA Request Intervals



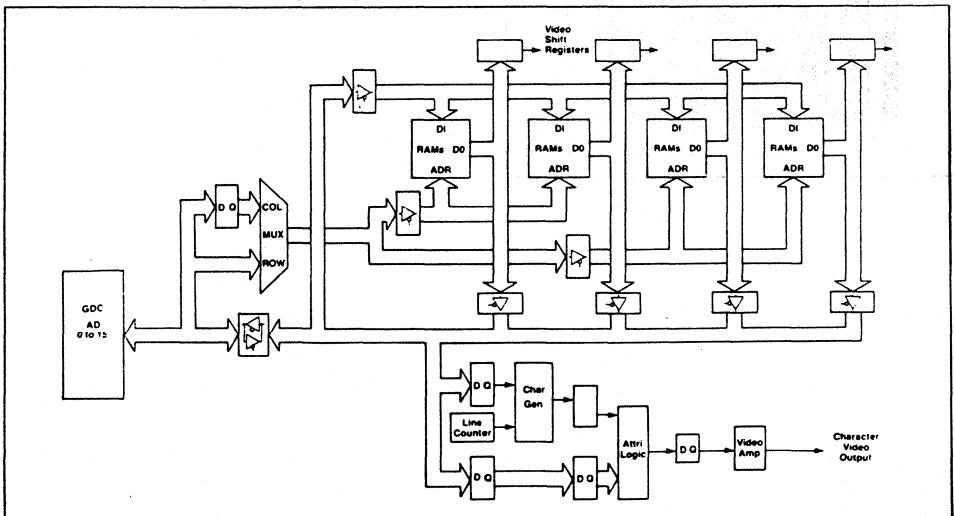
## Drawing Intervals



**Block Diagram of a Graphics Terminal**



**Multipane Display Memory Diagram**



The most significant difference between the current μPD7220 graphic display controller and the μPD7220A is the speed increase of the μPD7220A. Currently, the 7220 is available in 4.0-MHz, 5.0-MHz, and 5.5-MHz versions. The 7220A will be available in 6.0-MHz, 7.0-MHz, and 8.0-MHz versions. A number of other differences, in the form of corrections and enhancements, that exist between the parts are described below.

### Commands

Two new RESET command variations and one new BLANK command variation have been added to the GDC. All of the new commands allow a reset or display blank to be accomplished while preventing reinitialization of the internal sync generator by an external sync source (slave mode).

Command		Opcode
RESET1	Same as RESET in 7220	0 0 0 0 0 0 0 0
RESET2	Blank display; no resync	0 0 0 0 0 0 0 1
RESET3	Enable display; no resync	0 0 0 0 1 0 0 1
BLANK1	Same as BLANK in 7220	0 0 0 0 1 1 0 0
BLANK2	Blank display; no resync	0 0 0 0 0 1 0 1

Operation may be characterized as follows:

	Display Blanked	Reset Performed	External Sync (Slave Only)
RESET1	Yes	Yes	Accepted
RESET2	Yes	Yes	Ignored
RESET3	No	Yes	Ignored
BLANK1	Yes	No	Accepted
BLANK2	Yes	No	Ignored

### Flag Bits

Six additional operation flag bits have been added. They are located as shown in figure E-1 and their effect on GDC operation follows.

**PH Bit.** The width of the display pitch register has been increased from 8 to 9 bits. This additional bit (PH) is defined as the seventh bit of the fifth parameter following a SYNC or RESET command. Utilizing this bit, the GDC can handle pitches up to 511 words per line.

**DS Bit.** A drawing wait function has been added to the GDC. The Light Pen input (LPEN) may be used as a drawing wait control input. If this input is held high for a period greater than four 2xWCLK cycles drawing execution is halted. During the wait cycle, display address (DAD) is output instead of drawing address (EAD). See figure E-2.

Figure E-1. Additional Flag Bits

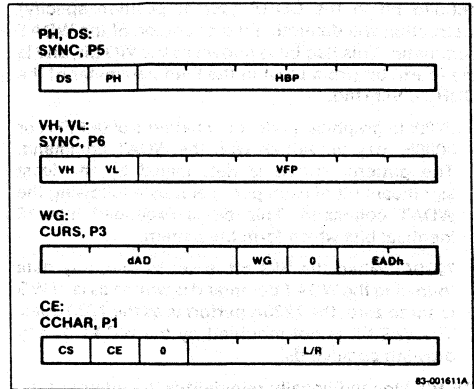
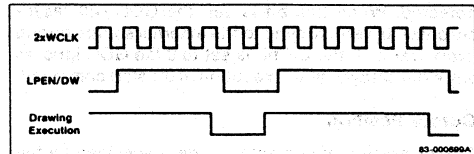


Figure E-2. Draw Wait Timing



**VH Bit.** A vertical blank status flag is available in the status register to replace the horizontal blank status flag if required.

7220: Outputs a horizontal blank status flag as DB6 in the status register.

7220A: User may select between the standard horizontal blank status flag and the new vertical blank status flag.

Selection is via the new VH flag. This flag is the eighth bit of the sixth parameter following a SYNC or RESET command. When VH is 1, vertical blank status is output. When VH is 0, operation is as in the 7220.

**VL Bit.** The number of display lines per video frame may be selected as odd or even when in interlaced mode if this bit is set.

7220: Allows only an odd number of lines per video frame.

7220A: Allows an even or odd number of lines per frame. Selection is via the VL flag created in the seventh bit of the sixth parameter byte following a SYNC or RESET command.

**WG Bit.** When in graphics mode, the state of a newly defined bit in the CURS (cursor position specify) instruction will determine the operation of the WDAT command. This flag bit is known as the WG bit, and is the fourth bit (from LSB) in the third parameter of the CURS command.

7220: In graphics mode, only patterns of 0FFFFH or 0000H may be written with the WDAT command. The pattern written is determined by the least significant bit of each parameter byte following the WDAT command. This bit is expanded into 16 identical bits which form the pattern.

7220A: When the WG bit is set to one, any data following the WDAT command is written as is. If WG is set to zero, the 7220A performs as the 7220 does. The WG bit is not modified by the WDAT or any drawing commands.

**CE Bit.** Unconditionally reinitializes the internal sync generator when the GDC is programmed for slave mode and the falling edge of an external sync input is detected. When this bit is set, the GDC will resynchronize its vertical sync to the external source during every frame. If the CE bit is set to 0 the GDC ignores any external sync after execution of the start command.

#### Cursor Position

Cursor position in character mode is specified by the lower 16 bits of address rather than the lower 13 bits as in the 7220.

#### Cursor Format

Cursor format restrictions in character mode or character areas in mixed mode have been removed.

#### Pins A<sub>16</sub> and A<sub>17</sub>

A<sub>16</sub> and A<sub>17</sub> will now provide static signals.

7220A: A<sub>16</sub> and A<sub>17</sub> are invalid during E4 of a read-modify-write cycle. Currently, an external latch is required.

7220: A<sub>16</sub> and A<sub>17</sub> will be valid during E4 and any other time. No external latch is required.

#### Display Word

Display word addresses are incremented during the AW and HPF periods, even during vertical blanking time.

#### ALE Signal

The ALE signal generated by the 7220A remains as per specifications even while zooming. Corrects problem 3 as described in Product Bulletin #29.

7220: ALE is lengthened by one half 2xWCLK cycle if horizontal blanking begins while in a zoomed display.

7220A: The problem no longer exists.

#### Resistors

On-chip pull-up/pull-down resistors have been added.

7220: All such resistors are external.

7220A: Added are a pull-up resistor for V/EXT SYNC and  $\overline{\text{DACK}}$ , and a pull-down resistor for LPEN/DH.

#### Software

In the area of software, only one modification to existing software is required. Bit 6 of parameter RAM bytes 3 and 7 must be set for the GDC to operate in image mode. These bits are known as the IM or image bits for display partitions one and two. Note that if the above circuitry is installed and these bits are not set, the display will not be correct as every other display word would be skipped.

The μPD72022 intelligent display processor (referred to as IDP below) is a display controller which processes texts, still and sprite images on a raster-scan CRT. The IDP performs high-level display processing function by means of commands and the sprite image display function which enables dynamic display position specification.

Through these functions, the IDP can control displays ranging from simple text and graphic display to screen division and sprite image screen display.

### 1.1 Features

- (1) Applicable to three display modes
  - Text mode
    - 80 characters x 50 lines
    - Character size is programmable (8 x 8 to 8 x 32).
    - Seven attributes are available.
  - Semi-graphic mode
    - 80 patterns x 50 lines
    - 2 types of display pattern formats can be specified.
  - Graphic mode
    - 640 dots x 200 dots or 640 dots x 400 dots
    - Bit image data display
    - 1, 2 or 4 bits/pixel
- (2) Divided screen display function (4 horizontal divisions)
- (3) Smooth scroll function
- (4) Sprite display function
- (5) Simultaneous 16-color display
- (6) Attribute adding function (up to a maximum of 7 attributes)
- (7) Interface display through external synchronization
- (8) Video memory address space with a maximum capacity of 256 K x 16 bits
- (9) DRAM refresh function built-in
- (10) Dual port RAM connection available.
- (11) Bus arbitration function

- (12) CRT control signal programmable
  - Horizontal display period, horizontal retrace period (right to left), horizontal synchronizing signal width, vertical display period, vertical retrace period, vertical synchronizing signal width, number of raster/line, and blinking period.
- (13) Display resolution variable
  - Horizontal display period: 640 dots max. (Max. dot rate 22 MHz)
  - Vertical display period: 512 dots max.
  - Display signal (4 bits/dot) serial output
- (14) Horizontal and vertical external synchronization functions
- (15) Twenty-two types of commands
- (16) CMOS
- (17) Single power supply: +5V

### 1.2 Order Information

Order Name	Package
μPD72022GF-3B9	80-pin plastics flat
μPD72022L	68-pin PLCC

### 1.3 Example of System Configuration

Figure 1-1 shows an example of a system using an IDP. The IDP can be connected as a peripheral device to the host CPU.

The IDP performs all video memory control functions. Through the use of the bus arbitration function of the IDP, the host CPU can also access the video memory directly.

The IDP fetches display data from the video memory or the character generator memory, creates pixel data for one raster, and outputs the created data as a serial display signal

A CRT can be connected directly to the system via a buffer. Therefore, peripheral display control circuits such as a parallel-serial converter circuit are not necessary.

- (1) Video memory configuration
  - This system can directly control a video memory (with a maximum capacity of 256 K x 16 bits) which stores character/graphic data and parameter tables for screen and sprite control.

A standard DRAM/SRAM can serve as this video memory which has a refresh function necessary for the DRAM.

The system has a dual port RAM interface capable of bidirection accessing, and automatically generate data transfer cycles necessary for serial accessing.



(2) Interfacing with the host system

The following three modes are available for interfacing with the host CPU, the main memory and the video memory.

(a) Stand-alone mode

The video memory space is independent from the main memory and the IDP uses the video memory exclusively as a local memory.

(b) Interleave mode

Display data access and command processing access are interrupted in reply to a request for access from the host CPU and the bus is temporarily opened.

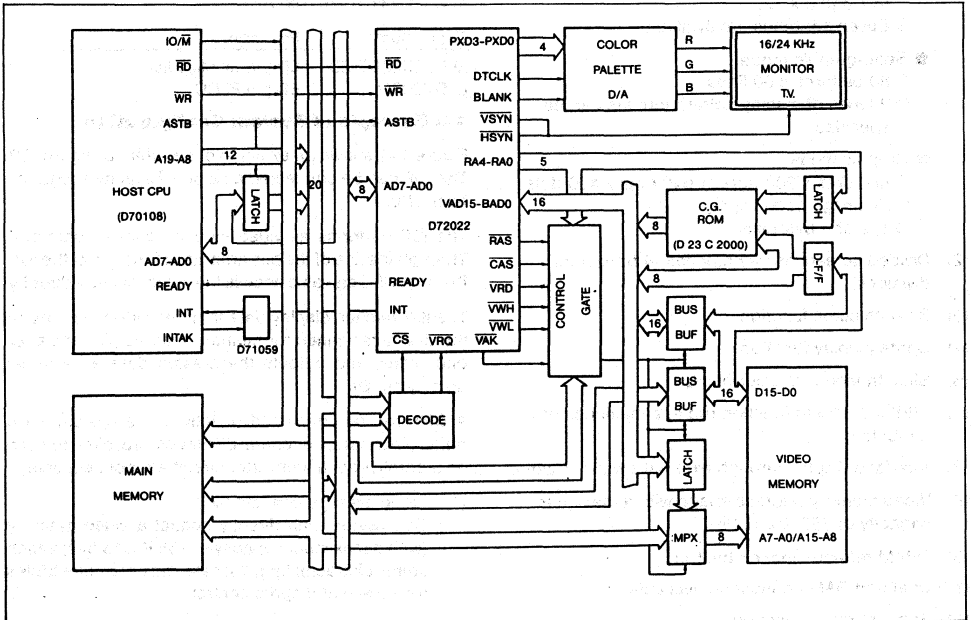
Because the host CPU can process display data directly, overhead during display data control can be alleviated.

(c) Dual port mode

Through the use of dual-port RAM serial access for display data access, random access processing can be carried out by requesting the host CPU for data processing by serial-port setup or command processing access.

In this mode, the host CPU can operate the video memory in a manner similar to that for the main memory and thus the system's maneuverability can be highly improved.

Figure 1-1 Example of System Configuration



### 1.4 Internal Configuration

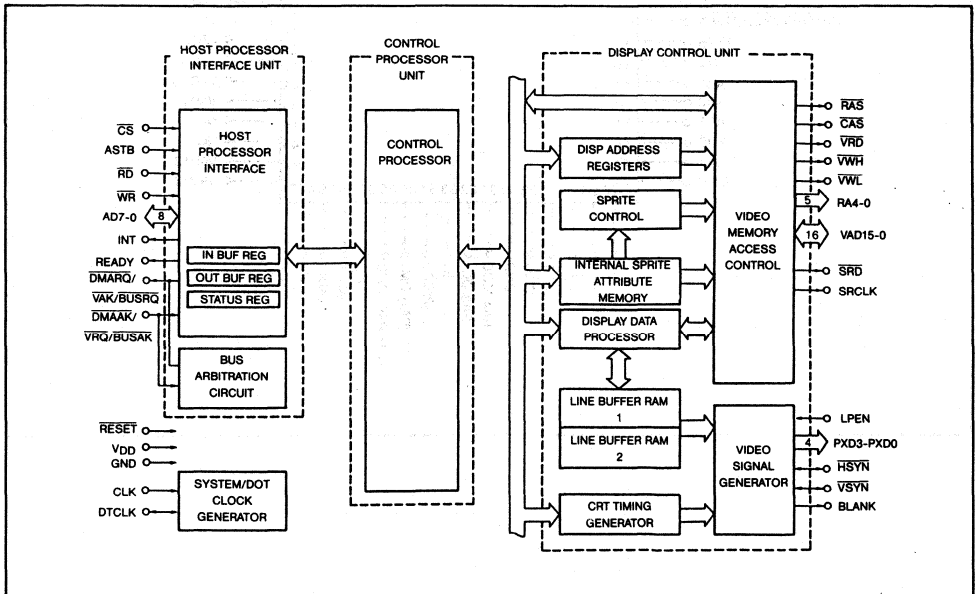
Figure 1-2 shows an internal block diagram of the IDP. The IDP consists of a host processor interface unit, a control processor unit and a display control unit. The functions of each unit are outlined:

- (1) Host processor interface unit  
This unit transmits and receives commands, parameters and data related to IDP internal processings to and from the host processor. It performs the asynchronizing bus interface control, DMA control and interrupt control functions.
- (2) Control processor unit  
This unit fetches commands and parameters via the host processor interface unit, interprets them and executes display format changing and video memory data processing according to the command functions.
- (3) Display control unit  
This unit generates video memory display addresses according to screen display formats, fetches display data from the video memory and the character generator memory, converts the data into color data corresponding directly to display pixels and outputs them in serial form.

The display control unit is equipped with a circuit for generating the synchronizing signal, the display timing signal and the video memory interface control signal so that it can perform the externally synchronizing function and the sprite display function in addition to basic display functions.

This unit also incorporates two pairs of 640 x 4 bit line buffers which convert display data into pixel data corresponding directly to display, edit the converted data and finally stores the edited data. While generating the data of n-raster, one line buffer accumulates the display data of (n+1)-raster into the other line buffer. Because it is not necessary under this method to read the video memory in synchronization with the CRT scanning system, the video memory bus is preferentially opened to the host memory if the host CPU makes a request for access to the video memory. Therefore, the host CPU can maintain its memory cycle efficiently even during display periods.

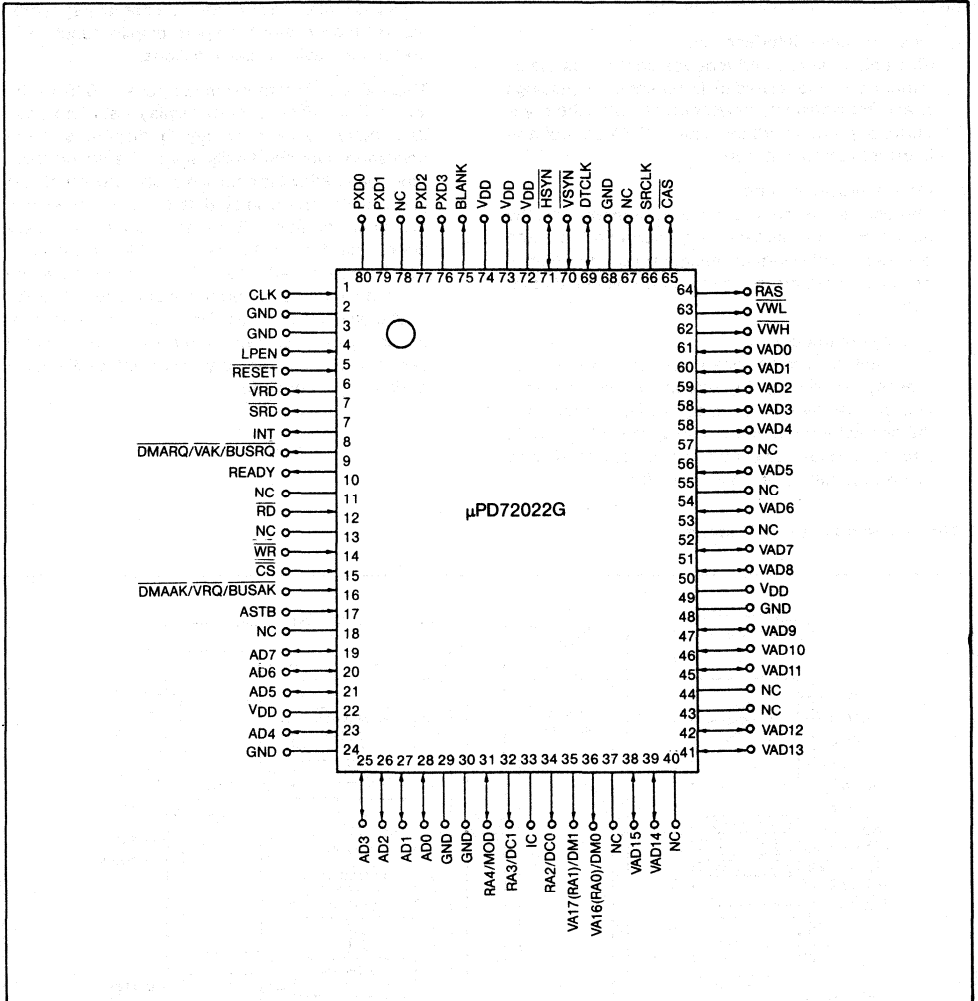
Figure 1-2 Internal Block Diagram



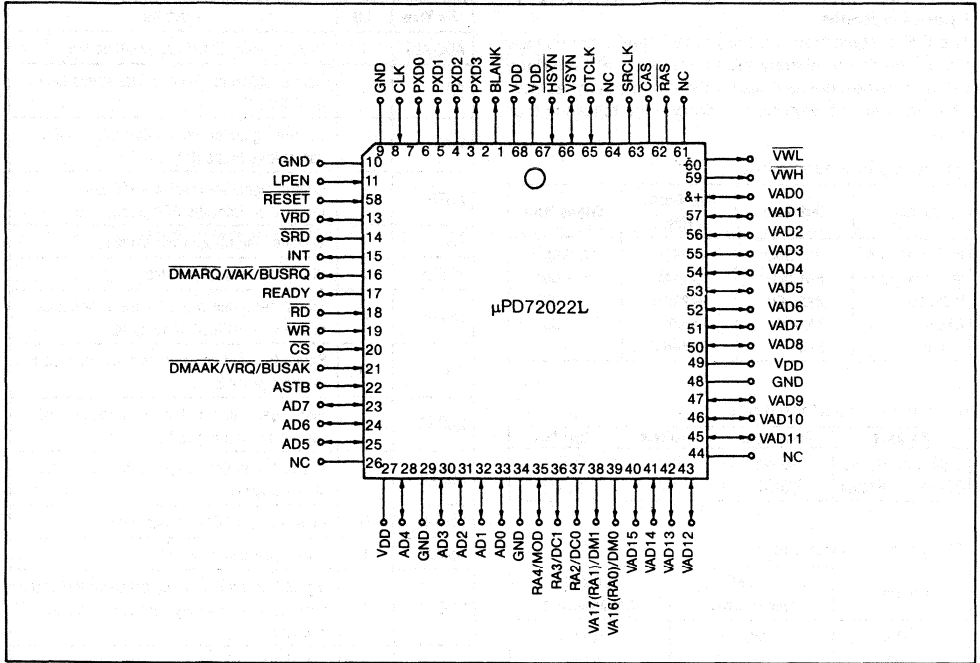
2.1 Pin Connection

Pin Connection Diagram (top view)

(1) 80-pin plastic flat package



(2) 68-pin PCCC



**CLK:** Clock  
**LPEN:** Light Pen Strobe  
**RESET:** Reset  
**VRD:** Video Memory Read  
**SRD:** Serial Read  
**INT:** Interrupt Request  
**DMARQ/VAK/BUSRQ:**  
 DMA Request/Video Memory Acknowledge/Bus Request  
**READY:** Ready  
**NC:** Non-Connection  
**RD:** Read  
**WR:** Write  
**CS:** Chip Select  
**DMAAK/VRQ/BUSAK:**  
 DMA Acknowledge/Video Memory Request/  
 Bus Acknowledge  
**ASTB:** Address Strobe  
**AD0-AD7:** Address/Data Bus 0-7  
**RA3/DC1:** Raster Address 3/Display Cycle 1  
**RA4/MOD:** Raster Address 4/Mode

**IC:** Internally Connected  
**RA2/DC0:** Raster Address 2/Display Cycle 0  
**VA17(RA1)/DM1:**  
 Video Memory Address 17 (Raster Address 1)/  
 Display Mode 1  
**VA16(RA0)/DM0:**  
 Video Memory Address 16 (Raster Address 0)/  
 Display Mode 0  
**VAD0-15:** Video Memory address 0-15/Data 0-15  
**VWH:** Video Memory Write High  
**VWL:** Video Memory Write Low  
**RAS:** Row Address Strobe  
**CAS:** Column Address Strobe  
**SRCLK:** Serial Read Clock  
**DTCLK:** Dot Clock  
**VSYN:** Vertical SYNC  
**HSYH:** Horizontal SYNC  
**BLANK:** Blanking Signal  
**PXD0-3:** Pixel data 0-3

**2.2 Pin Function**

**2.2.1 Pins with variable Functions According to Operation Modes**

The IDP is equipped with the pins whose functions vary according to the display mode, the host CPU interface mode, and the external/internal synchronization settings. The pins perform various functions in the following four modes:

(1) Display operation mode

Pin Name	Text Mode	Semi-Graphic Mode	Graphic Mode
VA16 (RA0)/DM0	RA0/DM0	RA0/DM0	VA16/DM0
VA17 (RA1)/DM1	RA1/DM1	RA1/DM1	VA17/DM1
RA2/DC0	RA2/DC0	RA2/DC0	DC0
RA3/DC1	RA3/DC1	RA3/DC1	DC1
RA4/MOD	RA4	RA4/MOD	-

(2) Host CPU interface mode

Pin Name	Stand-Alone	Interleave	Dual Port
DMARQ/VAK/BUSRQ	DMARQ	VAK	BUSRQ
DMAAK/VRQ/BUSAK	DMAAK	VRQ	BUSAK

(3) Synchronization mode

Pin Name	Internal Synchronization	External Synchronization
HSYN	Output	Input
VSYN	Output	Input

(4) Dot clock synchronization mode

Pin Name	Internal Synchronization	External Synchronization
DTCLK	Output	Input

**2.2.2 List of Pin Functions**

(1) Host system interface

Pin Name	I/O	Functions
AD0-AD7	I	3-state, bidirectional address/data bus
RD	I	Control signal for reading data/status flags from the IDP
WR	I	Control signal for writing data/commands/parameters to the IDP
ASTB	I	Control signal for inputting address information from the AD0 through AD7
CS	I	Validates the RD and WR signals.
RESET	I	Clears the inside of the chips.
READY	O	Signal generated when the IDP extends its memory or I/O read/write cycle.
INT	O	Signal transmitted to request an interrupt to the host processor.
BUSRQ	O	Signal generated by the IDP to request the system bus control right.
BUSAK	I	The IDP controls the system bus while this signal is active.
DMARQ	O	Generates the DMA request signal.
DMAAK	I	DMA cycle permit signal.
VRQ	I	Signal transmitted to request the IDP that the host processor directly controls the video memory.
VAK	O	Active while the host processor is permitted to directly control the video memory.
CLK	I	System clock input

### (2) Video memory interface

Pin Name	I/O	Functions
RAS	0	Generates the row-address strobe signal.
CAS	0	Generates the column-address strobe signal.
VAD0-15	I/O	Outputs the video memory address (lower 16 bits) and inputs/outputs the video memory data. Also outputs the DRAM refresh address.
VA16, 17	0	Outputs the video memory address (upper 2 bits) and display control data.
RA0-3	0	Outputs the raster address (lower 4 bits) and display control data.
RA4/MOD	I/O	Generates the raster address (most significant 4 bits) or inputs the mode switching signal (MOD).
VRD	0	Generates the read strobe signal for reading data from the video memory.
VWH, VWL	0	Generates the write strobe signal for writing data from the video memory.
SRCLK	0	Generates the serial read port control clock when the dual port memory is used.
SRD	0	Active while data is read from the serial port when the dual port memory is used.

REMARKS: RA0 and RA1 also serve as DM0 and DM1 respectively. RA2 and RA3 also serve as DC0 and DC1, respectively.

### (3) CRT interface

Pin Name	I/O	Functions
VSYN	I/O	Generates the vertical synchronizing signal when internal synchronization is specified.  Generates the vertical synchronizing signal when external synchronization is specified.
HSYN	I/O	Generates the horizontal synchronizing signal when internal synchronization is specified.  Generates the horizontal synchronizing signal when external synchronization is specified.
BLANK	0	Generates the display clearing signal
PXD0-3	0	Generates the display signal (4 bits/dot) in synchronization with the dot clock (DTCLK).
LPEN	I	Inputs the light pen detection signal.  Specifies the DTCLK pin function according to the level of this signal at the rising edge of the RESET signal.
DTCLK	I/O	Input/output pin for the display signal output timing clock.

### (4) Display data control signal

Pin Name	I/O	Functions
DM0, 1	0	Outputs the display mode of the still screen being processed by the IDP in code form.
DC0, 1	0	Outputs the display processing cycle in code from while the IDP is accessing the video memory.

REMARKS: DM0 and DM1 also serve as RA0 and RA1, respectively. DC0 and DC1 also serve as RA2 and RA3, respectively.

### 2.2.3 Description of Pin Functions

This section gives a supplementary description of section 2.2.2.

#### (1) Host system interface

##### (a) AD0 through AD7

3-stage bidirectional address/data buses. The following data control is carried out on the data bus according to the control signal status.

CS	DMAAK	WR	RD	Address			Data (AD7 thru AD0)
				AD2	AD1	AD0	
1	1	x	x	x	x	x	Non-selection (high impedance)
0	1	0	1	0	1	0	IDP command input
				1	1	1	IDP parameter input
				0	1	1	
1	0	0	1	x	x	x	Write by DMA transfer
0	1	1	0	0	1	0	IDP status output
				1	1	1	
				0	1	1	IDP parameter output
1	0	1	0	x	x	x	Read by DMA transfer

x: Don't care

#### (2) Video memory interface

##### (a) RAS

Generates the row-address strobe signal.

- Used as a row-address strobe signal for the DRAM which multiplexes the address.
- Used as a latch timing signal at its falling edge for the video memory address.

(b)  $\overline{\text{CAS}}$ 

Generates the column-address strobe signal.

- Used as a column-address strobe signal for the DRAM which multiplexes the address.
- Used as a latch timing signal at its falling edge for the video memory data and control data.
- Used as a column-address strobe signal for  $\overline{\text{RAS}}$  only refresh.
- Used as timing signal for identifying the video memory address cycle.

(c)  $\text{VAD0}$  through  $\text{VAD15}$ 

Outputs the video memory address (lower 16 bits) and inputs/outputs (16 bits) the video memory data.

- Outputs the video memory address while  $\overline{\text{RS}}$  and  $\overline{\text{CAS}}$  are at the low level
- Inputs/outputs the video memory data while  $\overline{\text{RAS}}$  and  $\overline{\text{CAS}}$  are at the low level.

Outputs the refresh address (16 bits) at the DRAM control refresh cycle.

Becomes high impedance when  $\overline{\text{VAR}}$  is set to the low level or at the serial read cycle.

(d)  $\text{VA16}$ / $\text{VA17}$ 

Outputs the video memory address (upper 2 bits) and the display control data.

Outputs the video memory address when  $\overline{\text{RAS}}$  and  $\overline{\text{CAS}}$  are at the high level.

(e)  $\text{RA0}$  through  $\text{RA3}$ 

Outputs the raster address (lower 4 bits) and the display control data in the text or semigraphic mode.

Outputs the raster address  $\overline{\text{RAS}}$  and  $\overline{\text{CAS}}$  are at the high level.

(f)  $\text{RA4}/\text{MOD}$ 

Outputs the most significant bit (RA4) of the raster address and inputs the semi-graphic mode switching signal (MOD).

- Outputs RA4 when  $\overline{\text{RAS}}$  and  $\overline{\text{CAS}}$  are at the high level.
- Inputs MOD when  $\overline{\text{RAS}}$  and  $\overline{\text{CAS}}$  are at the low level

(g)  $\overline{\text{VRD}}$ 

Outputs the read strobe signals for reading data from the video memory.

- Used as an output enable signal for the memory.
- Used as a signal for identifying the data transfer

cycle to the dual port RAM. At the falling edge of  $\overline{\text{RAS}}$  output, set to the low level during the data transfer cycle and to the high level during the read cycle.

- Used as a display data input timing signal. The IDP fetches display data while  $\overline{\text{VRD}}$  is at the low level.

(h)  $\overline{\text{VWH}}$ ,  $\overline{\text{VWL}}$ 

Outputs the write strobe signal for writing data to the video memory.

- Used as a write enable signal for the memory.

$\overline{\text{VWH}}$  and  $\overline{\text{VWL}}$  are output prior to the falling edge of the  $\overline{\text{CAS}}$  output. The early read cycle is set for the dynamic RAM.

- Used as a signal for identifying the byte/word access.

$\overline{\text{VWH}}$  becomes active in the high-byte access mode and  $\overline{\text{VWL}}$  becomes active in the low-byte access mode. In these cases, the upper and lower 8 bit data of the video memory bus are similar to each other.

$\overline{\text{VWH}}$  and  $\overline{\text{VWL}}$  becomes active simultaneously in the word access mode.

(i)  $\text{SRCLK}$ 

Outputs the serial read port control clock when the dual port memory is used.

The IDP fetches display data while the  $\text{SRCLK}$  output is at the low level.

(j)  $\text{SRD}$ 

Becomes active while data is read from the serial port when the dual port memory is used.

Becomes inactive during the data transfer or random read cycle.

## (3) CRT interface

(a)  $\text{BLANK}$ 

Outputs the display clearing signal.

Becomes active during the horizontal/vertical feedback or screen clearing.

(b)  $\text{PX0}$  through  $\text{PX3}$ 

Outputs the display signal (4 bits/dot) in synchronization with the dot clock  $\text{DTCLK}$ .

When the display clearing signal  $\text{BLANK}$  is active, the horizontal and vertical feedback signals are output.

- $\text{PX3}$ : Outputs the vertical feedback signal.
- $\text{PX2}$ : Outputs the horizontal feedback signal.
- $\text{PX1}$ ,  $\text{PX0}$ : Low level

When the DTCLK output is at the low level, PXD0 through PXD3 can be latched externally.

(c) LPEN

● Inputs the light pen detection signal. Inputting a high-level signal causes the X and Y positional coordinates on the screen to be latched and the IDP status LP (MSB) to be set.

Time required from inputting of the display signal on the CRT, the detection of display and characters by the light pen to the inputting of the LPEN signal to the IDP and time for internal detection are both delayed in detection. Therefore, the detection timings differ from the actual positional coordinates and calibration is necessary.

● Inputs the signal specified for the dot clock synchronizing modes (external and internal). The DTCLK pin functions are specified according to the level of this signal generated at the rising edge of the RESET signal.

H: Internal dot clock output mode

L: External dot clock input mode

(d) DTCLK

Input/output pin for the output timing clock of the display signal.

- Outputs the CLK divided clock in the internal dot clock output mode.
- Inputs the reference clock for the timing of the scanning system within the IDP in the external dot clock input mode.

(4) Display data control signal

The IDP outputs the control information (display data and control data) related to display processing from the DM0, DM1, DC0 and DC1 pins when CAS is at the low level.

The following operations can be carried out by latching the display data and control data at the rising edge of CAS.

- Display data selection
- Identification of video memory access cycle
- Control of synchronization with the IDP display processing cycle.

(a) DM0, DM1

Outputs the display mode on the still screen being processed in code form.

DM1	DM0	Display Mode
0	x	Text mode
1	0	Semi-graphic mode
1	1	Graphic mode

Used to select the read data from the video memory or the read data from the character generator to be input to the IDP according to the display mode.

DM0 and DM1 also serve as RA0 and RA1, respectively.

(b) DC0, DC1

Outputs the display cycle in code form during video memory accessing.

DC1	DC0	Processing Cycle
0	0	Cycles other than those listed below
0	1	Still image display cycle
1	0	Sprite display cycle
1	1	Screen start cycle

Used for the video memory access cycle for requesting display data.

DC0 and DC1 also serve as RA2 and RA3, respectively.



The IDP uses a total of twenty-two commands consisting to the initialization command, the display control command, the video memory operation command and the sprite control command.

**3.1 Types of Commands**

Classification	Name	Function
Initialization command	SYNC	Sets the display operation mode and the scan timing.
Display control command	DSPON	Sets the base address and border color of the screen control table to operate the display controller and starts display.
	DSPOF	Stops the display controller to clear the display.
	DSPDEF	Specifies the display screen configuration and display format.
	CURDEF	Specifies the cursor display format.
	ACTSCR	Selects the operation screen.
	CURS	Specifies the cursor display position to shift the cursor.
	LPNR	Reports the write pen position.
Video memory operation control command	DPLD	Specifies the operation address or address offset of the video memory.
	DPRD	Reports the video memory operation address.
	MASK	Sets the bit mask information for the data to be stored into the video memory.
	RDAT	Reads the contents of the video memory and reports them to the host processor.
	WDAT	Stores the transfer data into the video memory.
	BLKTOT	Reads the contents of the video memory and transfers them to the DMA.
	BLKTIN	Stores the DMA transferred data into the video memory.

Classification	Name	Function
Video memory operation control command (cont'd)	EXIT	Suspends the processing of the video memory operation command.
Sprite control command	SPRON	Activates the sprite controller to start sprite display.
	SPROF	Stops the sprite controller to clear sprite display.
	SPRSW	Specifies ON/OFF for each sprite display.
	SPRRD	Reads data from the sprite attribute table.
	SPRWR	Writes data into the sprite attribute table.
	SPROV	Reports the operation status of the sprite controller.

**3.2 Details of Commands**

**3.2.1 Initialization Command**

SYNC (SYNC FORMAT SPECIFY)

Command code

0	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---

10H

Parameter

RM	EL	EV	VM	0	DPM	ILM
0	RF	EF	ES	RV	RS	
0	0	LBL				
0	0	LBR				
HAD						
0	0	RBR				
0	0	RBL				
0	0	HS				
0	0	TBL				
0	0	TBR				
VAD(L)						
0	*	BBR				
0	0	BBL				
0	0	VS				

\*: VAD(H)

### Electrical Specification

Absolute Maximum Ratings (Ta = 25°C)

Parameter	Symbol	Test Conditions	Rating	Unit
Power Supply Voltage	VDD		-0.5 to +7.0	V
Input Voltage	VI		-0.5 to 7.0	V
Output Voltage	VO		-0.5 to +7.0	V
Operating Temperature	topt		-10 to +70	°C
Storage Temperature	tstg		-65 to +150	°C

DC Characteristics (Ta = -10 to 70°C, VDD = +5V ± 10%)

Parameter	Symbol	Test Conditions	Standard Value		Unit
			Min	Max	
Input Voltage Low	VIL	*1	-0.5	0.8	V
		*2	-0.5	0.6	V
Input Voltage High	VIH	*1	2.2	VDD+0.5	V
		*2	3.5	VDD+0.5	V
Output Voltage Low	VOL	IOL = 2.5 mA		0.4	V
Output Voltage High	VOH	IOH = -400μA	0.7VDD		V
Input Leakage Current Low	ILIL	VI = 0V		-10	μA
Input Leakage Current High	ILIH	VI = VDD		10	μA
Output Leakage Current Low	ILOL	VI = 0V		-10	μA
Output Leakage Current High	ILOH	VI = VDD		10	μA
Supply Current	IDD			150	mA

\*1: Except for CLK, DTCLK, RESET

\*2: CLK, DTCLK, RESET

Capacitance (Ta = 25°C, VDD = GND = 0V)

Parameter	Symbol	Test Conditions	Standard Value		Unit
			Min	Max	
Input Capacitance	CI	f = 1MHz Unmeasured Pins returned to 0V.		10	pF
Output Capacitance	CO			20	pF
I/O Capacitance	CI/O			20	pF
Clock Input Capacitance	CC			20	pF

AC Characteristics (Ta = -10 to +70°C, VDD = +5V ± 10%)

Clock and Reset

Parameter	Symbol	Test Conditions	Standard Value		Unit
			Min	Max	
System Clock Cycle	tCY		45	50	ns
System Clock high-level width	tKKH		18		ns
System Clock low-level width	tKKL		18		ns
Dot Clock Cycle	tCYDK	CL = 30pF	45	4tCY	ns
Dot Clock high-level width	tDKDKH		*		ns
Dot Clock low-level width	tDKDKL		*		ns
Dot Clock rise time	tDKR			*	ns
Dot Clock fall time	tDKF			*	ns
RESET Pulse width	tRSRSL				8tCY + 6tCYDK

\*: Under evaluation

CPU Read/Write Cycle

Parameter	Symbol	Test Conditions	Standard Value		Unit
			Min	Max	
ASTB Pulse width	tTSTH		45		ns
Address setup time (to ASTB <sup>↓</sup> )	tSAST		25		ns
Address hold time (to ASTB <sup>↓</sup> )	tHAST		10		ns
CS setup time (to RD <sup>↓</sup> , WR <sup>↓</sup> )	tSCSRW		0		ns
CS hold time (from RD <sup>↓</sup> , WR <sup>↓</sup> )	tHRWCS		0		ns
RD Pulse width	tRRL		170		ns
Data delay time (from RD <sup>↓</sup> )	tDRD		120		ns
Data hold time (from RD <sup>↑</sup> )	tHRD		0		ns
Data float time (from RD <sup>↑</sup> )	tFRD			55	ns
WR Pulse width	tWWL		180		ns
Data setup time (to WR <sup>↑</sup> )	tSDW		100		ns
Data hold time (from WR <sup>↑</sup> )	tHWD		10		ns
READY delay time from RD <sup>↓</sup> , WR <sup>↓</sup>	tDRWRDY			55	ns
READY delay time from VRQ <sup>↓</sup>	tDVRDY			60	ns
CS recovery time	tRVCS		*		ns
RD recovery time	tRVR		150		ns
WR recovery time	tRVW		150		ns
Read Access Cycle	tCYR			tRRL + 10tCY	ns
Write Access Cycle	tCYW			tWWL + 10tCY	ns

\*: Under evaluation

### VRAM Read/Write Cycle

Parameter	Symbol	Test Conditions	Standard value		Unit
			Min	Max	
Random read/write cycle	t <sub>CYRAS</sub>	*1	270		ns
		*2	360		ns
RAS high-level width	t <sub>RASRASH</sub>		95		ns
RAS low-level width	t <sub>RASRASL</sub>	*1	130		ns
		*2	210		ns
CAS high-level width	t <sub>CASCASH</sub>		110		ns
CAS low-level width	t <sub>CASCASL</sub>	*1	105		ns
		*2	185		ns
RAS <sub>i</sub> delay time from CAS <sub>i</sub> ↑	t <sub>DCASRASL</sub>		30		ns
CAS <sub>i</sub> delay time from RAS <sub>i</sub> ↓	t <sub>DRASCASL</sub>		40		ns
RAS <sub>i</sub> delay time from CAS <sub>i</sub> ↓	t <sub>DCASRASH</sub>	*1	60		ns
		*2	150		ns
Address setup time (to RAS <sub>i</sub> ↓)	t <sub>SVARAS</sub>		35		ns
Address hold time (from RAS <sub>i</sub> ↓)	t <sub>HRASVA</sub>		10		ns
Mode setup time (to CAS <sub>i</sub> ↓)	t <sub>SMDCAS</sub>		10		ns
Mode hold time (from CAS <sub>i</sub> ↓)	t <sub>HCASMD</sub>	*1	110		ns
		*2	185		ns

### VRAM read cycle

Parameter	Symbol	Test Conditions	Standard value		Unit
			Min	Max	
VRD <sub>i</sub> delay time from CAS <sub>i</sub> ↓	t <sub>DCASVR</sub>	*1	70		ns
		*2	150		ns
CAS <sub>i</sub> delay time from VRD <sub>i</sub> ↑	t <sub>DVRCAS</sub>		0		ns
VRD pulse width	t <sub>VRVRL</sub>	*1	70		ns
		*2	150		ns
CAS <sub>i</sub> delay time from VW <sub>i</sub> ↑	t <sub>DVWHCAS</sub>		70		ns
VW <sub>i</sub> delay time from RAS <sub>i</sub> ↑	t <sub>DRASHVW</sub>		35		ns
VW <sub>i</sub> delay time from CAS <sub>i</sub> ↑	t <sub>DCASVWL</sub>		15		ns
Input data setup time (to VRD <sub>i</sub> ↑)	t <sub>SDVR</sub>		40		ns
Input data hold time (from VRD <sub>i</sub> ↑)	t <sub>HVRD</sub>		0	30	ns
VRD <sub>i</sub> delay time from RAS <sub>i</sub> ↓	t <sub>DRASVRH</sub>	*1	130		ns
		*2	210		ns
RAS <sub>i</sub> setup time to VRD <sub>i</sub> ↑	t <sub>SVRRAS</sub>		15		ns
VRD <sub>i</sub> hold time from RAS <sub>i</sub> ↓	t <sub>HRASVR</sub>		25		ns

\*1: Text display cycle, semigraphic display cycle, display start cycle.

\*2: Graphic display cycle, sprite display cycle, command processing cycle.

VRAM write cycle

Parameter	Symbol	Test Conditions	Standard value		Unit
			Min	Max	
$\overline{RAS}$ ↓ delay time from $\overline{VRD}$ ↑	tDVRRAS		15		ns
$\overline{VRD}$ ↓ delay time from $\overline{RAS}$ ↓	tDRASVRL		130		ns
$\overline{RAS}$ ↓ delay time from $\overline{VW}$ ↑	tDVWRAS		35		ns
$\overline{VW}$ ↓ delay time from $\overline{RAS}$ ↓	tDRASLVW		-10		ns
$\overline{CAS}$ ↓ delay time from $\overline{VW}$ ↓	tDVWLCAS		10		ns
$\overline{VW}$ ↓ delay time from $\overline{CAS}$ ↓	tDCASVW		155		ns
$\overline{VW}$ Pulse width	tVWVWL		165		ns
Data setup time (to $\overline{CAS}$ ↓)	tSDCAS		10		ns
Data hold time (from $\overline{CAS}$ ↓)	tHCASD		155		ns

VRAM request

Parameter	Symbol	Test Conditions	Standard value		Unit
			Min	Max	
$\overline{VAK}$ set up time (to $\overline{CAS}$ ↑)	tSVACAS		20		ns
$\overline{VAK}$ hold time (from $\overline{CAS}$ ↓)	tHCASVA		10		ns
VRQ recovery time	tRVVQ				ns
READY delay time from $\overline{VRQ}$ ↓	tDVQRDY			60	ns

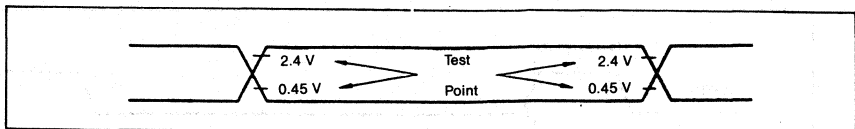
Display timing

Parameter	Symbol	Test Conditions	Standard value		Unit
			Min	Max	
Output *1 delay time from DTCLK↑	tDDKDSP	CL = 50pF	5	38	ns
Input*2 setup time (to DTCLK↑)	tSIDK		25		ns
Input*2 hold time (from DTCLK↑)	tHDKI		5		ns
Input*2 pulse width	tII		6tCYDK		ns

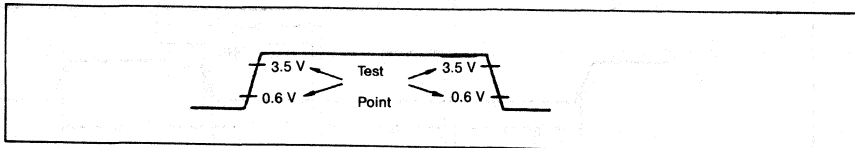
\*1: HSYN, VSYN, BLANK, PXD0-PXD3

\*2: HSYN, VSYN, LPEN

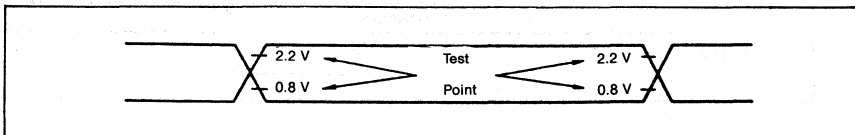
AC test I/O wave (except clock)



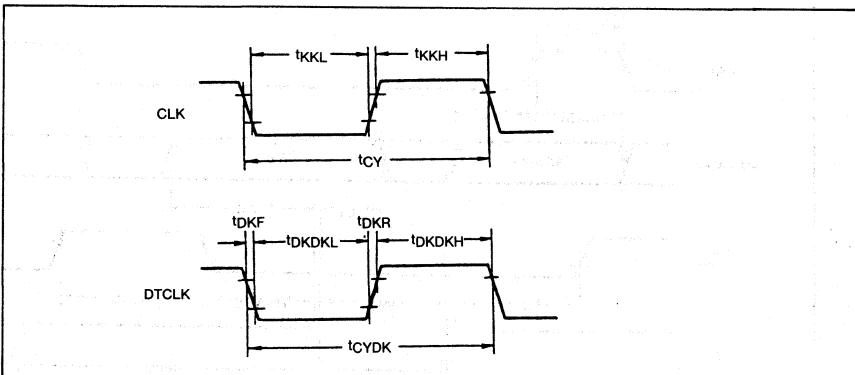
AC test clock input wave



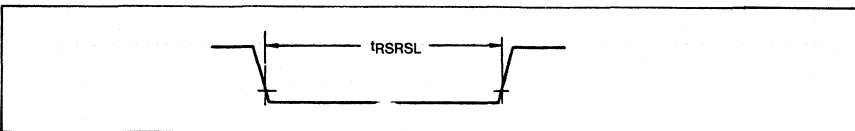
AC test output wave



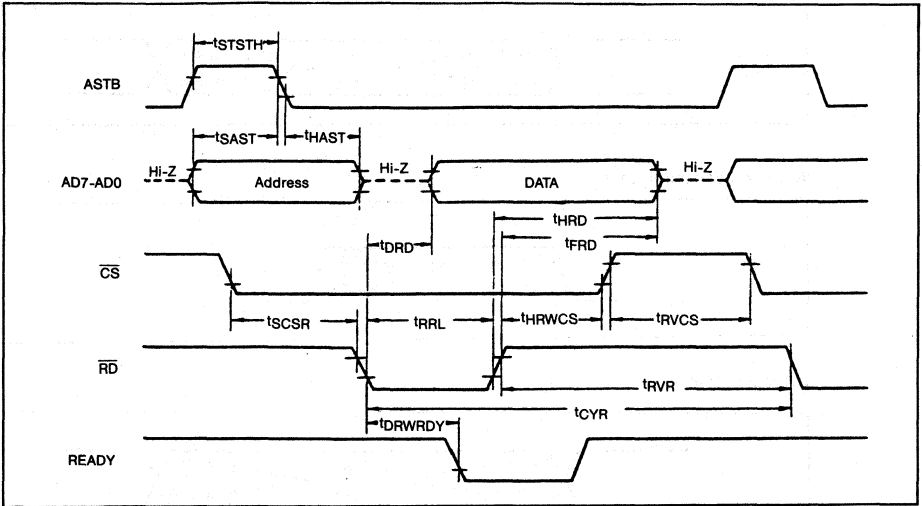
Clock wave



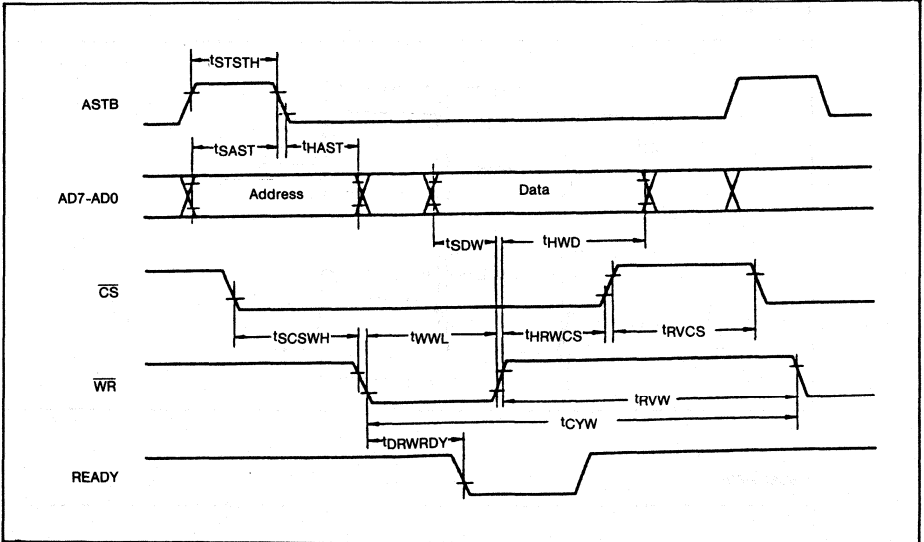
Reset wave



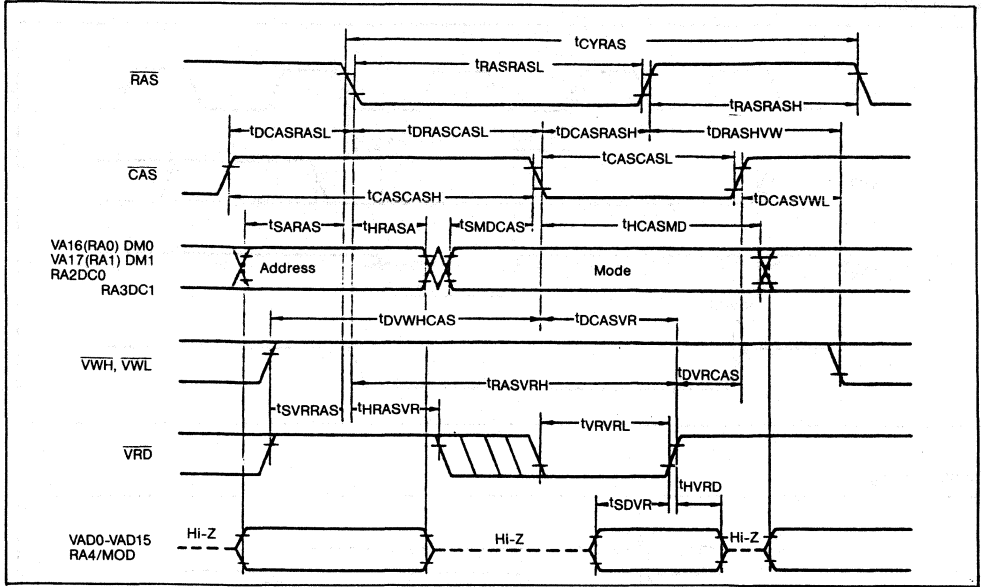
CPU read cycle



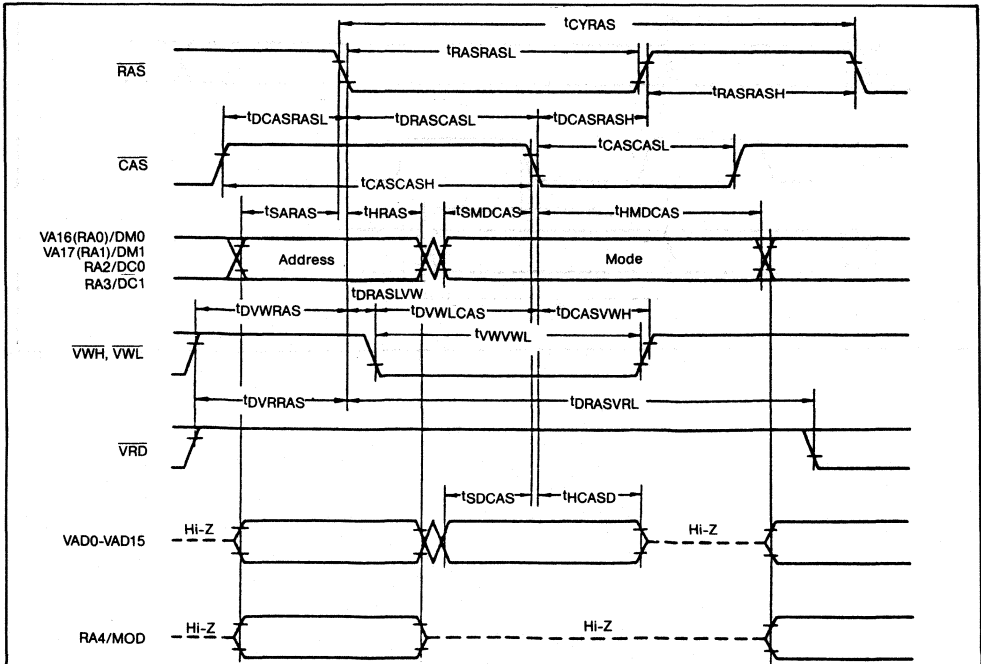
CPU write cycle



### VRAM read cycle

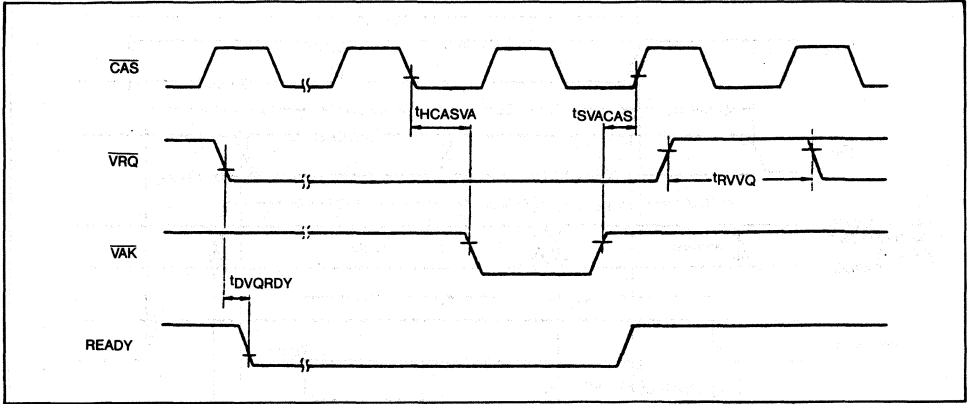


### VRAM write cycle

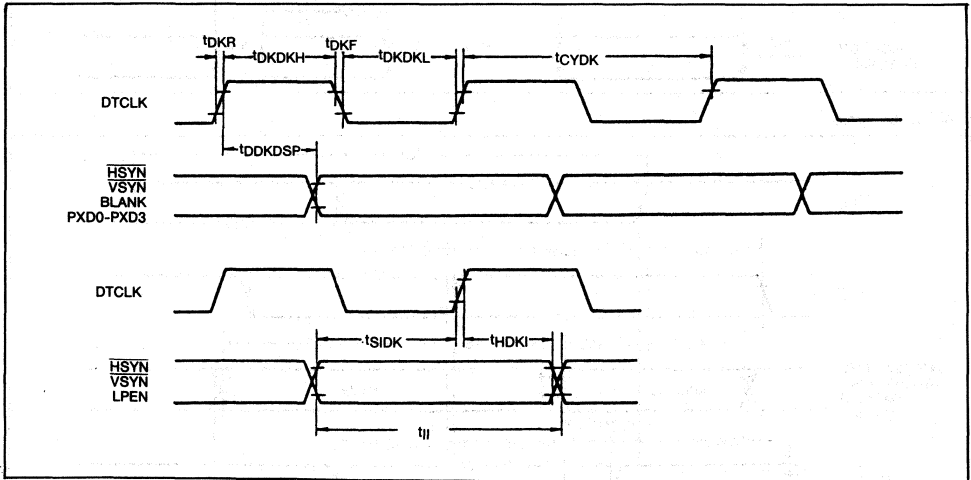




VRAM request



Display timing



### Overview

The μPD72120 Advanced Graphic Display Controller (AGDC) displays characters and graphics on a raster scan CRT according to commands and parameters received from a host processor or CPU. It has high speed graphic drawing capabilities, video timing signal generation, large capacity display memory control (including Video RAMs) and a versatile CPU interface. Also word processing systems can use the AGDC with a lot of advantages. Both packed pixel mode specially for high speed vector drawing and plane mode for full use of various bit block transfers (BIT BLT) with almost image processing power are available. Fill and paint algorithms with user definable patterns allow the AGDC to operate in 3-dimensional graphics systems.

### Features

\*High speed graphics drawing functions

- Graphics drawing
  - Dot, straight line, rectangle, circle, arc, sector, ellipse, ellipse arc and ellipse sector
  - Maximum drawing speed:
    - 500 ns/pixel (8MHz, pixel mode)
    - 500 ns/dot (8 MHz, plane mode)

- Painting (High speed processing in word units)
  - Non-arbitrary enclosed area painting (Fill): triangle, trapezoid and circle.
  - Arbitrary enclosed area painting (Paint): boundary dot retrieval

- Data transfers in display memory
  - Multiplane transfers
  - Data transformation (90°/180°/270° rotation and reversal)
  - Transfer speed: 500 ns/word max.

- Position specification by X-Y coordinates

- Logic operations between planes

\*Video timing signal generation

- High speed processing by two system clocks: display (for video sync signal generator) and graphics drawing clocks

- External synchronization

\*Large capacity display memory control

- Display memory bus interface (24-bit address and 16-bit data bus for addressing up to 16 Mwords, 16 bits/word)

- Video RAM (VRAM) control

- Display memory bus arbitration

\*Host Processor (CPU) Interface

- System bus interface – 20-bit address bus, 8 or 16 bit data bus

- System memory < – – > display memory data transfer with external DMA Controller

From system memory to display memory (PUT)  
From display memory to system memory (GET)

- High speed pipeline processing with preprocessor before drawing processor

- CPU memory or I/O mapping of internal registers and display memory – efficient system interface

\*8 MHz System Clock

\*CMOS Technology

\*Single +5 V Power Supply

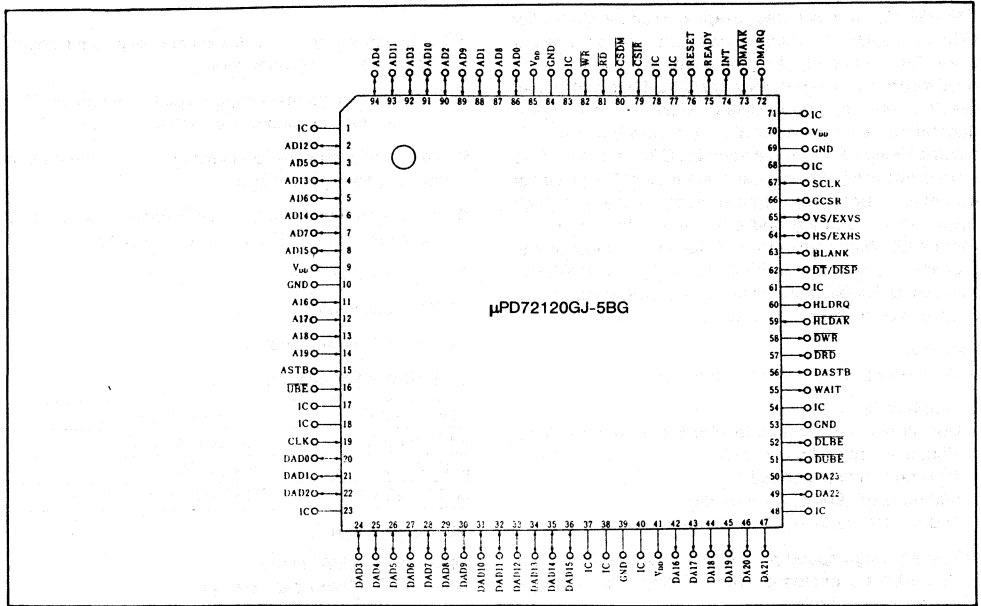
### Ordering Information:

Device Number	Package Type
μPD72120GJ	94-pin flat pack
μPD72120L	84-pin PLCC
μPD72120R	132-pin PGA

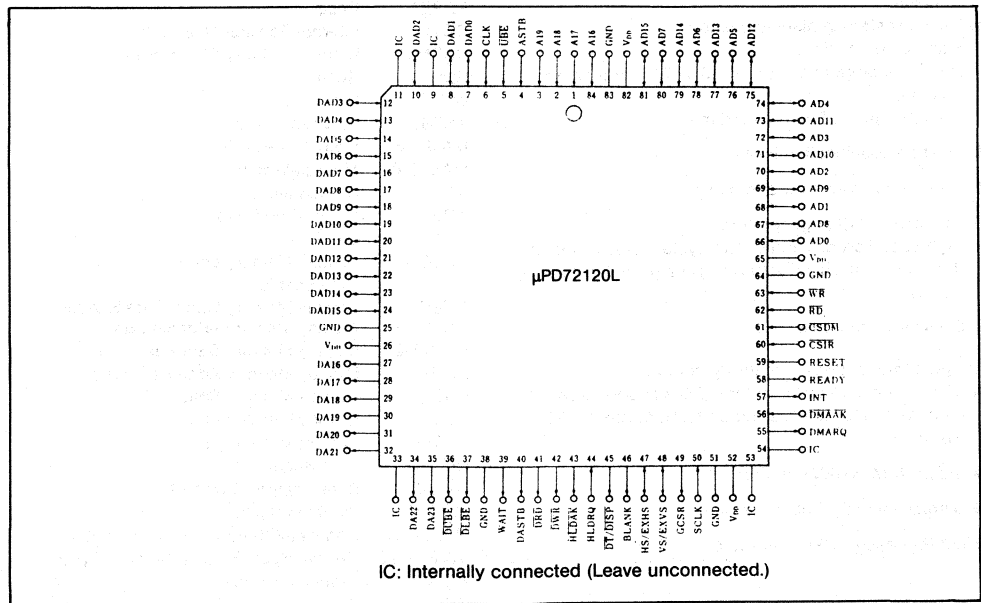
### Pin Configurations

DMARQ:	DMA Request
DMAAK:	DMA Acknowledge
INT:	Interrupt Request
READY:	Ready
RESET:	Reset
CSIR:	Internal Register Chip Select
CSDM:	Display Memory Chip Select
RD:	Read
WR:	Write
ASTB:	Address Strobe
MA19-16:	Main Address Bus
MAD15-0:	Main Data Bus
NC:	No connection
UBE:	Upper Byte Enable
CLK:	Clock
DLBE:	Display Memory Lower Byte Enable
DUBE:	Display Memory Upper Byte Enable
DA23-16:	Display Memory Address Bus
DAD15-0:	Display Memory Data Bus
DASTB:	Display Memory Address Strobe
DRD:	Display Memory Read
DWD:	Display Memory Write
HLDAK:	Hold Acknowledge
HLDRQ:	Hold Request
DT/DISP:	Data Transfer/Display Timing
BLANK:	Blanking Signal
HS/EXHS:	Horizontal Sync/External Horz. Sync
VS/EXVS:	Vertical Sync/External Vertical Sync
GCSR:	Graphics Cursor
SCLK:	Sync Generator Clock
GWAIT:	Graphics Wait

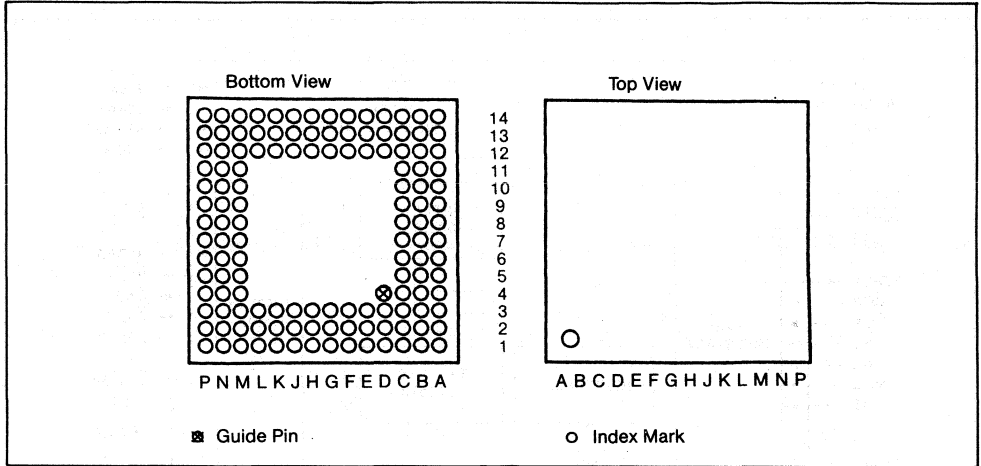
94-pin plastic QFP



84-pin PLCC



### Pin Configuration Pin Grid Array



Pin	Function	Pin	Function	Pin	Function	Pin	Function	Pin	Function	Pin	Function
A1		B9	WAIT	D3		H1	GND	L13	DAD4	N7	A16
A2		B10	DUBE	D12		H2	Vdd	L14	DAD7	N8	A19
A3		B11		D13	DA21	H3		M1		N9	
A4	GND	B12		D14	DA19	H12	DAD12	M2		N10	CLK
A5	GCSR	B13		E1		H13	DAD13	M3		N11	
A6	HS/EXHS	B14		E2	READY	H14	DAD14	M4		N12	
A7	HLDRQ	C1		E3	DMAAK	J1	AD0	M5	AD12	N13	
A8	HLDAK	C2		E12		J2	AD8	M6	AD14	N14	
A9	DASTB	C3		E13	DA18	J3	AD9	M7	GND	P1	
A10	GND	C4		E14	DA18	J12	DAD8	M8	ASTB	P2	
A11		C5		F1	CSDM	J13	DAD10	M9	DAD0	P3	AD5
A12	DA23	C6		F2	CSIR	J14	DAD11	M10		P4	AD6
A13		C7	BLANK	F3	RESET	K1	AD1	M11		P5	AD7
A14		C8	DRD	F12	DA17	K2	AD2	M12		P6	Vdd
B1		C9	DLBE	F13		K3	AD3	M13	DAD3	P7	A17
B2		C10	DA22	F14	Vdd	K12	DAD5	M14	DAD6	P8	A18
B3		C11		G1		K13		N1	AD4	P9	UBE
B4	Vdd	C12		G2	WR	K14	DAD9	N2		P10	
B5	SCLK	C13		G3	RD	L1	AD10	N3		P11	DAD1
B6	VS/EXVS	C14	DA20	G12	GND	L2	AD11	N4		P12	DAD2
B7	DT/DISP	D1	INT	G13		L3		N5	AD13	P13	
B8	DWR	D2	DMARQ	G14	DAD15	L12		N6	AD15	P14	



### Pin Functions

#### Clock Pins

Terminal Name	I/O	Active level	Function
CLK	I	-	Clock supplied to the circuits other than the sync signal generator and display processor. The drawing processor and preprocessor speed depend on this clock frequency.
SCLK	I	-	Clock supplied to the sync signal generator and the display processor. This clock frequency is determined by the CRT timing requirements - i.e., horizontal sync frequency, the number of dots per line, etc.

#### System Bus Control Pins

Terminal Name	I/O	Active level	Function															
MAD15-0	I/O	-	I/O bus to the CPU consisting of multiplexed 16-bit address and a bidirectional data bus.															
MA19-16	I	-	Upper four address bits of the 20-bit address.															
ASTB	I	H	Latches the address on MA19-16 and MAD15-0 on the falling edge.															
UBE	I	L	Together with MAD0 defines the data access format as shown below. UBE should be tied high when connected to an 8-bit CPU. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>MAD0</th> <th>UBE</th> <th>Data Access Format</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Even address word</td> </tr> <tr> <td>0</td> <td>1</td> <td>Even address byte</td> </tr> <tr> <td>1</td> <td>0</td> <td>Odd address byte</td> </tr> <tr> <td>1</td> <td>1</td> <td>Odd address byte</td> </tr> </tbody> </table>	MAD0	UBE	Data Access Format	0	0	Even address word	0	1	Even address byte	1	0	Odd address byte	1	1	Odd address byte
MAD0	UBE	Data Access Format																
0	0	Even address word																
0	1	Even address byte																
1	0	Odd address byte																
1	1	Odd address byte																

#### System Bus Control Pins (continued)

Terminal Name	I/O	Active level	Function
RD	I	L	Performs a read of data from the AGDC by the host CPU.
WR	I	L	Performs a write of data to the AGDC from the host CPU.
CSIR	I	L	Enables reading/writing of AGDC internal registers by the host CPU. The register is selected by the address input on MAD7-0.
CSDM	I	L	Enables reading/writing of display memory through the AGDC by the host CPU. The display memory address is generated by the address input on MA19-16 and MAD15-0 and by the bank register.
READY	O	H	Activated by the data access request (RD/WR) for the AGDC. During the access, the signal is low. RESET will set the READY line high.
INT	O	H	Signals an interrupt from the AGDC.
DMARQ	O	H	Indicates a request for data transfer (PUT/GET) to an external DMA controller. DMARQ will be low after RESET.
DMAAK	I	L	Acknowledgment of DMA request to the AGDC by the DMA controller.
RESET	I	H	Initializes operation of the AGDC. The internal parameter register is not cleared by RESET (it is initialized by data).

Display Memory Control Pins

Terminal Name	I/O	Active level	Function																								
DAD15-0	I/O	-	I/O pins for display memory – 16-bit address multiplexed with data.																								
DA23-16	0	-	Upper 8 bits of display memory address (the lower 16 bits of the 24-bit address are output on DAD15-0).																								
DASTB	0	H	Indicates that a display memory address is present on the falling edge.																								
DUBE DLBE	0	L	Defines the data format for accessing the display. RESET sets both pins low.  <table border="1" style="margin-left: 40px;"> <thead> <tr> <th></th> <th>DUBE</th> <th>DLBE</th> <th>Data access format</th> </tr> </thead> <tbody> <tr> <td>AGDC</td> <td>0</td> <td>0</td> <td>Word</td> </tr> <tr> <td>16-bit CPU</td> <td>0</td> <td>0</td> <td>Word</td> </tr> <tr> <td>8/16-bit CPU</td> <td>0</td> <td>1</td> <td>High (odd) byte</td> </tr> <tr> <td>16-bit CPU</td> <td>1</td> <td>0</td> <td>Low (even) byte</td> </tr> <tr> <td>8-bit CPU</td> <td>1</td> <td>1</td> <td>High (odd) byte</td> </tr> </tbody> </table>		DUBE	DLBE	Data access format	AGDC	0	0	Word	16-bit CPU	0	0	Word	8/16-bit CPU	0	1	High (odd) byte	16-bit CPU	1	0	Low (even) byte	8-bit CPU	1	1	High (odd) byte
	DUBE	DLBE	Data access format																								
AGDC	0	0	Word																								
16-bit CPU	0	0	Word																								
8/16-bit CPU	0	1	High (odd) byte																								
16-bit CPU	1	0	Low (even) byte																								
8-bit CPU	1	1	High (odd) byte																								
DRD	0	L	Controls reading of the display memory by the AGDC. Set high by RESET.																								
DWR	0	L	Controls writing to the display memory by the AGDC. Set high by RESET.																								
HLDRQ	I	H	Requests control of the display memory bus by an external device to transfer display data.																								
HLDAK	0	L	Indicates that the AGDC memory bus (DA23-26 and DAD15-0) is in the high impedance state so that an external device can have access to display memory bus. Set high by RESET.																								

Video Timing Signal Related Pins

Terminal Name	I/O	Active level	Function
VS/EXVS	I/O	H	When the AGDC operates as the master, VS is the vertical sync signal. When the AGDC operates as a slave, EXVS initializes the internal vertical sync signal on the rising edge.
HS/EXHS	I/O	H	When the AGDC operates as the master, HS is the horizontal sync signal output. When the AGDC operates as a slave, EXHS initializes the internal horizontal sync signal on the rising edge.

Display Signal Related Pins

Terminal Name	I/O	Active level	Function
BLANK	0	H	Used to blank the display.
DT/DISP	0	L	Set to DT in the DT mode (VRAMs used) and specifies the data transfer. In the cycle steal mode (VRAMs not used) indicates display cycle.
GCSR	0	H	Specifies the display of the graphic cursor.
GWAIT	0	H	Graphics wait signal.

### Power Supply and Ground Pins

Terminal Name	I/O	Active level	Function
VDD	-	-	+5 V power supply
GND	-	-	Ground

## Functional Description

### 2.1 Features

#### High Speed Graphics Drawing

- Graphic Drawing

The AGDC has graphic drawing commands to draw dots, straight lines, rectangles, circles and arcs, all of which are indispensable for CAD/CAM, office automation, document processing and printing. In addition, the AGDC supports the drawing of sectors, ellipses, ellipse arcs and ellipse sectors as advanced graphic primitives. These high speed graphics are drawn at maximum rate of 500 ns/pixel (8 MHz clock) for a straight line, 1 μs/pixel for a curved line (arc, etc.).

- Painting

The AGDC can paint or fill in a triangle, rectangle, trapezoid or circle as well as any enclosed area. This powerful feature is useful not only for document processing but advanced three dimensional graphics as well. Since the tiling pattern for painting can be freely set in the display memory, the areas to be painted can be drawn with any of a wide assortment of colors. In the past, painting was performed by software on the host CPU which is time consuming. For this reason, painting was used in only limited applications. The AGDC can upgrade the performance for painting speed by as much as several hundred times compared to painting done by a host CPU. This enables painting to be applied more extensively through the use of the AGDC.

- Transfer of Data in Display Memory (COPY)

The "COPY" commands refers to bit-block transfers (also known as bilbit). These commands transfer a rectangular area of any size and bit position to another similarly sized rectangle. The "COPY" command performs powerful character drawing and window control functions.

In the past, a character could be displayed only with a fixed size and at a set position (word boundary) on the screen. The word processing applications of today require more flexibility in the display of characters. As a result, it is necessary to display characters of different fonts, styles and sizes and to proportionally space characters. The "COPY" function of the AGDC

can satisfy all of these demands and at high speed – 500 ns/16 bits.

Personal computers are making use of multiple windows on display screens. The "COPY" command allows the user to easily specify the number, shape and size of multiple windows. Further, the "COPY" command of the AGDC can perform various logical operations between source and destination planes.

- Image Processing

The AGDC is able to do more than just copy data. It can slant copy, arbitrary angle rotation copy, 16/n enlargement copy or n/16 shrink copy (n any integer from 1 to 16). These copies transfer data from a rectangular source area to a non-rectangular destination.

The slant copy can be used for drawing italic characters. The arbitrary angle rotation copy is useful for document preparation because it can rotate characters graphics, images, etc. The enlarge and shrink copies are effective for editing and patching documents.

These AGDC commands relieve the host CPU of what was a software intensive function. The AGDC can quickly accomplish these image processing tasks and for a lower cost.

- Position Specification by X-Y Coordinates

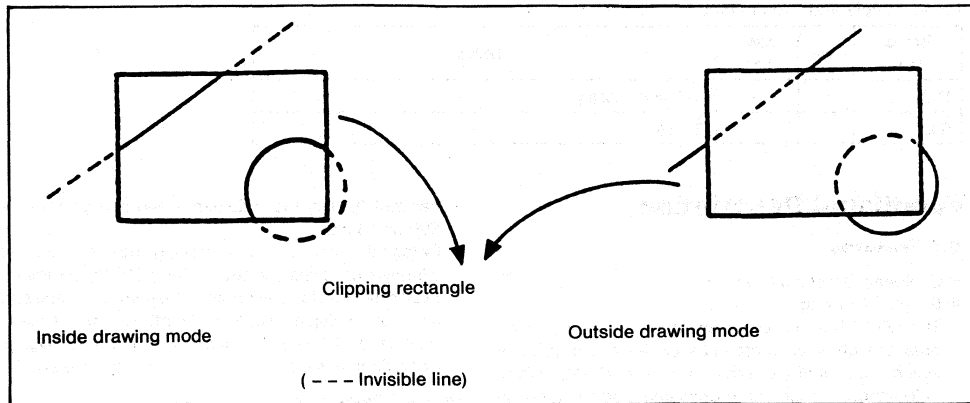
The graphics drawing and copy parameters can be given as X-Y coordinates thereby relieving the host CPU of the address calculation requirements.

- Hardware Clipping

Clipping is used to draw either inside or outside a rectangular window defined by the user. For example, a command to draw a straight line from the outside to the inside of a window will only be seen inside (or outside) of the window when clipping is used.

The AGDC implements clipping in hardware so that the user need only specify the coordinates of the rectangular window and the mode to indicate whether to clip inside or outside of the window.





**Video Timing Signal Generation**

- **High Speed Processing by Dual System Clocks**  
Dual system clocks can be input to the AGDC – the drawing clock and the display clock. In a graphics system, the display rate depends on the resolution of the CRT; the clock frequency of the display controller depends on the display rate. In the past, only a single clock was used for the display controller. It was often necessary to limit the clock to 5 MHz even though the capacity of the display controller was 8 MHz. Therefore, the AGDC was designed to have the display clock independent of the drawing clock so that drawing can be performed at the highest possible speed.
- **External Sync Input**  
In systems that incorporate a separate circuit to generate the video timing signals and use an AGDC only to draw graphics, or in systems that use multiple AGDCs to achieve higher performance, it is necessary to synchronize the AGDC with the external circuitry or the AGDC with other AGDCs. Therefore, the AGDC has external sync input capability to synchronize its operations with other AGDCs or with external circuitry.

**Large Capacity Display Memory Control**

- **Display Memory Bus Interface**  
With its 24-bit addressing and 16-bit data, 16 Mwords of 16 bits/word display memory can be configured with the AGDC. This means that up to 64 planes of 2048 x 2048 dots can be connected.
- **Video RAM (VRAM) Control**  
VRAMs are dual-ported DRAMs with internal line buffer shift registers (i.e., μPD41264). The VRAMs provide a separate port for display data so that when drawing, memory access time is greatly reduced. As a result, most of the entire memory cycle can be used as the drawing cycle to improve system throughput. The

AGDC supplies the signals needed to control VRAMs.

- **Display Memory Bus Arbitration**  
In a basic system, the display memory is accessed by the AGDC only. However, in a more advanced system, the display memory may be accessed by other processors. In other words, a local bus is established between the display memory and the AGDC or other processors. In this case, it is necessary for the AGDC to periodically provide refresh and display control and thus act as the local bus master. Other processors on the display memory bus (i.e., an image processor) are slaves and must request use of the bus from the AGDC. The AGDC controls the display memory bus and can grant the use of the bus to other processors. The AGDC incorporates display memory bus arbitration logic enabling higher performance systems to be constructed for a lower cost.

**CPU Interface**

- **System Memory Bus Interface**
- **CPU Mapping of Internal Registers and Display Memory**  
The AGDC system memory bus interface to the host CPU is independent of the display memory bus. The width of the address which the CPU can input to the AGDC is 20 bits and the data width is 8 or 16 bits. The 20-bit address is used when the display parameters, drawing parameters and commands are set and the CPU accesses the display memory directly. The internal AGDC registers and the display memory can be mapped in the CPU memory or I/O space. This allows the CPU to efficiently execute special drawing processing directly in the display memory. The mapping of AGDC registers in the CPU memory space provides for quick access of information to and from the AGDC.

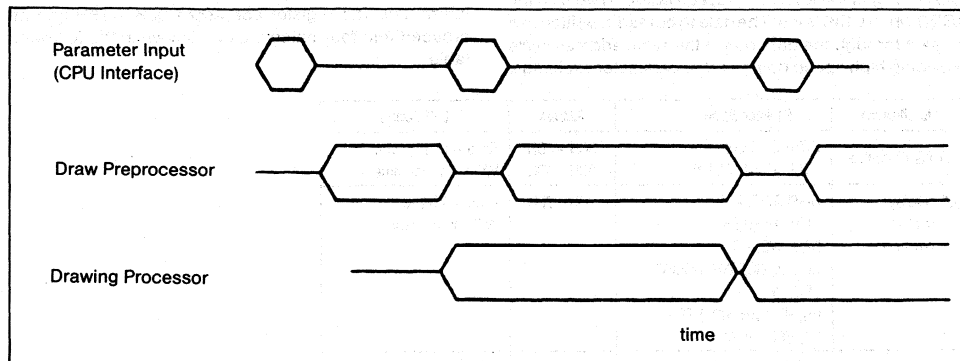
● System Memory < - - - > Display Memory Data Transfer

As described above, it is possible to map the display memory in the CPU memory space. It is also possible to execute PUT/GET commands - to transfer the data between the system memory and the display memory at high speed through the use of an external DMA controller. PUT is the command to transfer data from the system memory to the display memory. GET is the command to transfer data from the display memory to the system memory.

● High Speed Pipeline Drawing Processing

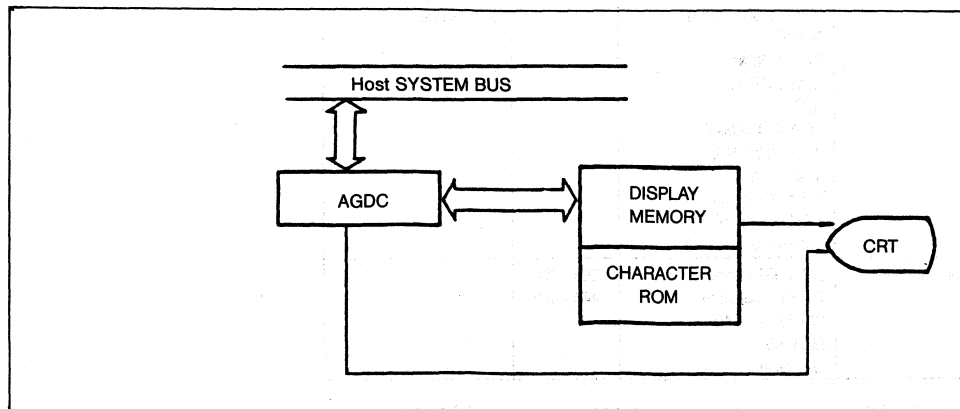
The parameters given by the CPU to the AGDC are greatly reduced by using X-Y coordinates. However, it is

necessary for the AGDC to calculate physical memory addresses for the drawing processor. The process to convert the X-Y coordinates from the CPU into physical addresses is done by the drawing preprocessor. The drawing processor executes the actual drawing commands independent of the preprocessor. Therefore the drawing processor and preprocessor can operate concurrently. As shown in the figure below, the throughput of the system is improved by the use of the drawing preprocessor and the drawing processor in a pipeline.



5

The figure below shows the basic configuration of a system using the AGDC.



**2.2 Command/Parameter Exchange  
Between the CPU and AGDC**

It is necessary for the CPU to set various parameters and commands to operate the drawing and display control functions of the AGDC. To set these parameters and commands, the CPU writes them directly to the AGDC internal registers.

The internal registers of the AGDC can be classified into registers the CPU can read and write directly and registers in which reading or writing is prohibited. The registers which can be read and written are assigned unique addresses. They are mapped in the CPU memory or I/O space and referred to as the "command/parameter table". These registers are selected by the lower 8 bits (00H - FFH) of the address input on MAD0 - MAD7 to the AGDC when CSIR is low. The data to be read or written can be sent through the data bus in the same address cycle providing high speed command or parameter exchange.

The AGDC incorporates a preprocessor for drawing preprocessing and a drawing processor for the actual drawing. The drawing preprocessor calculates the effective (physical) address from X-Y coordinates (logical address) given by the CPU and generates the parameters of the micro-level codes interpreted by the drawing processor. The drawing preprocessor decodes commands and performs the drawing preprocessing necessary to execute commands in the command/parameter table. It is not necessary to write all the parameters in the command/parameter table. Only the required parameters for the particular command need be written.

Of the addresses 00H - FFH in the internal register space, 80H - FFH are not presently used. These are reserved for future use. The register addresses 00H - 7FH can be grouped into four categories as shown in the following table.

Classification	Register Name	Address	CPU Access
(1) AGDC control	STATUS (Read) CTRL, BANK (Write)	3CH - 3DH 3CH - 3DH	Read - at any time Write - at any time
(2) Display related registers	DISPLAY FLAGS DISPLAY PITCH DAD, WC, AC GCSRX, GCSRYS, GCSRVE CRS, CE HS, HBP, HH, HD, HFP VS, VBP, L/F, VFP	70H - 7FH	Read - inhibited Write at any time
(3) Drawing related registers	EADORG, dADORG EAD1, dAD1 EAD2, dAD2 PDISPS, PDISPD PITCHS, PITCHD PMAX, MOD1, MOD0 PLANES PTNP, PTNCNT STACK, STMAX CLIP XCLMIN, XCLMAX YCLMIN, YCLMAX MACH, MAGV X, DX, XS, XE, XC, DH Y, DY, YS, YE, YC, DV COMMAND	00H - 1FH 40H - 6FH	Read at any time Write - 3 types of handshaking selectable A. status flag B. ready signal C. INT signal
(4) Data port	DMAPORT (during PUT/GET execution) DX (during READ DP/READ COL execution)	3EH - 3FH 44H - 45H	

Note: The DX register is used as the logical address (coordinate) setting register and at the same time as the data port during execution of the READ DP/READ COL command.

The registers in the first classification – STATUS, CTRL and BANK – are assigned address 3CH – 3DH. The contents of the STATUS register can be read at any time. The CTRL and BANK registers can be written to by the CPU at any time.

The registers in the second group – display related registers – are assigned addresses in the range 70H – 7FH. The contents of these registers cannot be read. Data can be written to these registers at any time. However, writing to these registers can disturb the CRT display. To prevent this, the display can be blanked while writing to these registers. First set the SD bit in the DISPLAY FLAGS register (address 70H – 71H) to “1”. Second, write the data into the display related register group. Third, set the SD bit from “1” back to “0”.

Follow these procedures to write data (SYNC parameters) such as HD, HBP, HH, HD, HFD, VS, VBP and L/F on the registers at address 7EH – 7FH:

1. Set the SYNC bit in the DISPLAY FLAGS register (70H – 71H) to “1”
2. Write, in the order listed, HS, HBP, HH, HD, HFP, VS, VBP, L/F AND VFP (address 7EH – 7FH).
3. Set the SYNC bit to “0”

The registers in the drawing related group are assigned addresses from 00H to 1FH and 40H to 6FH. The pre-processor resides in this register group. The read/write option of these registers are:

**– Read**

The CPU can read the contents of these registers at any time. The preprocessor stops for one clock while the contents of a register are read.

**– Write**

One of three writing procedures can be used:

1. Check the STATUS register PPBSY flag (address 3CH)
2. Use the READY signal
3. Use the INT pin to signal an interrupt to the CPU

The method described by (1) is the general method. The contents of the status flag are read to check that the preprocessor is not in operation. If the preprocessor is not busy, then data can be written to the command/parameter table.

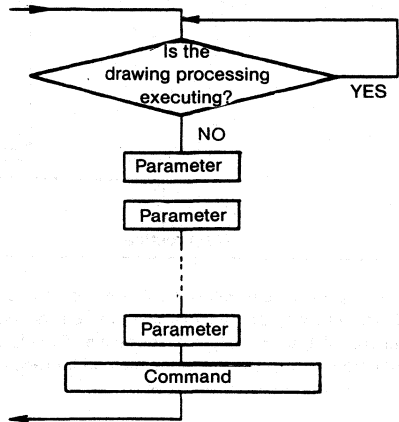
In the method described by (2), the CPU is tied to the AGDC's READY line. If the preprocessor is busy, the CPU cannot write the data until the preprocessor is finished and the READY line goes high. This procedure enables the command/parameter data to be sent at a higher speed.

In the method described by (3), the CPU is interrupted by the AGDC when the preprocessor is available. The AGDC,

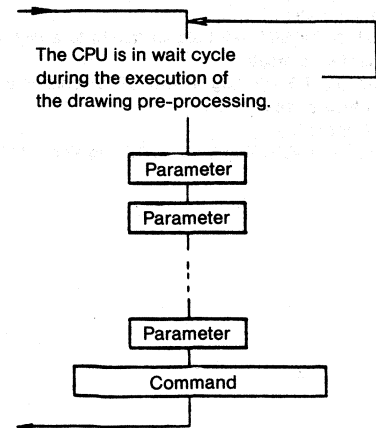
through its INT line, informs the CPU to send command/parameter data. The CPU interrupt routine should send a command and parameters to the ACDC. The “INT” line is enabled or disabled by a flag in the CTRL register (address 3DH).

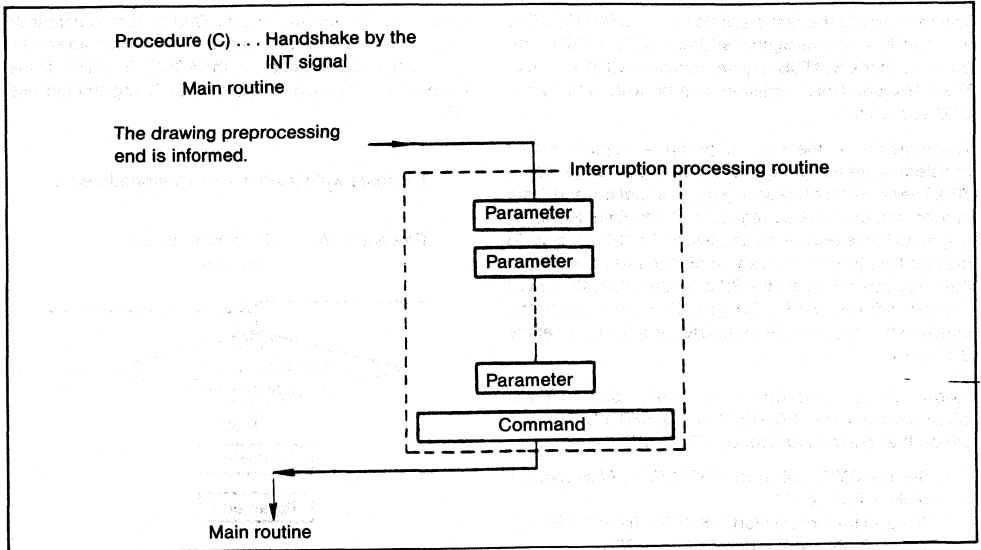
The three write methods are illustrated below.

**Procedure (A) . . . Handshake by the busy flag**



**Procedure (B) . . . Handshake by the READY signal**





The data port register, DMAPORT, is at address 3EH – 3FH. The drawing processor uses this register during execution of the drawing preprocessing. Therefore, to read/write this register, follow the method for reading or writing the drawing preprocessor.

**2.3 Reset and Abort Operation**

The AGDC resets or aborts processing by any of the following procedures:

- Reset Operation
  1. Set the RESET input signal (pin 5) to a high level (hardware reset)
  2. Set the RESET flag in the CTRL register to "1" (software reset).
- Abort Operation
  3. Set the ABORT flag in the CTRL register to "1".

Reset/Abort Operation	Operation
(1) Hardware Reset	<ul style="list-style-type: none"> <li>- sets the display stop flag SC in the DISPLAY FLAGS register (70H – 71H) to "1"</li> <li>- sets all bits in CTRL register (3DH) to "0"</li> <li>- all other registers maintain the same status</li> <li>- stops the preprocessor and the drawing processor</li> <li>- initializes the video timing signals</li> <li>- stops the image memory direct access</li> </ul>
(2) Software Reset	<ul style="list-style-type: none"> <li>- same as the hardware reset but does not set the display stop flag in the DISPLAY FLAGS register (70H – 71H) to "1" and does not initialize the video timing signals</li> </ul>
(3) Abort	<ul style="list-style-type: none"> <li>- all registers maintain their status</li> <li>- stops the preprocessor and drawing processor</li> </ul>

### Electric Characteristics (Target)

Caution: These standard are merely suggested targets and may differ from the electric characteristics of samples or mass-produced products.

Absolute Maximum Ratings (Ta = 25°C)

Parameter	Symbol	Test Conditions	Rating	Unit
Power Supply Voltage	VDD		-0.5 to +7.0	V
Input Voltage	VI		-0.5 to 7.0	V
Output Voltage	VO		-0.5 to +7.0	V
Operating Temperature	topt		-10 to +70	°C
Storage Temperature	tstg		-65 to +150	°C
Power dissipation	Pd		605	mW

DC Characteristics (Ta = -10 to 70°C, VDD = +5V ± 10%)

Parameter	Symbol	Test Conditions	Limits		Unit
			Min	Max	
Input Voltage Low	VIL	Except CLK or SCLK	-0.5	0.8	V
		CLK, SCLK	-0.5	0.6	V
Input Voltage High	VIH	Except CLK or SCLK	2.2	VDD+0.5	V
		CLK, SCLK	3.5	VDD+1.0	V
Output Voltage Low	VOL	IOL = 2.2 mA		0.45	V
Output Voltage High	VOH	IOH = -400μA	0.7VDD		V
Input Leakage Current Low	ILIL	VI = 0V		-10	μA
Input Leakage Current High	ILIH	VI = VDD		10	μA
Output Leakage Current Low	ILOL	VI = 0V		-10	μA
Output Leakage Current High	ILOH	VI = VDD		10	μA
Supply Current	IDD			110	mA

Capacitance (Ta = 25°C, VDD = GND = 0V)

Parameter	Symbol	Test Conditions	Standard Value		Unit
			Min	Max	
Input Capacitance	CI	f = 1MHz Unmeasured Pins returned to 0V.		10	pF
Output Capacitance	CO			20	pF
I/O Capacitance	CI/O			20	pF
Clock Input Capacitance	CC			20	pF

AC characteristics (Ta = -10°C to +70°C, VDD = +5V ± 10%)

Clock (CLK, SCLK)

Parameter	Symbol	Condition	Limits		Unit
			Min	Max	
Clock period	CLK	tCYK	125	600	ns
	SCLK	tCYSK	tCYK ≤ tCYSK	125	600
Clock width HIGH	CLK	tWKH	52		ns
	SCLK	tWSKH	52		ns
Clock width LOW	CLK	tWKL	52		ns
	SCLK	tWSKL	52		ns
Clock rise time	CLK	tKR		15	ns
	SCLK	tSKR		15	ns
Clock fall time	CLK	tKF		15	ns
	SCLK	tSKF		15	ns

Reset, interrupt

Parameter	Symbol	Condition	Limits		Unit
			Min	Max	
Reset pulse width	tRST		2		tCYSK
CLK ↑ → INT ↑ delay time	tDKI	CL = 50 pF		50	ns
RD ↓ → INT ↓ delay time	tDRI	STATUS read		3tCYK+50	ns

DMA read/write cycle

Parameter	Symbol	Condition	Limits		Unit
			Min	Max	
CLK↑ → DMARQ output delay time	tDKMQ	CL = 50 pF		50	ns
DMARQ set up time to DMAAK↓	tMAMQ		0		ns
DMAAK setup time to RD↓	tSRMA		0		ns
DMAAK hold time from RD↑	tHRMA		0		ns
DMAAK setup time from WR↓	tSWMA		0		ns
DMAAK hold time from WR↑	tHWMA		0		ns

### Image memory bus read cycle

Parameter	Symbol	Condition	Limits		Unit
			Min	Max	
CLK↑ → address output delay time	tDKA	CL = 50 pF		30	ns
Input data setup time to CLK↑	tSKD		20		ns
Input data hold time from CLK↑	tHKD		0		ns
CLK↑ → DASTB↑ delay time	tDKDSH	CL = 50 pF		30	ns
CLK↓ → DASTB↓ delay time	tDKDSL	CL = 50 pF		30	ns
CLK↑ → DRD delay time	tDKDR	CL = 50 pF		30	ns

### System bus write cycle

Parameter	Symbol	Condition	Limits		Unit
			Min	Max	
CS setup time to WR↓	tSWC		0		ns
CS hold time from WR↑	tHWC		0		ns
WR width LOW	tWWL		50		ns
WR width HIGH	tWWH		50		ns
Data setup time to WR↑	tSWD		30		ns
Data hold time from WR↑	tHWD		0		ns
WR↓ → READY↑ delay time	tDWRV	CL = 50 pF		30	ns
WR hold time from READY↑	tHRYW		30		ns
CLK↑ → READY↑ delay time	tDKRY	CL = 50 pF		40	ns
ASTB setup time to WR↓	tSWAS		0		ns

### HLDQR, HLDAK

Parameter	Symbol	Condition	Limits		Unit
			Min	Max	
CLK↑ → HLDAK delay time	tDKHA	CL=50 pF		50	ns
HLDQR setup time to CLK↑	tSKHQ		tCYK/2+20		ns

### Display memory bus write cycle

Parameter	Symbol	Condition	Limits		Unit
			Min	Max	
CLK↑ → DWR delay time	tDKDW	CL = 50 pf		30	ns
CLK↑ → data output delay time	tDKD	CL = 50 pF		30	ns



System bus read cycle

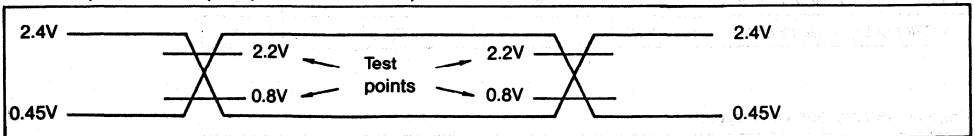
Parameter	Symbol	Condition	Limits		Unit
			Min	Max	
$\overline{CS}$ setup time to $\overline{RD}\downarrow$	tSRC		0		ns
$\overline{CS}$ hold time from $\overline{RD}\uparrow$	tHRC		0		ns
$\overline{RD}$ width HIGH	tWRH		50		ns
ASTB pulse width	tWAS		20		ns
ASTB setup time to $\overline{RD}\downarrow$	tSRAS		0		ns
Address setup time to $\overline{ASTB}\downarrow$	tSASA		20		ns
Address hold time from $\overline{ASTB}\downarrow$	tHASA		0		ns
Data setup time to $\overline{READY}\uparrow$	tSRYD		0		ns
Data float delay time from $\overline{RD}\uparrow$	tFRD		0	40	ns
$\overline{RD}\downarrow \rightarrow \overline{READY}\downarrow$ delay time	tDRRY	$C_L = 50 \text{ pF}$		30	ns
$\overline{RD}$ hold time from $\overline{READY}\uparrow$	tHRYR		0		ns
$\text{CLK}\uparrow \rightarrow \overline{READY}\uparrow$ delay time	tDKRY	$C_L = 50 \text{ pF}$		40	ns

Display cycle

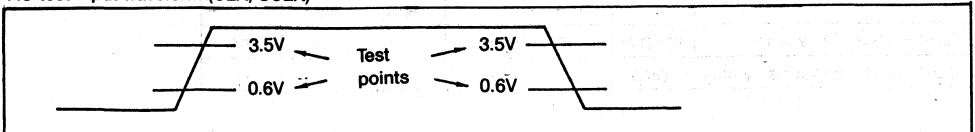
Parameter	Symbol	Condition	Limits		Unit
			Min	Max	
$\text{SCLK}\uparrow \rightarrow \overline{DASTB}\downarrow$ delay time	tDSKDASH	$C_L = 50 \text{ pF}$		30	ns
$\text{SCLK}\downarrow \rightarrow \overline{DASTB}\downarrow$ delay time	tDSKDASL	$C_L = 50 \text{ pF}$		30	ns
$\text{SCLK}\uparrow \rightarrow \overline{DT}/\overline{DISP}$ delay time	tDSKDT	$C_L = 50 \text{ pF}$		30	ns
$\text{SCLK}\uparrow \rightarrow$ address delay time	tDSKA	$C_L = 50 \text{ pF}$		30	ns
$\text{SCLK}\uparrow \rightarrow$ output signal (Note) delay time	tDSKO	$C_L = 50 \text{ pF}$		50	ns
$\text{SCLK}\uparrow \rightarrow \overline{WAIT}$ delay time	tDSKWT	$C_L = 50 \text{ pF}$		70	ns
$\overline{WAIT}$ pulse width	tWWT	$C_L = 50 \text{ pF}$	4tCYSK-70	50	ns

Note: The output signal denotes HS, VS, BLANK, or GCSR.

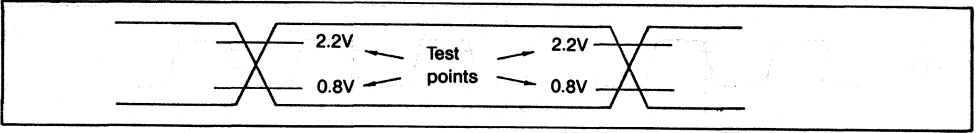
AC test input waveform (except for CLK or SCLK)



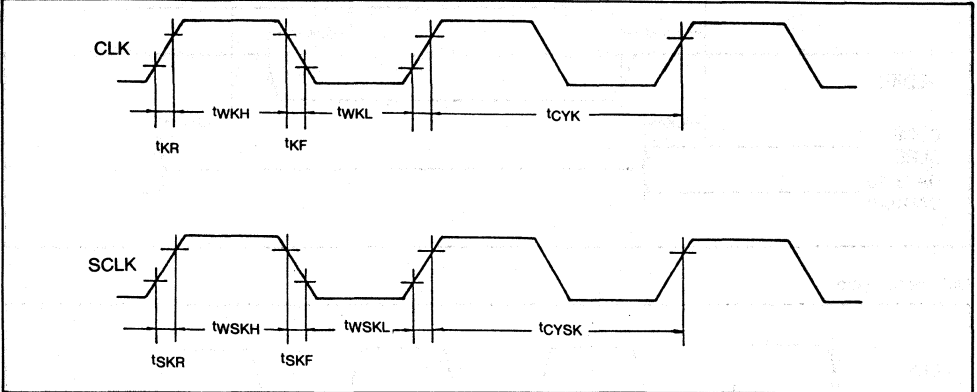
AC test input waveform (CLK, SCLK)



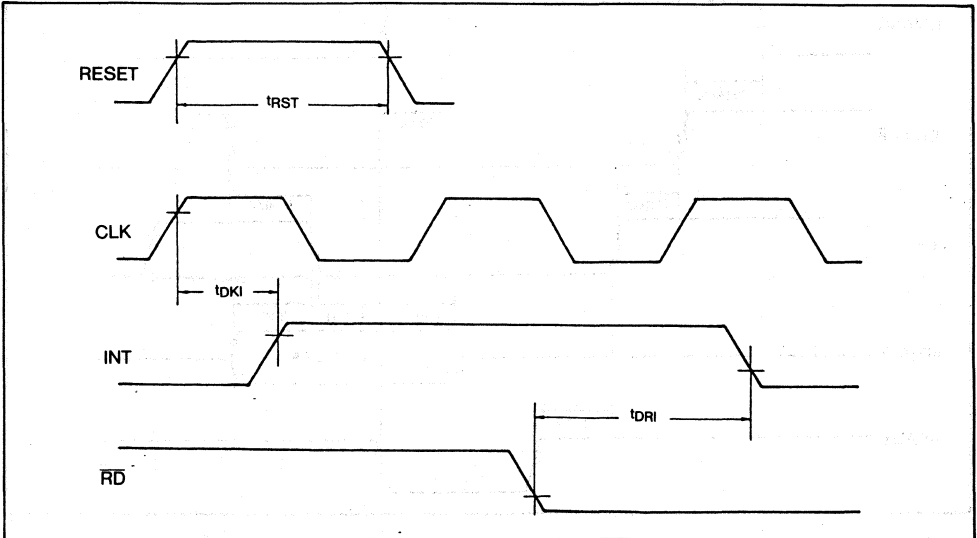
### AC test output waveform



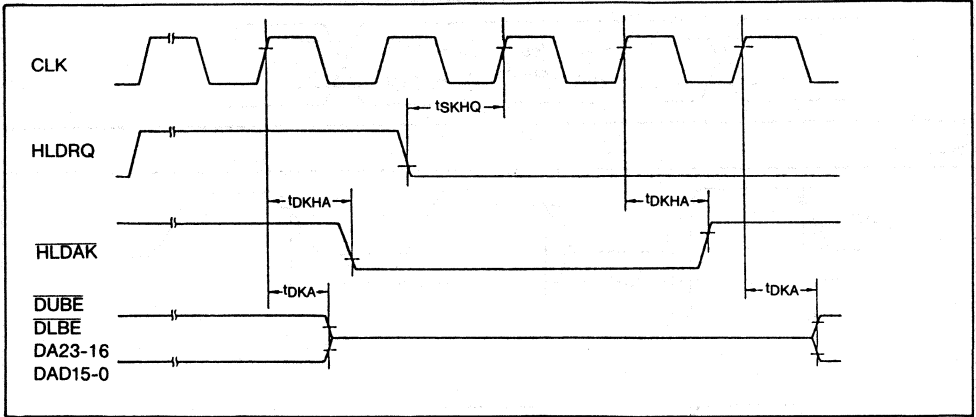
### Clock



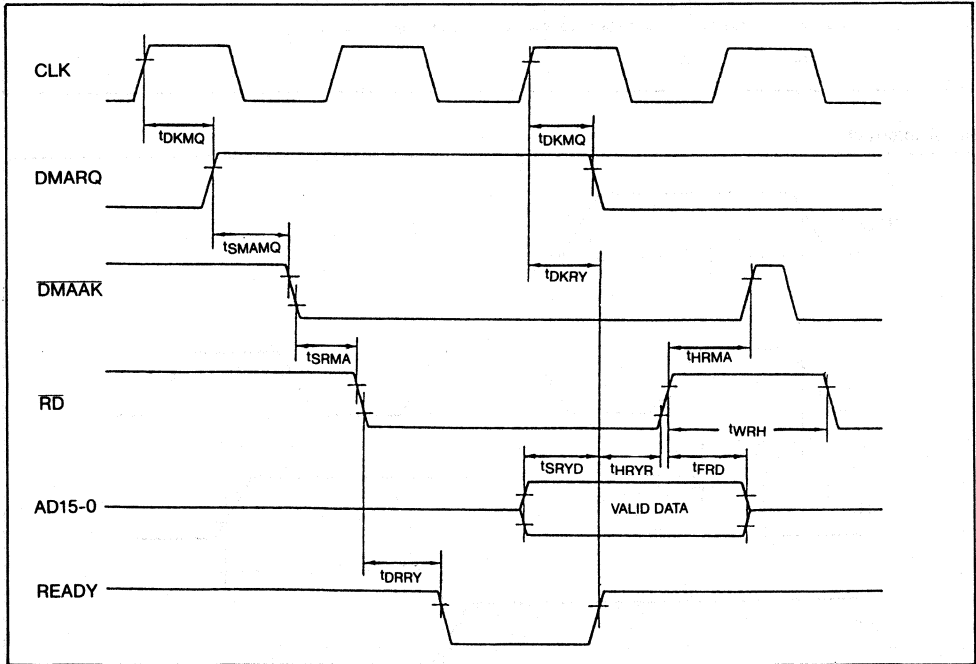
### Reset, interrupt



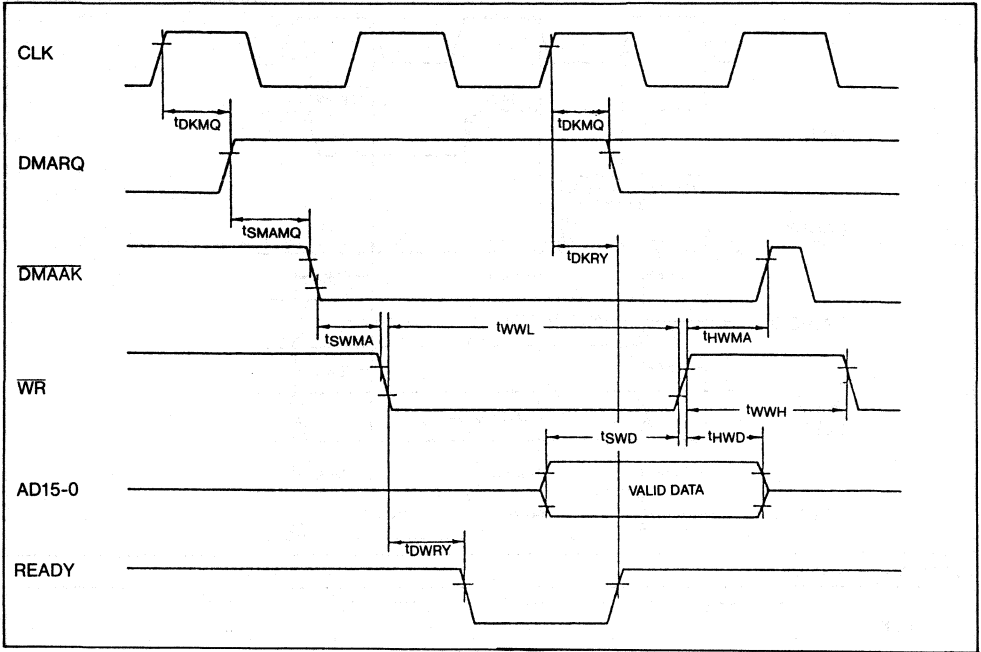
HLD $\overline{RQ}$ , HLD $\overline{A\overline{K}}$



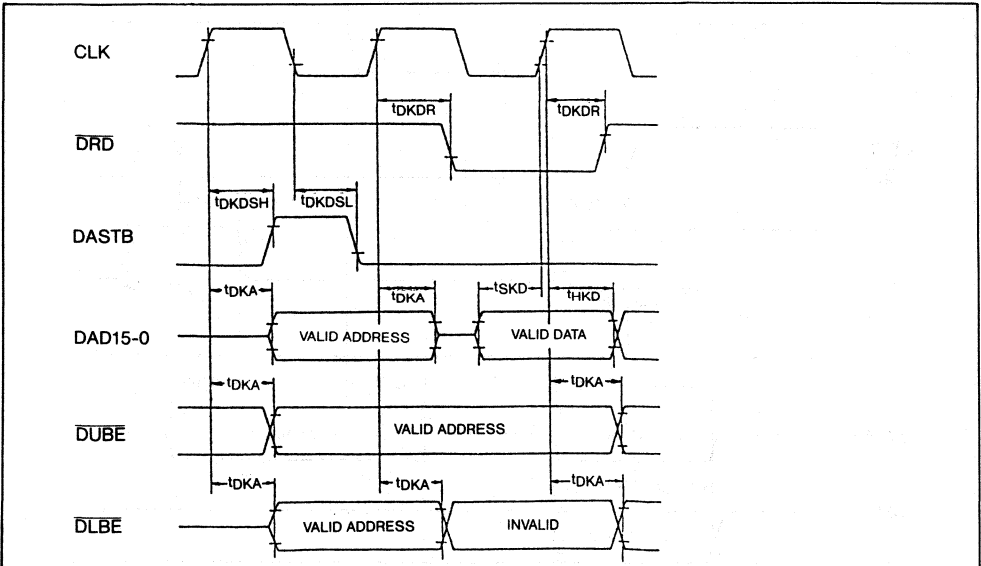
DMA read cycle



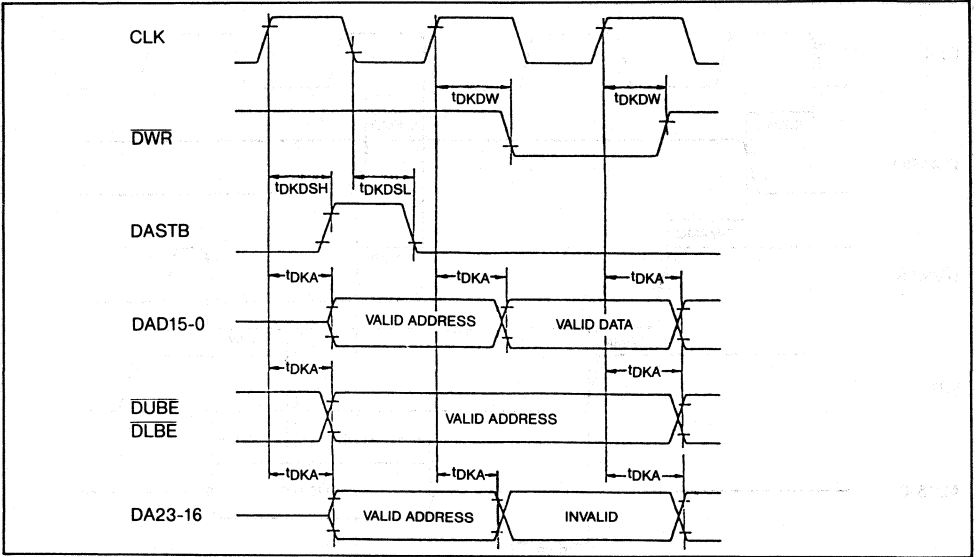
### DMA write cycle



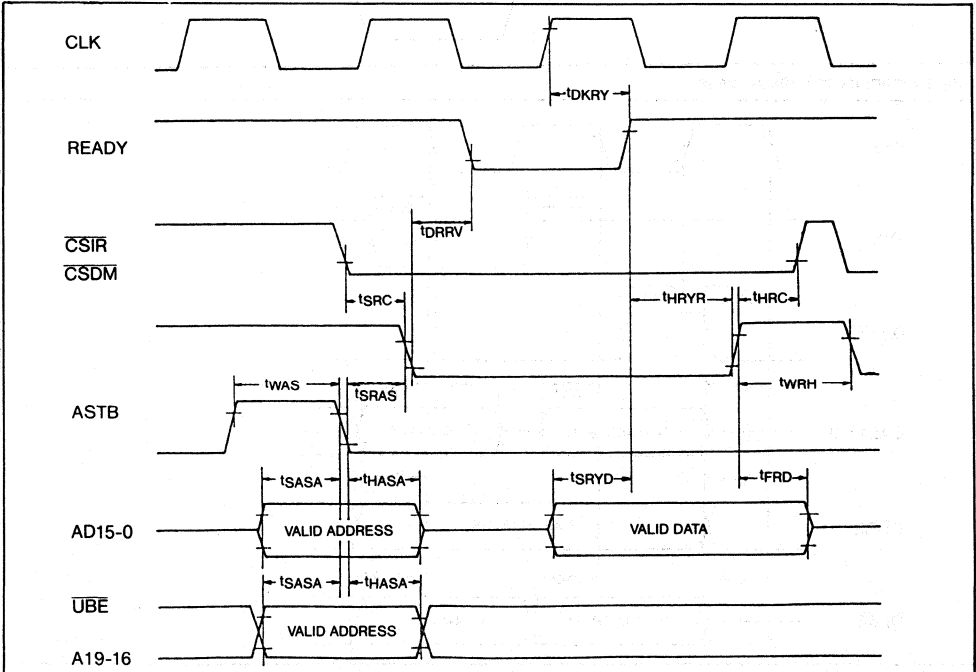
### Display memory bus ready cycle



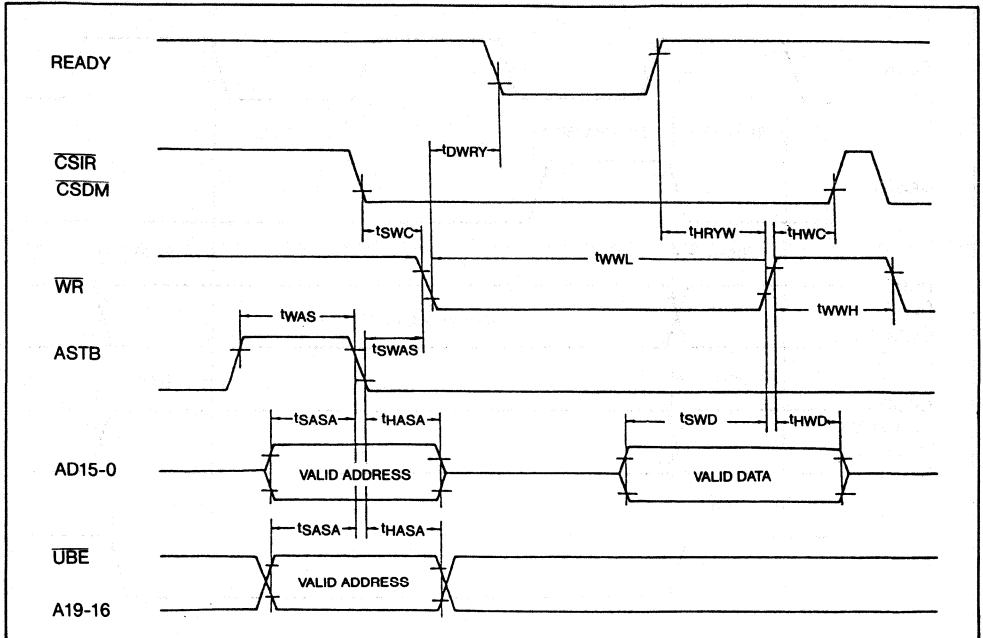
Display memory bus write cycle



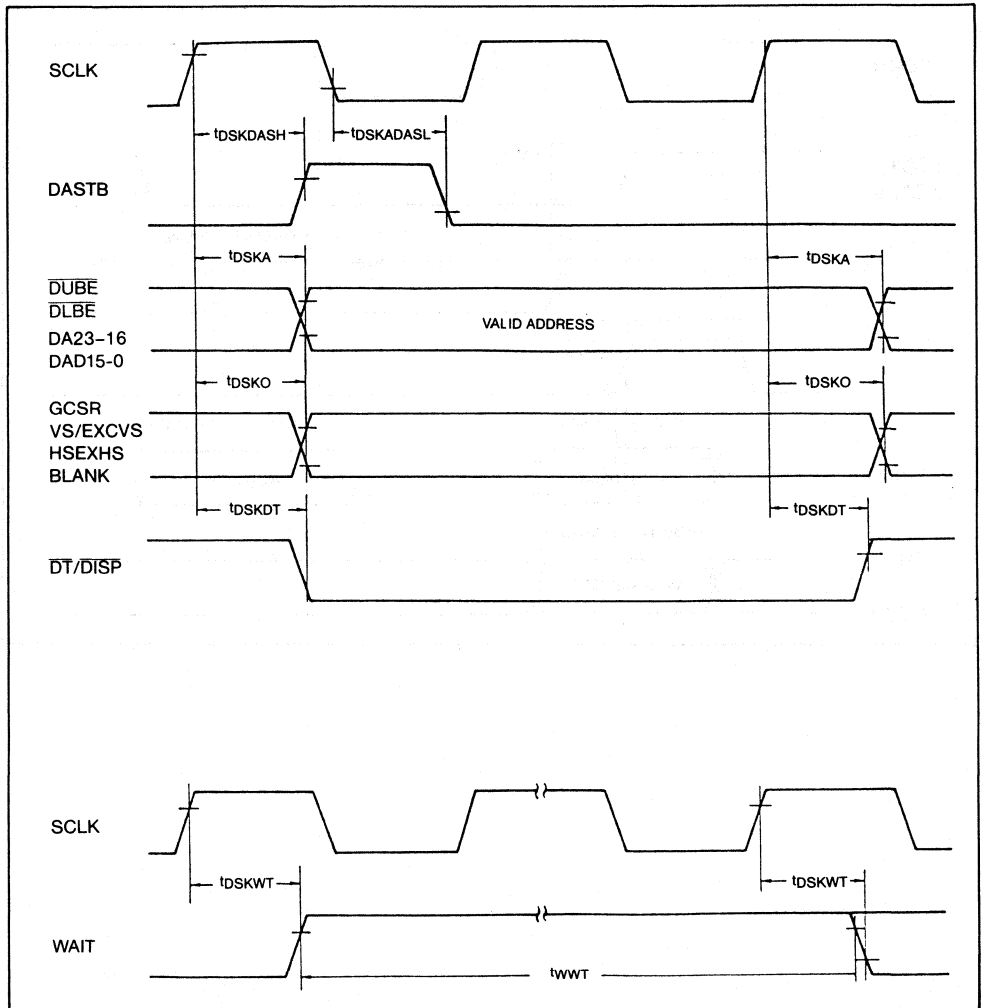
System bus read cycle



### System bus write cycle



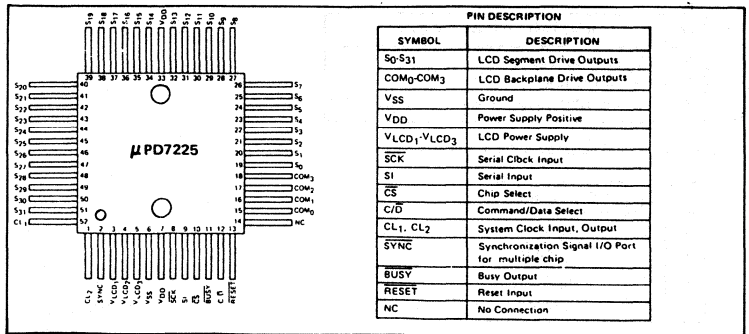
Display cycle



**DESCRIPTION** The μPD7225 is a programmable peripheral device containing all the circuitry necessary for interfacing a microprocessor to a wide variety of alpha-numeric Liquid Crystal Displays (LCDs). The display controller hardware automatically synchronizes the drive signals for any static or multiplexed LCD containing up to 4 backplanes, and up to 32 segments. The μPD7225 is fully compatible with most microprocessors, and communicates with them through a 2-line, 8-bit Serial port. It can be easily configured into multiple chip designs for larger LCD applications. In addition, the μPD7225 includes on board 8-segment Numeric and 15-segment Alpha-Numeric decoders, and programmable blinking capabilities. The μPD7225 is manufactured with a low-power single 5V CMOS process, and is available in a 52-pin plastic flat package.

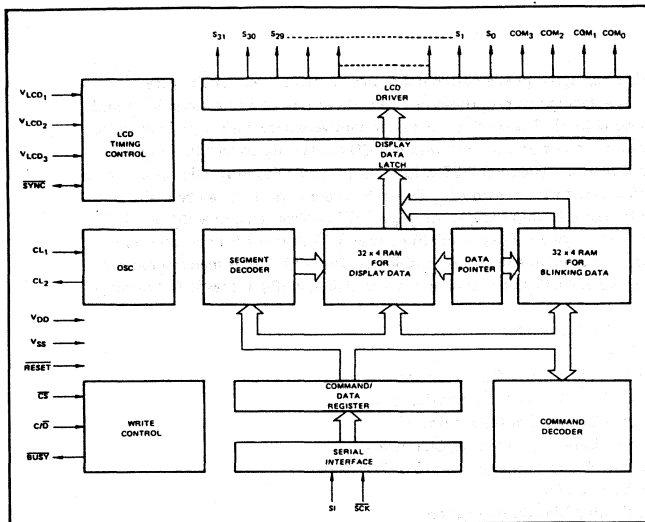
- FEATURES**
- Single Chip LCD Controller
  - Direct LCD Drive
  - Selectable Backplane Drive Configuration
    - Static; 2-, 3-, or 4-Backplane Multiplexed
  - Programmable Display Configurations
    - 8-Segment Numeric — up to 16 Characters
    - 15-Segment Alpha-Numeric — up to 8 Characters
  - 32-Segment Drive Lines
  - Selectable Display Bias Configuration
    - Static; 1/2 or 1/3
  - Automatic Synchronization of Segment and Backplane Drive Lines
  - Dual 32 x 4 Bit RAMs for Display Data Storage
  - Programmable Display Data Addressing
    - Individual Segment
    - 16-Character, 8-Segment Numeric Decoder
    - 64-Character, 15-Segment Alpha-Numeric Decoder
  - Programmable Blinking Capability
    - Individual Segment, Individual Character, or Entire Display
  - 8-Bit Serial Interface
  - Compatible with most 4-Bit, 8-Bit, and 16-Bit Microprocessors
  - Fully Cascadable for Larger LCD Applications
  - Single +5V Power Supply
  - CMOS Technology
  - 52-Pin Plastic Flat Package

**PIN CONFIGURATION**





BLOCK DIAGRAM



1. MODE SET

0	1	0	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	40-5F
---	---	---	----------------	----------------	----------------	----------------	----------------	-------

COMMAND DESCRIPTION

The MODE SET command sets up the Backplane Drive Configuration, the Display Bias Voltage Configuration, and the A/C Drive Frequency for the μPD7225.

The Backplane Drive Configuration is defined as follows:

D <sub>3</sub>	D <sub>2</sub>	Backplane Drive Configuration
0	1	Static (1-Backplane)
1	1	2-Backplane Multiplexed
1	0	3-Backplane Multiplexed
0	0	4-Backplane Multiplexed

The Display Bias Voltage Configuration is defined as follows:

D <sub>4</sub>	Display Bias Voltage Configuration
0	1/3 (three voltage)
1	1/2 (two voltage)
X	Static (single voltage; default when D <sub>3</sub> D <sub>2</sub> = 00)

The A/C Drive Frequency is defined as follows:

D <sub>1</sub>	D <sub>0</sub>	A/C Drive Frequency
0	0	f <sub>c</sub> /2 <sup>7</sup> Hz
0	1	f <sub>c</sub> /2 <sup>8</sup> Hz
1	0	f <sub>c</sub> /2 <sup>9</sup> Hz
1	1	f <sub>c</sub> /2 <sup>11</sup> Hz

Note: LCD Frame Frequency =  $\frac{\text{A/C Drive Frequency}}{\text{\# of active Backplane Drive lines}}$

2. UNSYNCHRONOUS DATA TRANSFER

0 0 1 1 0 0 0 0	30
-----------------	----

The Normal Transfer of data from the Display Data RAM to the segment output latches occurs with the rising edge of CS. The UNSYNCHRONOUS DATA TRANSFER command implements this mode of data transfer, and also disables the SYNCHRONOUS DATA TRANSFER operation.

3. SYNCHRONOUS DATA TRANSFER

0 0 1 1 0 0 0 1	31
-----------------	----

Data can also be transferred from the Display Data RAM to the segment output latches with the rising edge of  $f_c$ . The SYNCHRONOUS DATA TRANSFER command implements this mode of data transfer, and also disables the UNSYNCHRONOUS DATA TRANSFER operation.

4. INTERRUPT DATA TRANSFER

0 0 1 1 1 0 0 0	38
-----------------	----

Occasionally, the Host microprocessor system may experience events, such as prioritized Hardware interrupts, that may disrupt communications with the  $\mu$ PD7225. Display Data transfers to the  $\mu$ PD7225 may be interrupted, without disrupting the  $\mu$ PD7225 internal display data protocol, by issuing an INTERRUPT DATA TRANSFER command at the beginning of the interrupt service routine. Display data updating may be resumed in an orderly fashion after the interrupt service routine is completed.

5. CLEAR Display Data

0 0 1 0 0 0 0 0	20
-----------------	----

All locations in the Display Data RAM are set to zero by executing the CLEAR DISPLAY DATA command. The Data Pointer is also cleared, and set to its initial location.

6. CLEAR BLINKING DATA

0 0 0 0 0 0 0 0	00
-----------------	----

All locations in to Blinking Data RAM are set to zero by executing the CLEAR BLINKING DATA command. The Data Pointer is also cleared, and set to its initial location.

7. LOAD DATA POINTER

0 0 0 D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	E0-FF
--	-------

To access a particular location in either the Display Data RAM, or the BLINKING DATA RAM the Data Pointer must be given the corresponding address of that location. The LOAD DATA POINTER command transfers 5 bits of immediate data to the Data Pointer.

8. WRITE DISPLAY DATA

1 1 0 1 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	D0-DF
---	-------

The WRITE DISPLAY DATA command transfers 4 bits of immediate data to the Display Data RAM location addressed by the Data Pointer. After the transfer is complete, the Data Pointer is automatically incremented.

**9. WRITE BLINKING DATA**

1	1	0	0	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	CO-CF
---	---	---	---	----------------	----------------	----------------	----------------	-------

**COMMAND DESCRIPTION (CONT.)**

The WRITE BLINKING DATA command transfers 4 bits of immediate data to the Blinking Data RAM location addressed by the Data Pointer. After the transfer is complete, the Data Pointer is automatically incremented.

**10. ENABLE DISPLAY**

0	0	0	1	0	0	0	1	11
---	---	---	---	---	---	---	---	----

The ENABLE DISPLAY command turns on the LCD, and starts the automatic display controller hardware of the μPD7225.

**11. DISABLE DISPLAY**

0	0	0	1	0	0	0	0	10
---	---	---	---	---	---	---	---	----

The DISABLE DISPLAY command turns off the LCD, and stops the automatic display controller hardware of the μPD7225.

**12. ENABLE BLINKING**

0	0	0	1	1	0	1	D <sub>0</sub>	1A-1B
---	---	---	---	---	---	---	----------------	-------

If a particular LCD application requires blinking several segments, the appropriate information must have been transferred to the Blinking Data RAM previously. The ENABLE BLINKING command selects the Blinking frequency according to the value of D<sub>0</sub>, and turns the Blinking feature on.

D <sub>0</sub>	Blinking Frequency
0	$f_c/2^{16}$ Hz
1	$f_c/2^{17}$ Hz

**13. DISABLE BLINKING**

0	0	0	1	1	0	0	0	18
---	---	---	---	---	---	---	---	----

The DISABLE BLINKING command turns the Blinking feature OFF.

**14. ENABLE SEGMENT DECODER**

0	0	0	1	0	1	0	1	15
---	---	---	---	---	---	---	---	----

The μPD7225 has an internal 8-segment Numeric data decoder, and an internal 15-segment Alpha-Numeric data decoder. These decoders can be used for automatic display data addressing, by the Host microprocessor to absorb some of the system overhead required to decode display data for the μPD7225.

The ENABLE SEGMENT DECODER command implements this mode of display data addressing. Upon execution, display data received by the μPD7225 is diverted to one of the segment decoders. The segment decoder then writes display data to the Display Data RAM. The distinction between 8-segment decoding and 15-segment decoding is made by the MSB of the display data:

MSB	Decoding Selected
0	8-segment Numeric
1	15-segment Alpha-Numeric

### 15. DISABLE SEGMENT DECODER

0 0 0 1 0 1 0 0	14
-----------------	----

The DISABLE SEGMENT DECODER command stops the segment decode addressing, and enables the transfers of Display Data from the Host microprocessor directly to the Display Data RAM.

### 16. OR DISPLAY DATA

1 0 1 1 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	80-BF
---	-------

The OR DISPLAY DATA command performs a LOGICAL OR between the Display Data addressed by the Data Pointer, and 4 bits of immediate data. The result is written to the same Display Data location, and the Data Pointer is automatically incremented.

### 17. AND DISPLAY DATA

1 0 0 1 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	90-9F
---	-------

The AND DISPLAY DATA command performs a LOGICAL AND between the Display Data addressed by the Data Pointer, and 4 bits of immediate data. The result is written to the same Display Data location, and the Data Pointer is automatically incremented.

### 18. OR BLINKING DATA

1 0 1 0 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	A0-AF
---	-------

The OR BLINKING DATA command performs a LOGICAL OR between the Blinking Data addressed by the Data Pointer, and 4 bits of immediate data. The result is written to the same Blinking Data location, and the Data Pointer is automatically incremented.

### 19. AND BLINKING DATA

1 0 0 0 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	80-8F
---	-------

The AND BLINKING DATA command performs a LOGICAL AND between the Blinking Data addressed by the Data Pointer, and 4 bits of immediate data. The result is written to the same Blinking Data location, and the Data Pointer is automatically incremented.

**Command Summary**

COMMAND	DESCRIPTION	INSTRUCTION CODE									
		BINARY									
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	HEX	
1. Mode Set	Set up Driving Mode of LCD, including: 1) Backplane drive 2) Display Bias 3) LCD Frame Frequency	0	1	0	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	40-5F	
2. Unsynchronous Data Transfer	Synchronize writing of display data with CS	0	0	1	1	0	0	0	0	30	
3. Synchronous Data Transfer	Synchronize writing of display data with LCD Frame Frequency	0	0	1	1	0	0	0	1	31	
4. Interrupt Data Transfer	Interrupt writing of display data	0	0	1	1	1	0	0	0	38	
5. Clear Display Data	Clear the Display Data RAM and the Data Pointer	0	0	1	0	0	0	0	0	20	
6. Clear Blinking Data	Clear the Blinking Data RAM and the Data Pointer	0	0	0	0	0	0	0	0	00	
7. Load Data Pointer	Load Data Pointer with 5 Bits of Immediate Data	1	1	1	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	E0-FF	
8. Write Display Data	Write 4 Bits of Immediate Data to the Display Data Location addressed by the Data Pointer; Increment Data Pointer	1	1	0	1	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	D0-DF	
9. Write Blinking Data	Write 4 Bits of Immediate Data to the Blinking Data Location addressed by the Data Pointer; Increment Data Pointer	1	1	0	0	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	C0-CF	
10. Enable Display	Start Automatic LCD Controller Hardware	0	0	0	1	0	0	0	1	11	
11. Disable Display	Stop Automatic LCD Controller Hardware	0	0	0	1	0	0	0	0	10	
12. Enable Blinking	Start the Blinking Operation at the Frequency Specified by 1 Bit of Immediate Data	0	0	0	1	1	0	1	D <sub>0</sub>	1A-1B	
13. Disable Blinking	Stop Blinking Operation	0	0	0	1	1	0	0	0	18	
14. Enable Segment Decoder	Select 8-Segment Numeric or 15-Segment Alphanumeric Decoder Addressing	0	0	0	1	0	1	0	1	15	
15. Disable Segment Decoder	Stop Segment Decoder Addressing; Return to individual segment addressing	0	0	0	1	0	1	0	0	14	
16. OR Display Data	Perform a Logical OR between the Display Data addressed by the Data Pointer and 4 Bits of Immediate Data; Write Results to same Display Data Location; Increment Data Pointer	1	0	1	1	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	80-8F	
17. AND Display Data	Perform a Logical AND between the Display Data addressed by the Data Pointer and 4 Bits of Immediate Data; Write Result to same Display Data Location; Increment Data Pointer	1	0	0	1	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	90-9F	
18. OR Blinking Data	Perform a Logical OR between Blinking Data addressed by the Data Pointer and 4 Bits of Immediate Data; Write Result to same Blinking Data Location; Increment Data Pointer	1	0	1	0	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	A0-AF	
19. AND Blinking Data	Perform a Logical AND between Blinking Data addressed by the Data Pointer and 4 Bits of Immediate Data; Write Result to same Location; Increment Data Pointer	1	0	0	0	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	80-8F	

## Absolute Maximum Ratings\*

Supply Voltage	.....	-0.3~+ 7.0V
Input Voltage	.....	-0.3~V <sub>DD</sub> + 0.3V
Output Voltage	.....	-0.3~V <sub>DD</sub> + 0.3V
Operating Temperature	.....	-40~+85°C
Storage Temperature	.....	-40~+ 125 °C

COMMENT: Stress above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

\*T<sub>a</sub> = 25°C

## DC Electrical Characteristics

D. C. Characteristics (T<sub>a</sub> = -10~+70 °C, V<sub>DD</sub> = 5V ± 10 %

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Input High Voltage	V <sub>IH</sub>		0.7V <sub>DD</sub>		V <sub>DD</sub>	V
Input Low Voltage	V <sub>IL</sub>		0		0.3V <sub>DD</sub>	V
Output High Voltage	V <sub>OH</sub>	SYNC, BUSY, I <sub>OH</sub> = -10μA	V <sub>DD</sub> - 0.5			V
Output Low Voltage	V <sub>OL1</sub>	BUSY, I <sub>OL</sub> = 100μA			0.5	V
	V <sub>OL2</sub>	SYNC, I <sub>OL</sub> = 900μA			1.0	V
Output Short Circuit Current	I <sub>OS</sub>	SYNC, V <sub>O</sub> = 1V			-300	μA
Input Leakage High	I <sub>LIH</sub>	V <sub>I</sub> = 5.5V			2	μA
Input Leakage Low	I <sub>LIL</sub>	V <sub>I</sub> = 0V			-2	μA
Output Leakage High	I <sub>LOH</sub>	V <sub>O</sub> = V <sub>DD</sub>			2	μA
Output Leakage Low	I <sub>LOL</sub>	V <sub>O</sub> = 0V			-2	μA
Common Output Impedance	R <sub>COM</sub>	COM 0-COM 3, I <sub>DD</sub> ≅ V <sub>LCD</sub>		5	7	KΩ
Segment Output Impedance	R <sub>SEG</sub>	SO-S31, I <sub>DD</sub> ≅ V <sub>LCD</sub>		7	14	KΩ
Supply Current	I <sub>DD</sub>	CL1, External Clock, f <sub>c</sub> = 200KHz		100	250	μA

## AC Characteristics

A. C. Characteristics (T<sub>a</sub> = -10~+70 °C, V<sub>DD</sub> = 5V ± 10 %)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Operating Frequency	f <sub>c</sub>		50		200	KHz
Oscillation Frequency	f <sub>OSC</sub>	R = 180KΩ ± 5 %	85	130	175	KHz
Clock Pulse Width High	t <sub>WHC</sub>	CL1, External Clock	2		16	μs
Clock Pulse Width Low	t <sub>WLC</sub>	CL1, External Clock	2		16	μs
SCR Cycle	t <sub>CYC</sub>		1			μs
SCR Pulse Width High	t <sub>WHK</sub>		400			ns
SCR Pulse Width Low	t <sub>WLK</sub>		400			ns
BUSY ↑ → SCR ↓ Hold Time	t <sub>HBK</sub>		0			ns
SI Setup Time to SCR ↑	t <sub>SIK</sub>		250			ns
SI Hold Time after SCR ↑	t <sub>HK1</sub>		200			ns
8th SCK ↑ → BUSY ↓ Delay Time	t <sub>OKB</sub>	C <sub>L</sub> = 50pF			3	μs
CS ↓ → BUSY ↓ Delay Time	t <sub>OCB</sub>	C <sub>L</sub> = 50 pF			1.5	μs
C/D Setup Time to 8th SCK ↑	t <sub>SDK</sub>		9			μs
C/D Hold Time after 8th SCK ↑	t <sub>HKD</sub>		1			μs
CS Hold Time after 8th SCK ↑	t <sub>HCK</sub>		1			μs
CS Pulse Width High	t <sub>WCH</sub>	(2)				μs
CS Pulse Width Low	t <sub>WLC</sub>	(2)				μs
SYNC Load Capacitance	C <sub>LSY</sub>	t <sub>CYC</sub> = 5μs			50	pF

Notes (1) Applies to static, 1/2 and 1/3 multiplexed drive  
(2) 8<sup>1</sup>/<sub>C</sub>

**AC Characteristics**

A. C. Characteristics ( $T_a = 0 \sim +70^\circ\text{C}$ ,  $V_{DD} = 2.7\text{V} \sim 5.5\text{V}$ )

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Operating Frequency	$f_c$		50		140	KHz
Oscillation Frequency	$f_{OSC}$	$R=180\text{K}\Omega \pm 5\%$ , $V_{DD}=3\text{V} \pm 10\%$	50	40	140	KHz
Clock Pulse Width High	$t_{WHC}$	CL1, External Clock	3		16	$\mu\text{s}$
Clock Pulse Width Low	$t_{WLC}$	CL1, External Clock	3		16	$\mu\text{s}$
SCK Cycle	$t_{CYK}$		4			$\mu\text{s}$
SCK Pulse Width High	$t_{WHK}$		1.8			$\mu\text{s}$
SCK Pulse Width Low	$t_{WLK}$		1.8			$\mu\text{s}$
BUSY $\uparrow \rightarrow$ SCK $\downarrow$ Hold Time	$t_{HBK}$		0			ns
SI Setup Time to SCK $\uparrow$	$t_{SIK}$		1			$\mu\text{s}$
SI Hold Time after SCK $\uparrow$	$t_{HKL}$		1			$\mu\text{s}$
8th SCK $\uparrow \rightarrow$ BUSY $\downarrow$ Delay Time	$t_{DKB}$	$C_L = 50\text{pF}$			5	$\mu\text{s}$
CS $\downarrow \rightarrow$ BUSY $\downarrow$ Delay Time	$t_{DCB}$	$C_L = 50\text{ pf}$			5	$\mu\text{s}$
C/D Setup Time to 8th SCK $\uparrow$	$t_{SDK}$		18			$\mu\text{s}$
C/D Hold Time after 8th SCK $\uparrow$	$t_{HKD}$		1			$\mu\text{s}$
CS Hold Time after 8th SCK $\uparrow$	$t_{HCK}$		1			$\mu\text{s}$
CS Pulse Width High	$t_{WCH}$		(2)			$\mu\text{s}$
CS Pulse Width Low	$t_{WLC}$		(2)			$\mu\text{s}$
SYNC Load Capacitance	$C_{LSY}$	$t_{CYC} = 5\mu\text{s}$			50	pF

Notes (1) Applies to static and 1/3-multiplexed drive.  
(2) 8/1<sub>C</sub>

**DC Electrical Characteristics for LCD**

$2.7 \leq V_{LCD} \leq V_{DD}$

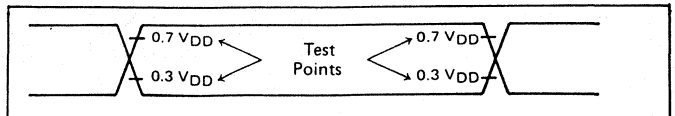
PARAMETER	SYMBOL	LIMITS			UNITS	CONDITIONS
		MIN	TYP	MAX		
Backplane Drive Output Impedance	R <sub>COM</sub>		2		k $\Omega$	COM <sub>0</sub> - COM <sub>3</sub> , Display Bias = 1/3 or Static
					k $\Omega$	COM <sub>0</sub> - COM <sub>3</sub> , Display Bias = 1/2
Segment Drive Output Impedance	R <sub>SEG</sub>		11		k $\Omega$	S <sub>0</sub> - S <sub>31</sub>

**Capacitance**

Capacitance ( $T_a = 25^\circ\text{C}$ ,  $V_{DD} = 0\text{V}$ )

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Input Capacitance	$C_{IN}$				10	pF
Output Capacitance	$C_{OUT1}$	Except BUSY			20	pF
Output Capacitance	$C_{OUT2}$	BUSY			15	pF
I/O Capacitance	$C_{IO}$	SYNC			15	pF
Clock Capacitance	$C_c$	CL1			30	pF

A. C. Timing Measurement Voltage



## EXTENDED TEMPERATURE RANGE (-40 to +85 °C)

### DC Electrical Characteristics

D. C. Characteristics (T<sub>a</sub> = -40 to +85 °C, V<sub>DD</sub> = 5V ± 10%)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Input High Voltage	V <sub>IH1</sub>	Except SCK	0.7V <sub>DD</sub>		V <sub>DD</sub>	V
	V <sub>IH2</sub>	SCK	0.75V <sub>DD</sub>		V <sub>DD</sub>	V
Input Low Voltage	V <sub>IL1</sub>	Except SCK	0		0.3V <sub>DD</sub>	V
	V <sub>IL2</sub>	SCK	0		0.25V <sub>DD</sub>	V
Output High Voltage	V <sub>OH</sub>	SYNC, BUSY, I <sub>OH</sub> = -7μA	V <sub>DD</sub> -0.5			V
Output Low Voltage	V <sub>OL1</sub>	BUSY, I <sub>OL</sub> = 100μA			0.5	V
	V <sub>OL2</sub>	SYNC, I <sub>OL</sub> = 400μA			0.5	V
Output Short Circuit Current	I <sub>OS</sub>	SYNC, V <sub>O</sub> = 0.5V (1)			-350	μA
Input Leakage High	I <sub>LIH</sub>	V <sub>I</sub> = 5.5V			2	μA
Input Leakage Low	I <sub>LIL</sub>	V <sub>I</sub> = 0V			-2	μA
Output Leakage High	I <sub>LOH</sub>	V <sub>O</sub> = V <sub>DD</sub>			2	μA
Output Leakage Low	I <sub>LOL</sub>	V <sub>O</sub> = 0V			-2	μA
Common Output	R <sub>COM</sub>	COM 0-COM 3, I <sub>I</sub> V <sub>DD</sub> ≥ V <sub>LCD</sub>		5	8	KΩ
Segment Output Impedance	R <sub>SEG</sub>	SO-S31, I <sub>I</sub> V <sub>DD</sub> ≥ V <sub>LCD</sub>		7	20	KΩ
Supply Current	I <sub>DD</sub>	CL 1, External Clock, V <sub>DD</sub> = 3V ± 10%, f <sub>c</sub> = 140KHz		90	250	μA

### AC Characteristics

A. C. Characteristics (T<sub>a</sub> = -40 to +85 °C, V<sub>DD</sub> = 5V ± 10%)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Operating Frequency	f <sub>c</sub>		75		180	KHz
Oscillation Frequency	f <sub>OSC</sub>	R = 180KΩ ± 5%	80	130	180	KHz
Clock Pulse Width High	t <sub>WHC</sub>	CL 1, External Clock	2		10	μs
Clock Pulse Width Low	t <sub>WLC</sub>	CL 1, External Clock	2		10	μs
SCK Cycle	t <sub>CYK</sub>		1.2			μs
SCK Pulse Width High	t <sub>WHK</sub>		500			ns
SCK Pulse Width Low	t <sub>WLK</sub>		500			ns
BUSY 1 → SCK ↓ Hold Time	t <sub>HBK</sub>		0			ns
S1 Setup Time to SCK ↑	t <sub>SIK</sub>		100			ns
S1 Hold Time after SCK ↓	t <sub>HKI</sub>		200			ns
8th SCK 1 → BUSY ↓ Delay Time	t <sub>DKB</sub>	C <sub>L</sub> = 50pF			3	μs
CS 1 → BUSY ↓ Delay Time	t <sub>DCB</sub>	C <sub>L</sub> = 50 pF			1.5	μs
C/D Setup Time to 8th SCK ↑	t <sub>SDK</sub>		9			μs
C/D Hold Time after 8th SCK ↓	t <sub>HKD</sub>		1			μs
CS Hold Time after 8th SCK ↓	t <sub>HKG</sub>		1			μs
CS Pulse Width High	t <sub>WCH</sub>		(3)			μs
CS Pulse Width Low	t <sub>WLC</sub>		(3)			μs
SYNC Load Capacitance	C <sub>LSY</sub>	t <sub>CYC</sub> = 5μs			50	pF

Notes (1) Output short circuit caused by conflict among SYNC outputs in multi-chip configuration  
 (2) Applies to static, 1/2- and 1/3-multiplexed drive  
 (3) 8/f<sub>c</sub>





### Introduction

The  $\mu$ PD7227 Intelligent Dot-matrix LCD Controller/Driver is a peripheral device designed to interface most microprocessors with a wide variety of dot matrix LCDs. It can directly drive any multiplexed LCD organized as 8 rows by 40 columns, and is easily cascaded up to 16 rows and 320 columns. The  $\mu$ PD7227 is equipped with several hardware logic blocks, such as an 8-bit serial interface, ASCII character generator, 40 x 16 static RAM with full read/write capability, and an LCD timing controller; all of which reduce microprocessor system software requirements. The  $\mu$ PD7227 is manufactured with a single 5V CMOS process, and is available in a space-saving 64-pin flat plastic package.

### Features

Single-chip LCD controller with direct LCD drive

Compatible with most microprocessors

Eight row drives

Designed for dot-matrix LCD configurations up to 320 dots

Designed for 5 x 7 dot-matrix character LCD configuration; up to 8 characters

Cascadable to 16 row drives

40 column drives

Cascadable to 320 column drives

Hardware logic blocks reduce system software requirements

8-bit serial interface for communication

ASCII 5 x 7 dot-matrix character generator with 64-character vocabulary

40 x 16 bit static RAM for data storage, retrieval, and complete back-up memory capability

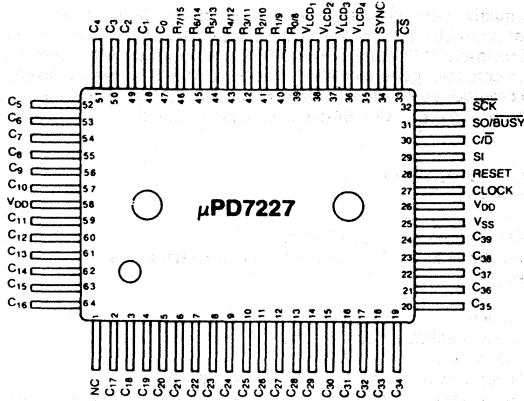
Voltage controller generates LCD bias voltages

Timing controller synchronizes column drives with sequentially-multiplexed row drives

Single +5V power supply

CMOS technology

Pin Configuration



Pin Description

Symbol	Description
R <sub>0/7</sub> -R <sub>8/15</sub>	LCD Row Drive Outputs
C <sub>0</sub> -C <sub>39</sub>	LCD Column Drive Outputs
$\overline{CS}$	Chip Select Input
C/D	Command/Data Select Input
SO/BUSY	Serial Output/Busy Output
SCK	Serial Clock Input
SI	Serial Input
SYNC	Synchronization Port for Cascaded Applications
CLOCK	System Clock Input
RESET	Reset Input
V <sub>DD</sub>	Power Supply Positive
V <sub>LCD1</sub> -V <sub>LCD4</sub>	LCD Power Supply
V <sub>SS</sub>	Ground
NC	No Connection

## LCD Panel

Dot Matrix		5 x 7 Dot Matrix Character		μPD7227s Required
8 Row	16 Row	8 Row	16 Row	
640 dots (8 x 80)	1280 dots (16 x 80)	16 characters plus 80 dots	2 rows of 16 characters plus 160 dots	2
960 dots (8 x 120)	1920 dots (16 x 120)	24 characters plus 120 dots	2 rows of 24 characters plus 240 dots	3
1280 dots (8 x 160)	2560 dots (16 x 160)	32 characters plus 160 dots	2 rows of 32 characters plus 320 dots	4
1600 dots (8 x 200)	3200 dots (16 x 200)	40 characters plus 200 dots	2 rows of 40 characters plus 400 dots	5
1920 dots (8 x 240)	3840 dots (16 x 240)	48 characters plus 240 dots	2 rows of 48 characters plus 480 dots	6
2240 dots (8 x 280)	4480 dots (16 x 280)	56 characters plus 280 dots	2 rows of 56 characters plus 560 dots	7
2560 dots (8 x 320)	5120 dots (16 x 320)	64 characters plus 320 dots	2 rows of 64 characters plus 640 dots	8

### Cascaded display configuration

One μPD7227 is selected as the master LCD controller, and the remaining μPD7227s become slave column drivers. You should connect the following pins of the master μPD7227 to the corresponding pins of each slave μPD7227.

CLOCK, RESET, V<sub>DD</sub>, V<sub>SS</sub>, VLCD1, VLCD2, VLCD3, VLCD4, SYNC,  
SO/BUSY C/D, SCK, SI

Each of the  $\overline{CS}$  inputs is brought out separately to the microprocessor.

Command Summary

Command	Description	Instruction Code							HEX	
		Binary								
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>		D <sub>0</sub>
1. MODE SET	Initialize the μPD7227, including selection of 1. LCD Drive Configuration 2. Row Driver Port Function 3. RAM Bank 4. SYNC Port Function	0	0	0	1	1	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	18-1F
2. FRAME FREQUENCY SET	Set LCD Frame Frequency	0	0	0	1	0	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	10-17
3. LOAD DATA POINTER	Load Data Pointer with 7 bits of Immediate Data	1	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	80-E7
4. WRITE MODE	Write Display Byte in Serial Register to RAM location addressed by Data Pointer; modify Data Pointer	0	1	1	0	0	1	D <sub>1</sub>	D <sub>0</sub>	64-67
5. READ MODE	Load RAM contents addressed by Data Pointer into Serial Register for output; modify Data Pointer	0	1	1	0	0	0	D <sub>1</sub>	D <sub>0</sub>	60-63
6. AND MODE	Perform a Logical AND between the display byte in the Serial Register and the RAM contents addressed by Data Pointer; write result to same RAM location; modify Data Pointer	0	1	1	0	1	1	D <sub>1</sub>	D <sub>0</sub>	6C-6F
7. OR MODE	Perform a Logical OR between the display byte in the Serial Register and the RAM contents addressed by Data Pointer; write Result to same RAM location; modify Data Pointer	0	1	1	0	1	0	D <sub>1</sub>	D <sub>0</sub>	68-6B
8. CHARACTER MODE	Decode display byte in Serial Register into 5 x 7 character with Character Generator; write character to RAM location addressed by Data Pointer; increment Data Pointer by 5	0	1	1	1	0	0	1	0	72
9. SET BIT	Set single bit of RAM location addressed by Data Pointer; modify Data Pointer	0	1	0	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	40-5F
10. RESET BIT	Reset single bit of RAM location addressed by Data Pointer; modify Data Pointer	0	0	1	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	20-3F
11. ENABLE DISPLAY	Turn on the LCD	0	0	0	0	1	0	0	1	09
12. DISABLE DISPLAY	Turn off the LCD	0	0	0	0	1	0	0	0	08

### Electrical Specification

#### Absolute Maximum Ratings\*

T<sub>a</sub> = 25°C

Power supply .....	-0.3V to + 7.0V
All inputs and outputs with respect to V <sub>SS</sub> .....	-0.3V to V <sub>DD</sub> +0.3
Storage temperature .....	-65°C to +150°C
Operating temperature .....	-10°C to +70°C

\*Comment: Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### DC Characteristics

**T<sub>a</sub> = -10°C to +70°C, V<sub>DD</sub> = +5.0V ± 10%**

Limits						
Parameter	Symbol	MIN	TYP	MAX	Units	Conditions
Input Voltage High	V <sub>IH</sub>	0.7 V <sub>DD</sub>		V <sub>DD</sub>	V	SI, $\overline{\text{SCK}}$ , C/D, $\overline{\text{CS}}$ , $\overline{\text{SYNC}}$ , RESET
Input Voltage Low	V <sub>IL</sub>	0		0.3 V <sub>DD</sub>	V	SI, $\overline{\text{SCK}}$ , C/D, $\overline{\text{CS}}$ , $\overline{\text{SYNC}}$ , RESET
Clock Voltage High	V <sub>βH</sub>	0.7 V <sub>DD</sub>		V <sub>DD</sub>	V	Clock, External Clock
Clock Voltage Low	V <sub>βL</sub>	0		0.3 V <sub>DD</sub>	V	Clock, External Clock
Input Leakage Current High	I <sub>LIH</sub>			10	μA	SI, $\overline{\text{SCK}}$ , C/D, $\overline{\text{CS}}$ , RESET V <sub>IH</sub> = V <sub>DD</sub>
Input Leakage Current Low	I <sub>LIL</sub>			- 10	μA	SI, SCK, C/D, CS, RESET V <sub>IH</sub> = 0V
Output Current High	I <sub>OH</sub>		- 2.0		mA	SO/BUSY, SYNC, V <sub>OH</sub> = V <sub>DD</sub> - 0.5V
Output Current Low	I <sub>OL</sub>		+ 3.5		mA	SO/ $\overline{\text{BUSY}}$ , V <sub>OL</sub> = 0.5V
LCD Operating Voltage	V <sub>LCD</sub>	- 3.0		V <sub>DD</sub>	V	8-Row Multiplexed LCD Drive Configuration
			V <sub>DD</sub>			16-Row Multiplexed LCD Drive Configuration
Supply Current	I <sub>DD</sub>		100	250	μA	No Load

DC Characteristics for LCD

Limits						
Parameter	Symbol	MIN	TYP	MAX	Units	Conditions
Row Drive Output Impedance	R <sub>ROW</sub>		4		KΩ	R <sub>0/8</sub> - R <sub>7/15</sub> 8-Row Multiplexed LCD Drive Configuration, 3.0 ≤ V <sub>LCD</sub> ≤ V <sub>DD</sub> ; 16-Row Multiplexed LCD Drive Configuration, V <sub>SS</sub> = V <sub>DD</sub> - V <sub>LCD</sub>
Column	R <sub>COLUMN</sub>		12		KΩ	C <sub>0</sub> -C <sub>39</sub> 8-Row Multiplexed LCD Drive Configuration, 3.0 ≤ V <sub>LCD</sub> ≤ V <sub>DD</sub> ; 16-Row Multiplexed LCD Drive Configuration V <sub>SS</sub> = V <sub>DD</sub> - V <sub>LCD</sub>

AC Characteristics

T<sub>a</sub> = -10°C to +70°C, V<sub>DD</sub> = +5.0V ± 10%

Limits						
Parameter	Symbol	MIN	TYP	MAX	Units	Conditions
Clock Frequency	f <sub>β</sub>	100	400	1000	kHz	
Clock Pulse Width High	t <sub>βWH</sub>	400			ns	External Clock
Clock Pulse Width Low	t <sub>βWL</sub>	400			μs	External Clock
SCK Cycle	t <sub>CYK</sub>	1			μs	
SCK Pulse Width High	t <sub>KWH</sub>	400			ns	
SCK Pulse Width Low	t <sub>KWL</sub>	400			ns	
SCK Hold Time	t <sub>KH</sub>	0			ns	After BUSY↓
SI Setup Time	t <sub>IS</sub>	300			ns	To SCK↑
SI Hold Time	t <sub>IH</sub>	250			ns	After SCK↑
BUSY Delay Time	t <sub>BDc</sub>			2	μs	After CS↓
	t <sub>BDd</sub>			2	μs	After C/D↑
	t <sub>BDk</sub>			3	μs	After 8th SCK↑
	t <sub>BDp</sub>			2	μs	After C/D↓
	t <sub>BDk</sub>		250		ns	After SCK↓
C/D Setup Time	t <sub>DS</sub>	2			μs	To 8th SCK↑
C/D Hold Time	t <sub>DH</sub>	2			μs	After 8th SCK↓
CS Hold Time	t <sub>CH</sub>	2			μs	After 8th SCK↑
CS Pulse Width High	t <sub>CWH</sub>	1			μs	
CS Pulse Width Low	t <sub>CWL</sub>	1			μs	

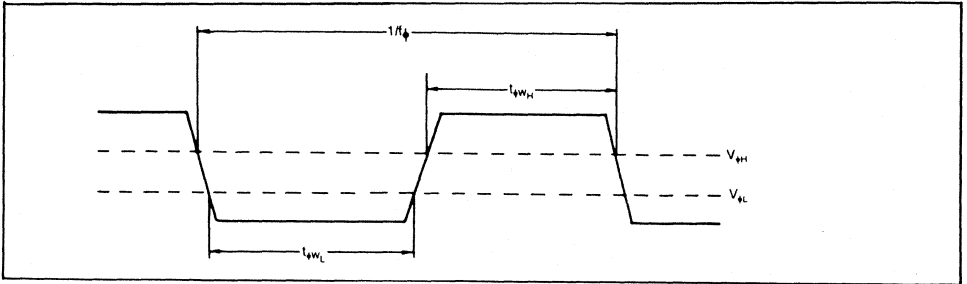
C<sub>LOAD</sub> = 50 pF

Capacitance

$T_a = 25^\circ\text{C}$

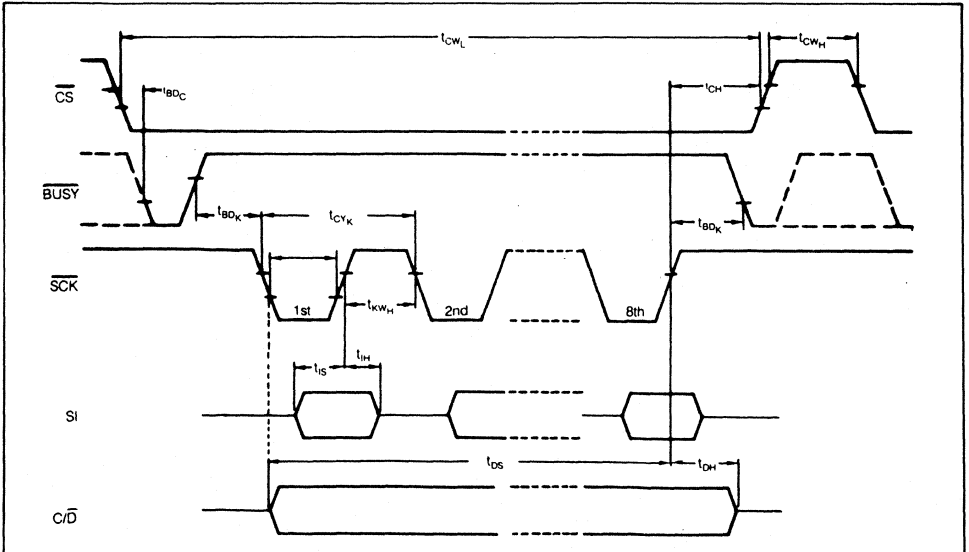
Parameter	Symbol	Limits			Units	Conditions
		MIN	TYP	MAX		
Input Capacitance	$C_i$		10		pF	SI, SCK, C/D, CS, RESET
Output Capacitance	$C_o$		25		pF	SO/BUSY, $R_{W8-R7/15}$ , $C_0-C_{39}$
Input/Output Capacitance	$C_{IO}$		20		pF	SYNC
Load Capacitance	$C_{LOAD}$		100		pF	SYNC
Clock Capacitance	$C_{CLOCK}$		10		pF	Clock Input

Clock Waveform



5

Serial Interface Timing Waveforms





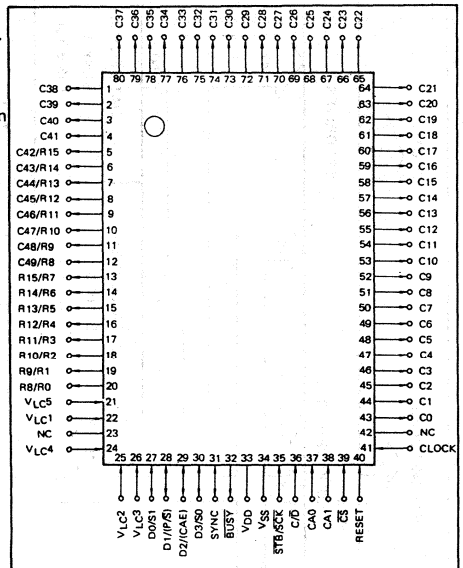


The μPD7228 is a programmable LCD controller/driver containing all the circuitry necessary for interfacing a microprocessor to LCD of 8-line multiplexing or 16-line multiplexing dot matrix configuration. The μPD7228 is able to drive the dot matrix LCD of up to 50 columns by 8-line multiplexing and up to 42 columns by 16-line multiplexing in the single chip configuration. When more than two chips are used in combination, large pattern LCD drive can easily be made. The μPD7228 contains a 5 x 7 dot matrix character generator conforming to ASCII/JIS so that alphanumerics and "kana" can easily be displayed. As an interface with the microprocessor, serial or 4 bit parallel interface can be selected and used.

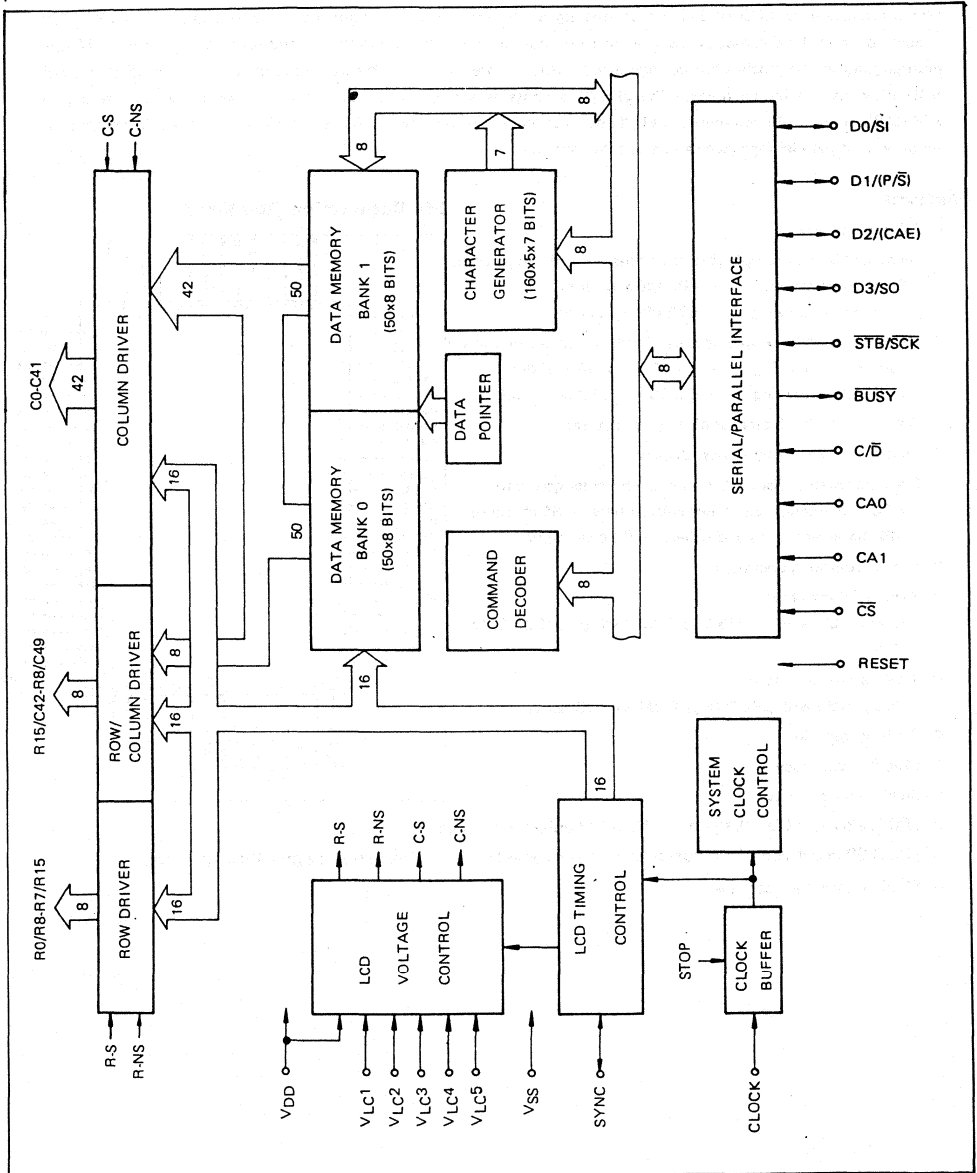
## Features

- LCD direct drive.
- 8-line or 16-line multiplexing drive possible with single chip.
  - 8-line multiplexing — 400 (50 x 8) dots
  - 16-line multiplexing — 672 (42 x 16) dots
- 8-line or 16-line multiplexing drive with n chip configuration
  - 8-line multiplexing — n x 400 (n x 50 x 8) dots
  - 16-line multiplexing — n x 800 (n x 50 x 16) dots
- RAM — 2 x 50 x 8 bits for display data storage
- Programmable dot (graphics) display
- 5 x 7 dot matrix display by on-chip character generator
  - ASCII characters (alphanumerics, others) — 64 characters
  - JIS characters (kana and others) — 96 characters
- Cursor operating command
- 8-bit serial interface
  - Compatible with μPD7500, μCOM-43N, μCOM-87/87LC Series
- 4-bit parallel interface
  - Compatible with μPD7500, μCOM-84/84C Series
- Standby function
- CMOS Technology
- Single power supply
- μPD7228A for LCD voltage up to 12.5V for enhanced contrast
- μPD7229 is available for a customized character set eg to get the European non ASCII characters.
- 80 pin plastic flat package

## Pin Connection (Top View)



μPD7228 Block Diagram



## 1. PIN FUNCTIONS

### 1.1 D0-D3 (DATA BUS) . . . 3 STATE INPUT AND OUTPUT

The data bus serves as an input/output pin for 4 bit parallel data in the parallel interface mode.

Data on D0-D3 lines is read at the rising edge of the  $\overline{STB}$  signal, and the 4 bit data read in the 1st  $\overline{STB}$  is loaded in the high-order 4 bits of the serial/parallel register, and the 2nd data is loaded in the low-order 4 bits.

The contents of serial/parallel register are output to D0-D3 pins synchronizing with the falling edge of the  $\overline{STB}$  signal. Just as in the case of read operation, the high-order 4 bits correspond to the 1st  $\overline{STB}$ , and the low-order 4 bits to the 2nd  $\overline{STB}$ .

In the serial interface mode, D0 serves as a serial data input pin (SI), and D3 as a serial data output pin (SO).

Also, D1 operates as a parallel/serial interface mode selection pin ( $P/\overline{S}$ ), and D2 as a chip address enable pin (CAE).

### 1.2 SI (SERIAL DATA IN) . . . INPUT COMMON TO D0

In the serial interface mode, SI operates as an input pin of serial data. Data on SI line is loaded in the serial/parallel register at the rising edge of  $\overline{SCK}$ .

The first data becomes MSB. To preclude the possibility of error operations due to noise, the Schmitt trigger input having hysteresis is employed.

### 1.3 SO (SERIAL DATA OUT) . . . OUTPUT COMMON TO D3

In the serial interface mode, SO operates as an output pin of serial data. The contents of serial/parallel register are output to SO pin sequentially as from MSB in synchronization with the falling edge of  $\overline{SCK}$ .

### 1.4 $P/\overline{S}$ (PARALLEL/SERIAL SELECT) . . . INPUT COMMON TO D1

If this input is high at the falling edge of the RESET signal (at the time of reset release), the parallel interface mode is set, and if it is low at the falling edge of the reset signal, the serial interface mode is set. The Schmitt trigger input design having hysteresis is employed to prevent error operations from being made due to noise.

### 1.5 CAE (CHIP ADDRESS ENABLE) . . . INPUT COMMON TO D2

CAE input is valid when  $P/\overline{S}$  input is low (serial interface mode) at the falling edge of the RESET signal, and if CAE input is high at the falling edge of RESET signal, the serial interface mode with the chip address function is set. Also, it is without the chip address function when CAE input is low. The Schmitt trigger input design with hysteresis is employed to prevent error operations from being caused by noise.

### 1.6 CA0, 1 (CHIP ADDRESS) . . . INPUT

These are input pins for addressing for selection of each individual μPD7228 in multi-chip configuration. CA0, 1 inputs are not related to CAE input in the parallel interface mode, and these inputs are compared with chip address data sent from the CPU.

In the serial interface mode, however, these are compared with the chip address data when the chip address selection function is enabled by CAE input.

Table 1-1 Specification of CA1 and CA0

MODE	SPECIFICATION
With Chip Address Function <ul style="list-style-type: none"> <li>* In Parallel</li> <li>* In Serial, CAE = 1</li> </ul>	CA1, CA0 = 00, 01, 10, or 11 (In Single-chip Configuration, CA1, CA0 = 00 only)
Without Chip Address Function <ul style="list-style-type: none"> <li>* In Serial, CAE = 0</li> </ul>	CA1, CA0 = 00 (always)

Note 1 : In the case of multi-chip configuration in the serial interface mode, if a proper decoded  $\overline{CS}$  signal is supplied to each chip, the chip address information is not needed.

In this case, CAE of every chip should be set to 0, and every CA1, 0 should be set to 00.

The CA0 and CA1 are composed of Schmitt trigger inputs to prevent error operations by noise.

### 1.7 $\overline{CS}$ (CHIP SELECT) . . . INPUT

This is a low-active chip select input pin.

When the chip address selection function is not in use,  $\overline{STB}/\overline{SCK}$  input and  $C/\overline{D}$  input are enabled if a low level is input into  $\overline{CS}$  input, thereby making the input of command and input/output of data possible.

When using the chip address select function, if  $\overline{CS}$  is brought low and that chip address data matches CA0, 1 inputs, then  $\overline{STB}/\overline{SCK}$  and  $C/\overline{D}$  input are to be enabled.

When  $\overline{CS}$  input is made high, D3–D0 and  $\overline{BUSY}$  pins are unconditionally placed into a high impedance state. The Schmitt trigger input design having hysteresis is employed to prevent error operations from being caused by noise.

### 1.8 $\overline{STB}/\overline{SCK}$ (STROBE/SERIAL CLOCK) . . . INPUT

In the parallel interface mode, this is the strobe signal input pin ( $\overline{STB}$ ) for 4 bit parallel data input and output. In the serial interface mode, this is the serial clock input pin ( $\overline{SCK}$ ) for serial data input and output.

### 1.9 $C/\overline{D}$ (COMMAND/DATA) . . . INPUT

This is a pin to specify whether parallel or serial input is a command or data. When a command is to be input,  $C/\overline{D}$  input has to be brought high, but when data is to be input,  $C/\overline{D}$  has to be made low.

In the case of command or data input, the contents of  $C/\overline{D}$  input are latched at the rising edge of the 2nd  $\overline{STB}$  in the parallel interface mode, but these are latched at the rising edge of the 8th  $\overline{SCK}$  in the serial interface mode.

In the parallel input, however, change of  $C/\overline{D}$  input should be performed prior to the falling edge of 1st  $\overline{STB}$ .

In addition, in the data output,  $C/\overline{D}$  input should be held low, whichever it is serial or parallel.

The Schmitt trigger input design with hysteresis is employed to prevent error operations from being operated by noise.

### 1.10 $\overline{BUSY}$ (BUSY) . . . 3-STATE OUTPUT

This is a pin for outputting the  $\overline{BUSY}$  signal warning the CPU a busy state during internal process of the  $\mu$ PD7228.

When this signal is low, the CPU can not make read/write operation to the  $\mu$ PD7228 because this chip is in a busy state.

In the parallel interface mode, the  $\overline{BUSY}$  signal is forced low at the rising edge of the 2nd  $\overline{STB}$ .

In the serial interface mode, it is forced low at the rising edge of the 8th  $\overline{SCK}$ .

In case of deselected chip ( $\overline{CS}$  = high or chip address mismatch), the  $\overline{BUSY}$  output is placed in a high impedance state.

### 1.11 SYNC (SYNCHRONOUS) . . . 3-STATE INPUT/OUTPUT

This is an input/output pin for synchronous signal to match the phase of LCD-drive AC signal (Row/Column signal) of all μPD7228 with the frame period in case of multi-chip configuration using a Row drive signal as a common signal.

With one of chips as a master, the SYNC pin of the master chip is placed in the output mode, and the remaining chips are all made slave chips. In this case, the SYNC pins of these chips are placed in the input mode.

Designation of either input or output is made by the SMM command.

The master chip designated for the output mode outputs a SYNC pulse in the last cycle of each frame. The slave chip reads the SYNC pulse from the SYNC input for synchronization with the master chip.

The output timing for the SYNC pulse in case of the 8-multiplexing or 16-multiplexing is shown in Fig. 1-1 and Fig. 1-2.

In case of single chip configuration, the SYNC pin may be set in either the input mode or output mode. However, if the input mode is set, the SYNC pin must be connected to  $V_{SS}$ . On the other hand, if the output mode is set, the SYNC pin must be open.

Fig. 1-1 SYNC SIGNAL BASED ON 8-LINE MULTIPLEXING

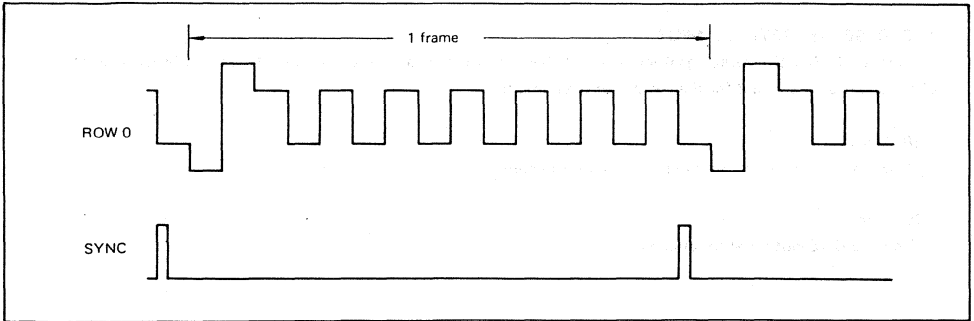
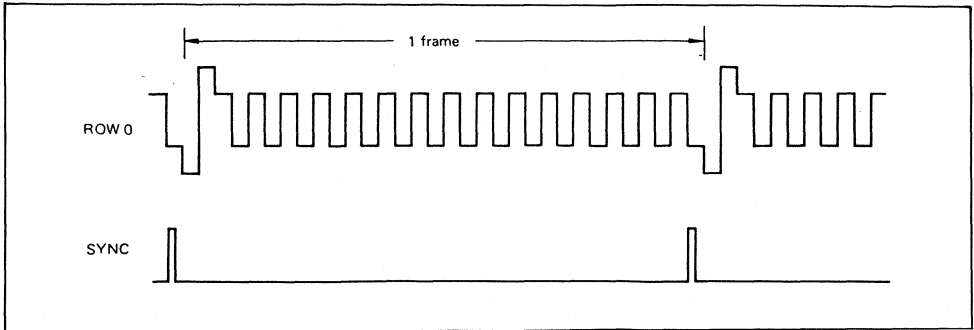


Fig. 1-2 SYNC SIGNAL BASED ON 16-LINE MULTIPLEXING



**1.12 C0–C41 (COLUMN) . . . OUTPUT**

These are output pins for the column drive signal of the LCD.

**1.13 R8/C49 – R15/C42 (ROW/COLUMN) . . . OUTPUT**

These are output pins for Row drive signals R8–R15 of the LCD or for column drive signals C49–C42. The output pins operate as Row drive signals or column drive signals depending upon the SMM command.

**1.14 R0/R8 – R7/R15 (ROW) . . . OUTPUT**

These are output pins for Row drive signals R0–R7 or R8–R15 of the LCD. The output pins operate as an either drive signal, depending upon the SMM command.

**1.15 VLC1 – VLC5 (LCD DRIVE VOLTAGE SUPPLY) . . . INPUT**

This is a reference voltage input pin for determining the voltage level of the Row/column drive signal of the LCD.

**1.16 CLOCK (CLOCK) . . . INPUT**

This is an external clock input pin.

**1.17 RESET (RESET) . . . INPUT**

This is a high-active reset signal input pin. It has priority over all the operations. It can also be used for release of the stand-by mode and for the low power data retention.

**1.18 VDD**

This is a positive power supply pin to VSS for circuits.

**1.19 VSS**

This is a GND potential for circuits.

## 2. FUNCTIONAL DESCRIPTION

### 2.1 SERIAL/PARALLEL INTERFACE

This circuit is provided with the functions of serial and parallel interfaces. It operates as an either function depending upon whether P/S input is high (parallel interface) or low (serial interface) at the falling edge of the RESET signal.

The write operation of command/data from the CPU, and the output of data to the CPU, are all done via this interface.

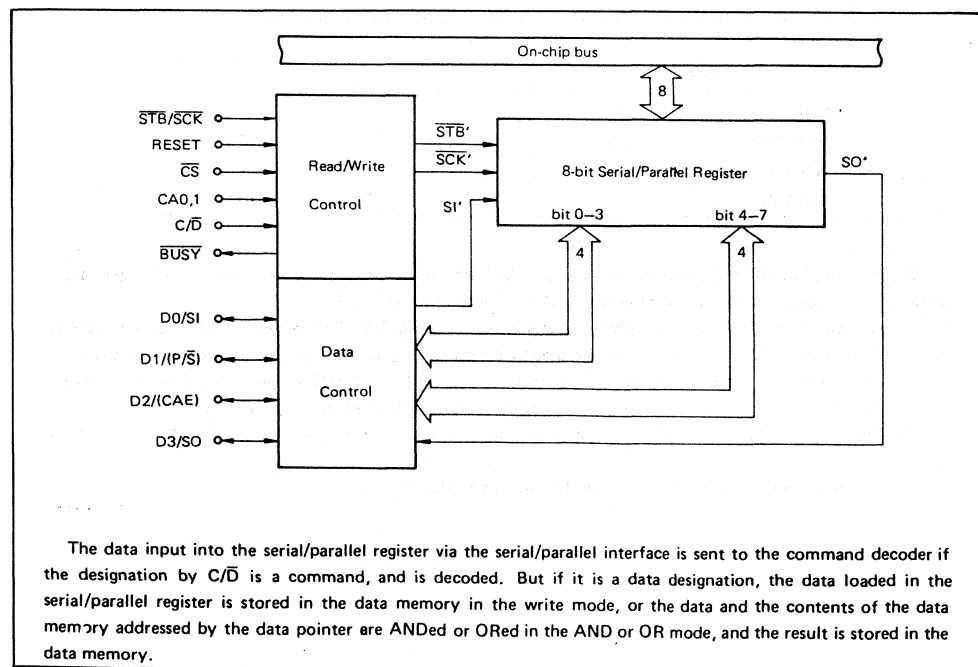
The operation of serial/parallel interface depends on a data processing mode. The data processing mode is initialized to the write mode by RESET input, to make command input possible. Thereafter, it is set to the data processing mode for the write group (Write, AND, OR, Character mode) or that for the read mode by the command setting a data processing mode.

When the write group is set as the data processing mode, the serial/parallel interface becomes the data input mode and takes data from the SI pin (serial data) or the D3–D0 pins (4-bit parallel data) in synchronization with the rising edge of SCK or STB.

Also, when the read mode is set, it becomes the data output mode and then outputs data from either the SO pin (serial data) or the D3–D0 pins (parallel data) in synchronization with the falling edge of SCK or STB.

The serial/parallel register operates as a buffer either between the 8-bit parallel data of data memory and the 8-bit serial data to be transferred via a serial input/output line (SI, SO) or between the 8-bit parallel data of data memory and 2x4-bit parallel data to be transferred via parallel input/output lines (D3–D0).

Fig. 2-1 SERIAL/PARALLEL INTERFACE



The data input into the serial/parallel register via the serial/parallel interface is sent to the command decoder if the designation by C/D is a command, and is decoded. But if it is a data designation, the data loaded in the serial/parallel register is stored in the data memory in the write mode, or the data and the contents of the data memory addressed by the data pointer are ANDed or ORed in the AND or OR mode, and the result is stored in the data memory.



In the character mode, if the designation by  $C/\bar{D}$  is a data, the data loaded in the serial/parallel register is regarded as either ASCII or JIS code, and is sent to the character generator for decoding into the character display pattern of 5x7 bit configuration, after that it is stored in 5 consecutive addresses of the data memory.

Only when the μPD7228 is set in the read mode, it performs data output operation. But if it is set in read mode, the μPD7228 always reads 8-bit data from the data memory in preparation for the next read cycle and keeps data in the serial/parallel register.

In the serial interface mode, the serial/parallel register data is output from SO pin sequentially as from MSB at the falling edge of SCK.

In the parallel interface mode, high-order 4 bits of the serial/parallel register are output at the falling edge of the 1st  $\overline{STB}$ , and the lower-order 4 bits of the serial/parallel register at the falling edge of the 2nd  $\overline{STB}$ . These data are output from D3–D0 pins.

In both serial and parallel interface mode, after 8-bit data is output, next 8-bit data is automatically read out from the data memory and is set in the serial/parallel register.

## 2.2 COMMAND DECODER

If the 8-bit data input via serial/parallel interface is the command designation ( $C/\bar{D}=1$ ), the data is taken in as a command and decoded to generate an internal control signal.

## 2.3 CHARACTER GENERATOR

The character generator is enabled when the mode setting commands (SCML, SCMR) are executed to interpret 8-bit data to be written via serial/parallel interface as ASCII code (alphanumerics and symbols) or JIS code (kana and symbols), and generate the corresponding 5x7 dot matrix pattern for transfer to five continuous addresses (7 bits x 5 times) of data memory.

The character generator contains 160 types of pattern data indicated below. (μPD7228/A)

ASCII		JIS	
Capital letters	26	Kana	55
Small letters	26	Symbols	9
Numerics	10		
Symbols	34		

Character (ASCII/JIS) codes and the corresponding 5x7 dot display patterns are shown in Fig. 2-3. 96 codes of 20H–7FH correspond to ASCII, and 64 codes of A0H–DFH correspond to JIS.

The LCD configuration using a character generator is shown in Fig. 2-2. The character is 5x7 dot type, and the most significant bit (bit 7) of data memory is not used by the character generator. So, the dot of LCD corresponding to the most significant bit (R7 in case of 8-line multiplexing, or R7 and R15 in case of 16-line multiplexing) can be used as a display pattern of the cursor or indicator independently of character generator.

Operation of the most significant bit is done by the cursor operating command (WRCURS and CLCURS) and so on.

The character generator of μPD7229 is to be defined by the user.

Fig. 2-2 LCD STRUCTURE IN CASE OF CHARACTER GENERATOR USE

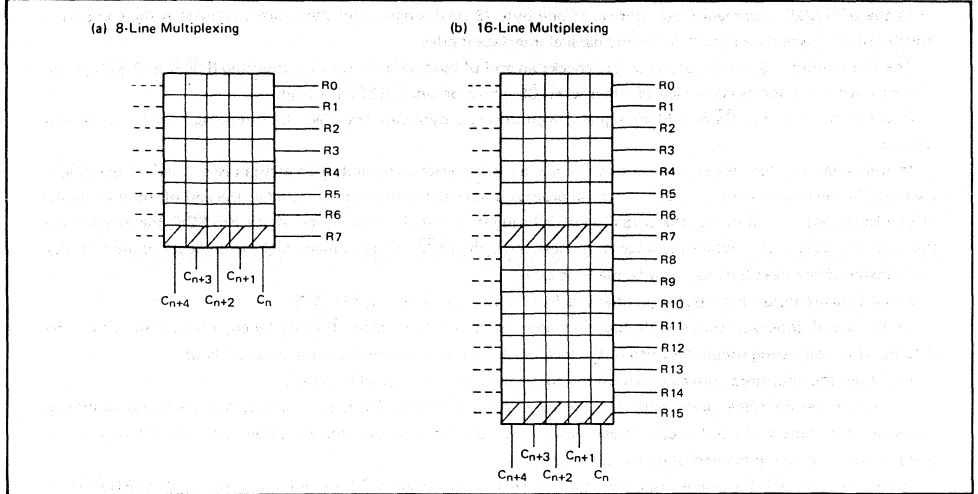
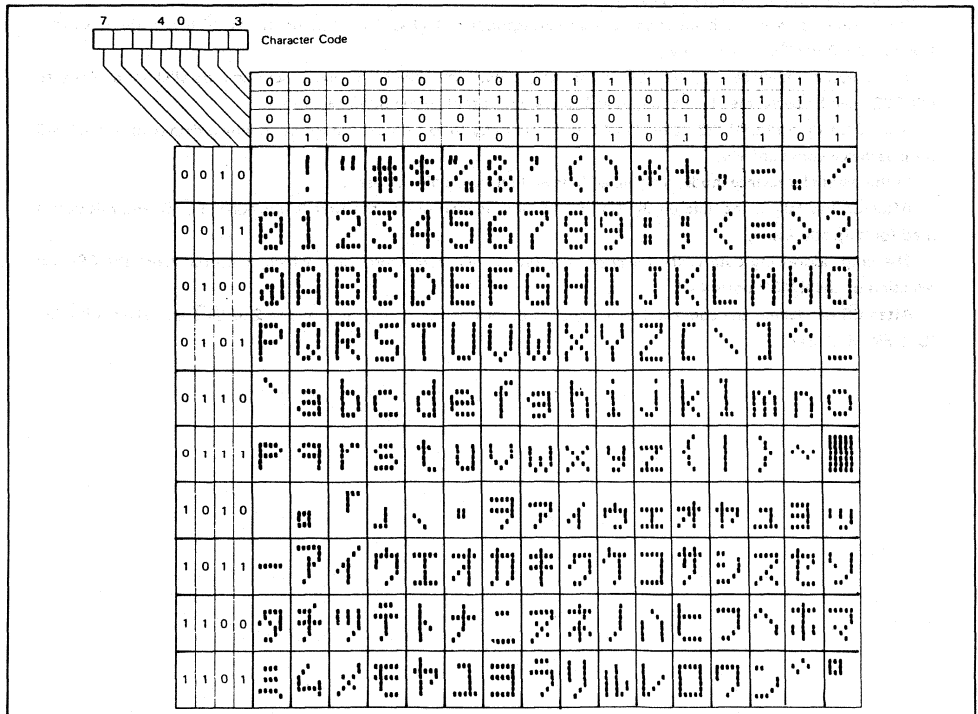


Fig. 2-3 CHARACTER CODES AND DISPLAY PATTERNS (μPD7228/A)



### 3. DATA INPUT AND OUTPUT OPERATIONS

In the  $\mu$ PD7228, command/data consists of one byte (8 bits) length, and the process is made at the end of data transfer of one byte data in both serial and parallel interface modes.

The byte counter (octal/binary counter) checks an end of byte-data transfer by counting 8  $\overline{\text{SCK}}$  or 2  $\overline{\text{STB}}$  pulses. This counter is unconditionally cleared either when  $\overline{\text{CS}}$  = high or when RESET = high.

Accordingly, if either  $\overline{\text{CS}}$  or RESET signal is high during a byte-data transfer, the byte-data transfer will not be achieved.

In the serial interface mode, the process is made an 8-bit serial data, and when eight pulses of serial clocks ( $\overline{\text{SCK}}$ ) are input in the chip selection state, the internal process is initiated as this is considered as the end of input or output of one byte data. In so doing, the  $\overline{\text{BUSY}}$  signal is brought low at the rising edge of the 8th  $\overline{\text{SCK}}$ , thereby warning the CPU the busy state. When the internal process ends, the  $\overline{\text{BUSY}}$  signal is made high, to indicate to the CPU that the transfer of the next byte can now be made possible.

Serial data are input or output sequentially as from MSB. (See Fig. 3-1, Fig. 3-2)

In the serial interface mode with the chip addressing function, after  $\overline{\text{CS}}$  has fallen, first of all, 8-bit serial data for chip addressing should be written (the lower-order 2 bits are latched as chip address data).

And then the addressed chip can perform the command input or data I/O operation.

In the parallel interface mode, data is processed as parallel data of 4 bits  $\times$  2 for data bus is 4-bit configuration, and when two pulses of parallel data strobe signal ( $\overline{\text{STB}}$ ) are input in the chip selection state, the internal process is made with it as being the end of data input or output.

In so doing, the  $\overline{\text{BUSY}}$  signal is made low at the rising edge of the 2nd  $\overline{\text{STB}}$ , to indicate to the CPU the  $\overline{\text{BUSY}}$  state.

When the internal process ends, the  $\overline{\text{BUSY}}$  signal is forced high to indicate to the CPU that the transfer of the next byte can now be made possible.

The parallel data of both input and output corresponds to high-order 4 bits of the 1st  $\overline{\text{STB}}$  and the low-order 4 bits of the 2nd  $\overline{\text{STB}}$ , respectively.

The parallel interface of  $\mu$ PD7228 is compatible with the 8243 I/O expander, and parallel data input to  $\mu$ PD7228 can be performed by the way that 4-bit data is output to the 8243 two times.

Also, 8-bit data of  $\mu$ PD7228 is read in 4 bits  $\times$  2 times by following the procedure in which data read of the 8243 is made twice consecutively.

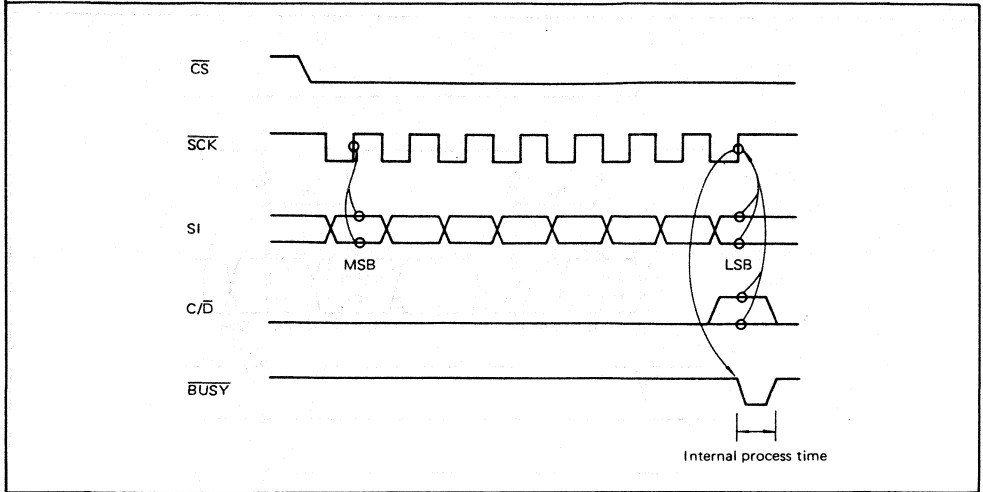
In the parallel interface mode, the chip addressing function is always used.

After  $\overline{\text{CS}}$  has fallen, the data on the D1 and D0 lines is read out at the first falling edge of  $\overline{\text{STB}}$ , and this data is used for chip addressing.

The chip addressing data is the lower-order 2 bits of the command code, which is output from the CPU for selection of the 8243's ports 4-7.

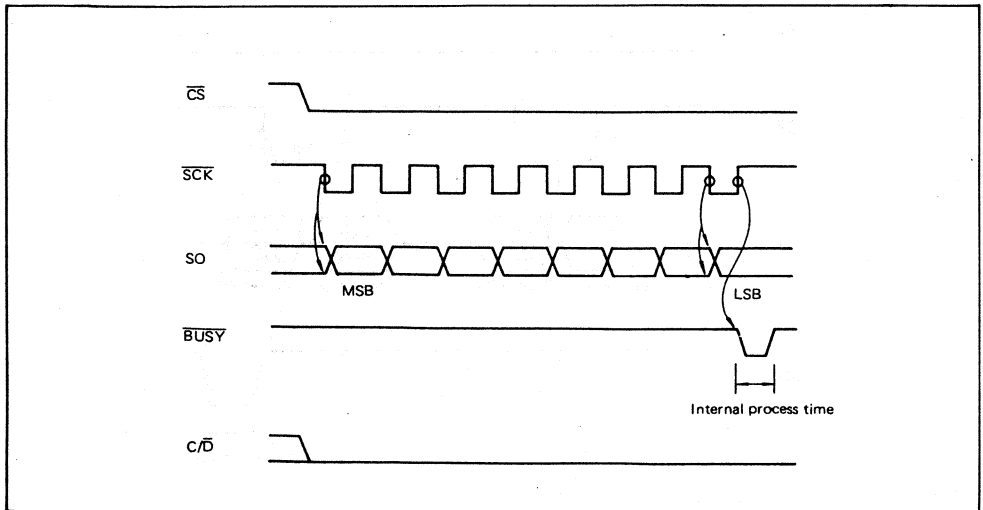
After  $\overline{\text{CS}}$  has fallen, the command code which is output at the falling edge of the 2nd  $\overline{\text{STB}}$  and after is invalid. (See Fig. 3-3, 3-4)

Fig. 3-1 SERIAL INPUT TIMING (WITHOUT CHIP ADDRESS SECTION FUNCTION)

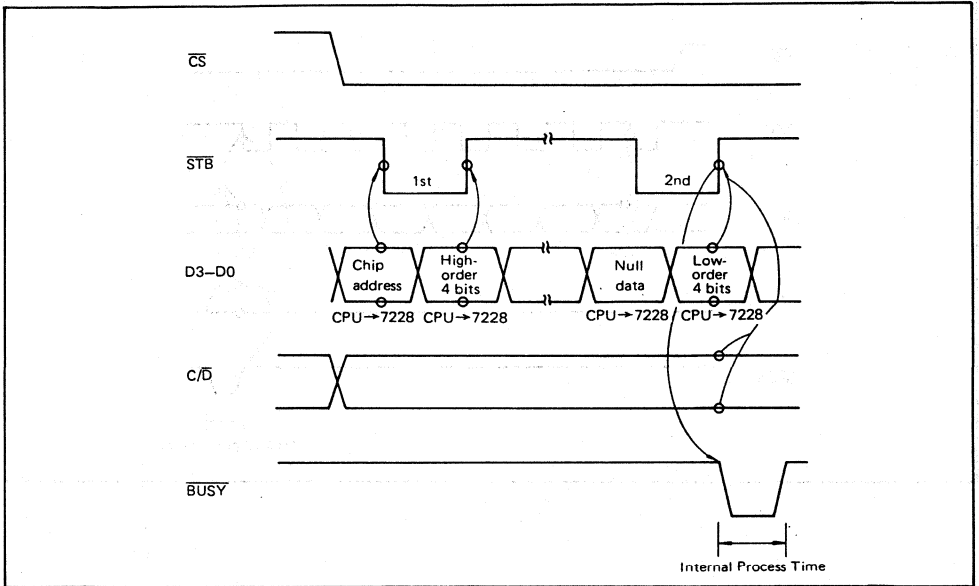


5

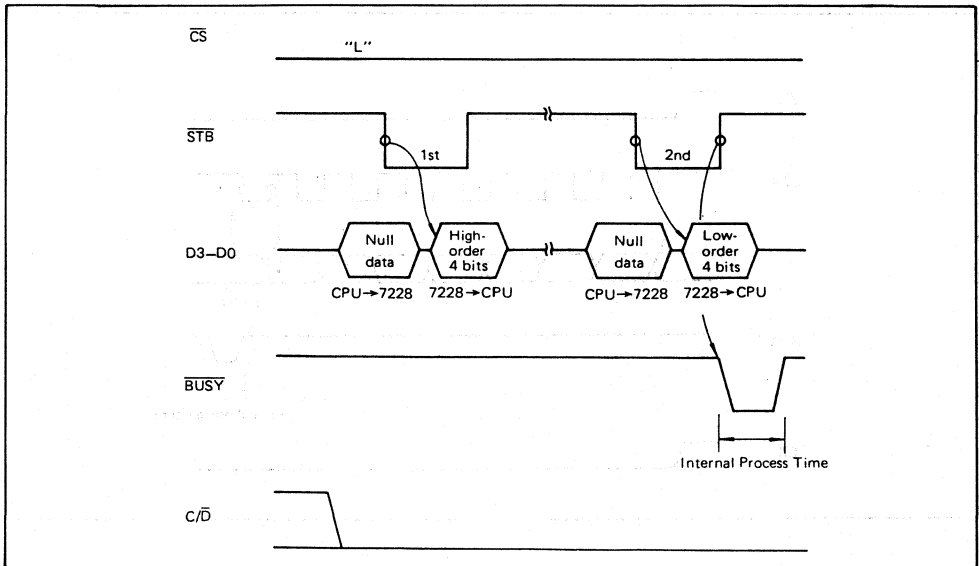
Fig. 3-2 SERIAL OUTPUT TIMING



**Fig. 3-3 PARALLEL INPUT TIMING**



**Fig. 3-4 PARALLEL OUTPUT TIMING**

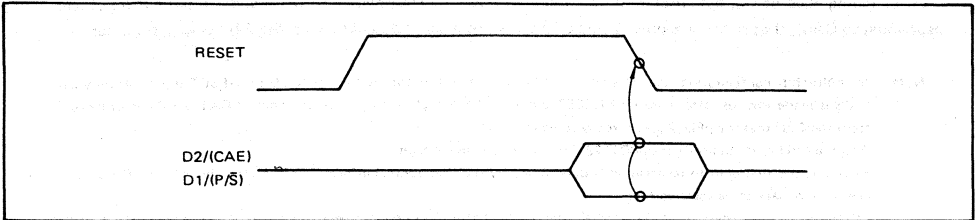


## 4. SELECTION OF INTERFACE BETWEEN μPD7228 AND CPU

Data length of the command/data of μPD7228 are 8 bits, however the μPD7228 is interfaced with the CPU in 8-bit serial or 4-bit parallel x2 configuration. Also, the μPD7228 is provided with the chip address selection function for multi-chip configuration purposes.

Designation of as to whether serial or parallel interface, and that of as to whether the chip address selection function is to be used or not, depends upon the data sent over D2/(CAE), D1/(P $\bar{S}$ ) at the release timing (falling edge) of the RESET signal of the μPD7228.

**Fig. 4-1 INTERFACE DESIGNATION TIMING**



**Table 4-1 INTERFACE DESIGNATION CODE**

D2/(CAE)	D1/(P $\bar{S}$ )	Serial/Parallel Designation	Chip Address Selection Function
0	0	Serial	Not provided
1	0		Provided
0/1	1	Parallel	

### 4.1 FUNCTION OF COMMON PINS

The function of common pins  $\overline{STB}/\overline{SCK}$ , D3/S0, D0/SI used for clock input and data input/output varies as shown in Table 4-2, depending upon whether the serial interface or parallel interface is designated.

**Table 4-2 FUNCTION OF COMMON PINS**

Pin Name	Serial (P $\bar{S}$ =0)	Parallel (P $\bar{S}$ =1)
$\overline{STB}/\overline{SCK}$	$\overline{SCK}$ input	$\overline{STB}$ input
D3/S0	S0 output	D3-D0 input and output (4-bit parallel data bus)
D2/(CAE)	-	
D1/(P $\bar{S}$ )	-	
D0/SI	SI input	

### 4.2 CHIP ADDRESS SLECTION FUNCTION

The chip address selection function compares the chip address assigned (by CA0, CA1 input) to each μPD7228 chip with chip address data (2 bits) sent from the CPU side in the serial or parallel data format and selects only the matched chips (this makes input and output of command/data possible). This function eliminates the need for sending a plurality of chip select signals ( $\overline{CS}$ ) from the CPU side when employing multi-chip configuration.

Parallel interface mode is unconditionally with chip address selection function, but serial interface mode is with chip address selection function only D2/(CAE)=1 at the release timing of the RESET signal of the μPD7228G.

(1) **Parallel Interface** (See Fig. 3-5, 3-6)

After falling edge of  $\overline{CS}$ , data to be read in D1 (corresponding to CA1) and D0 (corresponding to CA0) at the first falling edge of  $\overline{STB}$  becomes 2-bit chip address data.

In actuality, parallel interface is similar to that for the 8243 I/O Expander, connection should be made with the  $\mu$ PD7228 by using the 8243 interface function such as in the  $\mu$ PD7500 or  $\mu$ PD80C48, and the output instruction is executed for ports 4–7 of the 8243. In so doing, it is possible to obtain chip address data (0–3) of the  $\mu$ PD7228 at the falling edge of  $\overline{STB}$  on D1, D0.

(2) **Serial Interface** (See Fig. 3-3, 3-4)

After falling edge of  $\overline{CS}$ , data read in SI by 7th pulse of  $\overline{SCK}$  (corresponding to CA1) and 8th pulse there of (corresponding to CA0), that is, the low-order 2 bits of the first 8-bit serial data, becomes the 2-bit chip address data.

**Note:** In multi-chip configuration, the internal chip-address-comparison data is cleared to 00 by RESET input. Accordingly, if  $\overline{CS}$  is made low just after release of RESET input,  $\overline{BUSY} = \text{high}$  is output by the chip of CA1, 0 = 00, in order to inform the CPU that the  $\mu$ PD7228s are ready to be accessed.

And the CPU starts to access the  $\mu$ PD7228s by detecting  $\overline{BUSY} = \text{high}$ .

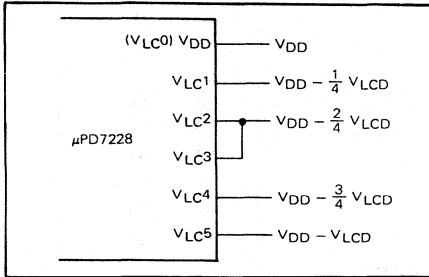
However, the first chip to be accessed is determined by the chip address information (i.e., the chip of CA1, 0 = 00 is not always necessary to be accessed first.).

In the multi-chip system using the parallel interface, just after  $\overline{CS}$  signal has been made low, if a read operation is wanted to perform for the chip that is specified in the read mode, the data pointer load command should be executed before the read operation. (i.e., the read operation must not be done during the first a couple of  $\overline{STB}$  signals just after  $\overline{CS}$  signal has been made low.)

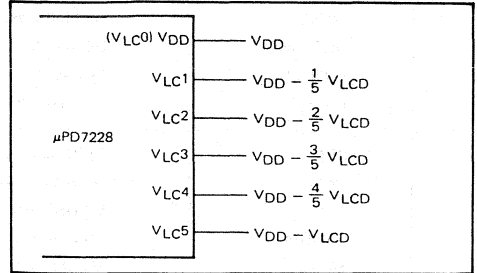
## 5. SUPPLY OF LCD DRIVE REFERENCE VOLTAGE

The value of LCD drive reference voltage to μPD7228 differs depending upon the number of multiplexings — 8-line multiplexing or 16-line multiplexing. These are shown in Fig. 5-1 and Fig. 5-2.

**Fig. 5-1 8-LINE MULTIPLEXING**



**Fig. 5-2 16-LINE MULTIPLEXING**

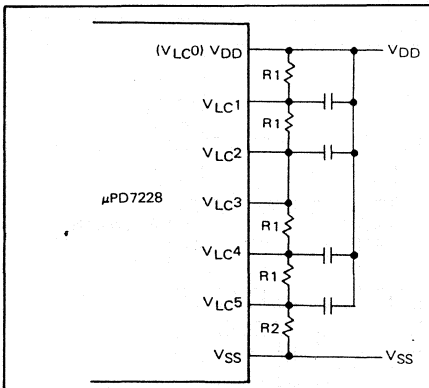


Note: LCD drive voltage ( $V_{LCD}$ ) should be below  $V_{DD}$  ( $V_{LCD} \leq V_{DD}$ ) in both cases of 8-Line multiplexing and 16-line multiplexing.

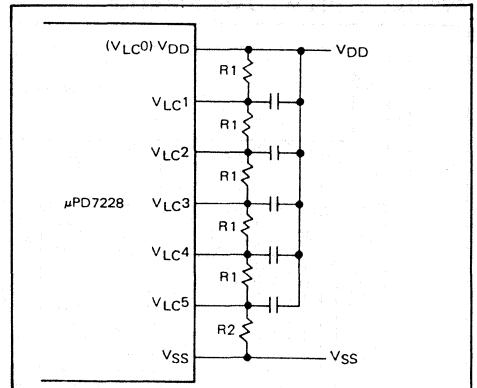
### 5.1 SUPPLY OF LCD DRIVE REFERENCE VOLTAGE BY RESISTOR DIVISION

An example of circuit for supplying the LCD drive reference voltage shown in Fig. 5-1 and Fig. 5-2 with the potential between  $V_{DD} - V_{SS}$  divided by resistors is shown in Fig. 5-3 and Fig. 5-4.

**Fig. 5-3 8-LINE MULTIPLEXING**



**Fig. 5-4 16-LINE MULTIPLEXING**



The values of dividing resistors  $R_1$ ,  $R_2$  in case of 8-line multiplexing and 16-line multiplexing are determined by the equation below.

$$R_1 = \frac{V_{LCD}}{4(V_{DD} - V_{LCD})} \times R_2 \quad : \text{ 8-line multiplexing}$$

$$R_1 = \frac{V_{LCD}}{5(V_{DD} - V_{LCD})} \times R_2 \quad : \text{ 16-line multiplexing}$$

The μPD7228A can drive LCD voltages up to 12.5V. In this case  $V_{SS}$  becomes -7.5V.



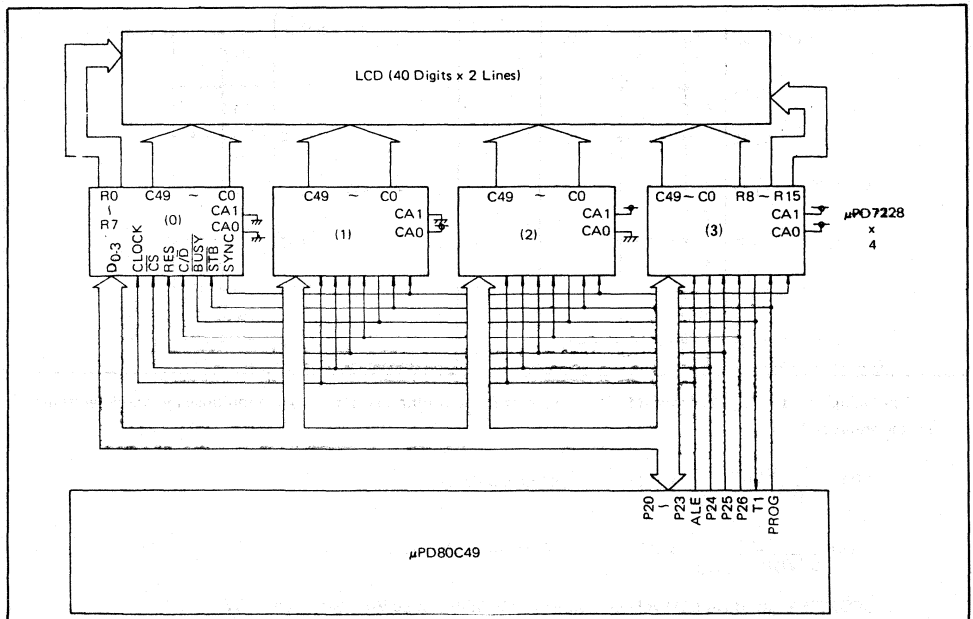
**COMMANDS OF μPD7228**

The μPD7228 is provided with sixteen types of commands, each command consisting of one byte (8 bits).

**LIST OF COMMANDS**

Mnemonic	Operation	Hexadecimal Code
SFF	Set Frame Frequency	10 - 14
SMM	Set Multiplexing Mode	18 - 1F
DISP OFF	Display Off	08
DISP ON	Display On	09
LDPI	Load Data Pointer with Immediate	80 - B1 C0 - F1
SRM	Set Read Mode	60 - 63
SWM	Set Write Mode	64 - 67
SORM	Set OR Mode	68 - 6B
SANDM	Set AND Mode	6C - 6F
SCML	Set Character Mode with Left entry	71
SCMR	Set Character Mode with Right entry	72
BRESET	Bit Reset	20 - 3F
BSET	Bit Set	40 - 5F
CLCURS	Clear Cursor	7C
WRCURS	Write Cursor	7D
STOP	Set Stop Mode	01

**TYPICAL SYSTEM CONFIGURATION (REFERENCE ONLY)**



**ABSOLUTE MAXIMUM RATINGS (Ta = 25 °C)**

Supply Voltage	V <sub>DD</sub>	-0.3 to +7.0	V
Input Voltage	V <sub>I</sub>	-0.3 to V <sub>DD</sub> +0.3	V
Output Voltage	V <sub>O</sub>	-0.3 to V <sub>DD</sub> +0.3	V
Operating Temperature	T <sub>opt</sub>	-10 to +70	°C
Storage Temperature	T <sub>stg</sub>	-65 to +150	°C

**DC CHARACTERISTICS for μPD7228/μPD7229**

(Ta = -10 to +70°C, V<sub>DD</sub> = +5 V ± 10%)

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
Input Voltage High	V <sub>IH1</sub>	0.7 V <sub>DD</sub>		V <sub>DD</sub>	V	Except $\overline{\text{SCK}}$
	V <sub>IH2</sub>	0.8 V <sub>DD</sub>		V <sub>DD</sub>	V	$\overline{\text{SCK}}$
Input Voltage Low	V <sub>IL</sub>	0		0.3 V <sub>DD</sub>	V	
Input Leakage Current High	I <sub>LIH</sub>			10	μA	V <sub>I</sub> =V <sub>DD</sub>
Input Leakage Current Low	I <sub>LIL</sub>			-10	μA	V <sub>I</sub> =0 V
Output Voltage High	V <sub>OH1</sub>	V <sub>DD</sub> -0.5			V	BUSY, D0-D3, I <sub>OH</sub> =-400 μA
	V <sub>OH2</sub>	V <sub>DD</sub> -0.5			V	SYNC, I <sub>OH</sub> =-100 μA
Output Voltage Low	V <sub>OL1</sub>			0.45	V	BUSY, D0-D3, I <sub>OL</sub> =1.7 mA
	V <sub>OL2</sub>			0.45	V	SYNC, I <sub>OL</sub> =100 μA
Output Leakage Current High	I <sub>LOH</sub>			10	μA	V <sub>O</sub> =V <sub>DD</sub>
Output Leakage Current Low	I <sub>LOL</sub>			-10	μA	V <sub>I</sub> =0 V
LCD Operating Voltage	V <sub>LCD</sub>	3.0		V <sub>DD</sub>	V	
Row Output Impedance	R <sub>ROW</sub>		4	8	kΩ	
Row/Column Output Impedance	R <sub>ROW/COL</sub>		5	10	kΩ	
Column Output Impedance	R <sub>COL</sub>		10	15	kΩ	
Supply Current	I <sub>DD1</sub>		200	400	μA	Operating Mode, f <sub>c</sub> =400 kHz
	I <sub>DD2</sub>			20	μA	STOP Mode, CLK=0 V

**CAPACITANCE (Ta = 25 °C, V<sub>DD</sub> = 0 V)**

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
Input Capacitance	C <sub>IN</sub>			10	pF	f = 1 MHz
Output Capacitance	C <sub>OUT</sub>			25	pF	Unmeasured pins returned to 0 V
I/O Capacitance	C <sub>IO</sub>			15	pF	

**AC Characteristics** (Ta = -10 to +70 °C, VDD = +5 V ± 10 %)

**COMMON OPERATION:**

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
Clock Frequency	f <sub>C</sub>	100		1100	kHz	
Clock Pulse Width High	t <sub>WHC</sub>	350			ns	
Clock Pulse Width Low	t <sub>WLC</sub>	350			ns	
BUSY Delay Time from CS↑	t <sub>DCSB</sub>			2	μs	C <sub>L</sub> =50 pF
CS↑ Delay Time to BUSY Floating	t <sub>DCSBF</sub>			4	μs	C <sub>L</sub> =50 pF
CS High Level Time	t <sub>WHCS</sub>	4			μs	
SYNC Load Capacitance	C <sub>LSY</sub>			100	pF	
Data Setup Time to RESET↓	t <sub>SDR</sub>	0			μs	
Data Hold Time from RESET↓	t <sub>HRD</sub>	4			μs	

**SERIAL INTERFACE OPERATION:**

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
SCK Cycle	t <sub>CYK</sub>	0.9			μs	
SCK Pulse Width High	t <sub>WHK</sub>	400			ns	
SCK Pulse Width Low	t <sub>WLK</sub>	400			ns	
SCK Hold Time from BUSY↑	t <sub>HBK</sub>	0			ns	
SI Setup Time to SCK↑	t <sub>SIK</sub>	100			ns	
SI Hold Time from SCK↑	t <sub>HKI</sub>	250			ns	
SO Delay Time from SCK↑	t <sub>DKO</sub>			320	ns	C <sub>L</sub> =50 pF
BUSY Delay Time from 8th SCK↑	t <sub>KDB</sub>			3	μs	C <sub>L</sub> =50 pF
BUSY Low Level Time	t <sub>WLB</sub>	18		64	1/f <sub>C</sub>	C <sub>L</sub> =50 pF
C/D Setup Time to 1st SCK↑	t <sub>SDK</sub>	0			μs	
C/D Hold Time from 8th SCK↑	t <sub>HKD</sub>	2			μs	
CS Hold Time from 8th SCK↑	t <sub>HKCS</sub>	2			μs	

**PARALLEL INTERFACE OPERATION:**

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
Input Command Setup Time to STB↓	t <sub>A</sub>	100			ns	C <sub>L</sub> =80 pF
Input Command Hold Time from STB↓	t <sub>B</sub>	90			ns	C <sub>L</sub> =20 pF
Input Data Setup Time to STB↑	t <sub>C</sub>	230			ns	C <sub>L</sub> =80 pF
Input Data Hold Time from STB↑	t <sub>D</sub>	50			ns	C <sub>L</sub> =20 pF
Output Data Delay Time	t <sub>ACC</sub>	90		650	ns	C <sub>L</sub> =80 pF
Output Data Hold Time	t <sub>H</sub>	0		150	ns	C <sub>L</sub> =20 pF
STB Pulse Width Low	t <sub>SL</sub>	700			ns	
STB High Level Time	t <sub>SH</sub>	1			μs	
STB Hold Time from BUSY↑	t <sub>HBS</sub>	0			μs	
BUSY Delay Time from 2nd STB↑	t <sub>DSB</sub>			3	μs	
C/D Setup Time to 1st STB↓	t <sub>SDS</sub>	0			μs	
C/D Hold Time from 2nd STB↑	t <sub>HSD</sub>	2			μs	
CS Hold Time from 2nd STB↑	t <sub>HSCS</sub>	2			μs	

### ABSOLUTE MAXIMUM RATINGS (Ta = 25 °C), μPD7228AG

Supply Voltage	V <sub>DD</sub>	-0.3 to +7.0	V
Input Voltage	V <sub>I</sub>	-0.3 to V <sub>DD</sub> +0.3	V
Output Voltage	V <sub>O</sub>	-0.3 to V <sub>DD</sub> +0.3	V
Operating Temperature	T <sub>opt</sub>	-10 to +70	°C
Storage Temperature	T <sub>stg</sub>	-65 to +150	°C
LCD Operating Voltage	V <sub>LCD</sub> *	4.5 to 12.5	V

### DC CHARACTERISTICS for μPD7228A

(Ta = -40 to +85°C, V<sub>DD</sub> = +5V ± 10%)

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
Input Voltage High	V <sub>IH1</sub>	0.7 V <sub>DD</sub>		V <sub>DD</sub>	V	Except SCK
	V <sub>IH2</sub>	0.8 V <sub>DD</sub>		V <sub>DD</sub>	V	SCK
Input Voltage Low	V <sub>IL</sub>	0		0.3 V <sub>DD</sub>	V	
Input Leakage Current High	I <sub>LIH</sub>			10	μA	V <sub>I</sub> =V <sub>DD</sub>
Input Leakage Current Low	I <sub>LIL</sub>			-10	μA	V <sub>I</sub> =0 V
Output Voltage High	V <sub>OH1</sub>	V <sub>DD</sub> -0.5			V	$\overline{\text{BUSY}}$ , D0-D3, I <sub>OH</sub> =-400 μA
	V <sub>OH2</sub>	V <sub>DD</sub> -0.5			V	SYNC, I <sub>OH</sub> =-100 μA
Output Voltage Low	V <sub>OL1</sub>			0.5	V	$\overline{\text{BUSY}}$ , D0-D3, I <sub>OL</sub> =1.7 mA
	V <sub>OL2</sub>			0.5	V	SYNC, I <sub>OL</sub> =100 μA
Output Leakage Current High	I <sub>LOH</sub>			10	μA	V <sub>O</sub> =V <sub>DD</sub>
Output Leakage Current Low	I <sub>LOL</sub>			-10	μA	V <sub>I</sub> =0 V
LCD Operating Voltage*	V <sub>LCD</sub>	4.5		12.5	V	
Row Output Impedance	R <sub>ROW</sub>		6	16	kΩ	
Row/Column Output Impedance	R <sub>ROW/COL</sub>		7.5	20	kΩ	
Column Output Impedance	R <sub>COL</sub>		15	30	kΩ	
Supply Current	I <sub>DD1</sub>		250	600	μA	Operating Mode, f <sub>c</sub> =400 kHz
	I <sub>DD2</sub>			25	μA	STOP Mode, CLK=0 V

\*V<sub>LCD</sub> = V<sub>DD</sub> - V<sub>LCS</sub>. V<sub>LCS</sub> ≤ V<sub>SS</sub>

### Capacitance (Ta = 25 °C, V<sub>DD</sub> = 0 V)

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
Input Capacitance	C <sub>IN</sub>			10	pF	f = 1 MHz Unmeasured pins returned to 0 V
Output Capacitance	C <sub>OUT</sub>			25	pF	
I/O Capacitance	C <sub>IO</sub>			15	pF	

AC CHARACTERISTICS ( $T_a = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD} = +5\text{V} \pm 10\%$ ),  $\mu$ PD7228AG

## COMMON OPERATION:

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
Clock Frequency	$f_C$	100		1100	kHz	
Clock Pulse Width High	$t_{WHC}$	350			ns	
Clock Pulse Width Low	$t_{WLC}$	350			ns	
BUS $\bar{Y}$ Delay Time from CS $\downarrow$	$t_{DCSB}$			3	$\mu\text{s}$	$C_L=50\text{ pF}$
CS $\uparrow$ Delay Time to BUS $\bar{Y}$ Floating	$t_{DCSBF}$			5	$\mu\text{s}$	$C_L=50\text{ pF}$
CS High Level Time	$t_{WHCS}$	4			$\mu\text{s}$	
SYNC Load Capacitance	$C_{LSY}$			100	pF	
Data Setup Time to RESET $\uparrow$	$t_{SDR}$	0			$\mu\text{s}$	
Data Hold Time from RESET $\uparrow$	$t_{HRD}$	5			$\mu\text{s}$	

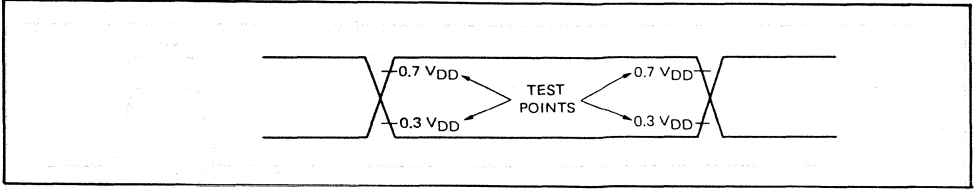
## SERIAL INTERFACE OPERATION:

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
SCK Cycle	$t_{CYK}$	0.9			$\mu\text{s}$	
SCK Pulse Width High	$t_{WHK}$	400			ns	
SCK Pulse Width Low	$t_{WLK}$	400			ns	
SCK Hold Time from BUS $\bar{Y}$ $\uparrow$	$t_{HBK}$	0			ns	
SI Setup Time to SCK $\uparrow$	$t_{SIK}$	120			ns	
SI Hold Time from SCK $\uparrow$	$t_{HKI}$	270			ns	
SO Delay Time from SCK $\uparrow$	$t_{DKO}$			320	ns	$C_L=50\text{ pF}$
BUS $\bar{Y}$ Delay Time from 8th SCK $\uparrow$	$t_{KDB}$			3	$\mu\text{s}$	$C_L=50\text{ pF}$
BUS $\bar{Y}$ Low Level Time	$t_{WLB}$	18		64	$1/f_C$	$C_L=50\text{ pF}$
C/D Setup Time to 1st SCK $\uparrow$	$t_{SDK}$	0			$\mu\text{s}$	
C/D Hold Time from 8th SCK $\uparrow$	$t_{HKD}$	3			$\mu\text{s}$	
CS Hold Time from 8th SCK $\uparrow$	$t_{HKCS}$	5			$\mu\text{s}$	

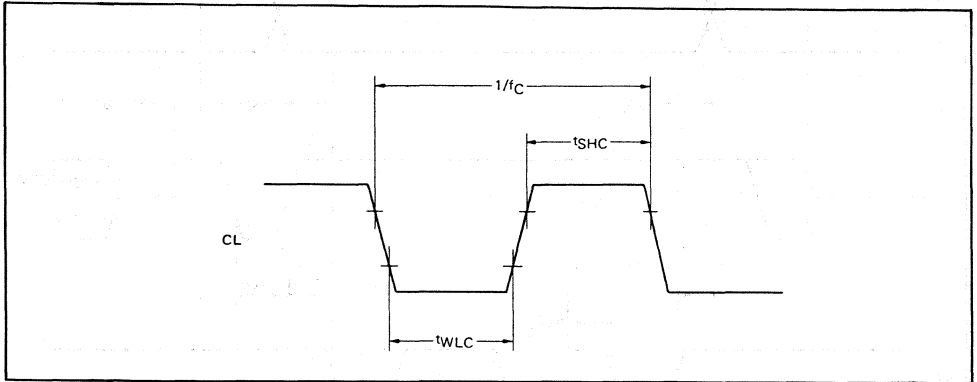
## PARALLEL INTERFACE OPERATION:

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
Input Command Setup Time to STB $\downarrow$	$t_A$	120			ns	$C_L=80\text{ pF}$
Input Command Hold Time from STB $\downarrow$	$t_B$	110			ns	$C_L=20\text{ pF}$
Input Data Setup Time to STB $\downarrow$	$t_C$	250			ns	$C_L=80\text{ pF}$
Input Data Hold Time from STB $\downarrow$	$t_D$	70			ns	$C_L=20\text{ pF}$
Output Data Delay Time	$t_{ACC}$	90		750	ns	$C_L=80\text{ pF}$
Output Data Hold Time	$t_H$	0		150	ns	$C_L=20\text{ pF}$
STB Pulse Width Low	$t_{SL}$	700			ns	
STB High Level Time	$t_{SH}$	1			$\mu\text{s}$	
STB Hold Time from BUS $\bar{Y}$ $\uparrow$	$t_{HBS}$	0			$\mu\text{s}$	
BUS $\bar{Y}$ Delay Time from 2nd STB $\uparrow$	$t_{DSB}$			4	$\mu\text{s}$	
C/D Setup Time to 1st STB $\uparrow$	$t_{SDS}$	0			$\mu\text{s}$	
C/D Hold Time from 2nd STB $\uparrow$	$t_{HSD}$	3			$\mu\text{s}$	
CS Hold Time from 2nd STB $\uparrow$	$t_{HSCS}$	3			$\mu\text{s}$	

## AC TEST POINTS

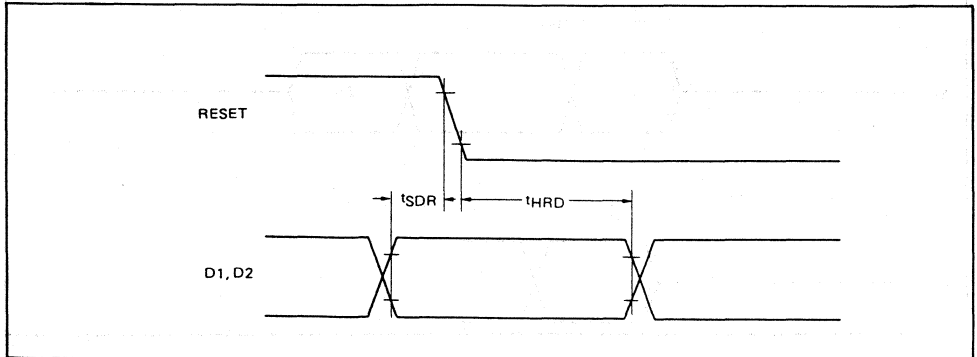


## CLOCK WAVEFORM



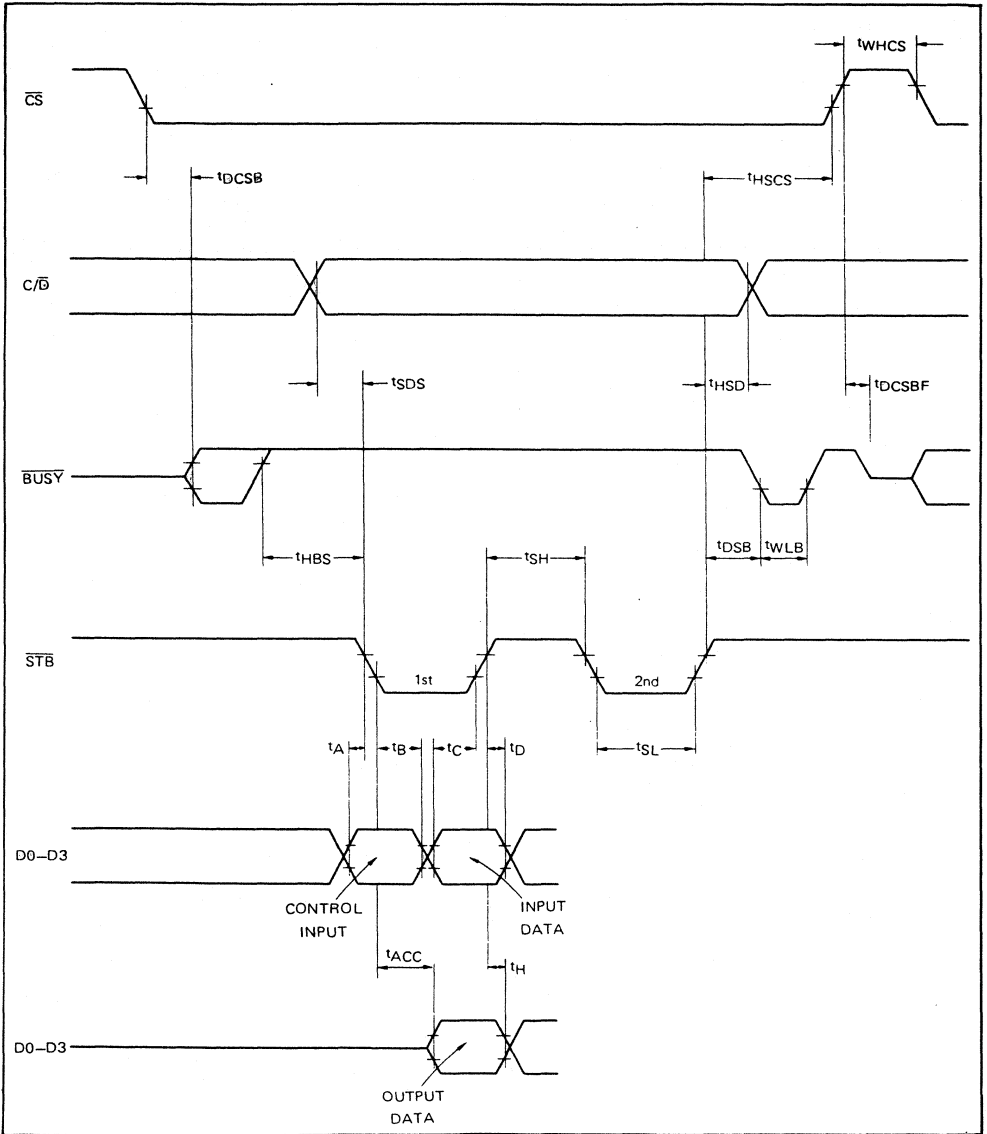
5

## INTERFACE TIMING WAVEFORMS





## Parallel Interface Timing Waveforms





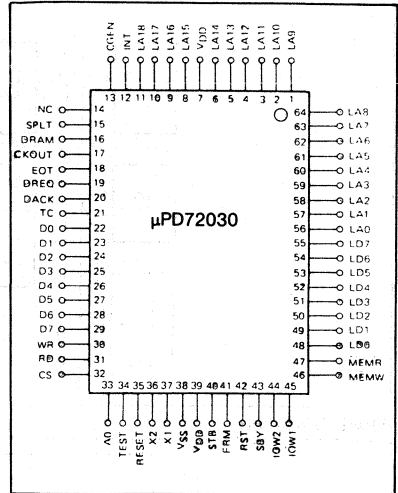


The μPD72030G is an LSI used to control the display of a heavy duty, dot-matrix LCD. Characters and graphics can be displayed on the LCD by providing commands through an 8085-compatible bus interface.

## Features

- Display duty 1/32 to 1/128
- Abundant functions for character display
- Internal character generator (5X7 dot matrix)
- 64 ASCII characters (alphanumeric characters and others)
- Expandable external character generator
- Use of character generator RAM possible
- Display memory: 64 KB MAX.
- Use of SRAM and DRAM possible
- Abundant control commands for easy data display
  - Cursor manipulation
- Vertical and horizontal movement
- Direct addressing
- Shift to home position
  - Editing
- Row/display screen clearing
  - Screen control
- Scrolling
  - Attribute functions
- Reverse display
- Underline
- Blinking display
- Secret
- Bus-compatible with other microcomputers such as the μCOM-87 and V series™
- Can be directly connected to the high-voltage output LCD driver μPD6307/6308G and is capable of 8-bit parallel high-speed transfer of display data.
- CMOS technology
- Single logic power supply +5 V
- 64-pin plastic flat package

## Pin Configurations (cont)



**1. System Configuration and General Functions**

The μPD72030G is an LCD controller which can be used to configure an LCD system by connecting a μPD6307G (LCD row driver), μPD6308G (LCD column driver), and a general-purpose RAM as the display memory to the μPD72030G. An external character generator may be connected as necessary.

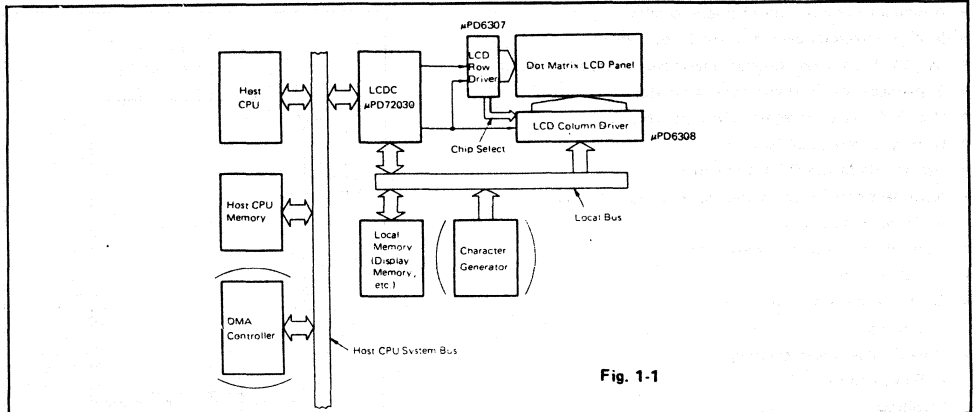
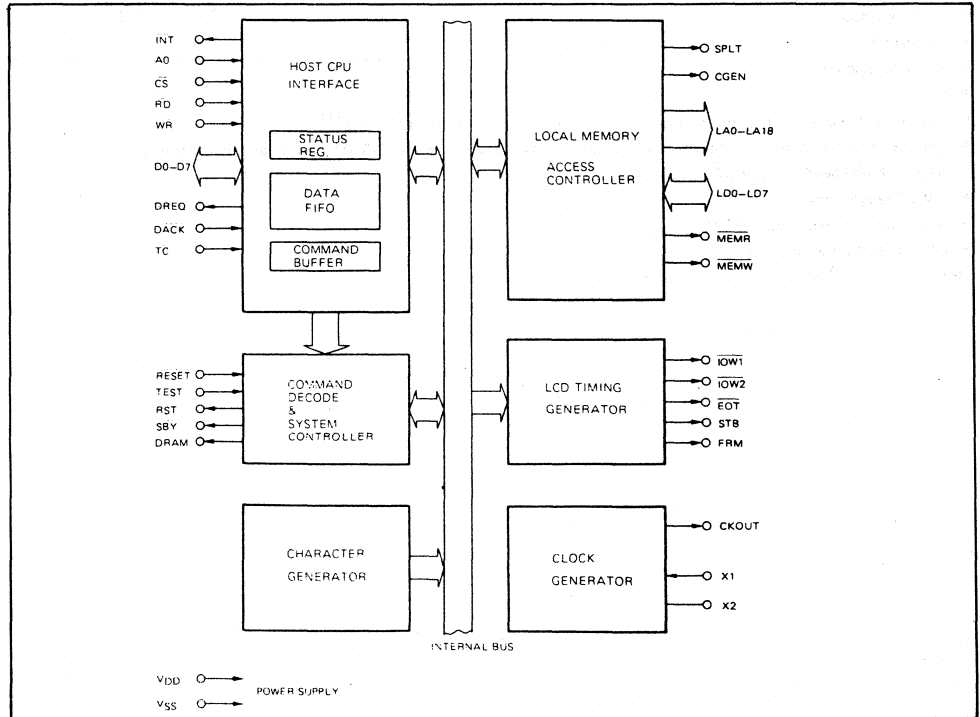


Fig. 1-1

**Block Diagram**

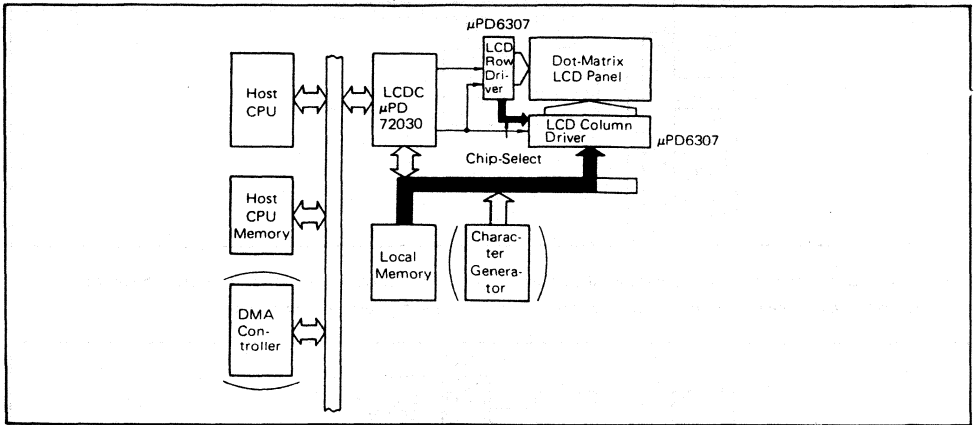


The μPD72030G carries out two processes: display and command. In the display process, the μPD72030G drives an LCD panel by sending display memory data to the column driver and timing signals to the row and column drivers. In the command process, the μPD72030G manipulates the display memory contents by commands sent from the host CPU.

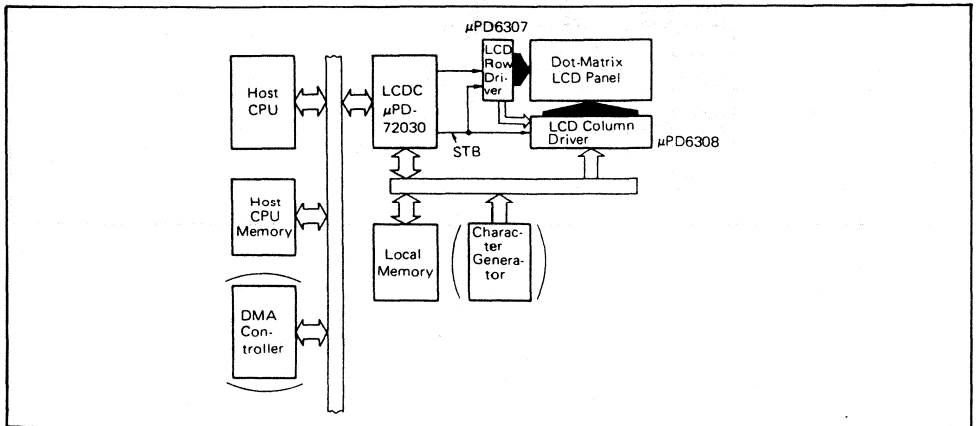
The following discusses these two processes in greater detail.

## 1.1 Display Process

- (1) The display memory contents are transferred directly to the column driver via the local bus at fixed timing intervals generated by the LCDC. When two or more column drivers are connected to the system, the chip select signal generated by the row driver determines the column driver to which the data are to be written.



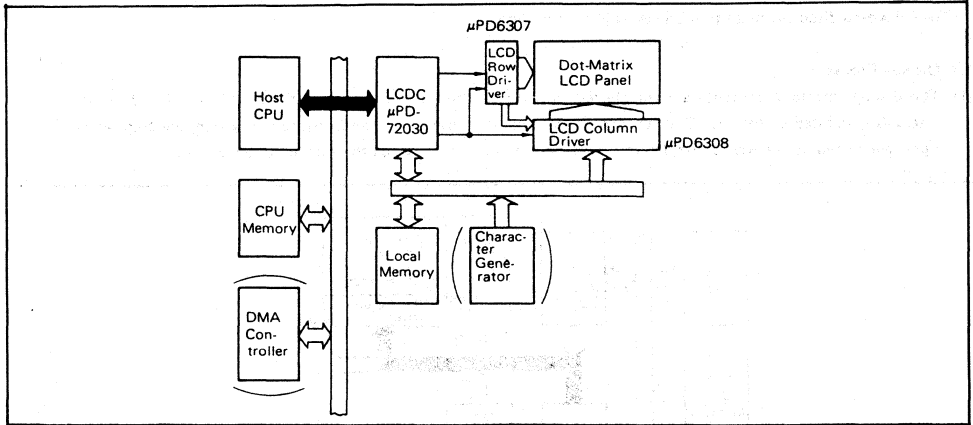
- (2) After display data are stored in each column driver, the STB signal is output. Then, one line of display data is output to the LCD panel. The row signals are scanned, and the above sequence is repeated to drive the LCD panel using the time-division method.



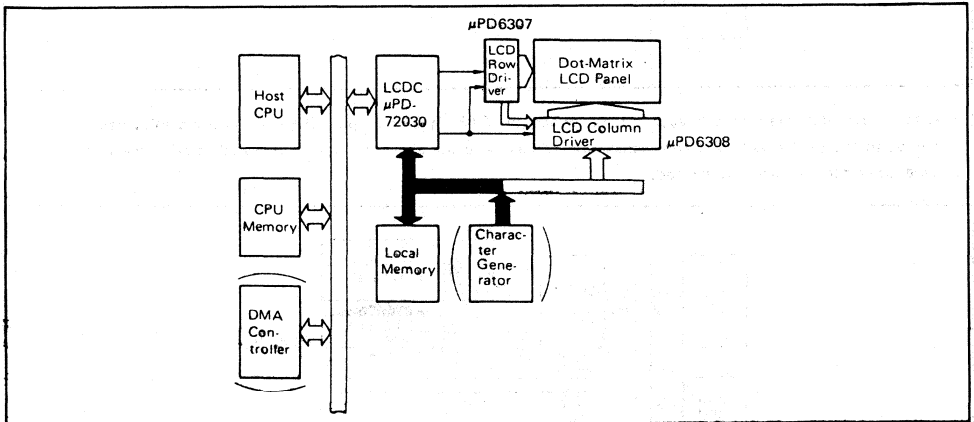
1.2 Command Process

(1) The host CPU reads and checks\* the status of the LCDC, then sends it an appropriate command.

(\* An interrupt may instead be used.)



(2) The LCDC interprets the given command, then executes it. To manipulate display data, the LCDC checks data such as those provided by the character generator, then manipulates the local memory contents.



## 2. PIN FUNCTIONS

### 2.1 D0 to D7 (Data Bus): Input/Output

This is the data bus for interfacing the host CPU. Writing commands, parameters, and data, or reading status and data are performed through this bus.

These pins connect the host CPU data bus.

### 2.2 A0 (Address 0) . . . Input

This pin inputs the host CPU commands and signals for function of the data bus (D0 to D7).

A0 = 0: data, parameter

A0 = 1: command, status

This pin connects the host CPU address bus.

### 2.3 $\overline{CS}$ (Chip Select) . . . Input

This pin inputs the host CPU interface-enable signal.  $\overline{CS} = 0$  enables the read/write function of the data bus.

This pin connects the host CPU address decode signal.

### 2.4 $\overline{RD}$ (Read Strobe) . . . Input

This pin inputs the status/data read strobe signal to the data bus. When the host CPU interface is enabled, with  $\overline{RD} = 0$ , status and data are read to the data bus.

This pin connects the host CPU read strobe signal.

### 2.5 $\overline{WR}$ (Write Strobe) . . . Input

This pin inputs the command/parameter write strobe signal from the data bus.

When the host CPU interface is enabled with  $\overline{WR} = 0$ , command/data on the data bus are written to the LCDC.

This pin connects the host CPU write strobe signal.

### 2.6 DREQ (DMA Request) . . . Output

This pin outputs a DMA service request signal for data block transfer.

When data block transfer is required between the host CPU memory and the LCDC local memory, if the data transfer is possible, DREQ is set to 1 to request DMA service.

This pin connects service request input of the DMA controller. If the block transfer function is not to be used, this pin should be left open.

### 2.7 $\overline{DACK}$ (DMA Acknowledge) . . . Input

This pin inputs an acknowledge signal for DMA service requests made when DREQ = 1.

When  $\overline{DACK}$  is set to 0,  $\overline{CS}$  and A0 are internally set to 0 regardless of their input level.

This pin connects the service acknowledge output of the DMA controller. If the block transfer function is not used, this pin should be pulled up to high level.

### 2.8 $\overline{TC}$ (Terminal Count) . . . Input

This pin inputs a termination signal for data block transfer.

When  $\overline{TC}$  is set to 0, termination of data block transfer is indicated.

This pin connects the DMA transfer termination output of the DMA controller. If the block transfer function is not used, this pin should be pulled up to high level.

### 2.9 LD0 to LD7 (Local Data Bus) . . . Input

These pins are the data bus to access the local memory. The display memory and external character generator are connected to the LCDC as local memories.

Communication with these memories is made through the local data bus.

**2.10 LA0 to LA18 (Local Address Bus) . . . Output**

These pins are the address bus to provide access to the local memory.

LA0 – 15: These pins are used for addressing the display memory and external character generator.

LA16 – 18: These pins are used for addressing the external character generator.

**2.11 CGEN (Character Generator Enable) . . . Output**

This is the external character generator enable signal output.

When CGEN = 1, the character address and scan address are output onto LA0 to LA18.

When CGEN = 0, the address to the display memory is output onto LA0 to LA15; LA16 and LA17 become don't care.

When the external generator is not used, this pin should be left open.

By combining with SPLT output,  $\overline{RFSH}$  output for a pseudo SRAM can be externally generated.

**2.12 SPLT (Split Screen Select) . . . Output**

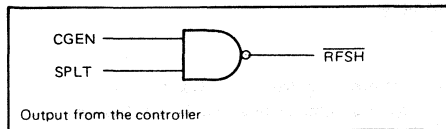
This pin is for the memory refresh select signal for the LCD screen partitioning matrix display.

SPLT = 0: Accesses the refresh memory for the upper portion of the screen.

SPLT = 1: Accesses the refresh memory for the lower portion of the screen.

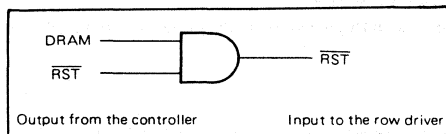
Refer to 7.1 for details.

By combining with CGEN output,  $\overline{RFSH}$  output for a pseudo SRAM can be externally generated.



**3.13 DRAM (DRAM Reset Enable) . . . Output**

When the DRAM (Dynamic RAM) is connected to the local memory, the DRAM reset enable signal and AND signal of  $\overline{RST}$  are input to the RST pin of the row driver. This prevents the loss of the display memory contents without refreshing the DRAM while the display is off.



**2.14 MEMR (Local Memory Read) . . . Output**

This is the read strobe to the local memory.

Pulses are output when the display memory contents are directly written to the display driver, with  $\overline{MEMR}$ ,  $\overline{IOW1}$ , and  $\overline{IOW2}$  set to low level. Pulses are also output when the LCDC reads the local memory contents, but with only  $\overline{MEMR}$  set to low level.

**2.15 MEMW (Local Memory Write) . . . Output**

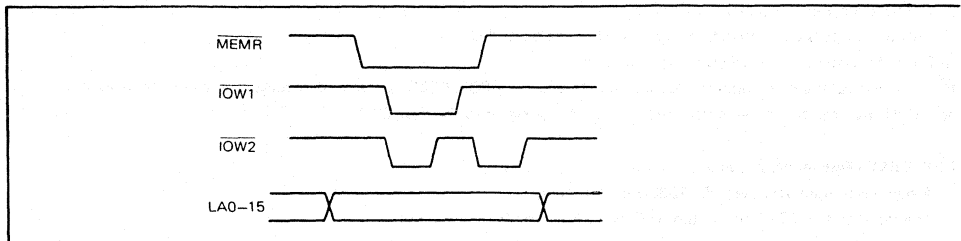
This is the write strobe to the local memory.

A low level pulse is output during data write to the local memory.

## 2.16 $\overline{IOW1}$ , $\overline{IOW2}$ (LCD Driver Write Strobe) . . . Output

These are the data write strobes to the LCD column (segment) driver.

$\overline{IOW1}$  generates one pulse and  $\overline{IOW2}$  generates two pulses for each  $\overline{MEMR}$  pulse. Therefore, these are selected according to how the LCD column driver is used.



## 2.17 STB (Strobe) . . . Output

This is the row drivers signal strobe output.

One pulse is output at the timing interval for display of one row.

The display duty is determined by the number of STB pulses output during each frame interval.

## 2.18 FRM (Frame) . . . Output

This is an AC-converted signal output to drive the LCD.

If it is high, a positive frame is displayed, and if it is low, a negative frame is displayed.

## 2.19 $\overline{EOT}$ (End of Transfer) . . . Output

This output indicates the end of display data transfer for one row.

When the display data transfer for one row is complete, a low-level pulse is output. This is used as clear signal for the chip select signal generator counter of the row driver.

## 2.20 $\overline{RST}$ (LCD Driver Reset) . . . Output

This is the LCD driver reset output.

When the LCD driver is being reset, a low-level signal is output at this pin.

For normal display, this is set to high level.

## 2.21 $\overline{SBY}$ (LCD Standby) . . . Output

This output sets the LCD driver to the standby state.

To stop the LCD display and put the LCD in the standby state, a low level signal is output during execution of the STOP command.

This is set to high level for normal display, and when a  $\overline{RESET}$  signal is input.

## 2.22 CKOUT (Clock Out) . . . Output

This is the divided output of the internal oscillator.

A clock whose frequency was divided down to 1/15 from the original oscillator is output. This is set to high level during execution of the STOP command.

## 2.23 X1, X2 (Xtal1, Xtal2) . . . Output

These pins are used to connect an external crystal.

The μPD72030G has a built-in high-gain amplifier. Thus, a functional clock can be generated simply by connecting a crystal or ceramic resonator and two capacitors to X1 and X2 pins.

When an external clock is used, X1 and X2 function as clock input pins.



### 2.24 RESET (Reset Input) . . . Input

This is the reset input for the μPD72030G.

When a low-level signal is input to this pin, the μPD72030G is initialized.

### 2.25 INT (Interrupt Request) . . . Output

This pin outputs an interrupt service request to the host CPU.

$\overline{\text{INT}} = 1$ : Indicates a command is being processed.

$\overline{\text{INT}} = 0$ : Indicates a command process is complete and the μPD72030G is ready to request a new command to the host CPU.  $\overline{\text{INT}}$  can be conveniently used as an interrupt signal to the host CPU.

### 2.26 TEST (Test Input) . . . Input

A high-level input sets the μPD72030G in test mode.

The input to the TEST pin should be fixed to low level for normal use.

To achieve this, this pin can be directly connected to the  $V_{SS}$ .

## 3. GENERAL DESCRIPTION OF INTERNAL BLOCK

### 3.1 Host CPU Interface Circuit

The μPD72030G sends/receives commands, parameters, status, data, etc., to/from the host CPU through an 8-bit data bus.

In addition, a DMA controller can be connected through a host CPU interface circuit. The data bus functions as follows:

	$\overline{\text{CS}}$	$\overline{\text{DACK}}$	A0	$\overline{\text{RD}}$	$\overline{\text{WR}}$	STATUS OF D0 TO D7 PINS	HOST CPU INTERFACE OPERATION
	1	1	X	X	X	Z	No operation
	X	X	X	1	1	Z	No operation
Command input	0	1	1	1	0	Z	Fetches data on D0 to D7 into command buffer
Status output	0	1	1	0	1	A	Places status on D0 to D7
Parameter input	0	1	0	1	0	Z	Fetches data on D0 to D7 into data FIFO
Data output	0	1	0	0	1	A	Places data on D0 to D7 from contents of data FIFO
Data block input	X	0	X	1	0	Z	Fetches data on D0 to D7 into data FIFO
Data block output	X	0	X	0	1	A	Places data on D0 to D7 from contents of data FIFO

Z: High impedance

A: Active

### 3.2 Command Decoder and System Controller

The internal circuit of the μPD72030G is controlled according to the command and/or parameter. This circuit provides reset and standby functions to the system, and generates reset and standby signals to the LCD driver.

### 3.3 Character Generator

This is a ROM in which 64 ASCII characters (5X7 dot) are stored. The character fonts are shown in Fig. 3-1.

### 3.4 Local Memory Access Controller

This circuit generates a memory read signal for read, write, and display of the local memory contents.

### 3.5 LCD Timing Generator

This circuit generates display timing and data latch signals to the LCD driver.

### 3.6 Clock Generator (Oscillator)

The clock generator generates a clock signal used in the μPD72030G through an external crystal or ceramic resonator by inputting an external clock signal.

	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	XA	XB	XC	XD	XE	XF
2X	Space	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3X	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4X	a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5X	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6X	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7X	P	q	r	s	t	u	v	w	x	y	z	{		}	~	

Fig. 3-1 Internal Character Generator Code and Font Reference Table

## 4. CONNECTION TO THE HOST COMPUTER

Connecting the host CPU is easily done using an 8-bit parallel data bus in the same manner as other general-purpose peripherals. (See Fig. 4-1)

The following registers can be accessed depending on the data bus interfaced to the host CPU.

- (1) Command buffer (write only) A0 = 1
- (2) Data FIFO (read/write) A0 = 0
- (3) Status (read only) A0 = 1

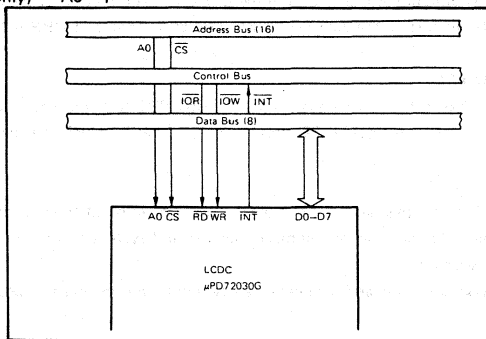


Fig. 4-1 Host CPU Connection

4.1 Command Buffer

The command buffer is an 8-bit register in which a command to be given to the LCDC is stored. Although a value can be set in the command buffer, the host CPU cannot read the value once it is set. When the command is written to the command buffer, CBF of the status flag is set to 1; when the LCDC receives the command and starts command execution, CBF is set back to 0.

Note that CBF = 0 does not mean that command process is terminated. The section 4.4 describes how to perform flag checks.

4.2 Data FIFO

For communication between the host CPU and the LCDC, an 8-byte bidirectional FIFO is used. The LCDC changes the FIFO direction depending on the command.

The status of the FIFO can be checked by the two flags IBR and OBR in the status register.

**IBR:** The master CPU can write the parameter and data to the LCDC's data FIFO only when this flag is set to 1.

When this is 0, write to the data FIFO should not be attempted.

**OBR:** The master CPU can read data from the LCDC's data FIFO only when this flag is set to 1.

If data are read from the data FIFO when this flag is 0, the data read out will be undefined.

**FIFO (First In, First Out)**

When a buffer register for two or more bytes of data exists between a data sender and receiver, after the sender writes the data to the buffer, the receiver reads the data in the same order they were written. This method is known as FIFO.

The opposite of this is called LIFO, Last In, First Out

4.3 Status

This is a register that indicates the functional status of the LCDC.

7	6	5	4	3	2	1	0
WRN	SBY	ERR	BSY1	CBF	BSY0	IBR	OBR

**OBR: Output Buffer Ready (bit 0)**

When the data FIFO direction is changed from the LCDC to the direction of the master CPU, and output data of one or more bytes exist in the data FIFO, this is set to 1. That is, if OBR is 1, data can be read from the data FIFO.

**IBR: Input Buffer Ready (bit 1)**

When the data FIFO direction is changed from the master CPU to the direction of the LCDC, and room for one or more bytes exists in the data FIFO, this is set to 1. That is, if IBR is 1, data can be written to the data FIFO.

**BSY0: Busy 0 (bit 2)**

When a command is given, this flag is set to 1 after a specific period of time. Then the LCDC provides its command and parameter to the internal command decoder and system controller. When command execution starts, this flag is set to 0.

If BSY0 = 0, the next command and parameter can be input from the master CPU.

**CBF: Command Buffer Full (bit 3)**

This flag indicates the status of the command buffer.

When CBF = 1: Indicates a command exists in the command buffer.

When CBF = 0: Indicates the command buffer is empty.

BSY0 and BSY1 require a certain period of time to be set to 1 after a command is written. However, CBF is set to 1 immediately after the command is written. Therefore, when giving a command, confirm that CBF = BSY0 = 0, or CBF = BSY1 = 0.

**BSY1: Busy 1 (bit 4)**

When a command is given, this flag is set to 1 after a specific period of time. When the LCDC completes execution of the command, this flag is set to 0. Note that Busy 0 is set to 0 when command execution starts. When Busy 1 is changed from 1 to 0, WRN and ERR flags can be checked.

**ERR: Error (bit 5)**

When there is an error in the parameter value, this flag is set to 1. When BSY1 = 0 or  $\overline{\text{INT}}$  pin is set to 0, the ERR flag can be checked.

This flag is automatically cleared upon start of the next command execution.

**SBY: Standby (bit 6)**

When the STOP command is executed and the μPD72030G enters the standby state, this flag is set to 1.

This flag is set to 0 when the μPD72030G leaves the standby state after the START command is given.

**WRN: Warning (bit 7)**

This flag is set to 1 if no memory exists to be cleared when the CLR LN, CLR FRM, or CLR GRP command is executed or no character code memory exists, when the TRNS command is executed. When BSY1 = 0 or  $\overline{\text{INT}}$  pin is set to 0, the WRN flag can be checked.

This flag is automatically cleared upon starting execution of the next command.

5. COMMANDS

Because the μPD72030G is an intelligent LCD controller, functions such as clearing of the display screen and scroll-up can be executed using simple commands issued by the host CPU. Consequently, the host CPU can perform other tasks while these functions are executed. The result is an increase in overall system efficiency and a reduction in host CPU overhead.

The commands are 1 byte in length. Data written to the command buffer register inside the LCDC are received by the LCDC and the corresponding process is executed.

After the process is over, another command can be given to the LCDC.

Fig. 6-1 shows command code table. Command codes are allocated in the non-character portion of the character code. All codes input to the buffer that are not commands are processed as character codes for display. That is, a character can be displayed by simply writing its code to the command buffer.

Commands are divided into the following four groups according to their function:

(1) Initialization commands

These commands specify the LCDC's functional mode corresponding to the LCD display system configuration.

(2) Function specifying commands

These commands specify functions such as temporary halt of the display, restart, and standby.

(3) Display control commands

These commands specify such things as cursor position and display mode.

(4) Display data manipulation commands

These commands manipulate display data and their attributes.

For execution, one or more parameters may be required depending on the command. These parameters are sent to the data FIFO.

When the display data are to be read upon command execution, the data are sent to the host CPU from the LCDC, again using the data FIFO.

80	START	(0)	8F	CURLT	(0)	9F	BLKTOT	(x)	EE	CLRCHR	(2)
81	STOP	(0)	90	CURON	(0)	E0	BLINK0	(0)	EF	CLRGRP	(1)
82	DSPLY1	(0)	91	CUROFF	(0)	E1	BLINK1	(0)	F0	DSPPPOS	(1)
83	BLANK	(0)	92	SELECTO	(0)	E2	BLINK2	(0)	F1	MEMADR	(8)
84	DSPLY2	(0)	93	SELCT1	(0)	E3	BLINK3	(0)	F2	MEMSIZ	(8)
85	DSPLY3	(0)	94	CURDR	(2)	E4	CURSOR	(1)	F3	SELECTCG	(1)
86	CLRLN	(0)	95	ATTR	(1)	E5	MODEC	(0)	F4	DISPL	(0)
87	CLRFRM	(0)	96	SYNC	(5)	E6	MODEG	(0)	20~7F, A0~DF		
88	DSPLY4	(0)	97	DSPDEF	(7)	E7	MODEM	(0)		CHRDSP	(1)
89	STOP2	(0)	98	DIVIDN	(0)	E8	READ	(x)	1B	ESC	(0)
8A	ATTROF	(0)	99	DIVIDU	(4)	E9	WRITE	(x)	0A	LF	(0)
8B	CURHM	(0)	9A	DIVIDD	(4)	EA	DRESET	(7)	0D	CR	(0)
8C	CURUP	(0)	9B	DIVIDB	(8)	EB	DSET	(7)	08	BS	(0)
8D	CURDN	(0)	9C	TRANS	(2)	EC	GET	(7)			
8E	CURRT	(0)	9E	BLKTIN	(x)	ED	COMP	(7)			

Note: Parentheses indicate the number of parameters.

Fig. 5-1 Command Code Table

## 6. LOCAL MEMORY

The LCDC has a function to control the memory used for LCD display in accordance with the commands received from the host CPU. This is the local memory of the LCDC. This memory is independent of the main CPU, which means CPU memory is not sacrificed for display. If direct manipulation of the display memory from the main CPU is required, this can be realized by allocating the local memory as part of the memory space of the main CPU. This can be done by adding a switching circuit between the local memory's address and data buses and the main CPU's address and data buses. Either SRAM or DRAM can be used as local memory. Type of memory to be used can be selected using the SYNC command.

The local memory consists of the display memory and the character generator. The display memory is divided into the following four memories:

- (1) Refresh memory
- (2) Character code memory
- (3) Character attribute memory
- (4) Graphics memory

The allocation of these memory spaces in the local memory is specified by the two commands, MEMADR and MEMSIZ. MEMADR specifies the start address and MEMSIZ specifies the byte length. Consequently, each memory can be independently allocated in the local memory in a desired manner. If a memory of sufficient capacity is mounted, the display of several pages can be easily switched by issuing the Display Modify command which modifies the start address.

In addition, the character generator also is connected to the local memory. However, by switching memory banks to separate the memory space from the display memories, a high capacity character generator for character sets such as Kanji can be connected. This character generator space is allocated by an address fixed by the hardware. Therefore, allocation by command is not necessary. The method of connection is discussed in section 6.5.

### 6.1 Refresh Memory

Data stored in the refresh memory are automatically transferred to the LCD driver periodically and displayed. The refresh memory stores data that is the logical sum (OR) of character display expanded to bit-image data and graphic display data.

The size of the refresh memory space is based on the number of pixels of the LCDC panel to be controlled.

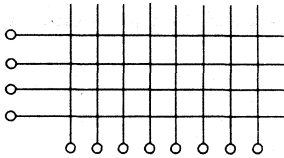
As was required for a 200X640 display with 1/100 duty, to perform a partitioned matrix display with a pixel number of twice the display duty, two refresh memories are necessary. One is for the upper screen and the other is for the lower screen. These two refresh memories simultaneously transfer display data for the two screens to each LCDC driver. The LCDC constructs the data so that character and graphics display assumes a continuous appearance between the upper and lower screens. Therefore, it is of no consequence to the host CPU that the screen is partitioned into upper and lower portions when commands are given.

Memories other than the refresh memory are not required to be partitioned into upper and lower portions.

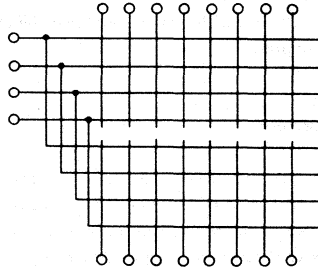
The SYNC command specifies whether or not a partitioned matrix is used.

**Partitioned Matrix Display**

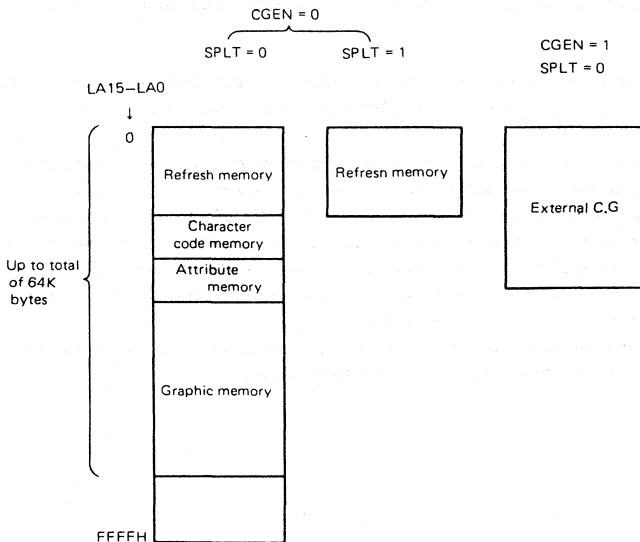
The number of pixels in the vertical direction on the LCD panel is limited by the number of time divisions of the display. While the number of time divisions has been increased with improved materials used for liquid crystal display, limitations are still imposed due to display quality. In partitioned matrix display, the number of pixels in the vertical direction can be twice the display duty; however, this requires that the number of pins for display in the column (segment) direction be twice the number of pixels in the horizontal direction.



Single Matrix



Partitioned Matrix



Typical Allocation of Local Memory

## 6.2 Character Code Memory

Either 8-bit or 16-bit character codes for character display can be stored in the character code memory. Whether the character code is to be 8-bit or 16-bit should be preset by DSPDEF command. This means that 1-byte and 2-byte codes cannot be mixed. When characters are not displayed, if FFFFH is specified as the start address of the character code memory, the LCDC interprets this as meaning that the character code memory does not exist.

The character memory can have a larger number of rows than that of the actual screen (up to 255 rows as set by MEMSIZ command.)

< Character code memory and scroll processing >

Fig. 6-3 below shows the relation between character code memory and LCD panel in scroll processing (screen movement in vertical direction).

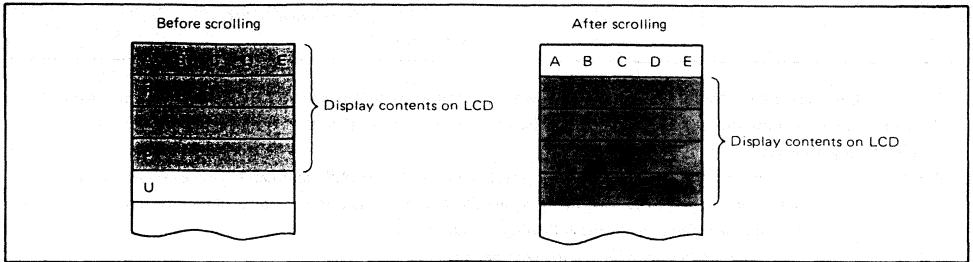


Fig. 6-3 Character Code Memory and Scroll Processing (I)

Fig. 6-4 below shows the relation between character code memory and LCD panel when the character code memory is scrolled up to the final line of the character code memory.

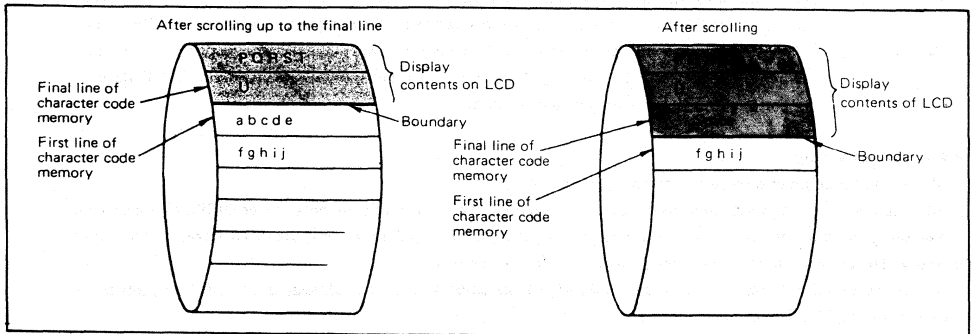


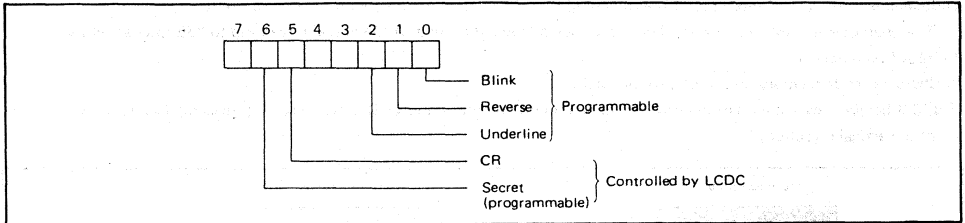
Fig. 6-4 Character Code Memory and Scroll Processing (II)

After scrolling, the LCD returns to the first line (next line down from the final line). At this time, the previous data is erased (a space code is written) and a blank line is displayed. This line is the final line of the character code memory. In this manner, although the addresses are non-continuous, the LCDC controls them as if they were continuous. The portion between the final line of the memory and the first line is called boundary.



### 6.3 Attribute Memory

This attribute memory is paired with the character code memory character by character, and stores attribute data (such as reverse display and underline) for each character. A byte of attribute memory is required for the display of each character. A single byte of the attribute memory stores the following information.



These attributes can be set by ATTR command. However, because the LCDC controls the screen, the attributes of CR and space are automatically attached when the character code is stored. (For the WRITE command, these are not automatically attached.)

**Blink** . . . . . This blinks the character at an interval specified by BLINK command. However, when the character is in the fixed row, the blink attribute is invalid and the character does not blink.

The blinking function has the following constraints:

- No blink takes place during the command process.
- The number of characters that can blink is limited due to the amount of process time required for blinking.

**Reverse** . . . . . This reverses the black and white portions of the character specified by the reverse attribute.

**Underline** . . . . . This draws an underline one pixel thick under the font of the specified character.

**CR** . . . . . This indicates the logical line end and is used for screen editing. When CR is input, the LCDC sets to 1 the CR bit of the attribute memory corresponding to the cursor location at CR input.

**Secret** . . . . . This indicates that the corresponding code is secret and no character is displayed on the display. When data is cleared by pressing the space key using CLRCHR, CLRLN, or CLRFRM command, the LCDC automatically adds a space.

### 6.4 Graphics Memory

Pixel data for graphics display are stored in this memory.

Manipulations of set, reset, and reverse for each pixel are performed to this memory. After DISPLAY1 command is executed, the character display is overlapped with the data in the graphics memory, then transferred to the refresh memory. The graphics display then changes (display-ON mixed mode).

When the graphics display is not needed, if FFFFH is specified as the start address, the LCDC interprets this as meaning the graphics memory does not exist.

One pixel on the LCD panel corresponds to one bit of the graphics memory. Fig. 6-5 shows the relation between the graphics memory addresses and LCD panel when the start address of the graphics memory is 1000H.

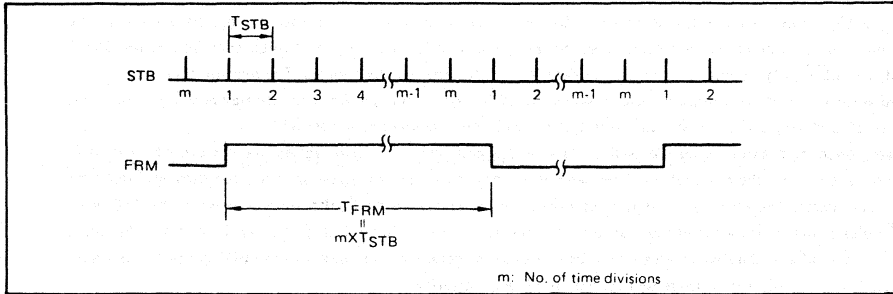




## 7. FUNCTIONAL TIMING

### 7.1 LCD Drive Timing

The LCD driver is controlled so that an LCD of up to a 1/128 duty cycle can be driven. The timing for this is as shown below.



$T_{STB}$  and  $T_{FRM}$  can be specified by SYNC command. See the chapter on commands for details.

$$T_{STB} : \left( \frac{1}{f_{osc}} \times 15 \right) \times n \dots \text{ (} f_{osc} \text{ is the original frequency of the oscillator; } n \text{ is an integer 1 to 256.)}$$

$$T_{FRM} : T_{STB} \times m \dots \text{ (} m \text{ is the number of time divisions, expressed as any integer from 1 to 128.)}$$

n: STB interval setting value

Here, value  $m$  is determined by the LCD panel used for display; however,  $n$  should be a value which satisfies the following condition:

$$T_{FRM} \leq \frac{1}{f_{FRM} \text{ (MIN.)}} \dots \text{ (A)}$$

The minimum display frame frequency ( $f_{FRM} \text{ (MIN.)}$ ) is determined according to the LCD panel to be used. However, lowering to too low a frequency causes flicker on the LCD display. Normally, 50 to 60 Hz is the minimum frequency. Data for one row (common) should be transferred to the column (segment) driver during  $T_{STB}$ . If  $T_{STB}$  is shortened, display may move to the next row before completion of display data transfer, and thus disturb the LCD display.

Display data transfer is discussed in Section 7.2;  $n$  should be a value which satisfies the formula in 7-2 (c). The SYNC command sets the values of  $m$  and  $n$ .

### 7.2 Local Memory Access Timing

The LCD performs the following three types of access to the local memory.

- (1) Display
- (2) Data read
- (3) Data write

When the STB pulse is output, display access begins so that data in the refresh memory are transferred to the LCD column (segment) driver. Once display access has been repeated for the number of bytes specified by the SYNC command, the  $\overline{EOT}$  pulse is output, and display data transfer for one row terminates. The next display access is not performed until the next STB pulse is output. While the LCDC is displaying a particular display and no command is executed (that is, during the command wait state), the LCDC performs the above functions.

However, once command execution begins, data read access and data write access are performed to modify display and to read from the external character generator. These two types of access are executed through interrupts to the display access. Therefore, when data read and data write access are made, the interval between the STB pulse and  $\overline{EOT}$  pulse is increased by the number of required interrupts. This is shown in Fig. 7-1. As the command is being processed, data read and data write access may be executed as necessary (even between the  $\overline{EOT}$  pulse and the next STB pulse). That is, execution of these two accesses has highest priority.

However, if a command is not being processed, and the LCDC is in command wait state, the LCDC does not access the local memory between the  $\overline{EOT}$  pulse and the next STB pulse. It is possible to use this timing to access the display memory directly from the host CPU by providing an external bus switching circuit.

During execution of the BLKTIN and BLKTOT commands, the frequency of the data read accesses and data write accesses will be at its highest. There is a 1:3 ratio between these two accesses and the display access.

Therefore, the time required to transfer data for one row (that is, the time ( $T_{TRNS}$ ) between the STB pulse and the  $\overline{EOT}$  pulse) is:

$$T_{TRNS} < \frac{4}{3} \ell \cdot T_{MCY} = \frac{10\ell}{f_{osc}}$$

To perform display normally:

$$T_{TRNS} < \frac{10\ell}{f_{osc}} \leq T_{STB}$$

$$\frac{10}{f_{osc}} \cdot \ell \leq T_{STB} \dots\dots\dots (B)$$

- Where  $\ell$  : Number of display data bytes in the column direction
- $f_{osc}$  : Original frequency of the oscillator
- $T_{MCY}$  : Time for one memory cycle  $T_{MCY} = 7.5/f_{osc}$

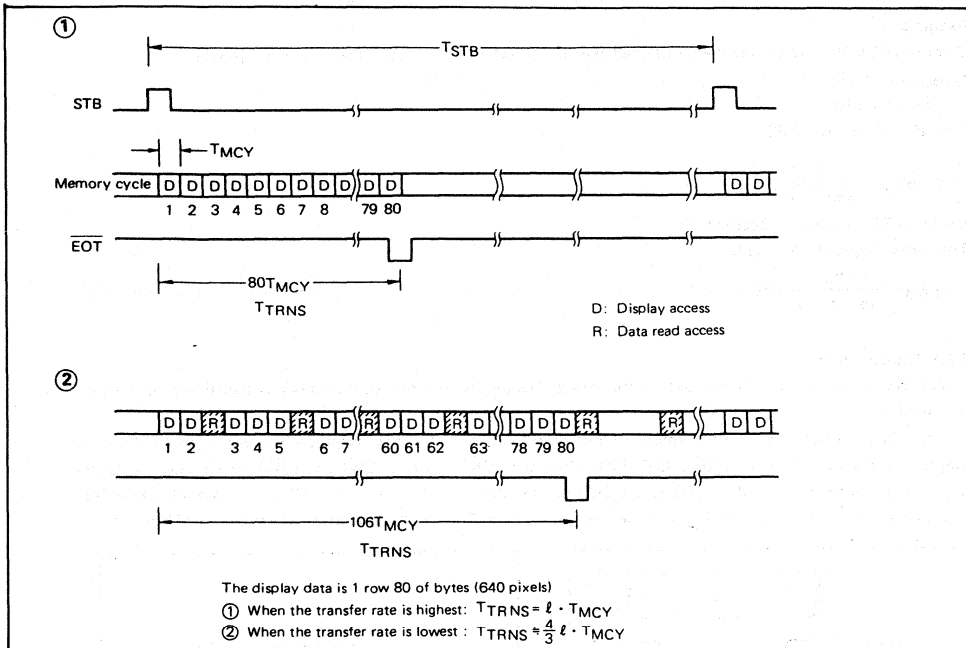


Fig. 7-1

Set the following values as the STB interval of the SYNC command according to the formulas given in 7-1 (A) and 7-2 (B).

If these values are not satisfied, the display may be disturbed or flicker will occur on the screen. This should be noted when setting the SYNC command.

$$\frac{2}{3} \ell < n \leq \frac{f_{osc}}{15 \cdot m \cdot f_{FRM(MIN.)}} \dots\dots\dots (C)$$

where,

- $\ell$  : Display data byte length in the column direction
- $n$  : Frequency dividing ratio to determine the STB pulse interval
- $m$  : Number of time divisions for the display
- $f_{osc}$  : Original frequency of the oscillator
- $f_{FRM(MIN.)}$  : Minimum display frame frequency

**Example:**

To drive 200X640 pixels using the partitioned matrix method at 1/100 duty, when  $f_{osc} = 6.0$  [MHz].

$$f_{FRM(MIN.)} = 60 \text{ [Hz]}$$

1 byte = 8 bits

$$\ell = 640 \div 8 = 80, m = 100$$

$$\frac{2}{3} \ell = 53.3, \frac{f_{osc}}{15mf_{FRM(MIN.)}} = 66.6$$

$n = 54$  to  $83$  (decimal) is specified.

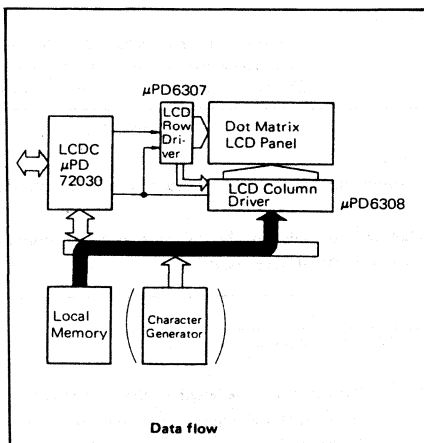
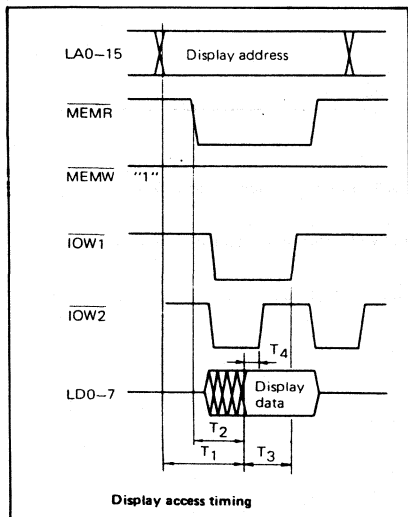
The frame frequency ( $f_{FRM}$ ) is:

$$f_{FRM} = \frac{f_{osc}}{15 \cdot m \cdot n} \text{ [Hz]}$$

**7.2.1 Display access**

For display access, the display data in the refresh memory within the local memory is transferred to the LCD column driver.

The LCDC outputs the refresh memory address onto LA0 to LA15, then outputs a low level signal to  $\overline{MEMR}$  to output the display data onto LD0 to LD7. Thereafter, the LCDC outputs  $\overline{IOW1}$  and  $\overline{IOW2}$ , and directly writes the display data placed onto LD0 to LD7 to the LCD column driver. In this case, the LCDC only outputs the address and the control pulse; the LCDC does not read the display data. This can be thought of as a type of DMA transfer.



- T<sub>1</sub>: Address access time
- T<sub>2</sub>: Read access time
- T<sub>3</sub>: LCD driver data set-up time (for  $\overline{IOW1}$ )
- T<sub>4</sub>: LCD driver data set-up time (for  $\overline{IOW2}$ )

The access time to the refresh memory ( $T_1$  and  $T_2$ ) should be set to correspond with the data setup time of the LCD column driver ( $T_3$  and  $T_4$ .)

The following is why two types of write signals to the LCD column driver ( $\overline{IOW1}$  and  $\overline{IOW2}$ ) are provided:

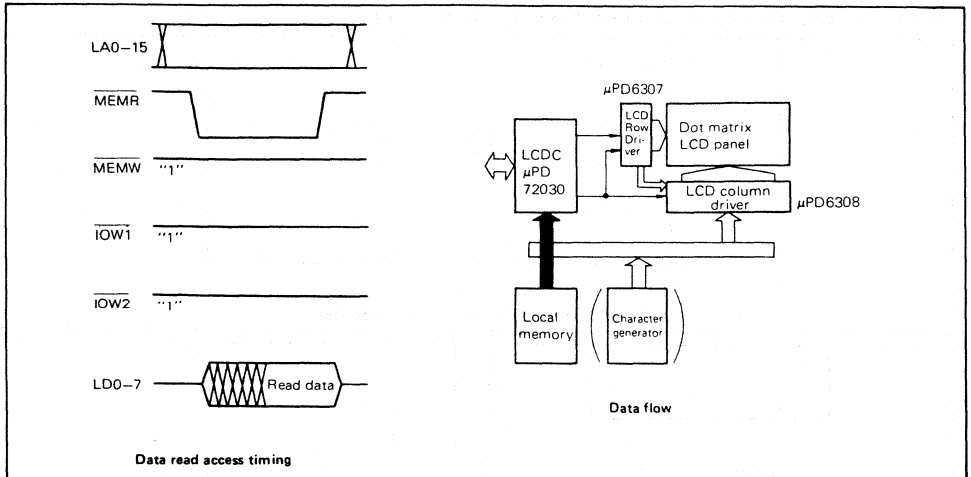
The display data are output in 8-bit units from the refresh memory. However, the LCD column driver (μPD6308G) accepts these data only in 4-bit units. Therefore,

- (1)  $\overline{IOW1}$  is input as the LCD driver write pulse ( $\overline{IOW}$ ) when an 8-bit display data is divided into two 4-bit units and written to the two μPD6308Gs simultaneously,
- (2)  $\overline{IOW2}$  is input as the LCD driver divided write pulse ( $\overline{IOW}$ ) when an 8-bit display data is divided into its upper and lower 4 bits, and written to one μPD6308G.

See the circuit shown in Figs. 11-2 and 11-5.

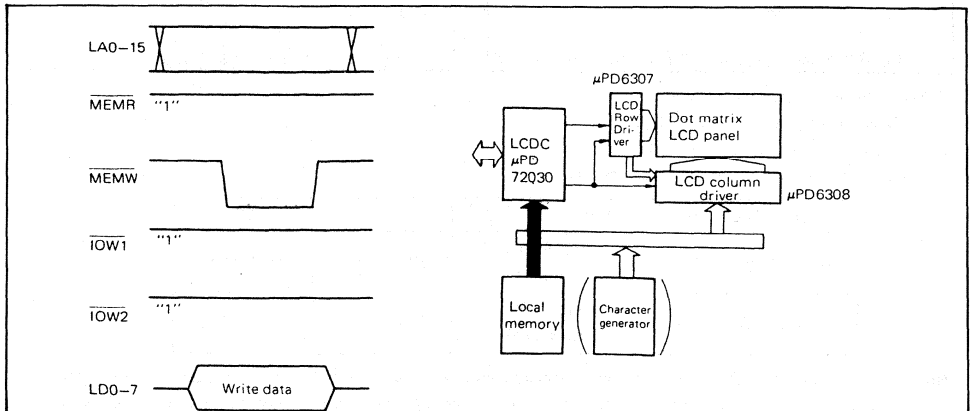
### 7.2.2 Data read access

During data read access, data in the local memory are read into the LCDC.



### 7.2.3 Data write access

During data write access, data are written to the local memory from the LCDC.





### 8. OSCILLATOR

The oscillator generates a reference clock for operation of the LCDC.

As shown in Fig. 8-1, a high gain amplifier for serial resonance is internally provided. A functional clock can be generated to the LCDC by connecting to the X1 and X2 pins a resonator (crystal or ceramic), and load capacitors. The inverter and amplifier inside the LCDC create self-oscillation. An external clock can be used; however, because  $V_{IH}$  of X1 and X2 are very high, the normal TTL level cannot drive this. In addition, when an external clock is used, caution should be taken concerning standby operation.

A crystal or ceramic resonator can be used as an external resonator for self-oscillation (see Fig. 8-2).

Since the oscillator is a high frequency analog circuit operating at a frequency in the range of several megahertz, take particular care in its wiring. The resonator and load capacitors should be located very close to the X1 and X2 pins, and wiring should be as short as possible. Grounding wire for the load capacitors (C1 and C2) should be a minimum length to the  $V_{SS}$  of the LCDC. By taking these points into consideration, adverse effects to the oscillator caused by inductance and capacitance of wiring can be minimized.

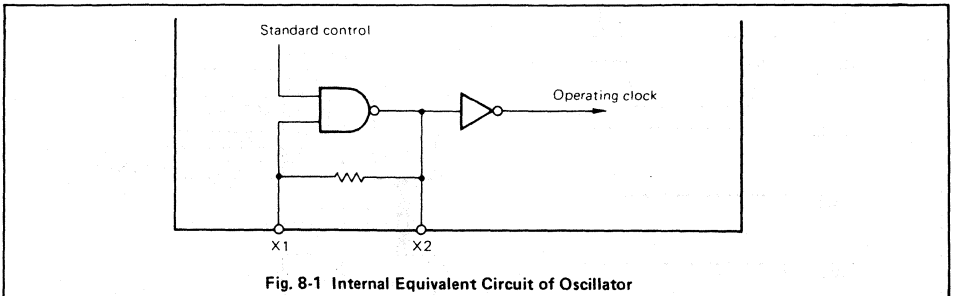


Fig. 8-1 Internal Equivalent Circuit of Oscillator

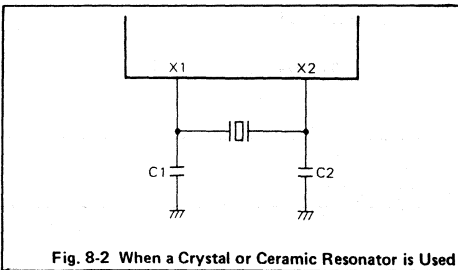


Fig. 8-2 When a Crystal or Ceramic Resonator is Used

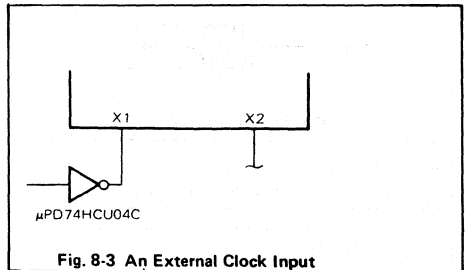


Fig. 8-3 An External Clock Input

The following crystals can be used:

External capacity:  $C1 = C2 \leq 50 \text{ pF}$

Equivalent series resistor:  $R_L < 100 \Omega$

The ceramic resonators listed below can be used.

MANUFACTURER	PRODUCT NAME	C1	C2
Murata Mfg. Co., Ltd.	CSA5.50M	30 pF ± 5 %	30 pF ± 5 %
	CSA6.0MT		
Kyocera Corporation	KBR-5.5M	33 pF ± 5 %	33 pF ± 5 %
	KBR-6.0M		

**Note:** When using a ceramic resonator, an externally connected return resistor is required. Stop current in this case increases by the portion of current that flows through the externally connected return resistor in addition to the rated value. When  $V_{DD}=5.5 \text{ V}$  the stop current increases by  $5.5 \mu\text{A}$  with  $R_f=1 \text{ M}\Omega$ .

## ELECTRICAL CHARACTERISTICS

### ABSOLUTE MAXIMUM RATINGS (T<sub>a</sub> = 25 °C)

Supply voltage	V <sub>DD</sub>	V <sub>SS</sub> -0.3 to +7	V
Input voltage	V <sub>I</sub>	V <sub>SS</sub> -0.3 to V <sub>DD</sub> +0.3	V
Output voltage	V <sub>O</sub>	V <sub>SS</sub> -0.3 to V <sub>DD</sub> +0.3	V
Operating temperature	T <sub>opt</sub>	-10 to +70	°C
Storage temperature	T <sub>stg</sub>	-65 to +125	°C

### OPERATING CONDITIONS

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITION
Supply voltage	V <sub>DD1</sub>	4.75	5.0	5.25	V	
Operating frequency	f <sub>opt1</sub>			6	MHz	-10 °C ≤ T <sub>a</sub> ≤ +70 °C
Supply voltage	V <sub>DD2</sub>	4.5	5.0	5.5	V	
Operating frequency	f <sub>opt2</sub>			6	MHz	-10 °C ≤ T <sub>a</sub> ≤ +50 °C

### DC CHARACTERISTICS (T<sub>a</sub> = -10 to +70 °C, V<sub>DD</sub> = 5 V ±5 %)

(T<sub>a</sub> = -10 to +50 °C, V<sub>DD</sub> = 5 V ±10 %)

CHARACTERISTIC	SYMBOL	RATED VALUE			UNIT	CONDITION
		MIN.	TYP.	MAX.		
Low-level input voltage	V <sub>IL</sub>	-0.3		0.7	V	
High-level input voltage	V <sub>IH1</sub>	2.2		V <sub>DD</sub> +0.3	V	Other than <u>RESET</u> , X1
	V <sub>IH2</sub>	V <sub>DD</sub> -1		V <sub>DD</sub> +0.3	V	<u>RESET</u> , X1
Low-level output voltage	V <sub>OL</sub>	0		0.45	V	<u>IOW1</u> , <u>IOW2</u> , LD0 to 7 CKOUT, MEMR, MEMW LA0 to 18, D0 to 7, <u>INT</u> , CGEN. SPLT, DRAM, DREQ FRM, STB, <u>RST</u> , <u>EOT</u> , <u>SBY</u> I <sub>OL</sub> = 1.8 mA
High-level output voltage	V <sub>OH</sub>	4.0			V	<u>IOW1</u> , <u>IOW2</u> , LD0 to 7 CKOUT, MEMR, MEMW LA0 to 18, D0 to 7, <u>INT</u> , CGEN. SPLT, DRAM, DREQ FRM, STB, <u>RST</u> , <u>EOT</u> , <u>SBY</u> I <sub>OH</sub> = -0.4 mA
Input current	I <sub>I</sub>	-25		-3	μA	<u>RESET</u> , V <sub>IN</sub> ≤ V <sub>IL</sub>
Input leakage current	I <sub>IL1</sub>	-3		+3	μA	<u>TC</u> , <u>DACK</u> , <u>CS</u> , <u>AO</u> , <u>RD</u> , <u>WR</u> V <sub>IN</sub> = 0 to V <sub>DD</sub>
Output leakage current	I <sub>LO</sub>	-3		+3	μA	D0 to 7, LD0 to 7 at high impedance
Current when stopped (1)	I <sub>STP1</sub>		1	12.0	μA	XTAL oscillation, no load, STOP command
Current when stopped (2)	I <sub>STP2</sub>		3	10	mA	f <sub>osc</sub> = 6.0 MHz, no load, STOP2 command
Operating current	I <sub>SS</sub>		5	14	mA	f <sub>osc</sub> = 6.0 MHz, no load

CAPACITY (T<sub>a</sub> = 25 °C)

CHARACTERISTIC	SYMBOL	RATED VALUE			UNIT	CONDITION
		MIN.	TYP.	MAX.		
Input capacity	C <sub>I</sub>		7	12	pF	f <sub>osc</sub> =1 MHz
Input/Output capacity	C <sub>IO</sub>		7	12	pF	V <sub>osc</sub> =0.1 V

AC CHARACTERISTICS FOR MPU (T<sub>a</sub> = -10 to +70 °C, V<sub>DD</sub> = 5 V ±5 %)  
(T<sub>a</sub> = -10 to +50 °C, V<sub>DD</sub> = 5 V ±10 %)

CHARACTERISTIC	SYMBOL	RATED VALUE			UNIT	CONDITION
		MIN.	TYP.	MAX.		
$\overline{\text{DACK}}, \overline{\text{CS}}, \text{A0}$ setup time for $\overline{\text{RD}}\downarrow$	t <sub>AR</sub>	0			ns	
$\overline{\text{DACK}}, \overline{\text{CS}}, \text{A0}$ hold time for $\overline{\text{RD}}\uparrow$	t <sub>RA</sub>	0			ns	
$\overline{\text{RD}}$ pulse width	t <sub>RR</sub>	250			ns	
$\overline{\text{DACK}}, \overline{\text{CS}}, \text{A0}$ →Data output delay	t <sub>AD</sub>			300	ns	C <sub>L</sub> =150 pF
$\overline{\text{RD}}$ →Data output delay	t <sub>RD</sub>			250	ns	
$\overline{\text{RD}}\uparrow$ →Data floating delay	t <sub>DF</sub>	10		120	ns	(Note)
$\overline{\text{DACK}}, \overline{\text{CS}}, \text{A0}$ setup time for $\overline{\text{WR}}\downarrow$	t <sub>AW</sub>	0			ns	
$\overline{\text{DACK}}, \overline{\text{CS}}, \text{A0}$ hold time for $\overline{\text{WR}}\uparrow$	t <sub>WA</sub>	0			ns	
$\overline{\text{WR}}$ pulse width	t <sub>WW</sub>	160		15/f <sub>osc</sub>	ns	
Data setup time for $\overline{\text{WR}}\uparrow$	t <sub>DW</sub>	100			ns	
Data hold time for $\overline{\text{WR}}\uparrow$	t <sub>WD</sub>	70			ns	
$\overline{\text{TC}}$ pulse width	t <sub>TC</sub>	100			ns	
$\overline{\text{TC}}\downarrow$ →D R E Q clear	t <sub>CDQ</sub>			200	ns	C <sub>L</sub> =80 pF
$\overline{\text{RD}}, \overline{\text{WR}}\downarrow$ → $\overline{\text{TC}}\uparrow$	t <sub>RT</sub>	100			ns	
$\overline{\text{TC}}\downarrow$ → $\overline{\text{RD}}, \overline{\text{WR}}\uparrow$	t <sub>TR</sub>	100			ns	
DREQ $\uparrow$ → $\overline{\text{DACK}}$	t <sub>DD</sub>	0			ns	C <sub>L</sub> =80 pF
$\overline{\text{DACK}}$ pulse width	t <sub>DACK</sub>	300			ns	

Note: Measuring point  
V<sub>OH</sub>=V<sub>DD</sub>-0.5 V  
V<sub>OL</sub>=0.5 V

## FOR LOCAL BUS

CHARACTERISTIC	SYMBOL	RATED VALUE			UNIT	CONDITION
		MIN.	TYP.	MAX.		
LA setup time for MEMR↑ *1	t <sub>LMR</sub>	60			ns	C <sub>L</sub> =100 pF
LA hold time for MEMR↑ *1	t <sub>MRL</sub>	100			ns	
MEMR pulse width	t <sub>MRR</sub>	660			ns	
MEMR cycle	t <sub>CYM</sub>		1250		ns	
MEMR↑ → MEMR↓	t <sub>MR(H)</sub>	250			ns	
LA → IOW↑	t <sub>AD(1)</sub>	610			ns	
MEMR↑ → IOW↑	t <sub>MRD(1)</sub>	400			ns	
MEMR hold time for IOW↑	t <sub>MRI1</sub>	50			ns	
MEMR setup time for IOW↓	t <sub>IMR1</sub>	160			ns	
IOW↑ pulse width	t <sub>I11</sub>	240			ns	
IOW↑ cycle	t <sub>CY1</sub>	1100			ns	
CKOUT pulse width	t <sub>FF</sub>	130			ns	
CKOUT cycle	t <sub>CY3</sub>		2500		ns	
MEMR↓ → CKOUT↓	t <sub>MRF</sub>	450			ns	
CKOUT↑ → MEMR↓	t <sub>FMR</sub>	200			ns	
MEMR↓ → data delay time	t <sub>MRD</sub>			430	ns	C <sub>L</sub> BUS=200 pF, C <sub>L</sub> =100 pF
MEMR↑ → data hold time	t <sub>MH</sub>	0			ns	
Data delay time for LA *	t <sub>AD1</sub>			500	ns	C <sub>L</sub> =100 pF
LA setup time for MEMW↓ *	t <sub>LMW</sub>	200			ns	
LA hold time for MEMW↑ *	t <sub>MWL</sub>	190			ns	
MEMW pulse width	t <sub>MWW</sub>	400			ns	C <sub>L</sub> BUS=200 pF, C <sub>L</sub> =100 pF
Data delay time for LA *	t <sub>AD2</sub>			270	ns	
Data setup time for MEMW↓	t <sub>DMW1</sub>	10			ns	
Data setup time for MEMW↑	t <sub>DMW2</sub>	580			ns	C <sub>L</sub> =100 pF *2
Data hold time for MEMW↑	t <sub>MWD</sub>	85			ns	
STB pulse width	t <sub>STB</sub>	360			ns	C <sub>L</sub> =100 pF
STB↑ → IOW↓	t <sub>SW1</sub>	530			ns	
IOW↑ → EOT	t <sub>W1E</sub>	0			ns	
EOT pulse width	t <sub>EOT</sub>	360			ns	
FRM (SBY, RST) → STB↓	t <sub>FS</sub>	720			ns	
LA → IOW2↑	t <sub>AD(2)</sub>	400			ns	
MEMR↓ → IOW2↑	t <sub>MRD(2)</sub>	220			ns	
MEMR↓ → IOW2↓	t <sub>MRI2</sub>	0			ns	
IOW2↑ → MEMR↓	t <sub>IMR2</sub>	140			ns	
IOW2 pulse width	t <sub>I12</sub>	150			ns	
IOW2 cycle	t <sub>CY2</sub>	500			ns	
IOW↑ → IOW2↓	t <sub>I12</sub>	50			ns	
IOW2↑ → IOW↑	t <sub>I21</sub>	70			ns	
STB↑ → IOW2↓	t <sub>SW2</sub>	450			ns	

\*1: Includes CGEN and SPLT pins

\*2: Measuring point  
 $V_{OH}=V_{DD}-0.5V$   
 $V_{OL}=0.5V$

TIMING DEFINITION OF BUS DEPENDENT ON tCY

CHARACTERISTIC	SYMBOL	RATED VALUE			UNIT	CONDITION
		MIN.	TYP.	MAX.		
LA setup time for MEMW↓ *	tLMR	1/f <sub>osc</sub> -106			ns	CL=100 pF
LA hold time for MEMR↑ *	tMRL	1.5/f <sub>osc</sub> -150			ns	
MEMR pulse width	tMRR	5/f <sub>osc</sub> -170			ns	
MEMR cycle	tCYM		7.5/f <sub>osc</sub>		ns	
MEMR↑ → MEMR↓	tMR(H)	2.5/f <sub>osc</sub> -166			ns	
LA → IOW↑	tAD(1)	5/f <sub>osc</sub> -223			ns	
MEMR↑ → TOW↑	tMRD(1)	4/f <sub>osc</sub> -266			ns	
MEMR setup time for IOW↓	tMRI1	1.5/f <sub>osc</sub> -200			ns	
MEMR hold time for IOW↑	tIMR1	1/f <sub>osc</sub> -13			ns	
IOW pulse width	tI1	2.5/f <sub>osc</sub> -176			ns	
IOW cycle	tCY1	7.5/f <sub>osc</sub> -150			ns	
CKOUT pulse width	tFF	1.5/f <sub>osc</sub> -120			ns	
CKOUT cycle	tCY3		15/f <sub>osc</sub>		ns	
MEMR↓ → CKOUT↓	tMRF	450			ns	
CKOUT↑ → MEMR↓	tFMR	200			ns	
MEMR↓ → Data delay time	tMRD			5/f <sub>osc</sub> -400	ns	CLBUS=200 pF, CL=100 pF
MEMR↑ → Data hold time	tMH	0			ns	
Data delay time for LA *	tAD1			7/f <sub>osc</sub> -666	ns	
LA setup time for MEMR↓ *	tLMW	2.5/f <sub>osc</sub> -216			ns	CL=100 pF
LA hold time for MEMW↑ *1	tMWL	1.5/f <sub>osc</sub> -60			ns	
MEMW pulse width	tMWW	3.5/f <sub>osc</sub> -183			ns	
Data delay time for LA *1	tAD2			1.5/f <sub>osc</sub> +20	ns	CLBUS=200 pF, CL=100 pF
Data setup time for MEMW↓	tDMW1	1/f <sub>osc</sub> -166			ns	
Data setup time for MEMW↑	tDMW2	4.5/f <sub>osc</sub> -170			ns	
Data hold time for MEMW↑	tMWD	1/f <sub>osc</sub> -81			ns	
STB pulse width	tSTB	3/f <sub>osc</sub> -140			ns	
STB↑ → IOW↓	tSW1	4/f <sub>osc</sub> -136			ns	
IOW↑ → EOT	tW1E	1/f <sub>osc</sub> -166			ns	
EOT pulse width	tEOT	3/f <sub>osc</sub> -140			ns	
FRM (SBY, RST) → STS↓	tFS	6/f <sub>osc</sub> -280			ns	
LA → IOW2↑	tAD(2)	4/f <sub>osc</sub> -266			ns	
MEMR↓ → IOW2↑	tMRD(2)	3/f <sub>osc</sub> -280			ns	
MEMR↓ → IOW2↓	tMRI2	1.5/f <sub>osc</sub> -250			ns	
IOW2↑ → MEMR↓	tIMR2	1/f <sub>osc</sub> -26			ns	
IOW2 pulse width	tI2	1.5/f <sub>osc</sub> -100			ns	
IOW2 cycle	tCY2	3.5/f <sub>osc</sub> -83			ns	
IOW1↑ → IOW2↓	tI12	1/f <sub>osc</sub> -116			ns	
IOW2↑ → IOW1↑	tI21	1/f <sub>osc</sub> -96			ns	
STB↑ → IOW2↓	tSW2	4/f <sub>osc</sub> -216			ns	

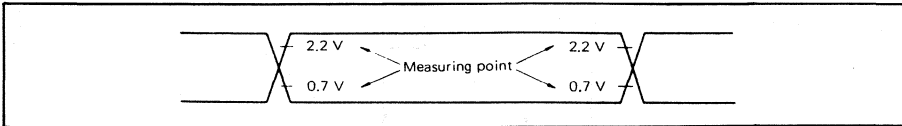
\*1: Includes CGEN and SPLIT pins

\*2: f<sub>osc</sub>=f<sub>opt1</sub>=f<sub>opt2</sub>

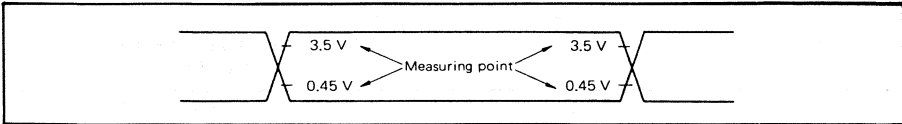
**FOR CLOCK**

CHARACTERISTIC	SYMBOL	CALCULATION FORMULA			UNIT	CONDITION
		MIN.	TYP.	MAX.		
X 1 cycle time	$t_{CY}$	166			ns	
X 1 pulse width (H)	$t_{CH}$	50			ns	
X 1 pulse width (L)	$t_{CL}$	80			ns	
X 1 rise time	$t_{CR}$			20	ns	
X 1 fall time	$t_{CF}$			20	ns	

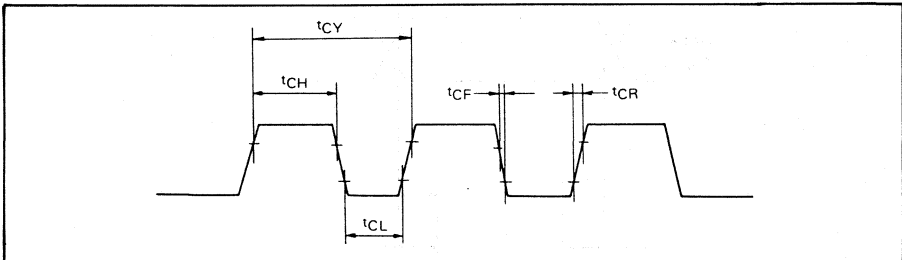
**Test Input Measuring Point (excluding X1)**



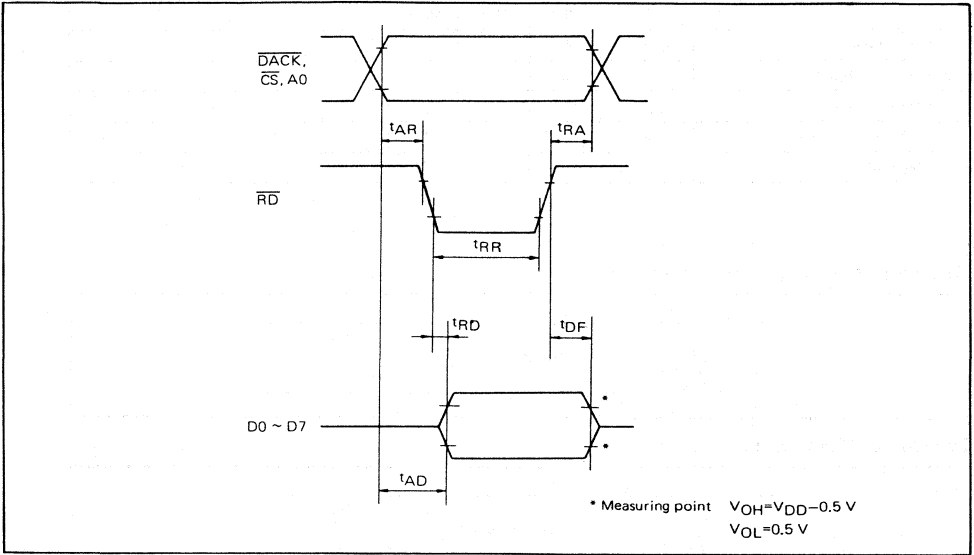
**Test Output Measuring Point**



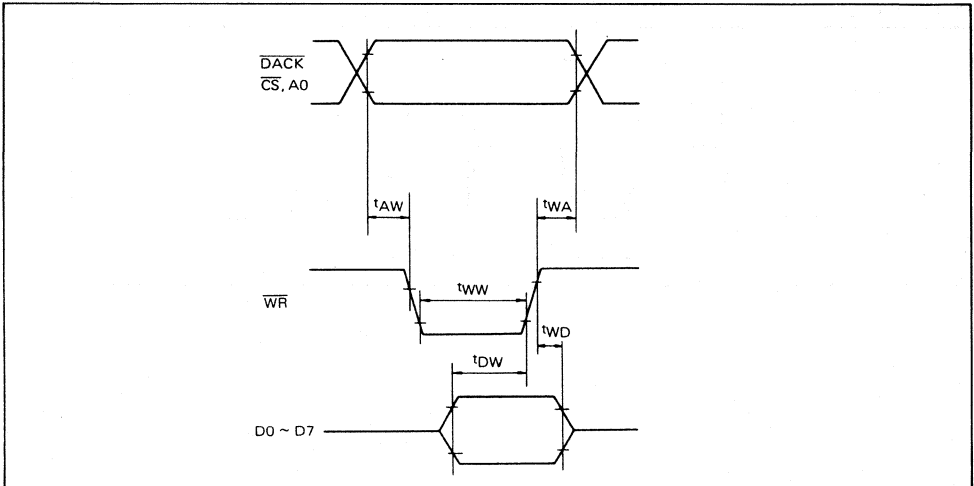
**Clock Timing**



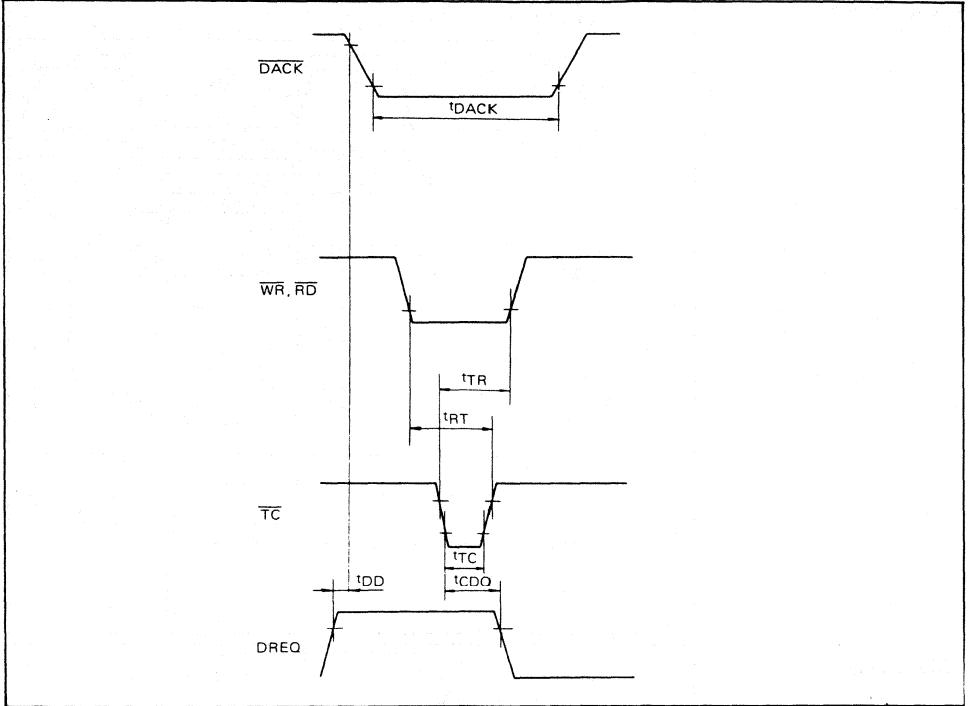
System Bus Timing  
Read timing



Write timing



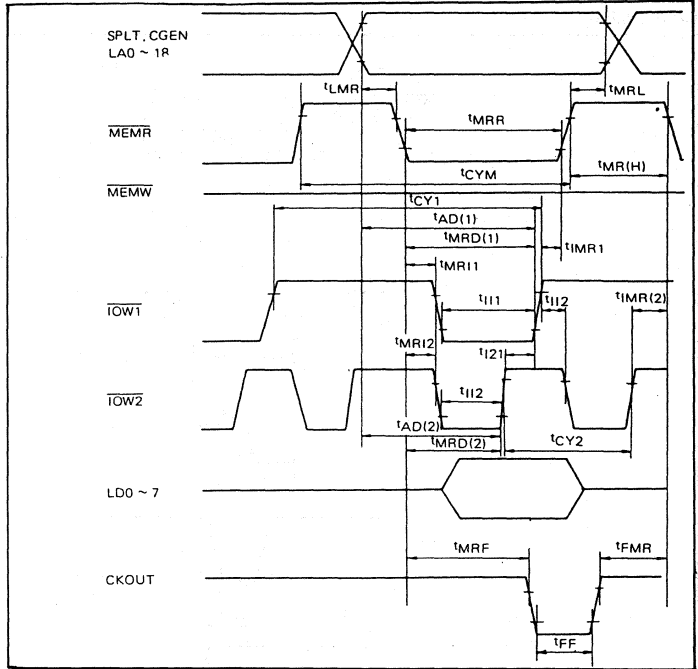
## DMA timing



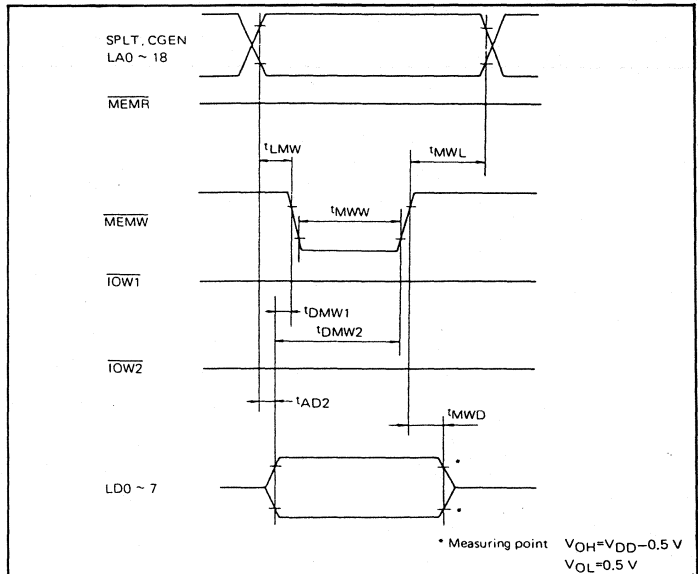


## μPD72030

### Local Bus Timing Display timina (I)

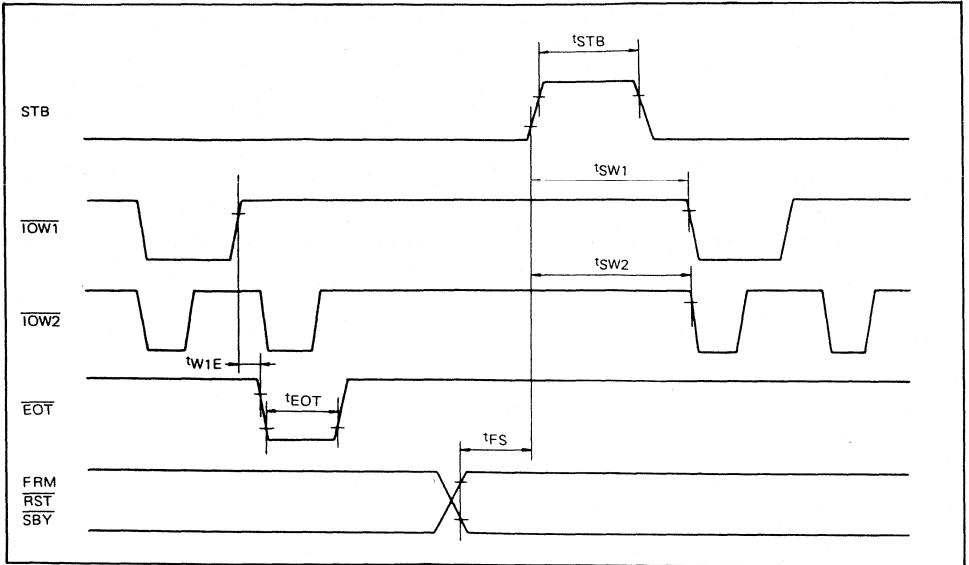


### Write timing

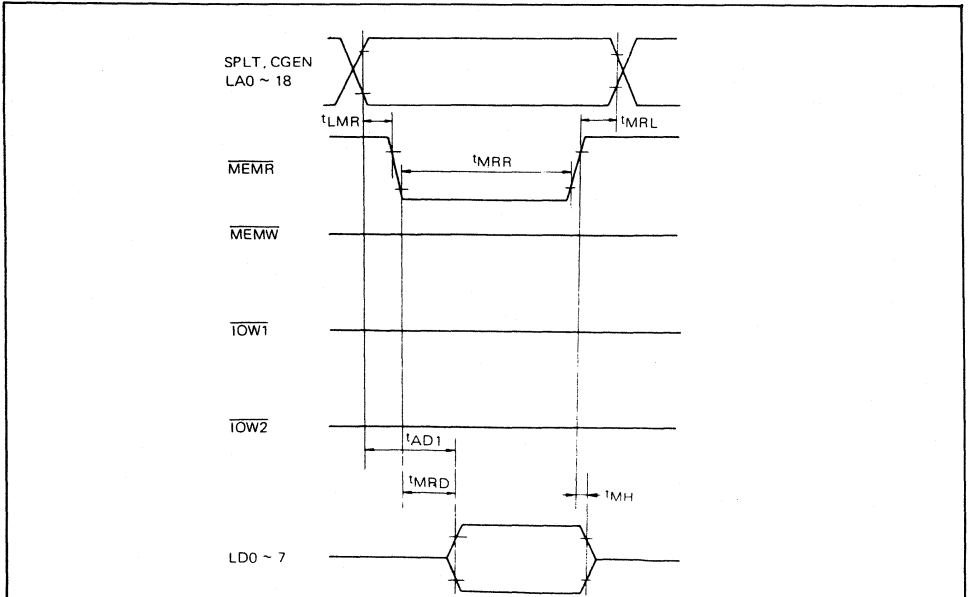


Note: Since this product is MOS IC, do not use in such environment with high charging property.

## Display timing (II)



## Read timing



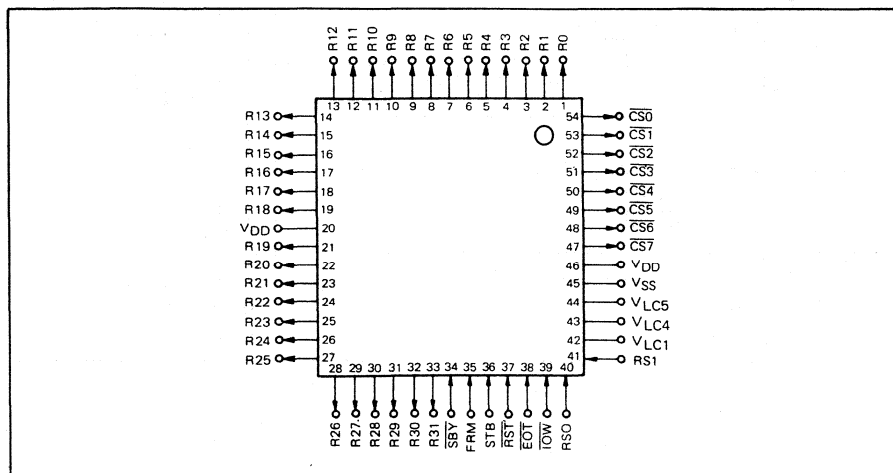


The μPD6307G is an IC for driving row signals of a heavy-duty, dot-matrix LCD. It is provided with 32 high-voltage outputs per package and is capable of driving up to a 128-row LCD by cascading four packages.

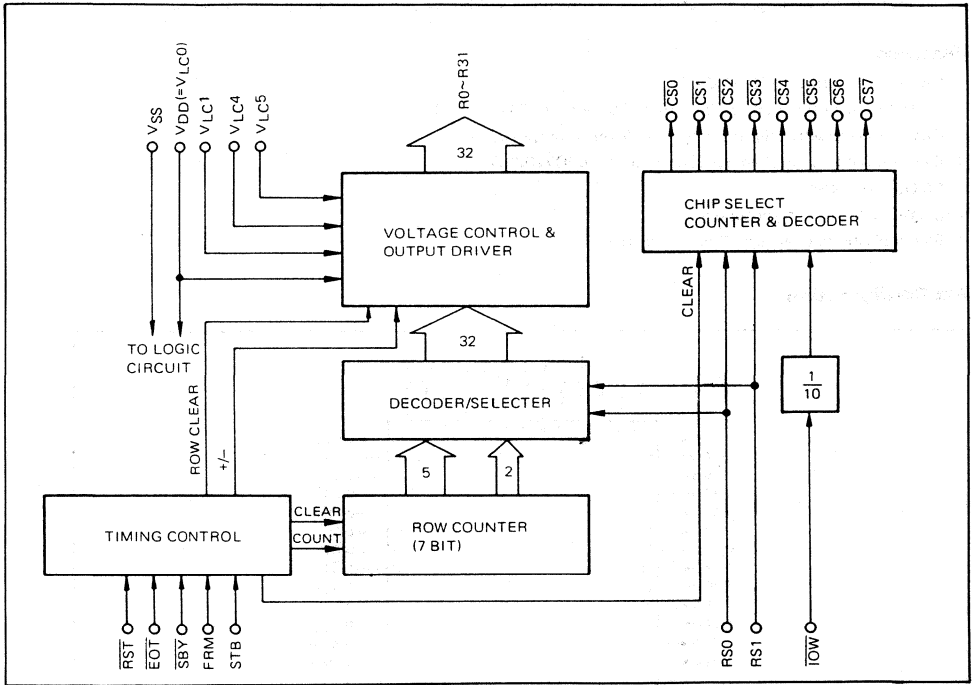
## Features

- High-voltage output: +20 V MAX.
- Up to four packages that can be cascaded to drive a 128-row LCD
- Generates chip-select signal for column driver: 8 outputs/chip
- Can be directly controlled by LCD controller μPD72030G
- CMOS technology
- Single +5 V logic power supply
- 54-pin plastic flat package (0.65-mm pitch)

## Pin Configuration



Block Diagram



## 1. Pin Description

### 1.1 FRM (Frame) . . . Input

This pin inputs row-drive AC signals. "H" displays the positive frame and "L" displays the negative frame. At the "H" to "L" or "L" to "H" transition, the row counter is cleared and the row-drive is started from row 0 (R0).

### 1.2 STB (Strobe) . . . Input

This pin inputs the row-drive signal strobe. Advances one row-drive signal at each leading edge of the STB input.

### 1.3 $\overline{IOW}$ (I/O Write) . . . Input

This pin inputs the chip-select count up signal. Advances one chip-select count after each group of 10  $\overline{IOW}$  pulses input.

### 1.4 $\overline{EOT}$ (End of Transfer) . . . Input

This pin inputs chip-select counter clear signal. Clears the chip-select counter by a low-level input.

### 1.5 RS0, RS1 (Row Select) . . . Input

This pin inputs row-driver cascade-connected select signals. Expandable up to 128 row-drive signal outputs and 32 chip-select outputs.

RS1	RS0	Row signal input	Chip-select output
0	0	R0 – R31	$\overline{CS0}$ – $\overline{CS7}$
0	1	R32 – R63	$\overline{CS8}$ – $\overline{CS15}$
1	0	R64 – R95	$\overline{CS16}$ – $\overline{CS23}$
1	1	R96 – R127	$\overline{CS24}$ – $\overline{CS31}$

### 1.6 $\overline{CS0}$ to $\overline{CS7}$ (Chip Select) . . . Output

This pin outputs chip-select signals for the column driver. These outputs are generated by the chip-select counter and RS0, RS1.

### 1.7 $\overline{RST}$ (Reset) . . . Input

This pin inputs the row-driver reset signal. When a low-level signal is input, the internal counter is cleared, all the row outputs R0 to R31 become non-select, and  $\overline{CS0}$  to  $\overline{CS7}$  outputs are set to high level.

### 1.8 $\overline{SBY}$ (Stand-by) . . . Input

This pin inputs the stand-by signal. When a low-level signal is input, the row outputs R0 to R31 are set to  $V_{LC0}$ . It is necessary to set all column-driver display data to 1 before entering the stand-by mode.

### 1.9 R0 to R31 (Row-Drive Outputs) . . . Output

This pin outputs row-drive signals. This is the LCD panel row (common) line-drive output.

### 1.10 $V_{LC1}$ , $V_{LC4}$ , $V_{LC5}$ (LCD Drive Supply) . . . Power supply

The LCD reference voltages reference the  $V_{DD}$  pin. These pins provide drive voltages for row-drive outputs.

### 1.11 $V_{DD}$ (= $V_{LC0}$ ) (Logic Supply and LCD Drive Supply) . . . Power supply

This is a pin common to the logic power supply and LCD reference voltage. Connect the power supply (+5 V) across pins  $V_{DD}$  and  $V_{SS}$  for logic circuit operation. This pin is also used for the row-drive signal output-drive voltage.

1.12 V<sub>SS</sub> (Logic Ground) . . . GND

This is logic ground pin.

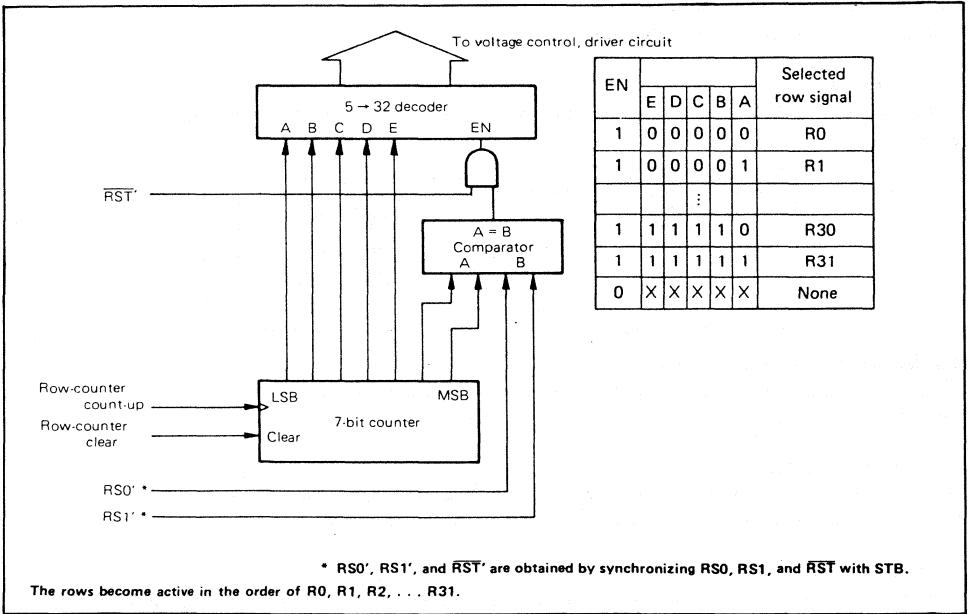
2. Internal Block Functions

2.1 Timing Control Circuit

This circuit controls the timing necessary to operate each of the internal blocks. FRM, RS0, RS1,  $\overline{RST}$ , and  $\overline{SBY}$  are sampled by this circuit at the leading edge of the STB, then supplied to the internal circuits.

2.2 Row-Counter, Decoder/Select Circuit

This circuit has a 7-bit counter to accommodate up to 128 segments. R0 to R31 become non-selected if the upper 2 bits of the counter do not match RS0 and RS1. If they match, one of R0 to R31, which is indicated by the lower 5 bits of the row counter, is selected. The remaining 31 outputs remain non-selected.



2.3 Voltage-Control Driver Circuit

This circuit generates the row (common) signals for driving the LCD panel with AC. The following levels are output on R0 to R31.

	. + (FRM'=1)	- (FRM'=0)
Select	V <sub>LC0</sub>	V <sub>LC5</sub>
Non-select	V <sub>LC4</sub>	V <sub>LC1</sub>

$\overline{RST}$  = Low level sets the output as non-select.

$\overline{SBY}$  = Low level sets the output V<sub>LC0</sub>.

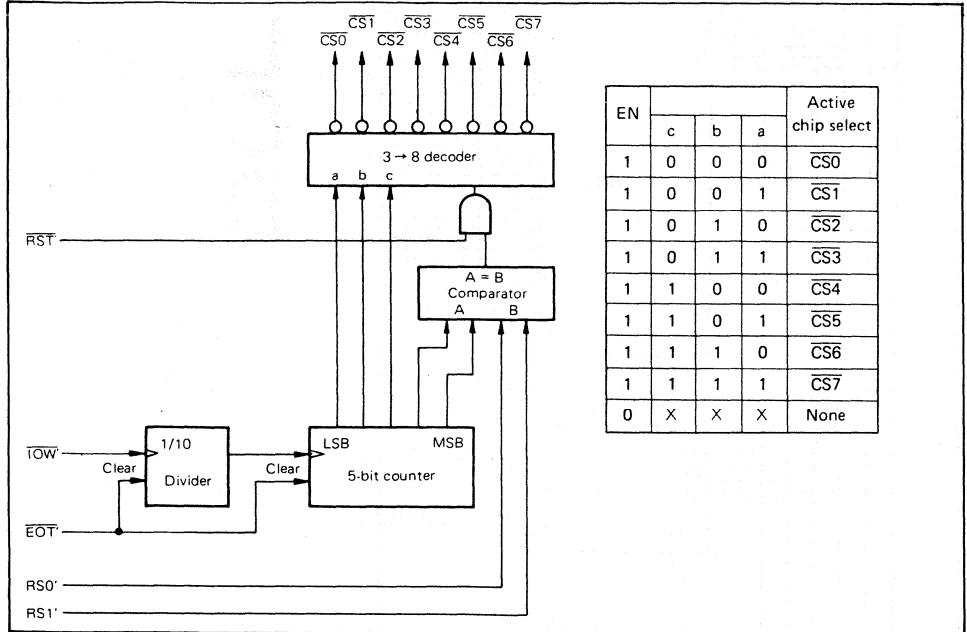
\* FRM' are obtained by internally synchronizing the FRM signal with the leading edge of the STB signal.

### 2.4 Chip-select Counter, Decoder Circuit

This circuit is the column-driver chip-select signal. It has a 5-bit counter so that up to 32 chip-select signals can be generated. A count signal of this counter obtained by dividing IOW by the 1/10 divider.

If the upper 2 bits of the chip-select counter do not match  $\overline{RS0}$  and  $\overline{RS1}$ , all the outputs  $\overline{CS0}$  to  $\overline{CS7}$  are set to high level. If they match, one of  $\overline{CS0}$  to  $\overline{CS7}$  (indicated by the lower 3 bits of the chip-select counter) becomes low level.

If  $\overline{RST}$  is low level, all  $\overline{CS0}$  to  $\overline{CS7}$  become high level.

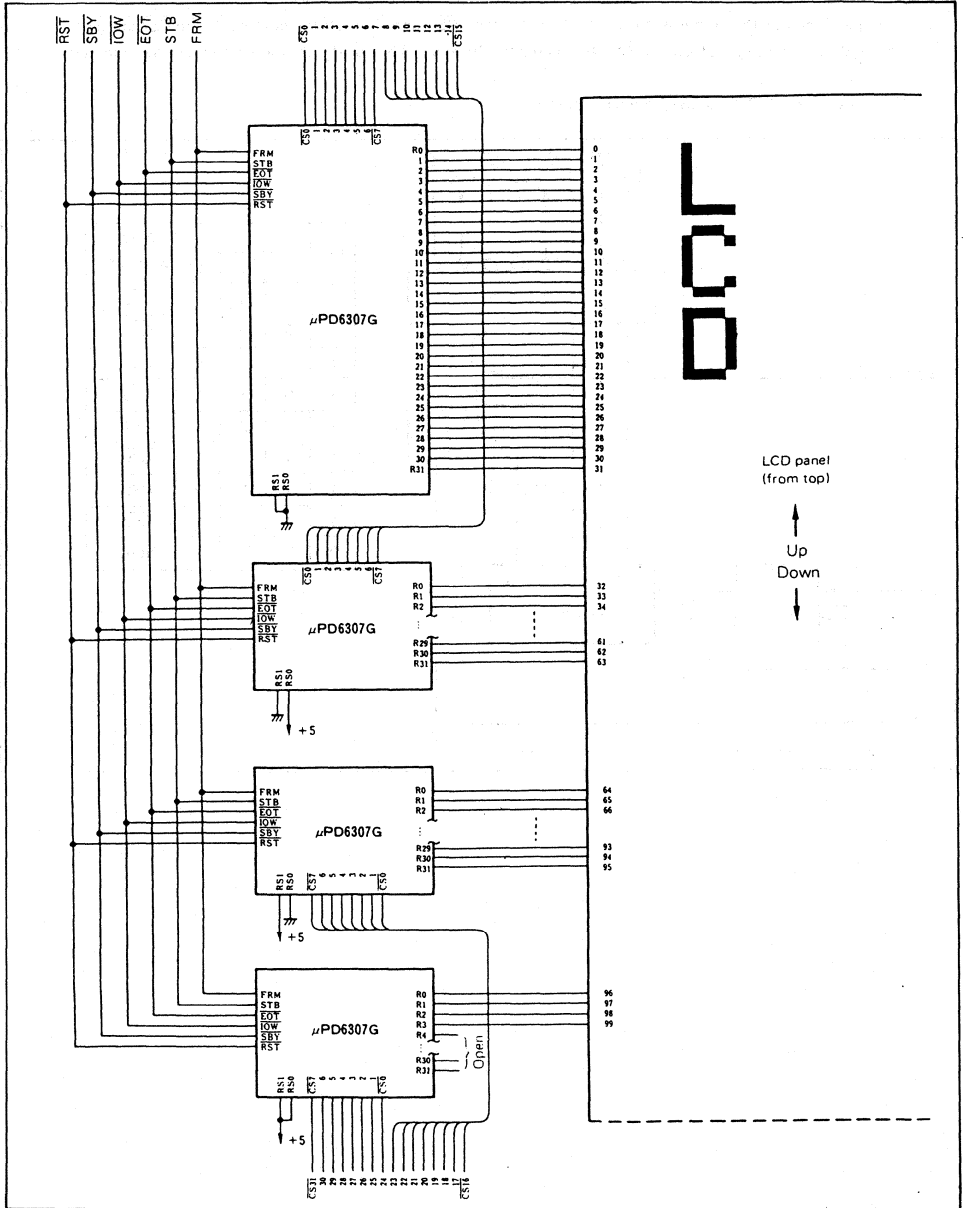




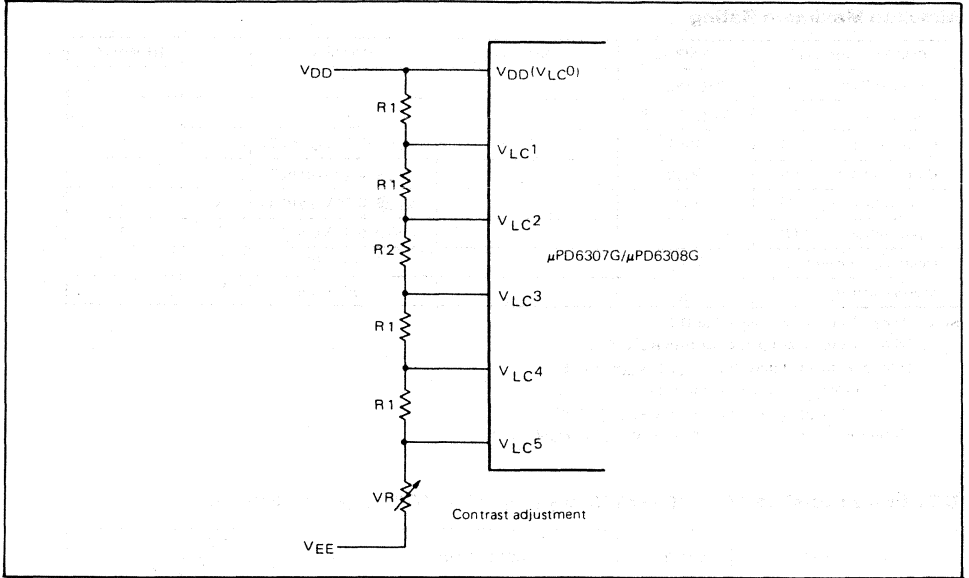
## μPD6307

### 3. Application Example

#### 3.1 Row (common) Signal Driver Circuit (1/100 duty)



### 3.2 Example of LCD Drive Power Supply Circuit (with split resistors)



- NOTE:**
1. R1 and R2 differ depending on the LCD panel to be used. For example, at 1/128 duty, R2 becomes 8 · R1
  2. It is necessary to keep the following relation:  $V_{DD} \geq V_{LC1} \geq V_{LC2} \geq V_{LC3} \geq V_{LC4} \geq V_{LC5} \geq V_{EE}$

**4. Electrical Specifications**

**Absolute Maximum Rating**

CHARACTERISTIC	SYMBOL	CONDITION	RATING	UNIT	REMARK
Supply voltage (1)	$V_{DD}-V_{SS}$		7	V	
Supply voltage (2)	$V_{DD}-V_{LC5}$		23	V	
Output voltage (1)	$V_{O1}$		$V_{LC5}-0.3\sim V_{DD}+0.3$	V	1
Output voltage (2)	$V_{O2}$		$V_{SS}-0.3\sim V_{DD}+0.3$	V	2
Input voltage (1)	$V_{I1}$		$V_{LC5}-0.3\sim V_{DD}+0.3$	V	3
Input voltage (2)	$V_{I2}$		$V_{SS}-0.3\sim V_{DD}+0.3$	V	4
Operating temperature	$T_{opt}$		-10~+70	°C	
Storage temperature	$T_{stg}$		-65~+125	°C	

- Note:** (1) Applies to output pins R0 to R31  
 (2) Applies to output pins other than R0 to R31  
 (3) Applies to input pins  $V_{LC0}$ ,  $V_{LC1}$ ,  $V_{LC4}$ ,  $V_{LC5}$   
 The following relation must be kept  
 $V_{LC0} \geq V_{LC1} \geq V_{LC2} \geq V_{LC3} \geq V_{LC4} \geq V_{LC5}$   
 (4) Applies to input pins other than  $V_{LC0}$ ,  $V_{LC1}$ ,  $V_{LC4}$ ,  $V_{LC5}$

**DC Characteristics** ( $T_a = -10 \sim +70$  °C,  $V_{DD}-V_{SS} = 5 \text{ V} \pm 10 \%$ ,  $V_{DD}-V_{LC5} = 8 \sim 20 \text{ V}$ )

CHARACTERISTIC	SYMBOL	CONDITION	RATING			UNIT
			MIN.	TYP.	MAX.	
High-level input voltage	$V_{IH1}$	Excluding FRM	0.7 $V_{DD}$		$V_{DD}$	V
High-level input voltage	$V_{IH2}$	FRM	3.8		$V_{DD}$	V
Low-level input voltage	$V_{IL}$		$V_{SS}$		0.3 $V_{DD}$	V
High-level output voltage	$V_{OH}$	$\overline{CS}$ , $I_{OH} = -500 \mu\text{A}$	4.0			V
Low-level output voltage	$V_{OL}$	$\overline{CS}$ , $I_{OL} = 500 \mu\text{A}$			0.5	V
Output-on resistor (R0~R31)	$R_{ON}$	Load = $\pm 100 \mu\text{A}$ , $V_{DD}-V_{LC5} = 20 \text{ V}$ $V_{Bias} = \frac{1}{2} (V_{DD}-V_{LC5})$			0.8	k $\Omega$
Input leakage current	$I_{L1}$	$V_{IN} = V_{DD}$ , $V_{SS}$	-3		+3	$\mu\text{A}$
ROW output leakage current	$I_{EL}$	$V_{DD} = 5 \text{ V}$ , $V_{DD}-V_{LC5} = 20 \text{ V}$			5	$\mu\text{A}$
Power consumption	P	No load			5	mW
Input capacity	$C_{IN}$	$V_{DD} = 5 \text{ V}$ , $f = 1 \text{ MHz}$ , $T_a = 25$ °C			4	pF

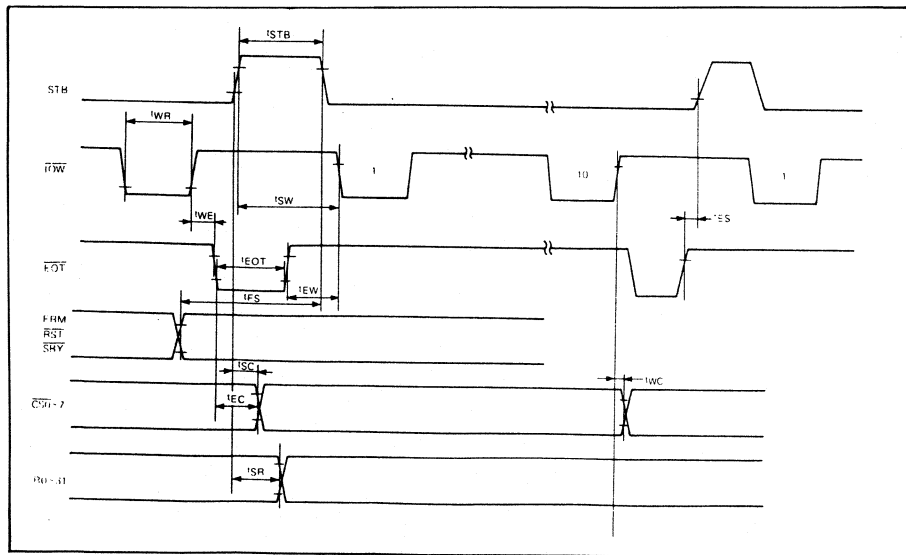
## AC Characteristics (T<sub>a</sub> = -10 ~ +70 °C, V<sub>DD</sub> - V<sub>SS</sub> = 5 V ± 10 %, V<sub>DD</sub> - V<sub>LC5</sub> = 8 ~ 20 V)

CHARACTERISTIC	SYMBOL	CONDITION	RATING			UNIT	RE-MARK
			MIN.	TYP.	MAX.		
Operating frequency	f <sub>C</sub>	IOW cycle			2.5	MHz	1
STB pulse width	t <sub>STB</sub>		250			ns	
FRM, RST, SBY → STB ↓	t <sub>FS</sub>		500			ns	
IOW pulse width	t <sub>WR</sub>		150			ns	
IOW ↑ → EOT ↓	t <sub>WE</sub>		0			ns	
EOT pulse width	t <sub>EOT</sub>		250			ns	
EOT ↓ → CS0 ↓	t <sub>EC</sub>	C <sub>L</sub> = 25 pF			850	ns	
STB ↑ → R0~R31 outputs	t <sub>SR</sub>	No load V <sub>DD</sub> - V <sub>LC5</sub> = 20 V			2.5	μs	
STB ↑ → CS0 ↓	t <sub>SC</sub>	C <sub>L</sub> = 25 pF			330	ns	
IOW ↑ → CS0~CS7 outputs	t <sub>WC</sub>	C <sub>L</sub> = 25 pF	100		850	ns	
STB ↑ → IOW ↓	t <sub>SW</sub>		300			ns	
EOT ↑ → STB ↑	t <sub>ES</sub>		350			ns	
EOT ↑ → IOW ↓	t <sub>EW</sub>		500			ns	

### Note:

(1) When using IOW2 of the LCD controller μPD72030G, use μPD72030G at its operating frequency f<sub>OSC</sub> = 3.0 MHz or lower.

## Timing Waveform



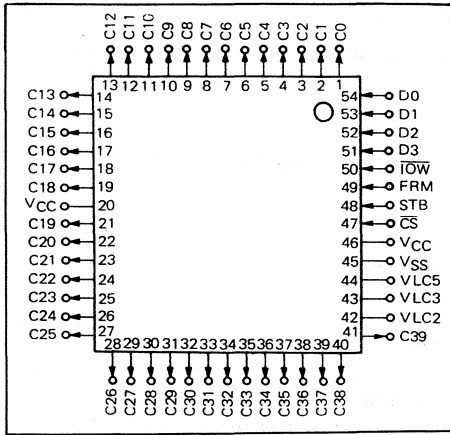


The μPD6308G LCD Column driver can drive multiplexed Dot-matrix LCD. It can directly drives any multiplexed LCD organized as up to 40 columns, and easily cascadable fitting to the user's system.

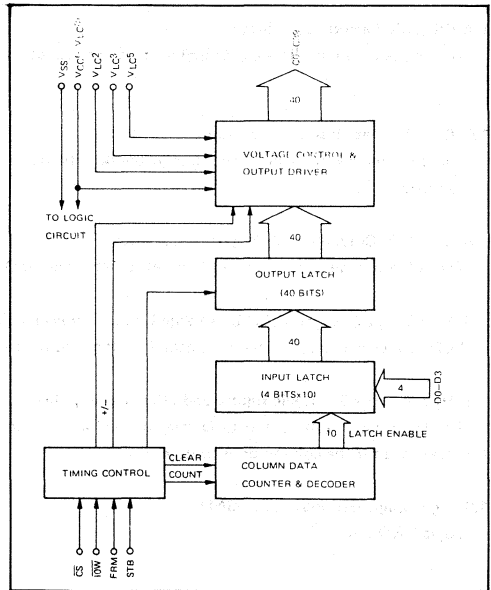
### Features

- High voltage output 21V max
- 40-column (segment) output and cascadable
- Can be directly controlled by the LCD controller, μPD72030G
- CMOS technology
- Single +5V logic power supply
- 54-pin plastic flat package (0.65 mm pitch)

### Pin Connection (Top View)



### μPD6308G Block Diagram



## 1. PIN FUNCTIONS

### 1.1 FRM (Frame) . . . . . input

The frame signal input: "H" displays the positive frame and "L" displays the negative frame.

### 1.2 STB (Strobe) . . . . . input

The column-drive signal strobe input: At the leading edge of the STB input, the 40-bits display data in the input latch is transferred to the output latch to appear in the column-drive output.

### 1.3 $\overline{IOW}$ (I/O Write) . . . . . input

The data-write signal: If the  $\overline{CS}$  is low-level, the data on D0–D3 is written into the input latch at the low-level of  $\overline{IOW}$ .

### 1.4 $\overline{CS}$ (Chip Select) . . . . . input

The chipselect signal: It is connected to the chipselect output of the  $\mu$ PD6307G row driver as the  $\overline{IOW}$ -enable signal.

### 1.5 D0–D3 (Data Input) . . . . . input

The display-data input bus: The data in the 40-bits input latch is written via this data bus 4bits at a time (a total of 10 times).

### 1.6 C0–C39 (Column-drive Output) . . . . . output

The column-drive output, i.e., LCD panel column (segment) line-drive output.

### 1.7 $V_{LC2}$ , $V_{LC3}$ , $V_{LC5}$ (LCD-Drive Supply) . . . . . power supply

LCD reference voltages: These voltages drive the column-drive outputs.

### 1.8 $V_{CC}$ (= $V_{LC0}$ ) (Logic supply and LCD-drive supply) . . . . . power supply

The logic power supply and LCD reference voltage. Connect the power supply (+5 V) between  $V_{CC}$  and  $V_{SS}$ . It is also used for the column-drive voltage input.

### 1.9 $V_{SS}$ (Logic Ground) . . . . . GND

Logic GND pin.

## 2. INTERNAL BLOCK FUNCTIONS

### 2.1 Timing Control Circuit

This circuit controls the timing necessary to operate each internal block.

### 2.2 Column-Data Counter/Decoder Circuit

This decimal counter/decoder circuit generates latch pulses to send to the input-latch circuit which latches 40-bits data, 4bits at a time (total of 10 times). Since one of the decoder output becomes active only when the  $\overline{CS}$  is at a low-level, the number of outputs can be increased by the control of the  $\overline{CS}$  by cascading the  $\mu$ PD6308G chips. The counter value is counted up at leading edge of the  $\overline{IOW}$ , and cleared when the  $\overline{CS}$  goes high-level.



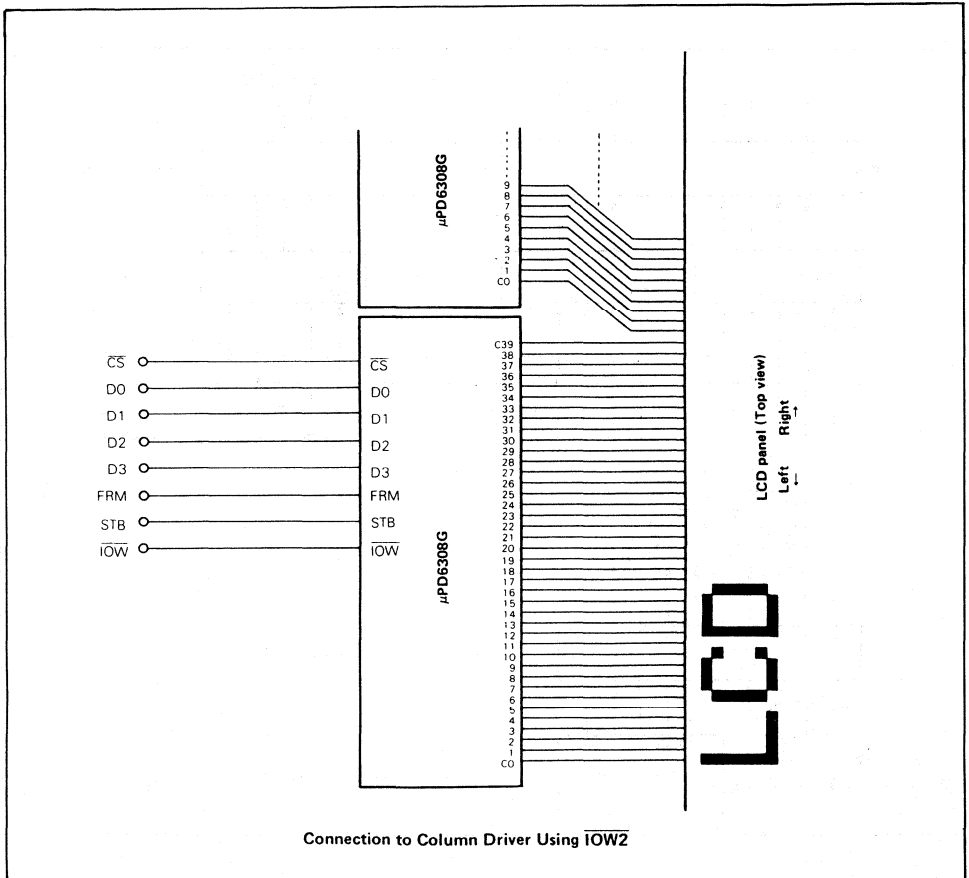


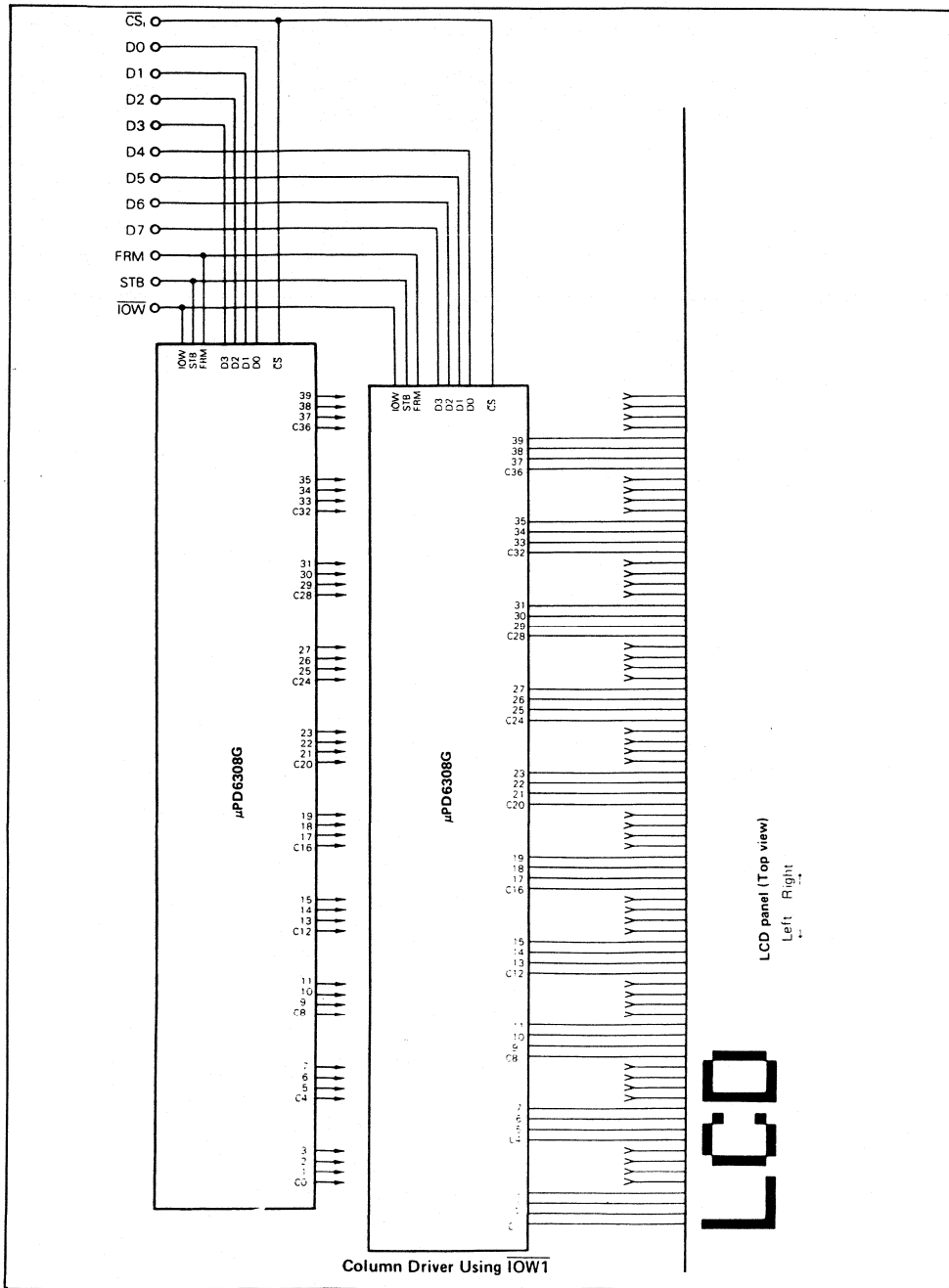
2.5 Voltage Control, Driver Circuit

Generates the column (segment) signals for the AC drive of the LCD panel. The level of outputs to C0–C39 are as follows:

	+ (FRM' = 1)	– (FRM' = 0)
Select (Data = 1)	V <sub>LC5</sub>	V <sub>LC0</sub>
Nonselect (Data = 0)	V <sub>LC3</sub>	V <sub>LC2</sub>

FRM' is obtained by internally synchronizing the FRM signal with the leading edge of the STB signal.





Column Driver Using IOW1

LCD panel (Top view)  
Left Right





## Description

The NEC μPD7281 Image Pipelined Processor is a high-speed digital signal processor specifically designed for digital image processing such as restoration, enhancement, compression, and pattern recognition. The μPD7281 employs token-based data-flow and pipelined architecture to achieve a very high throughput rate. A high-speed on-chip multiplier speeds calculations. More than one μPD7281 can easily be cascaded with a minimum amount of interface hardware to increase the throughput rate even further. The μPD7281 is designed to be used as a peripheral processor for minicomputers or microcomputers, thereby relieving the host processor from the burden of time-intensive computations. The μPD7281 has a very powerful instruction set designed specifically for digital image processing algorithms. The Image Pipelined Processor can also be used as either a general purpose digital signal processor or a numeric processor.

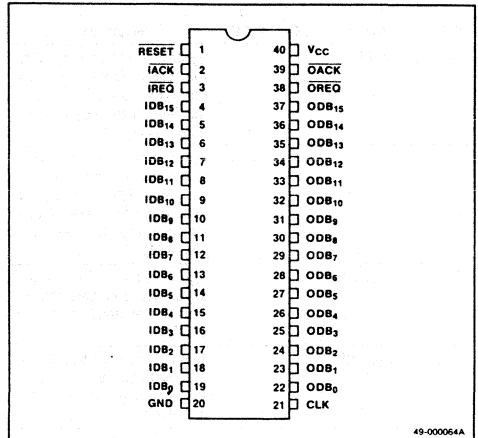
## Features

- Token-based data-flow architecture
- Internal pipelined ring architecture
- Powerful instruction set for image processing
- 17 x 17-bit (including sign bits) fast multiplier: 200 ns (target spec)
- High-speed data I/O handling
  - Asynchronous two-wire handshaking protocols
  - Separate data input and output pins
- Easy multiple-processor configuration
- Rewritable program stores
- On-chip memories:
  - Link Table (LT): 128 x 16 bits
  - Function Table (FT): 64 x 40 bits
  - Data Memory (DM): 512 x 18 bits
  - Data Queue (DQ): 32 x 60 bits
  - Generator Queue (GQ): 16 x 60 bits
  - Output Queue (OQ): 8 x 32 bits
- NMOS technology
- Single +5 V power supply
- 40-pin DIP

## Applications

- Digital image restoration
- Digital image enhancement
- Pattern recognition
- Digital image data compression
- Radar and sonar processing
- Fast Fourier Transforms (FFT)
- Digital filtering
- Speech processing
- Numeric processing

## Pin Configuration



## Performance Benchmarks

(Subject to change without notice.)

Operation	1 μPD7281	3 μPD7281s	Note
Rotation	1.5 sec	0.6 sec	512 x 512 binary image
1/2 Shrinking	80 ms	30 ms	512 x 512 binary image
Smoothing	1.1 sec	0.4 sec	512 x 512 binary image
3x3 Convolution	3.0 sec	1.1 sec	512 x 512 grey scale image
64-stage FIR Filter	50 μs	18 μs	17-bit fixed point
cos(x)	40 μs	15 μs	33-bit fixed point

**Pin Identification**

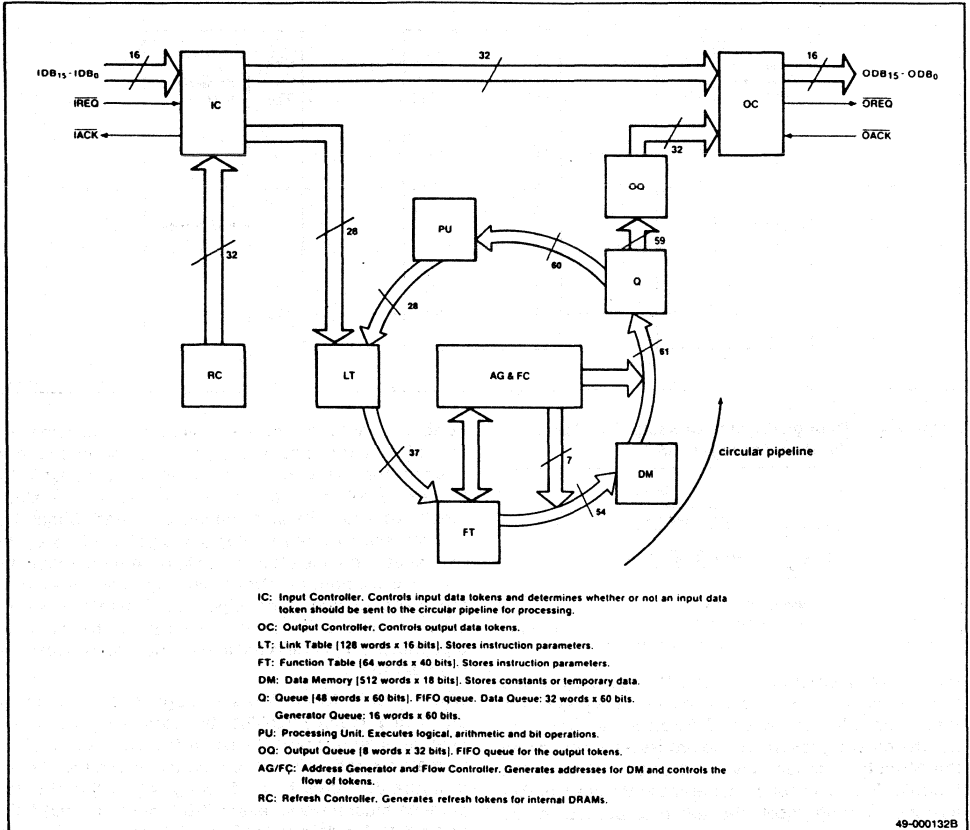
No.	Signal	I/O	At RESET	Description
1	RESET	In		System Reset: A low signal on this pin initializes μPD7281. During the reset, a 4-bit module number should be placed on IDB <sub>15</sub> - IDB <sub>12</sub> .
2	IACK	Out	High	Input Acknowledge: This acknowledge signal is output by the μPD7281 to notify the external data source that a 16-bit data transfer has been completed.
3	IREQ	In		Input Request: This input signal requests a data transfer from an external device to μPD7281.
4-19	IDB <sub>15</sub> - IDB <sub>0</sub>	In		16-bit input data bus: 32-bit input data tokens are input to the Input Controller as two 16-bit words.
20	GND			Power ground
21	CLK	In		System clock input (10 MHz: target spec)
22-37	ODB <sub>15</sub> - ODB <sub>0</sub>	Out	High Impedance	16-bit output data bus: 32-bit output data tokens are output by the Output Controller as two 16-bit words.
38	OREQ	Out	High	Output Request: This signal informs an external device that a 16-bit data word is ready to be transferred out of μPD7281.
39	OACK	In		Output Acknowledge: This acknowledge signal input by the external data destination notifies μPD7281 that a 16-bit data transfer may occur.
40	V <sub>CC</sub>			+5 V power supply

**Architecture**

The μPD7281 utilizes a token-based, data-flow architecture. This novel architecture not only provides multiprocessing capability without complex external hardware, but also offers high computational efficiency within each processor. Taking advantage of the multi-processing capability of data-flow architecture, almost any processing speed requirements can be satisfied by using as many μPD7281s as needed in the system. Within each μPD7281, the data-flow architecture provides high computational efficiency through concurrent operations. For example, while the Processing Unit (or ALU) spends its time for actual computations only, the internal memory address calculations, internal memory read and write operations and input/output operations are all being done concurrently. Furthermore, in contrast to conventional von Neumann processors, a data-flow processor doesn't fetch instructions, perform subroutine stack operations or do data transfers between registers. Therefore, it does not spend the time required for these operations.

The μPD7281 also utilizes an internally pipelined architecture. As shown in the block diagram, a circular pipeline is formed by five functional blocks: the Link Table (LT), the Function Table (FT), the Data Memory (DM), the Queue (Q), and the Processing Unit (PU). A token entered through the Input Controller (IC) is passed on to the Link Table to be processed around the pipelined ring as many times as needed. When a token is finished being processed, it is queued into Output Queue (OQ) and then output via the Output Controller (OC).

## Block Diagram



## Functional Description

As shown in the block diagram, the μPD7281 consists of 10 functional blocks. Before any processing occurs, the host processor down-loads the object code into the Link Table and the Function Table of the μPD7281 by using specially formatted input tokens. At this time, constants may also be sent to the Data Memory to be stored. The contents of the Link Table and the Function Table are closely related to a computational graph. When a computational process is represented graphically, it usually forms a directed data-flow graph. In such a graph, the arcs (or edges, links, etc.) represent the entries in the Link Table and the nodes

represent the entries in the Function Table. An arc between any two nodes has a data value, called a "token", and is identified by a corresponding entry in the Link Table. A node in the directed data-flow graph signifies an operation, and the type of operation is logged into the Function Table along with the identification information about the outgoing arc.

A minimal amount of interface hardware is required to configure μPD7281s in a multiprocessor system. As many as 14 μPD7281s can be cascaded together, as shown in figure 1. Each μPD7281 must be assigned a Module Number (MN) during reset. Figure 2 shows the timing diagram for assigning the module number.

Figure 1. Connecting Multiple μPD7281s

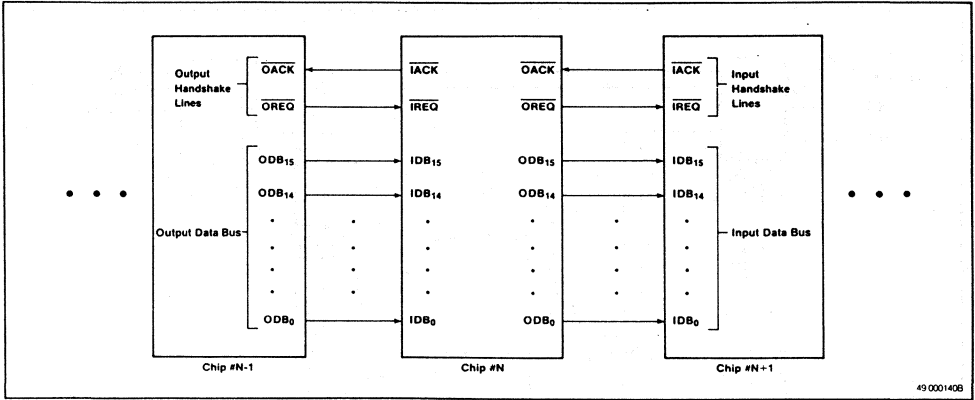
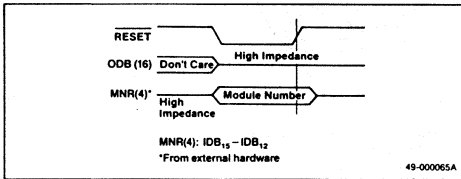


Figure 2. Timing Diagram for Assigning Module Numbers During RESET



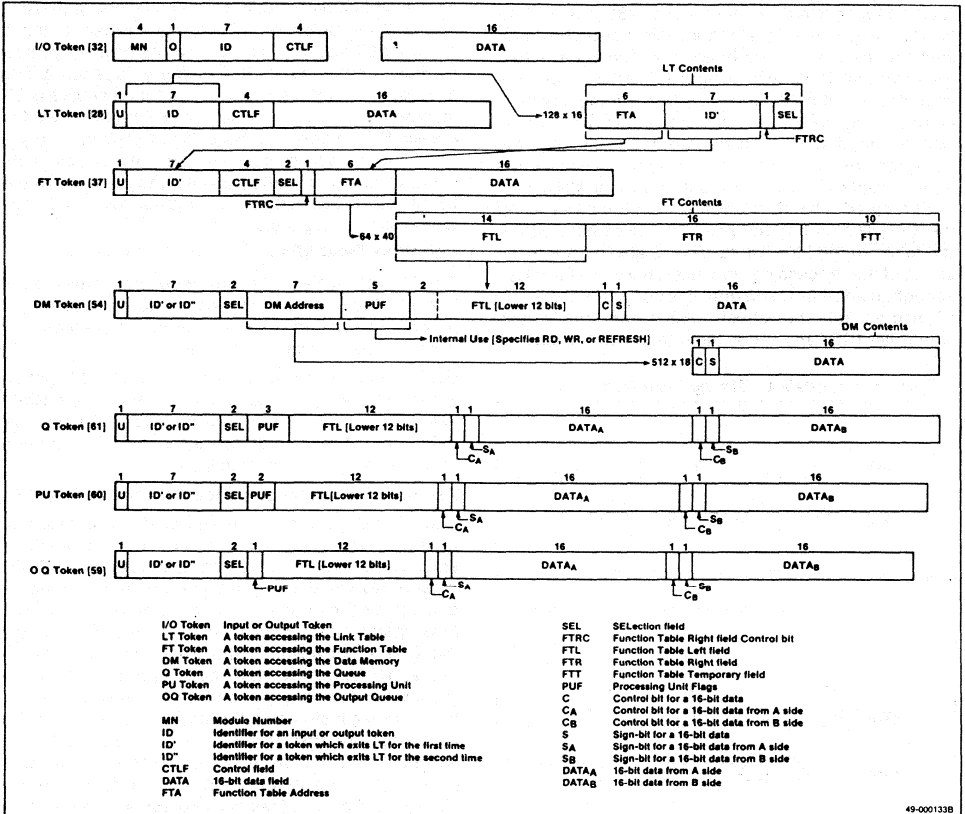
When any token enters a μPD7281, regardless of the total number of μPD7281s used in the system, the Input Controller of that μPD7281 discerns whether or not the entering token is to be processed by checking the Module Number (MN) field of the token. If the Module Number is not the same as the Module Number assigned during reset, the token is passed to the Output Controller so that it can be sent out via the Output Data Bus. However, if the token has the same Module Number, then the Input Controller strips off the MN field and sends the remaining part of the token to the Link Table for processing.

Once a token enters the circular pipeline by accessing the Link Table, it requires seven pipeline clock cycles for the token to fully circulate around the ring. One pipeline clock cycle is needed for the Link Table, the

Function Table, or the Data Memory to process an incoming token, and two pipeline clock cycles are needed for the Queue or the Processing Unit to process a token. The Queue requires one pipeline clock cycle to write and one cycle to read. Similarly, the Processing Unit requires one pipeline clock cycle to execute and one clock cycle to output the result. In other words, both the Processing Unit and the Queue are made of two-stage pipelines. Therefore, when seven tokens exist simultaneously in the circular pipeline, the pipeline is full and full parallel processing is achieved.

When a data token flows through each functional block in a given μPD7281, the format of the token changes significantly. The actual transitions of a token format through different functional blocks are shown in figure 3. A data token flowing within the circular pipeline must have at least a 7-bit Identifier (ID) field and an 18-bit data field. The ID field is used as an address to access the Link Table memory. When a token accesses the LT memory, the ID field of the token is replaced by a new ID (shown as ID' in figure 3) previously stored in the LT memory. As a result, every time a data token accesses LT memory, its ID field is renewed. The data field of a token consists of a control bit, a sign bit and a 16-bit data. A token may have up to two data fields, as well as other fields (OP code, control, etc.) if necessary.

Figure 3. Token Formats and Transitions



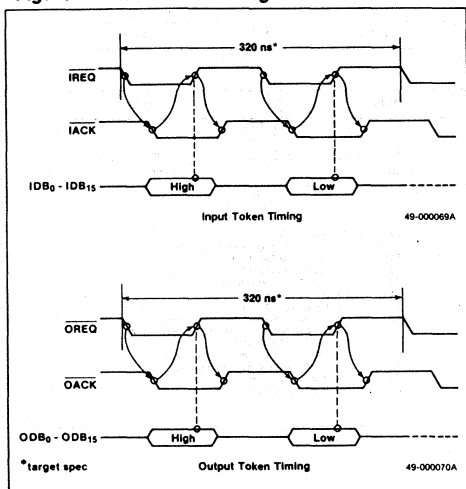
49-000133B



**Input Controller [IC]**

A 32-bit token is entered into a μPD7281 in two 16-bit halves using a two-signal request/acknowledge handshake method, as shown in figure 4. The input/output token format is shown in figure 7. After a token is accepted by the IC, the MN field of the token is compared to the Module Number of μPD7281 which was assigned at reset. If the Module Number of the accepted token is not the same, the IC passes the token directly to the Output Controller. If the MN field of the accepted token is the same, then the IC strips off the Module Number and sends the remaining part of the token to the Link Table. The IC also monitors the status of the Processing Unit. If it is busy, the IC delays accepting another token until it is no longer busy. The IC also accepts the refresh tokens from the Refresh Controller (RC) and sends them to the Link Table.

Figure 4. Handshake Timing Waveforms



**Output Controller [OC]**

The OC outputs 32-bit tokens in two 16-bit halves using a two-signal request/acknowledge handshake method, as shown in figure 4. The types of tokens output by the OC are as follows: output data tokens from the Output Queue, error status data tokens generated internally by OC, DUMP tokens, and passing data tokens from the Input Controller.

**Link Table [LT]**

The LT is a 128 x 16-bit dynamic RAM. The ID field of an incoming LT token is used to access the LT memory. The contents of an LT memory location

consist of a 6-bit Function Table Address (FTA), a 7-bit ID, a 1-bit Function Table Right Field Control (FTRC), and a 2-bit Selection (SEL) field. When a token accesses LT memory, its ID field is replaced by the new ID field contained in the memory location being accessed. Therefore, every time a token accesses LT memory, it is given a new ID. The FTA field is used to access FT memory locations. The FTRC bit and the SEL field are used to specify the type of instruction. By using specially formatted tokens, the contents of the LT can either be set during a program download or be read during a diagnosis.

**Function Table [FT]**

The FT is a 64 x 40-bit dynamic RAM. As for the case of the Link Table, the contents can either be set during a program download or be read during a diagnosis by using specially formatted tokens.

Each FT memory location consists of a 14-bit Function Table Left field (FTL), a 16-bit Function Table Right field (FTR), and a 10-bit Function Table Temporary field (FTT). These fields contain control information for different types of instructions.

**Address Generator and Flow Controller [AG/FC]**

The AG/FC generates the addresses to access the Data Memory (DM) and controls the writing of data to and the reading of data from the Data Memory. AG/FC determines whether the incoming token contains a one-operand instruction or a two-operand instruction. One-operand instruction tokens can be sent directly to the Queue. However, if the token contains a two-operand instruction, then both operands must be available before they can be sent to the Queue. For a two-operand instruction, the token which arrives at the Data Memory first is temporarily stored until the second operand token arrives. When the second operand token exits the Function Table, the AG/FC generates the Data Memory address which contains the first operand. Then, the second operand token and the first operand data read out from the Data Memory are sent to the Queue together.

**Data Memory [DM]**

The DM is a 512 x 18-bit dynamic RAM which is used to queue the first operand for a two-operand instruction until the second operand arrives. DM can also be used as a temporary memory or as a buffer memory for I/O data.

**Queue [Q]**

The Q is a FIFO memory configured with a 48 x 60-bit dynamic RAM. The Q is used to temporarily store the Processing Unit-bound and the Output Queue-bound tokens. The Q is further divided into two different FIFO memories: a 32 x 60-bit Data Queue (DQ) and a 16 x 60-bit Generator Queue (GQ). The DQ is used for the

PU, OUT and AG/FC instructions. The DQ temporarily stores the PU and AG/FC tokens before they are sent to the Processing Unit for processing. The DQ also temporarily stores the Output Queue tokens before they are sent to the Output Queue. The GQ is used for Generate (GE) instructions only. The DQ will not output tokens to the Output Queue if it is full, and the DQ or GQ will not output tokens to the Processing Unit if the Processing Unit is busy.

In order to control the number of tokens in the circular pipeline to prevent Q overflow, the Q is further restricted by the following two situation rules: when the DQ has eight or more tokens stored, the read from the GQ is inhibited, and when the DQ has fewer than eight tokens stored, the read from the GQ has a higher priority than the read from the DQ. Since instructions stored in the GQ generate tokens, restricting the number of GQ tokens is important in order to keep the Q from overflowing. In case the internal processing speed is slower than the rate of incoming data tokens, the DQ possesses a potential overflow condition. To prevent overflow, the processor is put into restrict/inhibit mode when the DQ reaches a level greater than 23.

### Output Queue [OQ]

The OQ is a first-in first-out (FIFO) memory configured in an 8 x 32-bit static RAM. The OQ is used to temporarily store the output data tokens from the Data Queue so that they can be output by the Output Controller via the output data bus. When OQ is full, it sends a signal to the Data Queue to delay accepting further tokens.

### Processing Unit [PU]

The PU executes two types of instructions: PU and GE. PU instructions include logical, arithmetic (add, subtract and multiply), barrel-shift, compare, data-exchange, bit-manipulation, bit-checking, data-conversion, double-precision adjust, and other operations. The control information for a PU instruction is contained in the Function Table Left field of the PU token. The GE instructions are used to generate a new token, multiple copies of a token, or block copies of a token. They can also be used to set the Control field (CTLF) of a token and to generate external memory addresses. If the current PU operation cannot be completed within a pipeline clock cycle, the PU sends a signal to the

Queue and the Input Controller to prevent them from releasing any more tokens.

### Refresh Controller [RC]

The RC automatically generates refresh tokens for the dynamic RAMs used in the circular pipeline, i.e. the LT, FT, DM, and Q. Each RC token, generated periodically, is sent to the Input Controller and is propagated through the LT, FT, DM and Q, in that order. The RC tokens are deleted after reaching the Q.

### Operation Modes

There are three different modes in which the μPD7281 can operate: Normal, Test, and Break (see figure 5). After an external hardware reset, the μPD7281 is in the Normal mode of operation. The μPD7281 can enter the Test mode for program debugging by inputting a SETBRK token (see figure 6) while the processor is in the Normal mode. If an overflow occurs in the Data Queue or the Generator Queue, the processor enters into the Break mode so that the internal contents of the processor can be examined; see table 1. Table 2 describes the effects of software and hardware resets.

**Table 1. DUMPD Output Token Format**

MN	Z	ID	CTLF	DATA (16-bit field)
0000	0	0000 000	0111	xxxxx(5) GQ Size(5 bits) DQ Size(6 bits)
0000	0	0000 001	0111	xxxx(4) u(1) ID(7) CTLF(4)
0000	0	0000 010	0111	DATA(16)
0000	0	0000 011	0111	xxx (3) u(1) ID(7) x(1) C <sub>B</sub> , S <sub>B</sub> , C <sub>A</sub> , S <sub>A</sub>
0000	0	0000 100	0111	xx(2) FTL (Lower 12 bits) xx(2)
0000	0	0000 101	0111	DATA <sub>A</sub> (16)
0000	0	0000 110	0111	DATA <sub>B</sub> (16)
0000	0	0000 111	0111	xxxxxxxxx(9) ID(7)

x: Don't care u: Unused

**Table 2. Effects of Reset Operation**

	Hardware Reset	Software Reset
MN	μPD7281 reads in MN	No Change
High/Low Word Flip-flop	Reset	No Change
Input Inhibit Control	Reset (No constraint)	No Change
LT Break State	Reset	Reset
Internal Operation	Stopped	Stopped
DQ, GQ, and OQ Pointers	Set to 0	Set to 0

Figure 5. μPD7281 Operation Modes

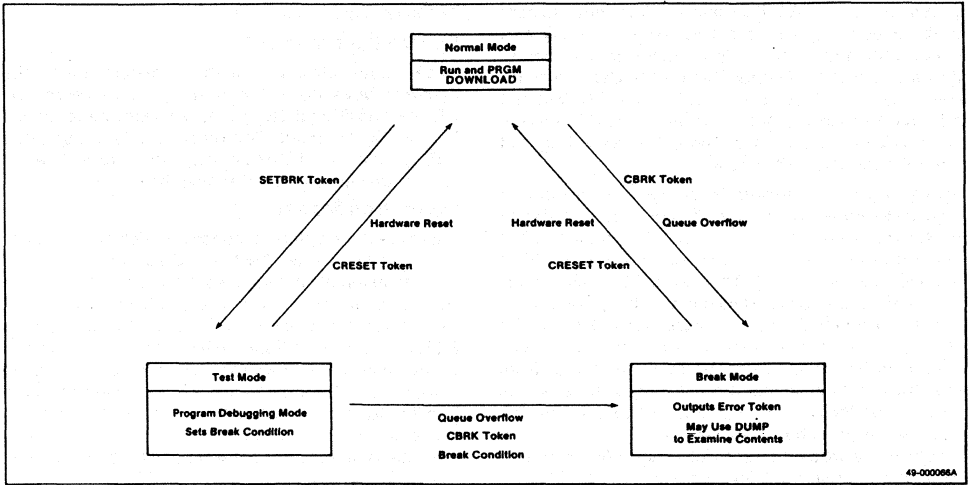
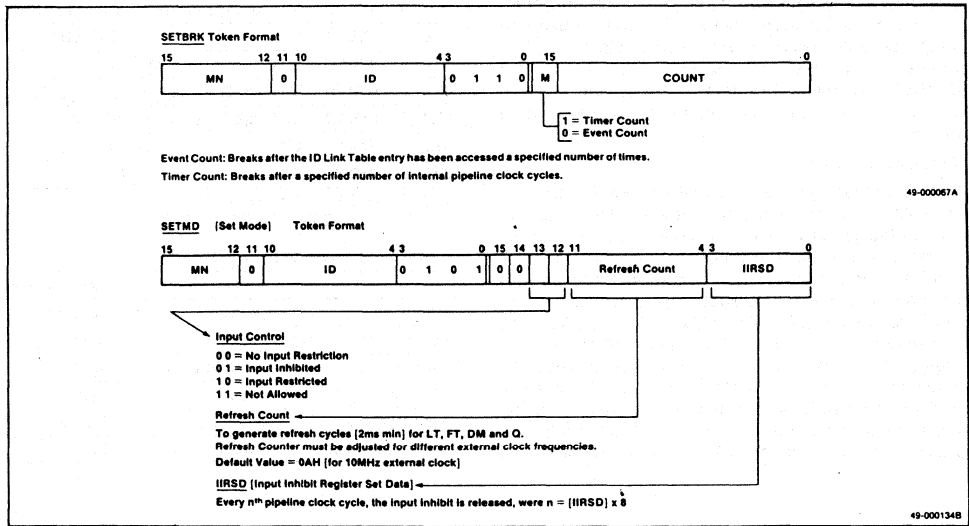


Figure 6. SETBRK (Set Break Condition) and SETMD (Set Mode) Token Formats



## Input/Output Tokens

The only way any external device can communicate with the μPD7281 is by using the I/O tokens (see figure 7). Both the input and the output tokens have the same format so that a token may flow through a series of multiple processors without a format change. A 32-bit I/O token is divided into upper and lower 16-bit words and input to or output from the μPD7281 a 16-bit word at a time. Object code is down-loaded into the Link

Table and the Function Table using SETLT, SETFTR, SETFTL and SETFTT input tokens. The contents of the Function Table and the Link Table can also be read using RDLT, RDFTR, RDFTL and RDFTT tokens. In order to write or read a value to and from the Data Memory, a program must be down-loaded and executed. Once object code is down-loaded into the μPD7281, data tokens are input to the processor, thereby initiating the processing. For a description of the input and output tokens, see tables 3 and 4.

Figure 7. Input/Output Token Format

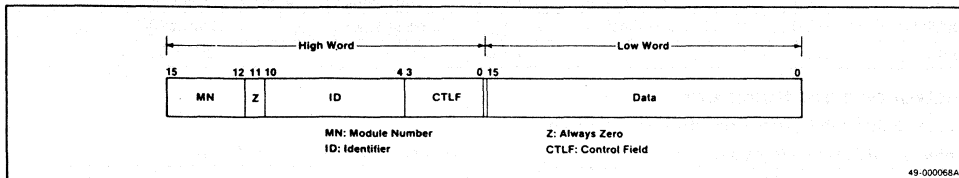


Table 3. Input Token Format

Input Token	High Word (16)							Low Word (16)							Remarks															
	MN (4)				Z (1)		ID (7)			CTLF (4)				DATA (16)																
	15	12	11	10	9	8	7	6	5	4	3	2	1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SETLT	MN	0					LT address		1	1	0	0		Data to be set in LT																Set LT
SETFTR	MN	0					FT address		1	1	0	1		Data to be set in FTR																Set FT Right Field
SETFTL	MN	0					FT address		1	1	1	0		Data to be set in FTL																Set FT Left Field
SETFTT	MN	0					FT address		1	1	1	1		Data to be set in FIT																Set FT Temporary Field
RDLT	MN	0					LT address		1	0	0	0																		Read LT
RDFTR	MN	0					FT address		1	0	0	1																		Read FT Right Field
RDFTL	MN	0					FT address		1	0	1	0																		Read FT Left Field
RDFTT	MN	0					FT address		1	0	1	1																		Read FT Temporary Field
CRESET	MN	0							0	1	0	0																		Command Reset
SETMD	MN	0							0	1	0	1		Mode set data																Set Operation Mode
SETBRK	MN	0					ID		0	1	1	0		M (1)		Count (15)														Set Break Condition
DUMP	MN	0					xxxx(4)	DUMP (3)	0	1	1	1																		Dump
CBRK	0	0	0	0	0				0	1	0	0																		Command Break
VAN	1	1	1	1	0																									Vanish Data
PASS	MN*	0																												Pass Data
EXEC	MN	0					ID		0	0	C	S		Data															Normal Execution Data	

\* When MN is not the current module number  
 x: Don't care

**Table 4. Output Token Format**

Output Token	Upper-Order Word (16)				Lower-Order Word (16)				Remarks
	MN (4) 15 12	Z (1) 11	ID (7) 10		CTLF (4) 4 3	DATA (16) 0 15	0		
LTRDD	0 0 0 0	0	LT address		1 0 0 0	Data read from LT		FT Read Data	
FTRRDD	0 0 0 0	0	FT address		1 0 0 1	Data read from FTR		FT Right Field Read Data	
FTLRDD	0 0 0 0	0	FT address		1 0 1 0	Data read from FTL		FT Left Field Read Data	
FTTRDD	0 0 0 0	0	FT address		1 0 1 1	Data read from FTT		FT Temporary Field Read Data	
PASSD	MN	0	ID		CTLFD	Data		Pass Data	
ERR	0 0 0 0	0	0 0 0 0 0 0 0 0		0 1 0 0	MN(4)/MODE(4) 0 0 0 STATUS(5)		Error Data	
DUMPD	0 0 0 0	0	0 0 0 0 0 DUMP(3)		0 1 1 1	Dump data		Dumped Data	
OUTD	MN	0	ID		0 0 C S	Data		Output Data	

### Instruction Set Summary

Tables 5 through 8 summarize the instruction set.

**Table 5. AG/FC Instructions**

Mnemonic	Instruction
QUEUE	Queue
RDCYCS	Read cyclic short
RDCYCL	Read cyclic long
WRCYCS	Write cyclic short
WRCYCL	Write cyclic long
RDWR	Read/Write Data Memory
RDIIX	Read Data Memory with index
PICKUP	Pickup data stream
COUNT	Count data stream
CONVO	Convolve
CNTGE	Count generation
DIVCYC	Divide cyclic
DIV	Divide
DIST	Distribute
SAVE	Save ID
CUT	Cut data stream

**Table 6. PU Instructions**

Mnemonic	Instruction
OR	Logical OR
AND	Logical AND
XOR	Logical EXCLUSIVE-OR
ANDNOT	Logical INVERT an operand then AND: ( $\bar{A} \cdot B$ )
NOT	Invert
ADD	Add
SUB	Subtract

**Table 6. PU Instructions (cont)**

Mnemonic	Instruction
MUL	Multiply
NOP	No operation
ADDSC	Add and shift count
SUBSC	Subtract and shift count
MULSC	Multiply and shift count
NOPSC	NOP and shift count
INC	Increment
DEC	Decrement
SHR	Shift right
SHL	Shift left
SHRBRV	Shift right with bit reverse
SHLBRV	Shift left with bit reverse
CMPNOM	Compare and normalize
CMP	Compare
CMPXCH	Compare and exchange
GET1	Get one bit
SET1	Set one bit
CLR1	Clear one bit
ANDMSK	Mask a word with logical AND
ORMSK	Mask a word with logical OR
CVT2AB	Convert 2's complement to sign-magnitude
CVTAB2	Convert sign-magnitude to 2's complement
ADJL	Adjust long (for double precision numbers)
ACC	Accumulate
COPYC	Copy control bit

**Table 7. GE Instructions**

Mnemonic	Instruction
COPYBK	Copy block
COPYM	Copy multiple
SETCTL	Set control field

**Table 8. OUT Instructions**

Mnemonic	Instruction
OUT1	Output 1 token
OUT2	Output 2 tokens

There are four different types of instructions which can be specified by the SEL field of an FT token. See table 9.

**Table 9. SEL Field of an FT Token**

SEL Type	Description
11 AG/FC	Executes instructions specified by the Function Table Right field while monitoring the Function Table Temporary field.
01 PU	Performs arithmetic, logical, barrel-shift, bit-manipulation, data-conversion, etc.
10 GE	Generates a block or multiple new tokens from a token. Sets the control field of a token. Increments or decrements the data field of a token.
00 OUT	Outputs data tokens from the circular pipeline to the Output Queue after the tokens are finished being processed.

## AG/FC Instructions

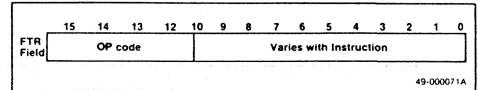
There are 16 AG/FC instructions (see table 10). They can be grouped into three types: Address Generator (AG), Flow Controller (FC), and AG/FC type.

AG type: RDCYCS, RDCYCL, WRCYCS, WRCYCL, RDWR, RDIDX

FC type: PICKUP, COUNT, CUT, DIVCYC, DIV, DIST, CONVO, SAVE, CNTGE

AG/FC type: QUEUE

A 4-bit OP code in the Function Table right field specifies the instruction to be executed.

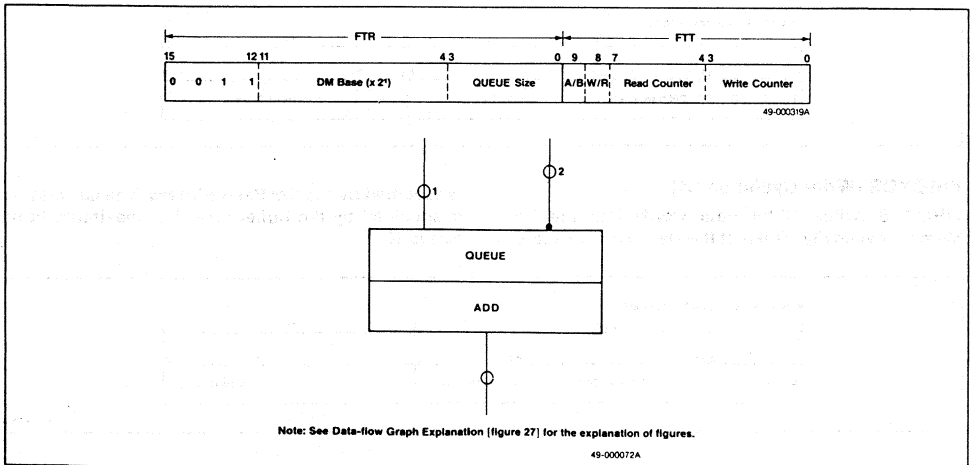


## QUEUE

For a two-operand instruction, a QUEUE instruction is used to temporarily store the first operand token in the Data Memory until the second operand token arrives. The maximum Queue size is 16. See figure 8.

5

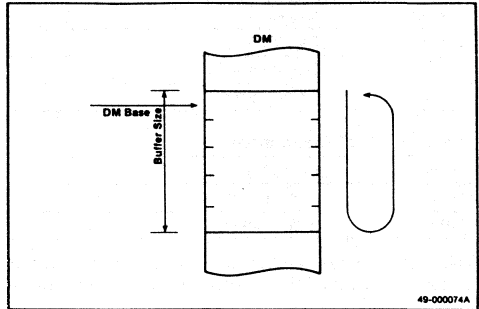
**Figure 8. QUEUE Instruction**



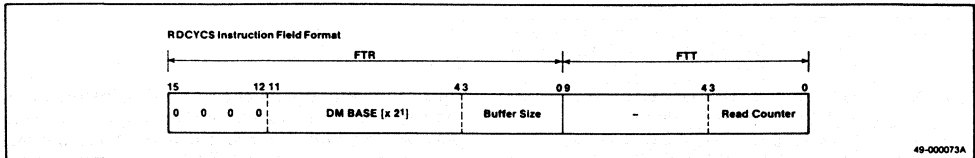
**RDCYCS [Read Cyclic Short]**

RDCYCS reads 18-bit data words from the Data Memory cyclically (see figure 9). The first data to be read is specified by the DM Base address. The last data to be read is specified by the buffer size. The Read Counter (RC) contains the offset address from Data Memory Base (DMB) address. It is incremented each time the Data Memory is accessed. The maximum buffer size is 16.

Figure 9. RDCYCS Instruction Operation



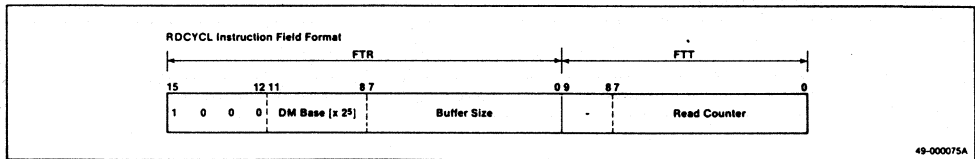
49-00074A



**RDCYCL [Read Cyclic Long]**

RDCYCL reads 18-bit data words from the Data Memory in a cyclic manner like RDCYCS but has a longer cyclic

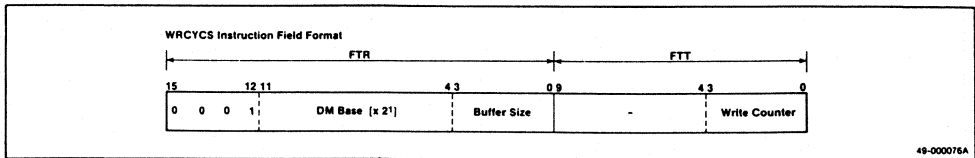
range. The first data to be read is specified by the DM Base address. The last data to be read is specified by the buffer size. The maximum buffer size is 256.



**WRCYCS [Write Cyclic Short]**

WRCYCS writes 18-bit data words into the Data Memory cyclically. The first the Data Memory address

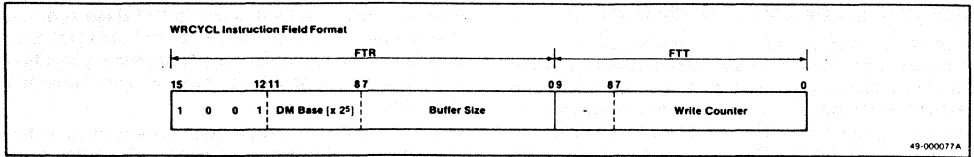
is specified by the DM Base address. The last address is specified by the buffer size. The maximum buffer size is 16.



**WRCYCL [Write Cyclic Long]**

WRCYCL writes 18-bit data words into the data memory in a cyclic manner similar to WRCYCS but has a longer

cyclic range. The first DM address is specified by the DM Base address. The last address is specified by the buffer size. The maximum buffer size is 256.



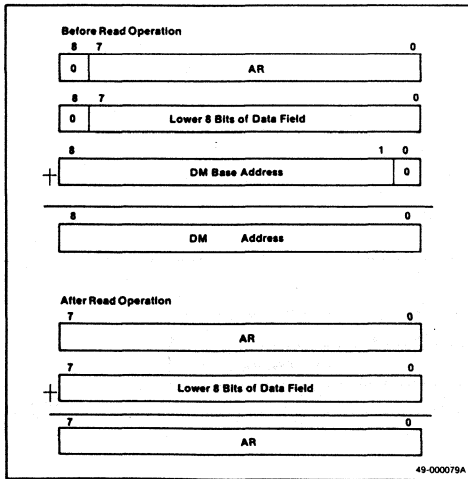
### RDWR [Read/Write Data Memory]

RDWR is used to write or read data to and from the Data Memory. This instruction reads/modifies/writes the Data Memory with the Address Register as index.

If a token arriving at the instruction has FTRC bit = 0, then the instruction performs a DM read operation. If it has FTRC bit = 1, then the instruction performs a DM write operation.

For a token with the FTRC bit = 0, the actual DM address location to be read is determined by the sum of the following three values: 8-bit Address Register (AR),

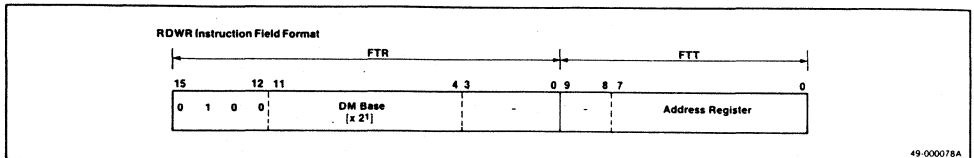
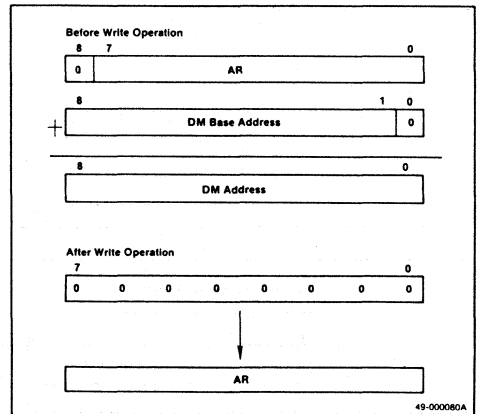
**FTRC = 0**



the lower eight bits of the data field of the token, and the DM Base address. After the read operation, the lower eight bits of the token's data field is added to the value of AR. Additionally, the data field of the token is replaced by the contents read from the Data Memory location.

If a token with FTRC bit = 1 is used along with RDWR, a write operation is performed. The Data Memory address location is determined by the sum of 8-bit AR and DM Base address. The 18-bit data from the token is written into the DM address calculated. After the write operation, AR is reset to 00H.

**FTRC = 1**





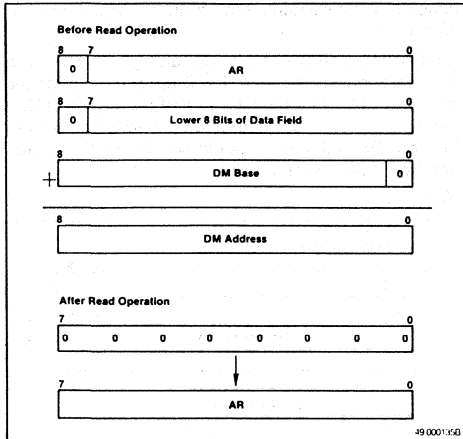
**RDIDX [Read Data Memory with Index]**

RDIDX is used to read the contents of the Data Memory. This instruction is most useful when a part of the Data Memory is used as a look-up table. The RDIDX instruction performs different operations depending upon the FTRC bit of the token using the instruction. If the FTRC bit = 0, then the instruction reads a Data Memory location. The DM address location to be read is determined by the sum of the following three values: the 8-bit AR, the lower eight bits

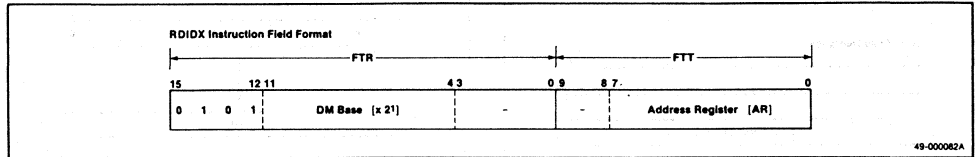
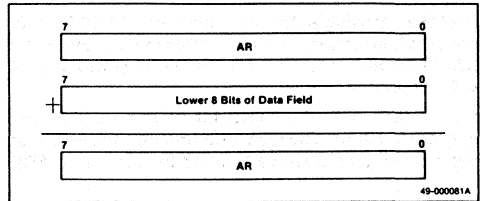
of the token's data field, and the DM Base address. After the read operation, the data field of the token is replaced by the contents of the Data Memory location read. The value of AR is reset to zero after the operation.

If the FTRC bit = 1, no operation is performed on the Data Memory. However, the token's AR contents are replaced by the modulo-256 sum of the lower eight bits of data field and the current contents of AR.

**FTRC = 0**



**FTRC = 1**



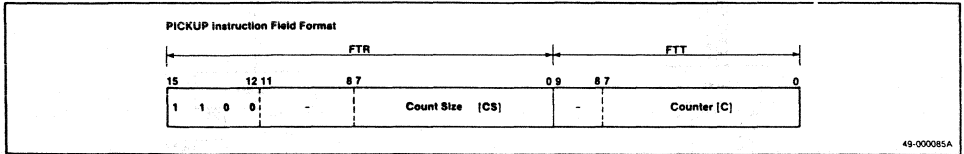
### PICKUP [Pickup Data Stream]

This instruction picks up every  $(n+1)^{th}$  token from a stream of incoming tokens and increments the  $(n+1)^{th}$  token's ID field by one. The number  $n$  is specified by the Count

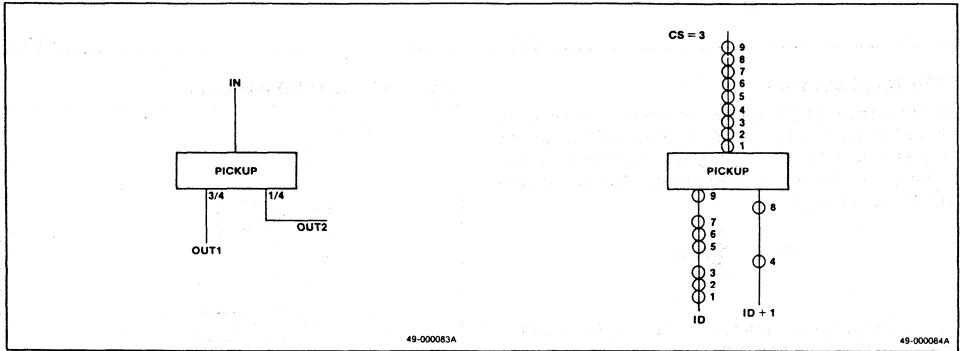
Size (CS) of the Function Table Right field.

Figure 10 illustrates the PICKUP instruction with CS = 3.

**Note:** These figures use the data-flow graph convention. See figure 27, Data-flow Graph Explanation for the explanation of figures.



**Figure 10. Pickup Instruction**



### COUNT [Count Data Stream]

COUNT copies every  $(n+1)^{th}$  token from a stream of incoming tokens and increments the copied token's ID

field by one. The number  $n$  is specified by CS of the Function Table Right field. Figure 11 illustrates the COUNT instruction with CS = 3.

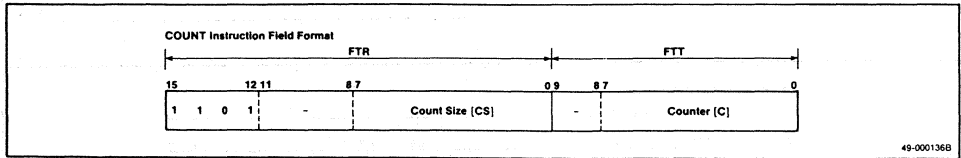
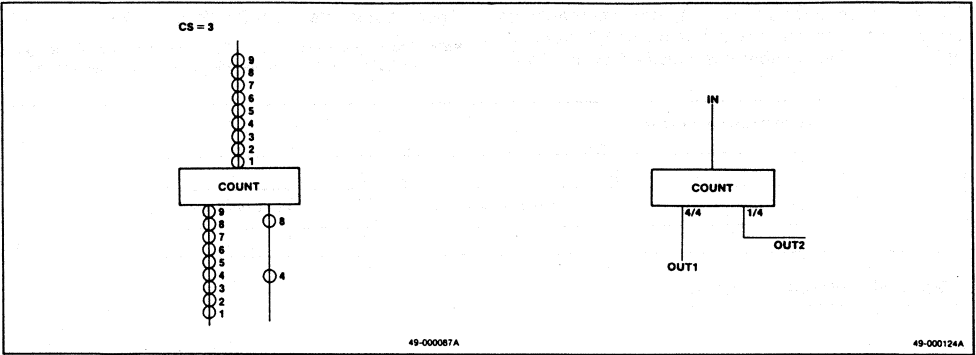


Figure 11. COUNT Instruction



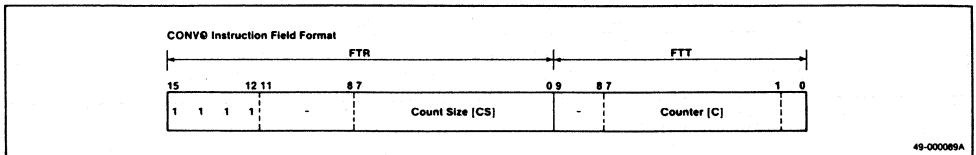
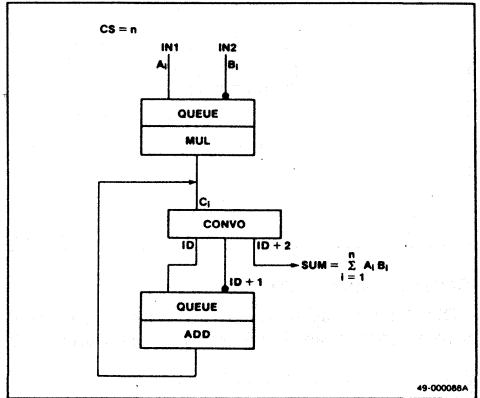
**CONVO [Convolve]**

CONVO instruction is used to perform cumulative operations such as  $\sum A_i$  or  $\prod A_i$ . The CONVO instruction is best suited for convolving two sequences of the same length. Figure 12 illustrates the CONVO instruction by computing

$$SUM = \sum_{i=1}^n A_i B_i$$

The  $A_i$  sequence is input to IN1 while the  $B_i$  sequence is input to IN2. Together they are queued and multiplied to form the  $C_i$  sequence. The  $C_i$ 's arriving at CONVO instruction are queued and added together to form the final answer SUM. The length of the summation,  $n$ , is specified by the CS.

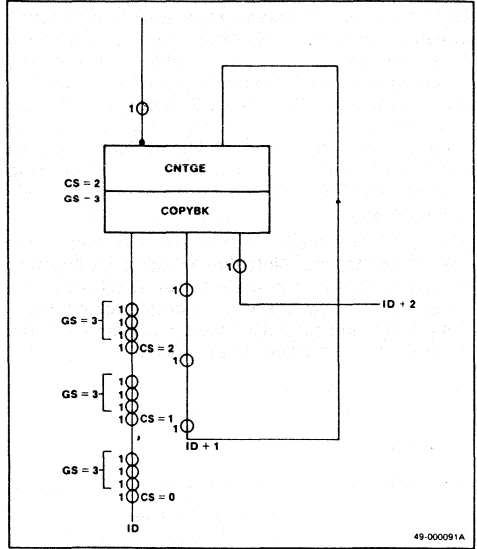
Figure 12. CONVO Instruction



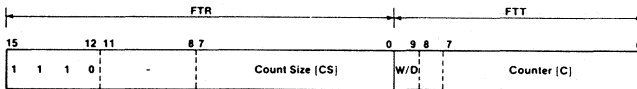
### CNTGE [Count Generation]

CNTGE is normally used with COPYBK (Copy Block) to generate more than 16 copies of a single token (see figure 13). This instruction has both the dead (inactive) state and the wait (active) state. The instruction starts in the dead state. The FTRC bit = 0 tokens that arrive during the dead state of instruction are output to the ID + 2 token stream. It enters the wait state when a token with FTRC bit = 1 arrives and the token is output to ID token stream. Once the instruction is in the wait state, it counts the number of tokens arriving with FTRC bit = 0, outputting them to the ID + 2 token stream, until the number exceeds the number specified by CS. If Counter (C) reaches the number specified by Count Size (CS), the instruction automatically enters the dead state. Tokens with the FTRC bit = 1 arriving at CNTGE while the instruction is in the wait state are deleted by the instruction. Once the instruction enters the dead state, it can be reactivated by the arrival of a token with FTRC bit = 1.

Figure 13. CNTGE Instruction



CNTGE Instruction Field Format



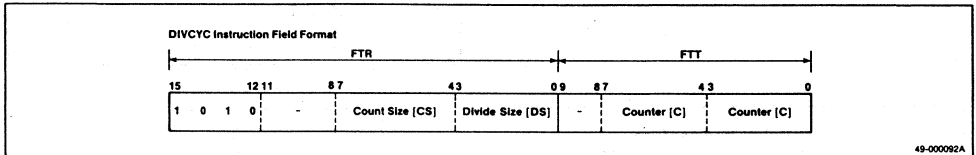
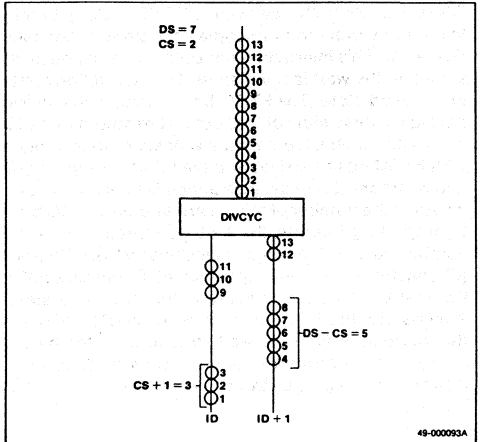
49-000090A

**DIVCYC [ Divide Cyclic ]**

DIVCYC divides an incoming stream of tokens into two streams of tokens: an ID token stream and an ID + 1 token stream. The pattern in which the incoming tokens are divided is specified by the Divide Size (DS) and Count Size (CS). The DS specifies cycle size whereas CS specifies the number of consecutive tokens to be in the ID stream. The first CS + 1 tokens are output to the ID token stream. The following consecutive (DS - CS) tokens are output to the ID + 1 token stream.

Figure 14 illustrates the DIVCYC instruction with DS = 7 and CS = 2. Note that an incoming stream of tokens is divided into a stream of ID tokens and a stream of ID + 1 tokens with a cycle of 8 tokens. Since CS = 2, the number of ID tokens in one cycle is 3, the number of ID + 1 tokens in a cycle is 5.

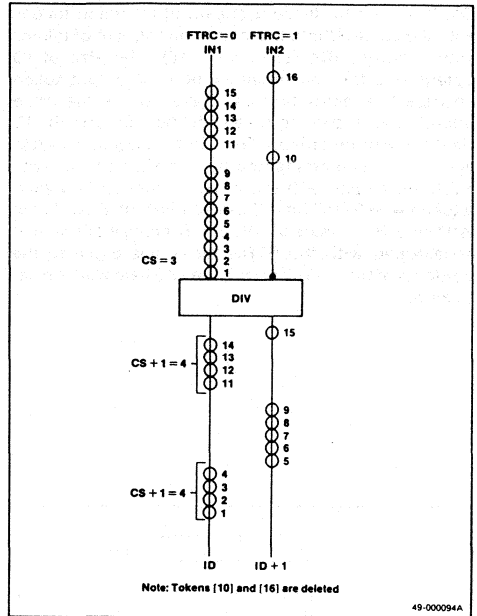
**Figure 14. DIVCYC Instruction**



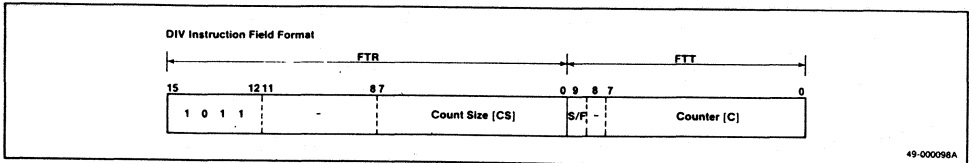
## DIV [Divide]

DIV with CS = n divides an incoming stream of tokens with FTRC bit = 0 into two streams of tokens: ID tokens and ID + 1 tokens. The first (n + 1) incoming tokens with FTRC bit = 0 are output as the ID tokens, and the rest of the incoming tokens with FTRC bit = 0 are output as ID + 1 tokens. An incoming token with FTRC bit = 1 is used to reinitialize the DIV instruction. The stream of input tokens with FTRC bit = 0 after the reinitialization is again divided into a stream of (n + 1) ID tokens followed by ID + 1 tokens. A token with FTRC bit = 1 which reinitializes the DIV instruction is deleted from the output token stream. A DIV instruction with CS = 3 is illustrated in figure 15. The 10th and 16th input tokens have FTRC bit = 1, so they reinitialize the DIV instruction.

Figure 15. DIV Instruction



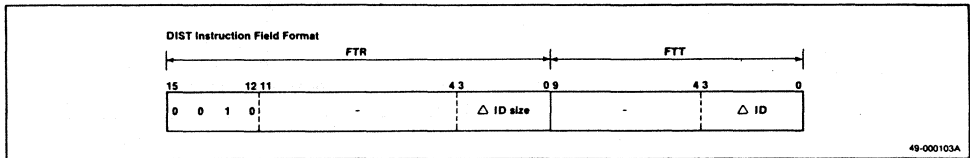
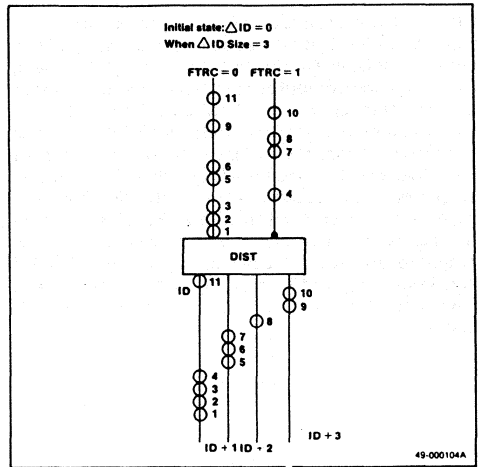
5



**DIST [Distribute]**

DIST is used to divide a stream of incoming tokens with the same ID into more than one stream of tokens with different IDs (see figure 16). The size of ID determines the maximum number of output token streams the instruction can have.  $\Delta ID$  is the value added to an incoming token's ID field to form the ID field of the output token. The  $\Delta ID$  field is initially set to zero, and it is incremented by one after a token with FTRC bit = 1 passes through the instruction. However, a token with FTRC bit = 0 has no effect on the value of  $\Delta ID$  field. If the value of  $\Delta ID$  before being incremented by a token with the FTRC bit = 1 is equal to the contents of the  $\Delta ID$  size field, the  $\Delta ID$  field will be reset to zero.

**Figure 16. DIST Instruction**

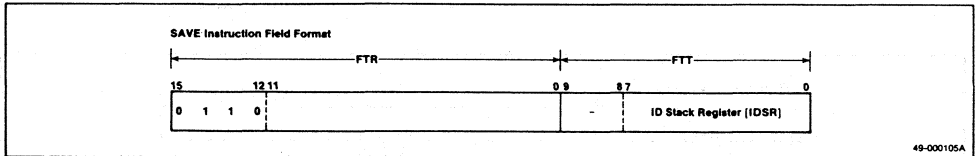
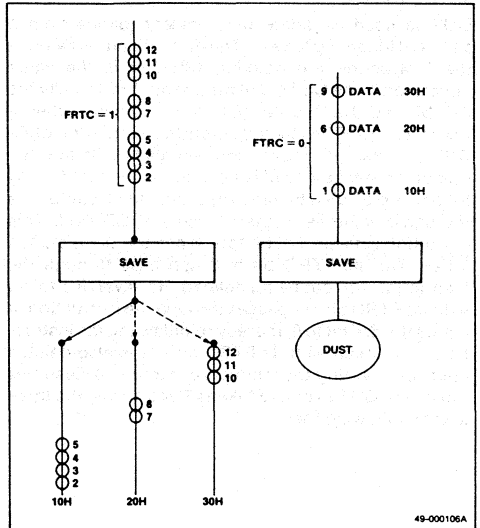


## SAVE [Save ID]

SAVE is used to set the value of the ID field of a token. The instruction performs two different operations depending on whether the token's FTRC bit is 1 or 0. If the token's FTRC bit = 0, the instruction copies the lower eight bits of the data field into the Identifier Stack Register (IDSR) field. However, if the token's FTRC bit is 1, the instruction replaces the token's ID field with the contents of IDSR.

Figure 17 illustrates the use of the SAVE instruction. Token 1 assigns an ID field value of 10H to tokens 2, 3, 4 and 5, token 6 assigns an ID field value of 20H to tokens 7 and 8, and token 9 assigns an ID field value of 30H to tokens 10, 11 and 12. In this example, tokens 1, 6 and 9 are deleted after SAVE instruction.

Figure 17. SAVE Instruction

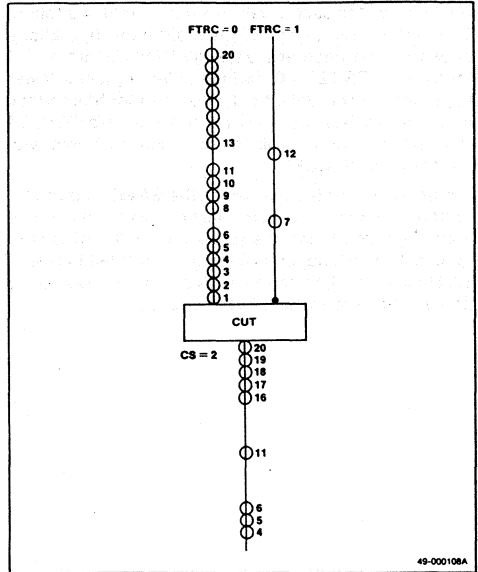




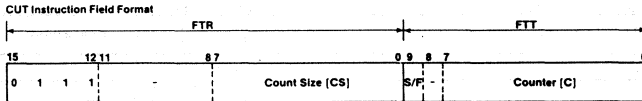
**CUT [Cut Data Stream]**

CUT is used to delete unnecessary tokens from a series of incoming tokens. The first  $n$  tokens arriving at the instruction are deleted, where  $n$  is the value contained in the CS field of the instruction. Initially the S/F bit and the Counter (C) are set to zero. When a token with its FTRC bit = 0 enters the instruction while S/F bit is zero, the token increments the Counter by one and the token itself is deleted. As the first  $(n + 1)$  tokens are deleted by the instruction, the Counter has the same value as  $n$ , the contents of CS field. This condition sets the S/F bit to 1. When the S/F bit is 1, a token with its FTRC bit = 0 can pass through the instruction without being deleted. However, if a token with its FTRC bit = 1 passes through the instruction, it resets the S/F bit to 0, thereby reinitializing the instruction. The token with its FTRC bit = 1 is also deleted after reinitializing the instruction. Figure 18 illustrates the use of CUT to delete tokens 7 and 12 and the three tokens following them.

**Figure 18. CUT Instruction**



49-000108A

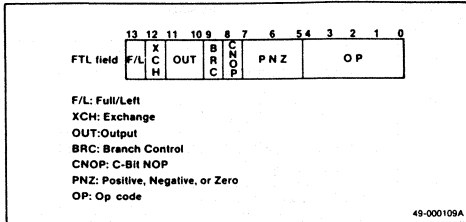


49-000107A

**Table 10. AG and FC Instructions**

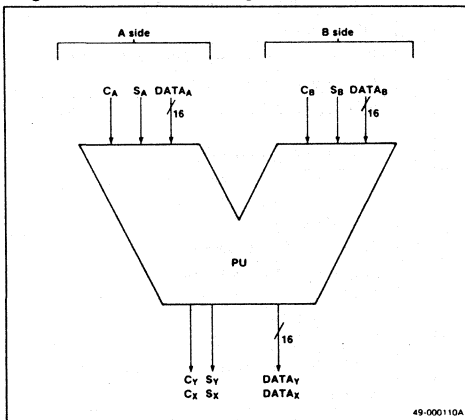
Mnemonic	FTR (16)												FTT (10)										FTRC (1)	Operation	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4			3
QUEUE	0	0	1	1	DM Base (x 2 <sup>1</sup> ) (8)				Queue Size (4)				A	W	Read Counter (4)				Write Counter (4)			Synchronize two tokens			
RDCYCS	0	0	0	0	DM Base (x 2 <sup>1</sup> ) (DMB) (8)				Buffer Size (BS) (4)				(6)				Read Counter (RC) (4)		0	DATA - (DMB + RC), RC - RC + 1					
					1															1	DATA - (DMB + RC), RC - RC + 1, when BS = RC, copy with ID + 1				
RDCYCL	1	0	0	0	DM Base (x 2 <sup>5</sup> ) (4)				Buffer Size (8)				(2)		Read Counter (8)				0	DATA - (DMB + RC), RC - RC + 1					
					1															1	DATA - (DMB + RC), RC - RC + 1, when BS = RC, copy with ID + 1				
WRCYCS	0	0	0	1	Base (x 2 <sup>1</sup> ) (8)				Buffer Size (4)				(6)				Write Counter (WC) (4)		0	(DMB + WC) - DATA, WC - WC + 1, delete token					
					1															1	(DMB + WC) - DATA, WC - WC + 1, when BS = WC, token not deleted				
WRCYCL	1	0	0	1	DM Base (x 2 <sup>5</sup> ) (4)				Buffer Size (8)				(2)		Write Counter (8)				0	(DMB + WC) - DATA, WC - WC + 1, delete token					
					1															1	(DMB + WC) - DATA, WC - WC + 1, when BS = WC, token not deleted				
RDWR	0	1	0	0	DM Base (x 2 <sup>1</sup> ) (8)				(4)				(2)		Address Register (AR) (8)				0	DATA - (DMB + AR + DATA), AR - AR + DATA					
					1															1	(DMB + AR) - DATA, AR - 0				
RDIDX	0	1	0	1	DM Base (x 2 <sup>1</sup> ) (8)				(4)				(2)		Address Register (8)				0	DATA - (DMB + AR + DATA), AR - 0					
					1															1	AR - AR + DATA				
PICKUP	1	1	0	0	(4)				Count Size (CS) (8)				(2)		Counter (C) (8)				0	When CS ≠ C, C - C + 1; when CS = C, distribute, C - 0					
					1															1	C - C + DATA, token deleted				
COUNT	1	1	0	1	(4)				Count Size (8)				(2)		Counter (8)				0	When CS ≠ C, C - C + 1; when CS = C, copy token, C - 0					
					1															1	C - C + DATA, token deleted				
CUT	0	1	1	1	(4)				Count Size (8)				S	/	F	(1)	Counter (8)				0	When S/F = 0 and C ≤ CS, C - C + 1, delete token; when S/F = 0 and C > CS, or when S/F = 1, C - C + 1, token not deleted			
					1															1	S/F - 0, C - 0, token deleted				
DIVCYC	1	0	1	0	Count Size (4)		Divide Size (4)		(2)		Counter (4)		Counter (4)						0	When C ≤ CS, C - C + 1; when C > CS, distribute, C - C + 1; C - C. When C = DS, C - 0					
					1															1	C - C + DATA, token deleted				
DIV	1	0	1	1	(4)				Count Size (8)				S	/	F	(1)	Counter (8)				0	When S/F = 0 and C ≤ CS, C - C + 1; when S/F = 0 and C > CS, or when S/F = 1, distribute, C - C + 1;			
					1															1	S/F - 0, C - 0, token deleted				
DIST	0	0	1	0	(8)				Δ ID Size (4)				(6)				Δ ID (4)		0	ID - (ID + Δ ID) modulo Δ ID size					
					1															1	When Δ ID ≠ Δ ID size, ID - (ID + Δ ID) modulo Δ ID size, Δ ID - Δ ID + 1. When Δ ID = Δ ID size, Δ ID - 0				
CONVO	1	1	1	1	(4)				Count Size (8)				(2)		Counter (7)				(1)	When CS ≠ C, ID - ID + C (modulo 2), C - C + 1; when CS = C, ID - ID + 2, C - 0					
					1															1	ID - IDSR				
SAVE	0	1	1	0	(12)				(2)		ID Stack Register (8) (IDSR)								0	IDSR - Lower 8-bit of DATA					
					1															1	ID - IDSR				
CNTGE	1	1	1	0	(4)				Count Size (8)				W	/	D	(1)	Counter (8)				0	When dead, ID - ID + 2; when wait, if C = CS, C - 0, W/D = 0; when wait, if C ≠ CS, C - C + 1			
					1															1	When dead, initialization; when wait, delete token				

**PU Instructions**



PU instructions (see table 20) are stored in the Function Table Left field of the Function Table memory. The bits 0 through 11 are used as control information for the Processing Unit. The bits 12 and 13 are deleted before the token arrives at the Processing Unit. Two operands from the A and B sides are operated on by the Processing Unit and the result is output to the X and Y sides (see figure 19).

**Figure 19. The Processing Unit**



**Bit Assignments**

**F/L [Full/Left]:** F/L bit = 0 indicates that the PU instruction is a one-operand instruction, and only the Function Table Left field is meaningful. F/L bit = 1 indicates that the PU instruction is a two-operand instruction, and both the Function Table Left field and the Function Table Right field are meaningful. Therefore, when F/L bit = 1, the PU instruction is used in conjunction with an AG/FC instruction.

**XCH [Exchange]:** This bit controls the exchange operation. Operands will be exchanged just before the two tokens enter the QUEUE when XCH = 1.

**OUT [Output]:** There are four different PU output token formats. The two OUT bits specify the output token format. See table 11.

**Table 11. OUT Bits**

OUT Bits	No. of Outputs	First Output ID	First Output DATA, C, S	Second Output ID	Second Output DATA, C, S
00	1	ID	X <sup>1</sup>		
01	1	ID	Y <sup>2</sup>		
10	2	ID	X	ID + 1	X
11	2	ID	X	ID + 1	Y

- Notes:**
1. This is the 18-bit result of the operation output to the X side. It includes the C<sub>X</sub> and S<sub>X</sub> bits.
  2. This is the 18-bit result of the operation output to the Y side. It includes the C<sub>Y</sub> and S<sub>Y</sub> bits.

**BRC [Branch Control]:** The BRC bit controls the flow of the PU output data token. The output data token may be output to either the ID token stream or the ID + 1 token stream. When the BRC bit is set to 1 and the C bit of the PU output data token is also 1, the output data token is sent to the ID + 1 token stream. But when the BRC bit is set to 1 and the C bit of the output data token is 0, the token is sent to the ID token stream. Therefore, using the BRC bit implements a conditional branch on C.

**CNOP Bit:** This bit informs the Processing Unit whether or not the incoming token should be processed. If the CNOP bit is set, and the C<sub>A</sub> bit is not equal to the C<sub>B</sub> bit, then the token passes through the Processing Unit with no operation performed. See table 12.

**Table 12. CNOP Bit**

C <sub>A</sub>	C <sub>B</sub>	PU Operation
0	0	Processing specified by the OP code is performed.
0	1	Token passes through the Processing Unit as NOP.
1	0	Token passes through the Processing Unit as NOP.
1	1	Processing specified by the OP code is performed.

**PNZ [Positive, Negative, Zero] Field:** The PNZ field is used to test the resulting condition of the PU operation. If the resulting condition matches the condition set by the PNZ field, then the C bit of the output data token is set to 1. See table 13.

**Table 13. PNZ Field**

P	N	Z	Condition	C <sub>X</sub>	C <sub>Y</sub>	Assembler Description	
0	0	0	No condition set	C <sub>A</sub>	C <sub>B</sub>		
0	0	1	Result of operation = 0	1	1	EQ	True
			Result of operation ≠ 0	0	0		False
0	1	0	Result of operation < 0	1	1	LT	True
			Result of operation ≥ 0	0	0		False
0	1	1	Result of operation ≤ 0	1	1	LE	True
			Result of operation > 0	0	0		False
1	0	0	Result of operation > 0	1	1	GT	True
			Result of operation ≤ 0	0	0		False
1	0	1	Result of operation ≥ 0	1	1	GE	True
			Result of operation < 0	0	0		False
1	1	0	Result of operation ≠ 0	1	1	NE	True
			Result of operation = 0	0	0		False
1	1	1	Overflow generated	1	1	OVF	True
			No overflow generated	0	0		False

**OP Code Field:** This 5-bit OP code field specifies the PU operations to be performed. See table 14

**Table 14. OP Code Field**

Instruction	Mnemonic	Opcode
Logical	OR	00000
	AND	00001
	XOR	00010
	ANDNOT	00011
	NOT	01100
Arithmetic	ADD	11000
	ADDSC	11100
	SUB	11001
	SUBSC	11101
	MUL	11010
	MULSC	11110
	NOP	11011
	NOPSC	11111
	INC	01010
	DEC	01011
	Shift	SHL
SHLBRV		00101
SHR		00110
SHRBRV		00111
Compare	CMPNOM	01000
	CMP	01001
	CMPXCH	10001
Bit manipulation	GET1	10101
	SET1	10110
	CLR1	10111
Bit check	ANDMSK	01101
	ORMSK	10000
Data conversion	CVT2AB	01110
	CVTAB2	01111
Double precision adjust	ADJL	10100
Accumulative addition	ACC	10010
C bit copy	COPYC	10011

### Logical Instructions

These instructions perform 16-bit logical operations on DATA<sub>A</sub> and DATA<sub>B</sub>. Usually there are no changes in C and S bits between the input token and the output token, however C bits can be affected by PNZ condition when specified.

**OR, AND, XOR:** These instructions perform 16-bit logical OR, AND, and XOR operations using input data tokens from the A and B sides of the Processing Unit. The 16 bit result is output to the X side.

**ANDNOT:** This instruction first complements DATA<sub>A</sub> and then performs logical AND operation with DATA<sub>B</sub>. The 16-bit result is output to the X side.

**NOT:** This is a one-operand instruction which requires 16-bit data input from the A side only. The B side input is ignored. This instruction complements the 16-bit input data from the A side. The 16-bit result is output to the X side.

### Arithmetic Instructions

These instructions perform 17-bit (including the sign bit) arithmetic operations on DATA<sub>A</sub> and DATA<sub>B</sub>. When a PNZ condition is specified, the C bits of output data, C<sub>X</sub> and C<sub>Y</sub>, reflect the setting. However, if no PNZ condition is specified (i.e., PNZ = 000), then C<sub>X</sub> ← C<sub>A</sub> and C<sub>Y</sub> ← C<sub>B</sub>.

**ADD, SUB:** These instructions perform addition or subtraction on DATA<sub>A</sub> and DATA<sub>B</sub> along with the sign bits, S<sub>A</sub> and S<sub>B</sub>. The result is output to the X side. DATA<sub>Y</sub> is normally 0000H. However, if an overflow occurs, then DATA<sub>Y</sub> is equal to +0001H (S<sub>Y</sub> = 0). If an underflow occurs, then the DATA<sub>Y</sub> is equal to -0001H (S<sub>Y</sub> = 1).

**MUL:** This instruction multiplies DATA<sub>A</sub> and DATA<sub>B</sub>. The correct sign bit for the product is determined from S<sub>A</sub> and S<sub>B</sub>. The 33-bit result including a sign bit is output as two 17-bit words, S<sub>X</sub> and DATA<sub>X</sub>, followed by S<sub>Y</sub> and DATA<sub>Y</sub>. DATA<sub>X</sub> is the upper 16-bit word and DATA<sub>Y</sub> is the lower 16-bit word. S<sub>X</sub> holds the resulting sign bit, and S<sub>Y</sub> is a mere duplicate of S<sub>X</sub>.

**NOP:** This instruction performs no operation on the input token. The input data from A and B sides are output to the X and Y sides, respectively, without any change in their contents. If any control other than the OP code (such as PNZ control, BRC control, etc.) has been specified, the output complies with the control.

### Shift Count Instructions

These four Shift Count (SC) instructions first perform the normal operations, then count the number of leading zeros in DATA<sub>X</sub> of the result, and finally output

the number of zeros as DATA<sub>Y</sub> (see table 15). These instructions are provided for easy floating point processing.

**ADDSC, SUBSC, NOPSC:** These instructions perform addition, subtraction, or no operation. The number of preceding zeros in DATA<sub>X</sub> of the result is output as DATA<sub>Y</sub>. If an overflow or an underflow occurs as a result of an operation, DATA<sub>Y</sub> contains +0001H (S<sub>Y</sub> = 0) or -0001H (S<sub>Y</sub> = 1), respectively.

**MULSC:** This instruction performs a normal multiplication operation using the two 17-bit data. The upper order 16-bit data and its sign bit are output as DATA<sub>X</sub> and S<sub>X</sub>, but the lower 16-bit data is not output as DATA<sub>Y</sub>. Instead, the number of preceding zeros in DATA<sub>X</sub> are counted and output as DATA<sub>Y</sub>. The S<sub>Y</sub> bit is always zero.

Table 15. Shift Count Operation

DATA <sub>X</sub> After Operation																SC Output (Y)	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	S <sub>Y</sub>	Y Data
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 1 0 H
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0 0 0 F H
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	x	0	0 0 0 E H
0	0	0	0	0	0	0	0	0	0	0	0	0	1	x	x	0	0 0 0 D H
0	0	0	0	0	0	0	0	0	0	0	1	x	x	x	x	0	0 0 0 C H
0	0	0	0	0	0	0	0	0	0	1	x	x	x	x	x	0	0 0 0 B H
0	0	0	0	0	0	0	0	1	x	x	x	x	x	x	x	0	0 0 0 A H
0	0	0	0	0	0	0	0	1	x	x	x	x	x	x	x	0	0 0 0 9 H
0	0	0	0	0	0	0	1	x	x	x	x	x	x	x	x	0	0 0 0 8 H
0	0	0	0	0	0	1	x	x	x	x	x	x	x	x	x	0	0 0 0 7 H
0	0	0	0	0	1	x	x	x	x	x	x	x	x	x	x	0	0 0 0 6 H
0	0	0	0	1	x	x	x	x	x	x	x	x	x	x	x	0	0 0 0 5 H
0	0	0	1	x	x	x	x	x	x	x	x	x	x	x	x	0	0 0 0 4 H
0	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0 0 0 3 H
0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0 0 0 2 H
0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0 0 0 1 H
1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0 0 0 0 H*

Notes: \* When an overflow or underflow has occurred  
x don't care

### Increment and Decrement Instructions

**INC, DEC:** These instructions increment or decrement the 17-bit data from the A side (S<sub>A</sub> and DATA<sub>A</sub>), and outputs the result to X side as S<sub>X</sub> and DATA<sub>X</sub>. The S<sub>Y</sub> and DATA<sub>Y</sub> are normally zero. However, if an overflow or an underflow occurs, then the Y side outputs +0001H (S<sub>Y</sub> = 0) or -0001H (S<sub>Y</sub> = 1), respectively. See figure 20 for detailed operation explanations.

### Shift Instructions

**SHR [Shift Right], SHL [Shift Left]:** SHR or SHL instructions perform a barrel-shifting operation on the 16-bit data, DATA<sub>A</sub>. The actual number of shifts and the direction is further specified by the lower five bits of DATA<sub>B</sub> and S<sub>B</sub>, respectively. See figure 20 for detailed operation explanations.

Figure 20. SHR and SHL

Right Shift [SHR execution]

S <sub>n</sub>	Lower 5 bits of DATA <sub>n</sub> (No. of shifts)	DATA <sub>x</sub>	DATA <sub>y</sub>
0	00000	A <sub>15</sub> A <sub>14</sub> ...A <sub>1</sub> A <sub>0</sub>	0...0
0	00001	0 A <sub>15</sub> A <sub>14</sub> ...A <sub>1</sub>	0 0...0
0	00010	00 A <sub>15</sub> ...A <sub>2</sub>	AA 0 0...0
0	00011	0...0 A <sub>15</sub> ...A <sub>3</sub>	A <sub>2</sub> A <sub>0</sub> 0...0
0	00100	0...0 A <sub>15</sub> ...A <sub>4</sub>	A <sub>3</sub> ...A <sub>0</sub> 0...0
0	00101	0...0 A <sub>15</sub> ...A <sub>5</sub>	A <sub>4</sub> ...A <sub>0</sub> 0...0
0	00110	0...0 A <sub>15</sub> ...A <sub>6</sub>	A <sub>5</sub> ...A <sub>0</sub> 0...0
0	00111	0...0 A <sub>15</sub> ...A <sub>7</sub>	A <sub>6</sub> ...A <sub>0</sub> 0...0
0	01000	0...0 A <sub>15</sub> ...A <sub>8</sub>	A <sub>7</sub> ...A <sub>0</sub> 0...0
0	01001	0...0 A <sub>15</sub> ...A <sub>9</sub>	A <sub>8</sub> ...A <sub>0</sub> 0...0
0	01010	0...0 A <sub>15</sub> ...A <sub>10</sub>	A <sub>9</sub> ...A <sub>0</sub> 0...0
0	01011	0...0 A <sub>15</sub> ...A <sub>11</sub>	A <sub>10</sub> ...A <sub>0</sub> 0...0
0	01100	0...0 A <sub>15</sub> ...A <sub>12</sub>	A <sub>11</sub> ...A <sub>0</sub> 0...0
0	01101	0...0 A <sub>15</sub> A <sub>14</sub>	A <sub>12</sub> ...A <sub>0</sub> 0...0
0	01110	0...0 AA 15 14	A <sub>13</sub> ...A <sub>0</sub> 0 0
0	01111	0...0 A 15	A <sub>14</sub> ...A <sub>0</sub> 0
0	1XXXX	0...0	A <sub>15</sub> ...A <sub>1</sub> A <sub>0</sub>
1	00000	A <sub>15</sub> A <sub>14</sub> ...A <sub>1</sub> A <sub>0</sub>	0...0
1	00001	A <sub>14</sub> ...A <sub>0</sub> 0	0...0 A 15
1	00010	A <sub>13</sub> ...A <sub>0</sub> 0 0	0...0 AA 15 14
1	00011	A <sub>12</sub> ...A <sub>0</sub> 0...0	0...0 A <sub>2</sub> A <sub>0</sub> 15 13
1	00100	A <sub>11</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>12</sub> ...A <sub>11</sub>
1	00101	A <sub>10</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>11</sub>
1	00110	A <sub>9</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>10</sub>
1	00111	A <sub>8</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>9</sub>
1	01000	A <sub>7</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>8</sub>
1	01001	A <sub>6</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>7</sub>
1	01010	A <sub>5</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>6</sub>
1	01011	A <sub>4</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>5</sub>
1	01100	A <sub>3</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>4</sub>
1	01101	A <sub>2</sub> A <sub>0</sub> 0...0	0 0...0 A <sub>15</sub> ...A <sub>3</sub>
1	01110	AA 1 0 0...0	0 0...0 A <sub>15</sub> ...A <sub>2</sub>
1	01111	A 0 0...0	0...0 A <sub>15</sub> ...A <sub>1</sub>
1	1XXXX	0...0	A <sub>15</sub> ...A <sub>0</sub>

49-000137C

Left Shift [SHL execution]

S <sub>n</sub>	Lower 5 bits of DATA <sub>n</sub> (No. of shifts)	DATA <sub>x</sub>	DATA <sub>y</sub>
0	00000	A <sub>15</sub> A <sub>14</sub> ...A <sub>1</sub> A <sub>0</sub>	0...0
0	00001	A <sub>14</sub> ...A <sub>0</sub> 0	0...0 A 15
0	00010	A <sub>13</sub> ...A <sub>0</sub> 0 0	0...0 AA 15 14
0	00011	A <sub>12</sub> ...A <sub>0</sub> 0...0	0...0 A <sub>2</sub> A <sub>0</sub> 15 13
0	00100	A <sub>11</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>12</sub>
0	00101	A <sub>10</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>11</sub>
0	00110	A <sub>9</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>10</sub>
0	00111	A <sub>8</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>9</sub>
0	01000	A <sub>7</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>8</sub>
0	01001	A <sub>6</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>7</sub>
0	01010	A <sub>5</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>6</sub>
0	01011	A <sub>4</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>5</sub>
0	01100	A <sub>3</sub> ...A <sub>0</sub> 0...0	0...0 0...0 A <sub>15</sub> ...A <sub>4</sub>
0	01101	A <sub>2</sub> A <sub>0</sub> 0...0	0 0...0 A <sub>15</sub> ...A <sub>3</sub>
0	01110	AA 1 0 0...0	0 0...0 A <sub>15</sub> ...A <sub>2</sub>
0	01111	A 0 0...0	0...0 A <sub>15</sub> ...A <sub>1</sub>
0	1XXXX	0...0	A <sub>15</sub> ...A <sub>0</sub>
1	00000	A <sub>15</sub> A <sub>14</sub> ...A <sub>1</sub> A <sub>0</sub>	0...0
1	00001	0 A <sub>15</sub> A <sub>14</sub> ...A <sub>1</sub>	A 0 0...0
1	00010	0 0 A <sub>15</sub> ...A <sub>2</sub>	AA 1 0 0...0
1	00011	0...0 A <sub>15</sub> ...A <sub>3</sub>	A <sub>2</sub> A <sub>0</sub> 0...0
1	00100	0...0 A <sub>15</sub> ...A <sub>4</sub>	A <sub>3</sub> ...A <sub>0</sub> 0...0
1	00101	0...0 A <sub>15</sub> ...A <sub>5</sub>	A <sub>4</sub> ...A <sub>0</sub> 0...0
1	00110	0...0 A <sub>15</sub> ...A <sub>6</sub>	A <sub>5</sub> ...A <sub>0</sub> 0...0
1	00111	0...0 A <sub>15</sub> ...A <sub>7</sub>	A <sub>6</sub> ...A <sub>0</sub> 0...0
1	01000	0...0 A <sub>15</sub> ...A <sub>8</sub>	A <sub>7</sub> ...A <sub>0</sub> 0...0
1	01001	0...0 A <sub>15</sub> ...A <sub>9</sub>	A <sub>8</sub> ...A <sub>0</sub> 0...0
1	01010	0...0 A <sub>15</sub> ...A <sub>10</sub>	A <sub>9</sub> ...A <sub>0</sub> 0...0
1	01011	0...0 A <sub>15</sub> ...A <sub>11</sub>	A <sub>10</sub> ...A <sub>0</sub> 0...0
1	01100	0...0 A <sub>15</sub> ...A <sub>12</sub>	A <sub>11</sub> ...A <sub>0</sub> 0...0
1	01101	0...0 A 15 13	A <sub>12</sub> ...A <sub>0</sub> 0...0
1	01110	0...0 AA 15 14	A <sub>13</sub> ...A <sub>0</sub> 0 0
1	01111	0...0 A 15	A <sub>14</sub> ...A <sub>0</sub> 0
1	1XXXX	0...0	A <sub>15</sub> ...A <sub>1</sub> A <sub>0</sub>

FD00137C

**SHRBRV [Shift Right with Bit Reverse], SHLBRV [Shift Left with Bit Reverse]:** SHRBRV or SHLBRV first reverses the order of the bits in DATA<sub>A</sub> and then performs a normal SHR or SHL operation, respectively. See figure 21.

**Compare Instructions**

The Compare instructions (see table 16) are different from other PU instructions in that PNZ conditions must be specified along with the instructions. When a compare instruction is used along with a specified PNZ field, the Processing Unit performs a subtract operation. This subtract operation produces a set of PNZ flags, which are compared against the PNZ field specified by the instruction. When these two PNZ fields coincide, the specified PNZ conditions are said to be true. When they do not coincide, the specified PNZ conditions are said to be false (see table 17). The output data from the Processing Unit differs significantly depending on the PNZ conditions. The following three instructions compare the 17-bit data (S<sub>A</sub> and DATA<sub>A</sub>) from the A side against the 17-bit data (S<sub>B</sub> and DATA<sub>B</sub>) from the B side.

**CMPNOM [Compare and normalize]:** If the specified PNZ conditions are false, then the control bits, sign bits and data for both the X and Y sides are set to zero. If the PNZ conditions are true, then C<sub>X</sub> and C<sub>Y</sub> are set to one, S<sub>X</sub> and S<sub>Y</sub> are set to zero, DATA<sub>X</sub> is set to 0001H, and DATA<sub>Y</sub> is set to 0000H.

**CMP [Compare]:** This instruction outputs the 17-bit data words from the A and B sides to the X and Y sides without any change in their contents. It only alters the control bits. If the specified PNZ conditions are true, then C<sub>X</sub> and C<sub>Y</sub> are set to one. If the PNZ conditions are false, then C<sub>X</sub> is set to one and C<sub>Y</sub> is set to zero.

**CMPXCH [Compare and exchange]:** If the specified PNZ conditions are true, then both the input data from the A side and B side are unchanged and output to the X side and Y side, respectively, including their sign bits and the control bits. However, if the PNZ conditions are false, then the input data from the A side is exchanged with the input data from the B side, including the control and sign bits.

Figure 21. Bit Reversal Operations in SHRBRV and SHLBRV

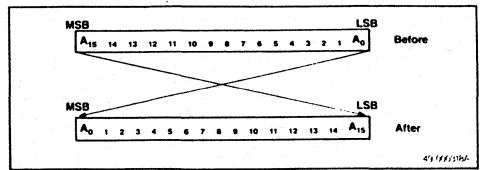


Table 17. PNZ Field Conditions for Compare Instructions

P	N	Z	Condition	True/ False	Function	Mnemonic
0	0	1	S <sub>A</sub> DATA <sub>A</sub> = S <sub>B</sub> DATA <sub>B</sub>	True	Equal	EQ
			S <sub>A</sub> DATA <sub>A</sub> ≠ S <sub>B</sub> DATA <sub>B</sub>	False	Not equal	
0	1	0	S <sub>A</sub> DATA <sub>A</sub> < S <sub>B</sub> DATA <sub>B</sub>	True	Less than	LT
			S <sub>A</sub> DATA <sub>A</sub> ≥ S <sub>B</sub> DATA <sub>B</sub>	False	Greater or equal	
0	1	1	S <sub>A</sub> DATA <sub>A</sub> ≤ S <sub>B</sub> DATA <sub>B</sub>	True	Less or equal	LE
			S <sub>A</sub> DATA <sub>A</sub> > S <sub>B</sub> DATA <sub>B</sub>	False	Greater than	
1	0	0	S <sub>A</sub> DATA <sub>A</sub> > S <sub>B</sub> DATA <sub>B</sub>	True	Greater than	GT
			S <sub>A</sub> DATA <sub>A</sub> ≤ S <sub>B</sub> DATA <sub>B</sub>	False	Less or equal	
1	0	1	S <sub>A</sub> DATA <sub>A</sub> ≥ S <sub>B</sub> DATA <sub>B</sub>	True	Greater or equal	GE
			S <sub>A</sub> DATA <sub>A</sub> < S <sub>B</sub> DATA <sub>B</sub>	False	Less than	
1	1	0	S <sub>A</sub> DATA <sub>A</sub> ≠ S <sub>B</sub> DATA <sub>B</sub>	True	Not equal	NE
			S <sub>A</sub> DATA <sub>A</sub> = S <sub>B</sub> DATA <sub>B</sub>	False	Equal	

**Note:** The significance of the PNZ bits when Compare instructions are executed differs from that of other instructions. Here, the use of PNZ = 111 or 000 is prohibited.

Table 16. Compare Instructions

Mnemonic	Input								Output				Notes
	C <sub>A</sub>	S <sub>A</sub>	DATA <sub>A</sub>	C <sub>B</sub>	S <sub>B</sub>	DATA <sub>B</sub>	C <sub>X</sub>	S <sub>X</sub>	DATA <sub>X</sub>	C <sub>Y</sub>	S <sub>Y</sub>	DATA <sub>Y</sub>	
CMPNOM	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	0	0	0000H	0	0	0000H	When PNZ is False
	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	1	0	0001H	1	0	0000H	When PNZ is true
CMP	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	0	S <sub>A</sub>	A	0	S <sub>B</sub>	B	When PNZ is false
	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	1	S <sub>A</sub>	A	1	S <sub>B</sub>	B	When PNZ is true
CMPXCH	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	When PNZ is true
	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>B</sub>	S <sub>B</sub>	A	C <sub>A</sub>	S <sub>B</sub>	A	When PNZ is false

### Bit Manipulation Instructions

**GET1 [Get one bit]:** This instruction is used to read a particular bit from DATA<sub>A</sub> (see table 18). A bit of DATA<sub>A</sub> specified by the lower 4 bits of DATA<sub>B</sub> is output as the least significant bit of DATA<sub>X</sub>. All other bits of DATA<sub>X</sub> are set to zero. DATA<sub>Y</sub> is also set to zero. The control bits and the sign bits of DATA<sub>X</sub> and DATA<sub>Y</sub> are as follows: C<sub>X</sub> - C<sub>A</sub>, C<sub>Y</sub> - C<sub>B</sub>, S<sub>X</sub> - S<sub>A</sub>, S<sub>Y</sub> - 0.

**SET1 [Set one bit]:** This instruction is used to set a particular bit of DATA<sub>A</sub>. The bit of DATA<sub>A</sub> to be set is specified by the lower 4 bits of DATA<sub>B</sub>. After the bit is set, the 16-bit result is output as DATA<sub>X</sub>. DATA<sub>Y</sub> is always output as zero. The control bits and the sign bits of DATA<sub>X</sub> and DATA<sub>Y</sub> are as follows: C<sub>X</sub> - C<sub>A</sub>, C<sub>Y</sub> - C<sub>B</sub>, S<sub>X</sub> - S<sub>A</sub>, S<sub>Y</sub> - 0.

**CLR1 [Clear one bit]:** This instruction is used to reset a particular bit of DATA<sub>A</sub>. The bit of DATA<sub>A</sub> to be reset is specified by the lower 4 bits of DATA<sub>B</sub>. After the bit is reset (cleared), the 16-bit result is output as DATA<sub>X</sub>. DATA<sub>Y</sub> is always output as zero. The control bits and the sign bits of DATA<sub>X</sub> and DATA<sub>Y</sub> are as follows: C<sub>X</sub> - C<sub>A</sub>, C<sub>Y</sub> - C<sub>B</sub>, S<sub>X</sub> - S<sub>A</sub>, S<sub>Y</sub> - 0.

**Table 18. Bit Addressing**

DATA <sub>B</sub> BIT				DATA <sub>A</sub> Bit Position
3	2	1	0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

### Bit Check Instructions

**ANDMSK [Mask a word with logical AND]:** This instruction tests certain bits in DATA<sub>A</sub>. The bits in DATA<sub>A</sub> to be tested are first masked with a bit pattern in DATA<sub>B</sub>. Only those bits in DATA<sub>A</sub> corresponding to the one bits of DATA<sub>B</sub> are considered. Then only those masked bits

of DATA<sub>A</sub> are ANDed together to set or reset the control bits, C<sub>X</sub> and C<sub>Y</sub>. If the result of the AND operation is 1, then both the C<sub>X</sub> and C<sub>Y</sub> are set to 1. If the result of the operation is 0, then the both C<sub>X</sub> and C<sub>Y</sub> are set to 0. The rest of the output data fields are the following: S<sub>X</sub> - S<sub>A</sub>, S<sub>Y</sub> - S<sub>B</sub>, DATA<sub>X</sub> - DATA<sub>A</sub>, DATA<sub>Y</sub> - DATA<sub>B</sub>.

**ORMSK [Mask a word with logical OR]:** This instruction tests certain bits in DATA<sub>A</sub>. The bits in DATA<sub>A</sub> to be tested are first masked with a bit pattern in DATA<sub>B</sub>. Only those bits in DATA<sub>A</sub> corresponding to the one bits of DATA<sub>B</sub> are considered. Then only those masked bits of DATA<sub>A</sub> are ORed together to set or reset the control bits, C<sub>X</sub> and C<sub>Y</sub>. If the result of the OR operation is 1, then both C<sub>X</sub> and C<sub>Y</sub> are set to 1. If the result of the operation is 0, then the both C<sub>X</sub> and C<sub>Y</sub> are set to 0. The rest of the output data fields are the following: S<sub>X</sub> - S<sub>A</sub>, S<sub>Y</sub> - S<sub>B</sub>, DATA<sub>X</sub> - DATA<sub>A</sub>, DATA<sub>Y</sub> - DATA<sub>B</sub>.

### Data Conversion Instructions

**CVT2AB [Convert two's complement to sign-magnitude]:** This instruction converts a 16-bit number in two's complement form to a 17-bit number in sign-magnitude form. The sign of the two's complement number is output as the S<sub>X</sub> bit.

**CVTAB2 [Convert sign-magnitude to two's complement]:** This instruction converts a 17-bit number in sign-magnitude form to a 16-bit number in two's complement form. This operation has the potential danger of an overflow or an underflow. If an overflow or an underflow occurs, the C<sub>X</sub> bit is set to 1.

### Double Precision Adjustment Instruction

**ADJL [Adjust long]:** This instruction is used to adjust a double precision number, in which the sign bits of the upper and lower words are different. This situation may occur after a double precision arithmetic operation. The examples in table 19 illustrate the adjustments of double precision numbers.

**Table 19. Double Precision Adjustment Examples**

	Input/Output	Sign	Data	
Input	High (A data)	0	1234H	
	Low (B data)	0	5678H	
	Output	High (X data)	0	1234H
		Low (Y data)	0	5678H
Input	High (A data)	0	1234H	
	Low (B data)	1	5678H	
	Output	High (X data)	0	1233H
		Low (Y data)	0	A988H
Input	High (A data)	1	1234H	
	Low (B data)	0	5678H	
	Output	High (X data)	1	1233H
		Low (Y data)	1	A988H



**Accumulative Addition Instruction**

**ACC [Accumulate]:** This instruction (see figure 22) performs cumulative additions of incoming tokens' data fields. The incoming tokens are classified into type 1 and type 2 tokens. A type 1 token is deleted after the ACC operation, but a type 2 token is not. Moreover, a type 2 token reads the contents of the ACC register, which contains the accumulated sum of tokens. When a type 2 token reads the contents of the ACC register, the ID field of the token is unchanged. However, if an overflow has occurred prior to the arrival of a type 2 token, the ID field is incremented by one. Only the following three tokens qualify as type 2 tokens.

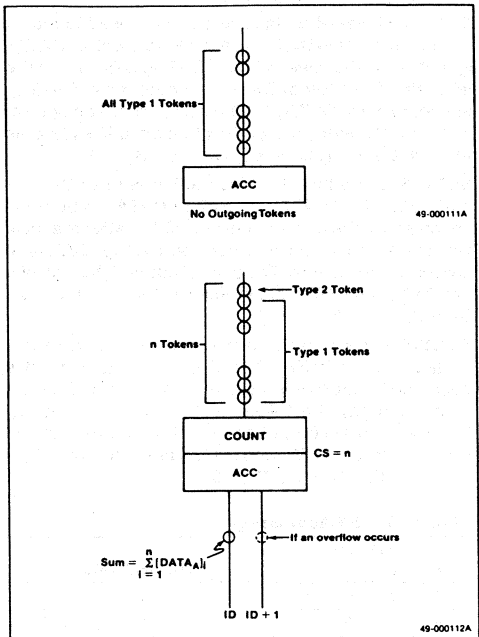
1. If the ACC instruction is used along with RDCYCS instruction, and the token's FTRC bit = 1, and the Buffer Size and Read Counter of RDCYCS instruction are equal.
2. If the ACC instruction is used along with RDCYCL instruction, and the token's FTRC bit = 1, and the Buffer size and Read Counter of RDCYCL instruction are equal.
3. If the ACC instruction is used along with COUNT instruction, and the token's FTRC bit = 0, and the Count Size and Counter of COUNT instruction are equal.

**C Bit Copy Instruction**

**COPYC [Copy control bit]:** This instruction copies the control bit of the A side and outputs it as C<sub>Y</sub>.

$C_X \leftarrow C_A, S_X \leftarrow S_A, DATA_X \leftarrow DATA_A, C_Y \leftarrow C_A, S_Y \leftarrow S_B, DATA_Y \leftarrow DATA_B.$

Figure 22. ACC Instruction



**Table 20. PU Instruction**

Mnemonic	OP Code	Input						Output						Notes
		C <sub>A</sub>	S <sub>A</sub>	DATA <sub>A</sub>	C <sub>B</sub>	S <sub>B</sub>	DATA <sub>B</sub>	C <sub>X</sub>	S <sub>X</sub>	DATA <sub>X</sub>	C <sub>Y</sub>	S <sub>Y</sub>	DATA <sub>Y</sub>	
<b>Logical Operations</b>														
OR	00000	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>X</sub>	S <sub>A</sub>	A OR B	C <sub>Y</sub>	0	0000H	
AND	00001	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>X</sub>	S <sub>A</sub>	A AND B	C <sub>Y</sub>	0	0000H	
XOR	00010	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>X</sub>	S <sub>A</sub>	A XOR B	C <sub>Y</sub>	0	0000H	
ANDNOT	00011	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>X</sub>	S <sub>A</sub>	$\bar{A}$ AND B	C <sub>Y</sub>	0	0000H	
NOT	01100	C <sub>A</sub>	S <sub>A</sub>	A				C <sub>X</sub>	S <sub>A</sub>	$\bar{A}$	C <sub>Y</sub>	0	0000H	
<b>Arithmetic Operations</b>														
ADD	11000	C <sub>A</sub>	0	A	C <sub>B</sub>	0	B	C <sub>X</sub>	0	A + B	C <sub>Y</sub>	0	*	
		C <sub>A</sub>	0	A	C <sub>B</sub>	1	B	C <sub>X</sub>	0	A - B	C <sub>Y</sub>	0	0000H	When A ≥ B, S <sub>X</sub> = 0
								C <sub>X</sub>	1	B - A	C <sub>Y</sub>	1	0000H	When A < B, S <sub>X</sub> = 1
		C <sub>A</sub>	1	A	C <sub>B</sub>	0	B	C <sub>X</sub>	0	B - A	C <sub>Y</sub>	0	0000H	When A < B, S <sub>X</sub> = 0
								C <sub>X</sub>	1	A - B	C <sub>Y</sub>	1	0000H	When A ≥ B, S <sub>X</sub> = 1
ADDSC	11100	C <sub>A</sub>	0	A	C <sub>B</sub>	0	B	C <sub>X</sub>	0	A + B	C <sub>Y</sub>	S <sub>S</sub>	No. of shifts †	
		C <sub>A</sub>	0	A	C <sub>B</sub>	1	B	C <sub>X</sub>	0	A - B	C <sub>Y</sub>	S <sub>S</sub>	*	When A ≥ B, S <sub>X</sub> = 0
								C <sub>X</sub>	1	B - A	C <sub>Y</sub>	S <sub>S</sub>	No. of shifts †	When A < B, S <sub>X</sub> = 1
		C <sub>A</sub>	1	A	C <sub>B</sub>	0	B	C <sub>X</sub>	0	B - A	C <sub>Y</sub>	S <sub>S</sub>	*	When A < B, S <sub>X</sub> = 0
								C <sub>X</sub>	1	A - B	C <sub>Y</sub>	S <sub>S</sub>	No. of shifts †	When A ≥ B, S <sub>X</sub> = 1
		C <sub>A</sub>	1	A	C <sub>B</sub>	1	B	C <sub>X</sub>	1	A + B	C <sub>Y</sub>	S <sub>S</sub>	No. of shifts †	
SUB	11001	C <sub>A</sub>	0	A	C <sub>B</sub>	0	B	C <sub>X</sub>	0	A - B	C <sub>Y</sub>	0	0000H	When A > B, S <sub>X</sub> = 0
								C <sub>X</sub>	1	B - A	C <sub>Y</sub>	1	0000H	When A < B, S <sub>X</sub> = 1
		C <sub>A</sub>	0	A	C <sub>B</sub>	1	B	C <sub>X</sub>	0	A + B	C <sub>Y</sub>	0	*	
		C <sub>A</sub>	1	A	C <sub>B</sub>	0	B	C <sub>X</sub>	1	A + B	C <sub>Y</sub>	1	*	
		C <sub>A</sub>	1	A	C <sub>B</sub>	1	B	C <sub>X</sub>	0	B - A	C <sub>Y</sub>	0	0000H	When A < B, S <sub>X</sub> = 0
								C <sub>X</sub>	1	A - B	C <sub>Y</sub>	1	0000H	When A ≥ B, S <sub>X</sub> = 1
SUBSC	11101	C <sub>A</sub>	0	A	C <sub>B</sub>	0	B	C <sub>X</sub>	0	A - B	C <sub>Y</sub>	S <sub>S</sub>	No. of shifts †	When A ≥ B, S <sub>X</sub> = 0
								C <sub>X</sub>	1	B - A	C <sub>Y</sub>	S <sub>S</sub>	No. of shifts †	When A < B, S <sub>X</sub> = 1
		C <sub>A</sub>	0	A	C <sub>B</sub>	1	B	C <sub>X</sub>	0	A + B	C <sub>Y</sub>	S <sub>S</sub>	No. of shifts †	
		C <sub>A</sub>	1	A	C <sub>B</sub>	0	B	C <sub>X</sub>	1	A + B	C <sub>Y</sub>	S <sub>S</sub>	No. of shifts †	
		C <sub>A</sub>	1	A	C <sub>B</sub>	1	B	C <sub>X</sub>	0	B - A	C <sub>Y</sub>	S <sub>S</sub>	No. of shifts †	When A < B, S <sub>X</sub> = 0
								C <sub>X</sub>	1	A - B	C <sub>Y</sub>	S <sub>S</sub>	No. of shifts †	When A ≥ B, S <sub>X</sub> = 1
MUL	11010	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>X</sub>	S <sub>X</sub>	A x B High	C <sub>Y</sub>	S <sub>X</sub>	A x B Low	S <sub>X</sub> = S <sub>A</sub> OR S <sub>B</sub> (logical OR)
MULSC	11110	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>X</sub>	S <sub>X</sub>	A x B High	C <sub>Y</sub>	S <sub>S</sub>	No. of shifts †	S <sub>X</sub> = S <sub>A</sub> OR S <sub>B</sub> (logical OR)

5

Table 20. PU Instruction (cont)

Mnemonic	OP code	Input							Output						Notes
		C <sub>A</sub>	S <sub>A</sub>	DATA <sub>A</sub>	C <sub>B</sub>	S <sub>B</sub>	DATA <sub>B</sub>	C <sub>X</sub>	S <sub>X</sub>	DATA <sub>X</sub>	C <sub>Y</sub>	S <sub>Y</sub>	DATA <sub>Y</sub>		
<b>Arithmetic Operations</b>															
NOP	1 1 0 1 1	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>X</sub>	S <sub>A</sub>	A	C <sub>Y</sub>	S <sub>B</sub>	B		
NOPSC	1 1 1 1 1	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>X</sub>	S <sub>A</sub>	A	C <sub>Y</sub>	S <sub>S</sub>	No. of shifts ↑		
INC	0 1 0 1 0	C <sub>A</sub>	0	A				C <sub>X</sub>	0	A + 1	C <sub>Y</sub>	0	*		
		C <sub>A</sub>	1	A				C <sub>X</sub>	0	1	C <sub>Y</sub>	0	0000H	When A = 0, S <sub>X</sub> = 0	
DEC	0 1 0 1 1	C <sub>A</sub>	0	A				C <sub>X</sub>	0	A - 1	C <sub>Y</sub>	0	0000H	When A ≥ 0, S <sub>X</sub> = 0	
		C <sub>A</sub>	1	A				C <sub>X</sub>	1	1	C <sub>Y</sub>	1	0000H	When A = 0, S <sub>X</sub> = 1	
C <sub>A</sub>	1	A					C <sub>X</sub>	1	A + 1	C <sub>Y</sub>	1	*			
<b>Shift</b>															
SHL	0 0 1 0 0	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	0	No. of shifts	C <sub>X</sub>	S <sub>A</sub>	Shift A left	C <sub>Y</sub>	S <sub>A</sub>	Shift A left		
		C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	1	No. of shifts	C <sub>X</sub>	S <sub>A</sub>	Shift A right	C <sub>Y</sub>	S <sub>A</sub>	Shift A right		
SHLBRV	0 0 1 0 1	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	0	No. of shifts	C <sub>X</sub>	S <sub>A</sub>	Invert A shift left	C <sub>Y</sub>	S <sub>A</sub>	Invert A shift left		
		C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	1	No. of shifts	C <sub>X</sub>	S <sub>A</sub>	Invert A shift right	C <sub>Y</sub>	S <sub>A</sub>	Invert A shift right		
SHR	0 0 1 1 0	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	0	No. of shifts	C <sub>X</sub>	S <sub>A</sub>	Shift A right	C <sub>Y</sub>	S <sub>A</sub>	Shift A right		
		C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	1	No. of shifts	C <sub>X</sub>	S <sub>A</sub>	Shift A left	C <sub>Y</sub>	S <sub>A</sub>	Shift A left		
SHRBRV	0 0 1 1 1	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	0	No. of shifts	C <sub>X</sub>	S <sub>A</sub>	Invert A shift right	C <sub>Y</sub>	S <sub>A</sub>	Invert A shift right		
		C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	1	No. of shifts	C <sub>X</sub>	S <sub>A</sub>	Invert A shift left	C <sub>Y</sub>	S <sub>A</sub>	Invert A shift left		
<b>Comparison</b>															
CMPNOM	0 1 0 0 0	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	0	0	0000H	0	0	0000H	When PNZ is false	
		C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	1	0	0001H	1	0	0000H	When PNZ is true	
CMP	0 1 0 0 1	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	0	S <sub>A</sub>	A	0	S <sub>B</sub>	B	When PNZ is false	
		C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	1	S <sub>A</sub>	A	1	S <sub>B</sub>	B	When PNZ is true	
CMPXCH	1 0 0 0 1	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	When PNZ is true	
		C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>A</sub>	S <sub>A</sub>	A	When PNZ is false	
<b>Accumulative Addition</b>															
ACC	1 0 0 1 0	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>X</sub>	S <sub>X</sub>	ΣA				Used as a pair with AG & FC instruction COUNT	
<b>C Bit Copy</b>															
COPYC	1 0 0 1 1	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>A</sub>	S <sub>B</sub>	B		

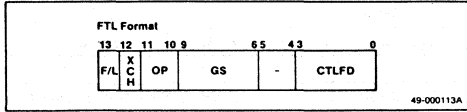
**Table 20. PU Instruction (cont)**

Mnemonic	OP code	Input						Output						Notes
		C <sub>A</sub>	S <sub>A</sub>	DATA <sub>A</sub>	C <sub>B</sub>	S <sub>B</sub>	DATA <sub>B</sub>	C <sub>X</sub>	S <sub>X</sub>	DATA <sub>X</sub>	C <sub>Y</sub>	S <sub>Y</sub>	DATA <sub>Y</sub>	
<b>Bit Operations</b>														
GET1	1 0 1 0 1	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	Bit position	C <sub>X</sub>	S <sub>A</sub>	0000H	C <sub>Y</sub>	0	0000H	When the bit specified by the lower 4 bits of DATA <sub>B</sub> is 0
		C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	Bit position	C <sub>X</sub>	S <sub>A</sub>	0001H	C <sub>Y</sub>	0	0000H	
SET1	1 0 1 1 0	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	Bit position	C <sub>X</sub>	S <sub>A</sub>	A bit in DATA <sub>A</sub> is set	C <sub>Y</sub>	0	0000H	Bit specification by the lower 4 bits of DATA <sub>B</sub>
CLR1	1 0 1 1 1	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	Bit position	C <sub>X</sub>	S <sub>A</sub>	A bit in DATA <sub>A</sub> is cleared	C <sub>Y</sub>	0	0000H	Bit specification by the lower 4 bits of DATA <sub>B</sub>
<b>Bit Check</b>														
ANDMSK	0 1 1 0 1	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	0	S <sub>A</sub>	A	0	S <sub>B</sub>	B	If ANDMSK = 0
		C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	1	S <sub>A</sub>	A	1	S <sub>B</sub>	B	If ANDMSK = 1
ORMSK	1 0 0 0 0	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	0	S <sub>A</sub>	A	0	S <sub>B</sub>	B	If ORMSK = 0
		C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	1	S <sub>A</sub>	A	1	S <sub>B</sub>	B	If ORMSK = 1
<b>Data Conversion</b>														
CVT2AB	0 1 1 1 0	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>X</sub>	S <sub>X</sub>	Converted A data	C <sub>Y</sub>	0	0000H	Absolute value — twos complement
CVTAB2	0 1 1 1 1	C <sub>A</sub>	S <sub>A</sub>	A	C <sub>B</sub>	S <sub>B</sub>	B	C <sub>X</sub>	S <sub>X</sub>	Converted A data	C <sub>Y</sub>	0	0000H	Twos complement — absolute value
<b>Adjustment of Double Precision Numbers</b>														
ADJL	1 0 1 0 0	C <sub>A</sub>	0	A	C <sub>B</sub>	1	B	C <sub>X</sub>	0	A - 1	C <sub>Y</sub>	0	0000H-B	A ≠ 0 AND B ≠ 0
		C <sub>A</sub>	1	A	C <sub>B</sub>	0	B	C <sub>X</sub>	1	A - 1	C <sub>Y</sub>	1	0000H-B	A ≠ 0 AND B ≠ 0
		C <sub>A</sub>	0	A	C <sub>B</sub>	1	0000H	C <sub>X</sub>	0	A	C <sub>Y</sub>	0	0000H	
		C <sub>A</sub>	0	0000H	C <sub>B</sub>	1	B	C <sub>X</sub>	1	0000H	C <sub>Y</sub>	1	B	B ≠ 0
		C <sub>A</sub>	1	A	C <sub>B</sub>	0	0000H	C <sub>X</sub>	1	A	C <sub>Y</sub>	1	0000H	
		C <sub>A</sub>	1	0000H	C <sub>B</sub>	0	B	C <sub>X</sub>	0	0000H	C <sub>Y</sub>	0	B	B ≠ 0
		C <sub>A</sub>	0	A	C <sub>B</sub>	0	B	C <sub>X</sub>	0	A	C <sub>Y</sub>	0	B	
		C <sub>A</sub>	1	A	C <sub>B</sub>	1	B	C <sub>X</sub>	1	A	C <sub>Y</sub>	1	B	

**Notes:** \* If an overflow occurs as the result of A + B, DATA<sub>Y</sub> = 0001H and if no overflow, DATA<sub>Y</sub> = 0000H.

† This indicates the number of consecutive zeros from the MS<sub>B</sub> of DATA<sub>X</sub>. This number is used to calculate the number of shifts to be performed by subsequent processing.

**GE Instructions**



**Bit Assignments**

**F/L [Full/Left]:** F/L bit = 0 indicates that the GE instruction is used alone, whereas F/L bit = 1 indicates that the GE instruction is used in conjunction with an AG/FC instruction.

**XCH [Exchange]:** XCH bit = 1 indicates that the data from A side and B side are to be exchanged before the two data tokens enter the Queue.

**OP [OP code]:** These two bits select an operation to be performed. See table 21.

**Table 21. OP Bits**

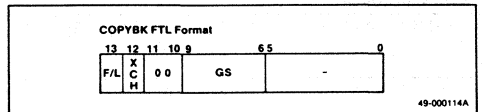
OP	Operation
00	COPYBK (Copy block)
01	COPYM (Copy multiple)
11	SETCTL (Set control field)

**GS [Generation Size]:** These four bits determine the number of copies of a token to be made. A minimum of 2 and a maximum of 17 copies can be made using a GE instruction.

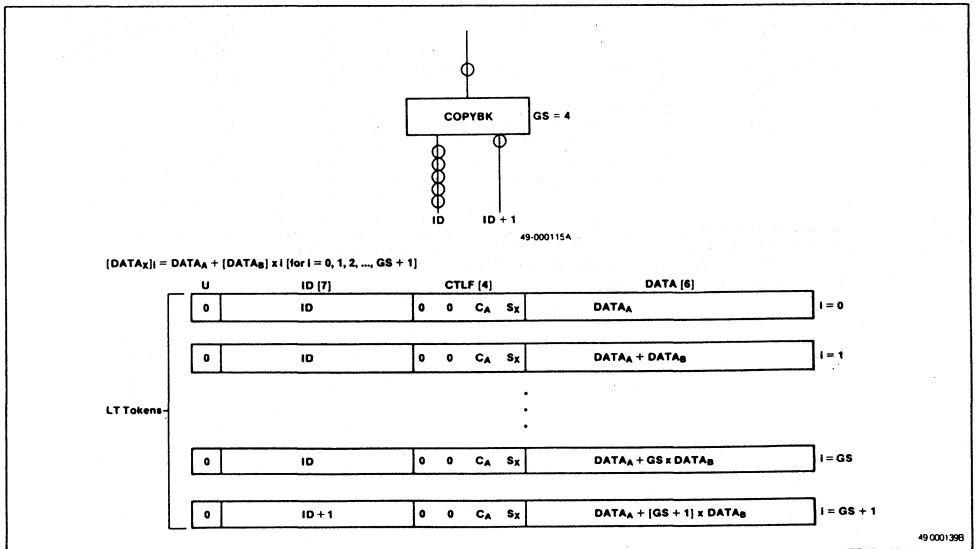
**CTLFD [Control Field]:** This field is used with Set Control Field (SETCTL) instruction. The data in CTLFD field further specifies the types of operations to be performed by the SETCTL instruction.

**COPYBK [Copy Block]**

COPYBK is used to duplicate a block of tokens from a single token. These duplicated tokens have exactly the same ID as the original token except the token copied last which has the original token's ID plus one. The number of tokens to be generated is specified by the GS field, and the COPYBK instruction generates exactly GS + 2 tokens. The data fields of the tokens being duplicated can also be incremented or decremented in a systematic manner. The incremental (or decremental) step value is contained in DATA<sub>B</sub>. The tokens generated are sent to the Link Table. The series of LT tokens output by the instruction is shown in figure 23.



**Figure 23. COPYBK Instruction Output**



### COPYM [Copy Multiple]

COPYM is used to generate multiple tokens from a single token. Each generated token has a different ID value. The number of tokens generated from the original token is  $GS + 2$ . The data field of the tokens being generated can also be incremented or decremented in a systematic manner. The incremental (or decremental) step value is contained in  $DATA_B$ . The

generated tokens are sent to the Link Table as LT tokens. The series of LT tokens output by the COPYM instruction is shown in figure 24.

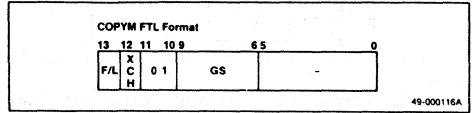
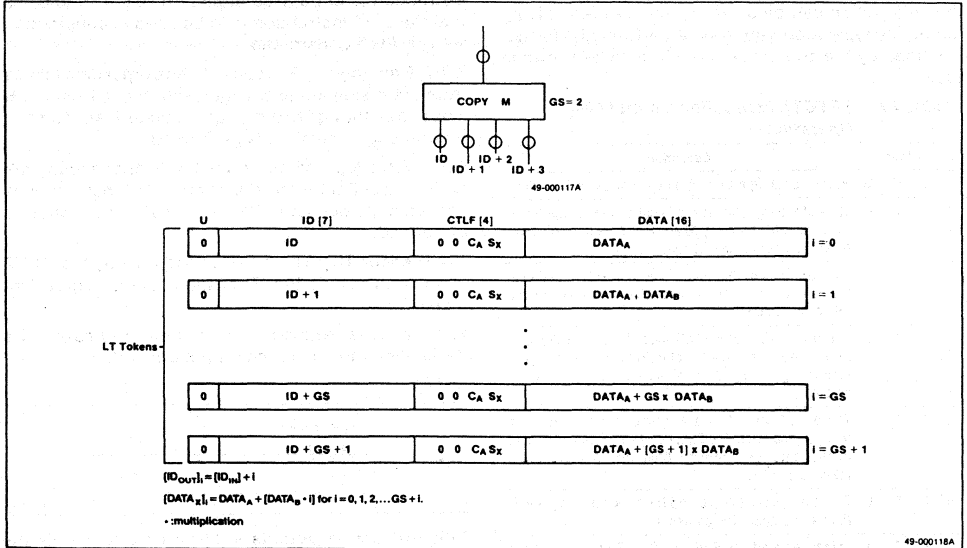
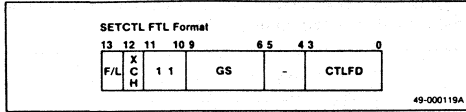


Figure 24. COPYM Instruction Output Tokens



**SETCTL [Set Control Field]**



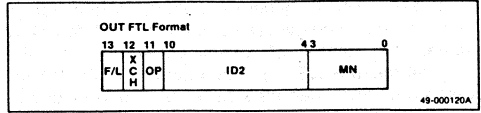
SETCTL is used to read and rewrite the contents of the Link Table and the Function Table. Since it can change the contents of the Link Table and the Function Table, this instruction can be used to write a self-modifying code. The type of operation to be performed is further specified by the contents of CTLFD field, as shown in table 22.

**Table 22. SETCTL Instruction Control Field Operation**

CTLFD	Operation
0 0 C S	Normal data. Operation is exactly the same as COPYM.
1 1 0 0	The data field of this token is used to set a location in the Link Table memory (C and S bits are not included.) After the data is set, the token is deleted.
1 1 0 1	The data field of this token is used to set a location in the Function Table Right field. After the data is set, the token is deleted.
1 1 1 0	The lower 14 bits of the data field of this token are used to set a location in the Function Table Left field (higher bits are ignored.) After the data is set, the token is deleted.
1 1 1 1	The lower 10 bits of the data field of this token are used to set a location in the Function Table Temporary field (higher bits are ignored.) After the data is set, the token is deleted.
1 0 0 0	This token reads the LT address indicated by the ID field and outputs the contents.
1 0 0 1	This token reads the Function Table Right field address indicated by the ID field and outputs the contents.
1 0 1 0	This token reads the Function Table Left field address indicated by the ID field and outputs the contents.
1 0 1 1	This token reads the Function Table Temporary field address indicated by the ID field and outputs the contents.
0 1 0 0	These tokens should not be generated by the Processing
0 1 0 1	Unit. They are operating-mode-related tokens.
0 1 1 0	
0 1 1 1	

**Note:** The set or write operation is performed at the address indicated by the ID field of the token.

**OUT Instructions**



**Bit Assignments**

**F/L [Full/Left]:** F/L bit = 0 indicates that the OUT instruction is to be used alone. F/L bit = 1 indicates that the OUT instruction is to be used in conjunction with an AG/FC instruction.

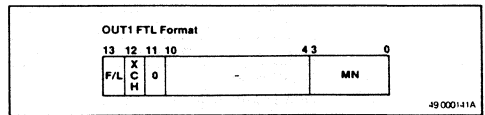
**XCH [Exchange]:** If XCH bit = 1, the output data tokens from the A side are exchanged with those from the B side before they go to the Output Queue. If XCH bit = 0, no exchange operation is performed.

**OP [OP Code]:** This bit is used to further specify the OUT instruction. If OP = 0, then OUT1 instruction is performed, whereas if OP = 1, OUT2 instruction is performed.

**ID2 [Second ID]:** This field is used only by the OUT2 instruction. ID2 is the ID of the second output data token.

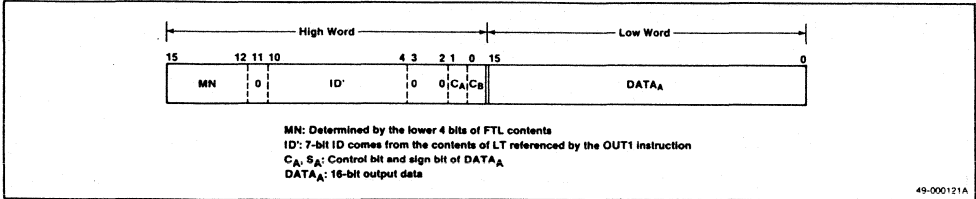
**MN [Module Number]:** This field indicates the destination module of the output data token.

**OUT1**



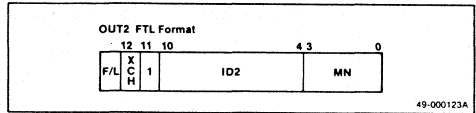
This instruction outputs a 32-bit data token via the Output Data Bus (ODB). Since the size of the ODB is 16 bits, a 32-bit output data token is divided into two 16-bit words and output one 16-bit word at a time. The format of an output data token is shown in figure 24.

**Figure 25. OUT1 Output Token Format**



## OUT2

This instruction outputs two 32-bit data tokens via ODB. Since the ODB is 16 bits wide, each 32-bit token is divided into two 16-bit words and output one 16-bit word at a time. This instruction is useful when a double precision number is to be output. The formats of two output data tokens are shown in figure 25.



**Figure 26. OUT2 Output Tokens Format**

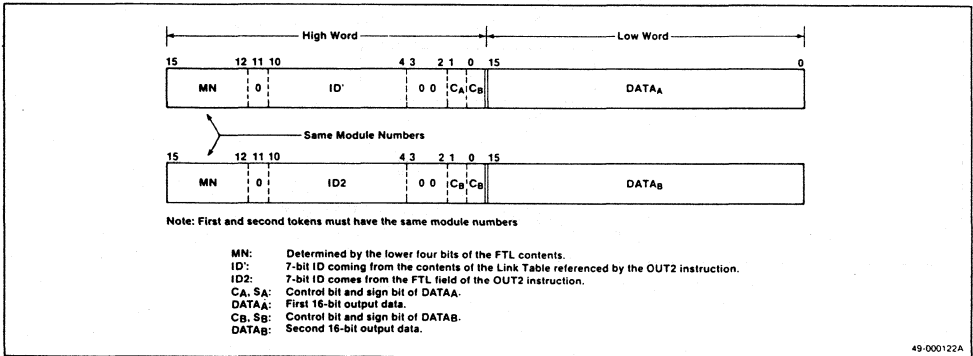




Figure 27. Data-Flow Graph Explanation

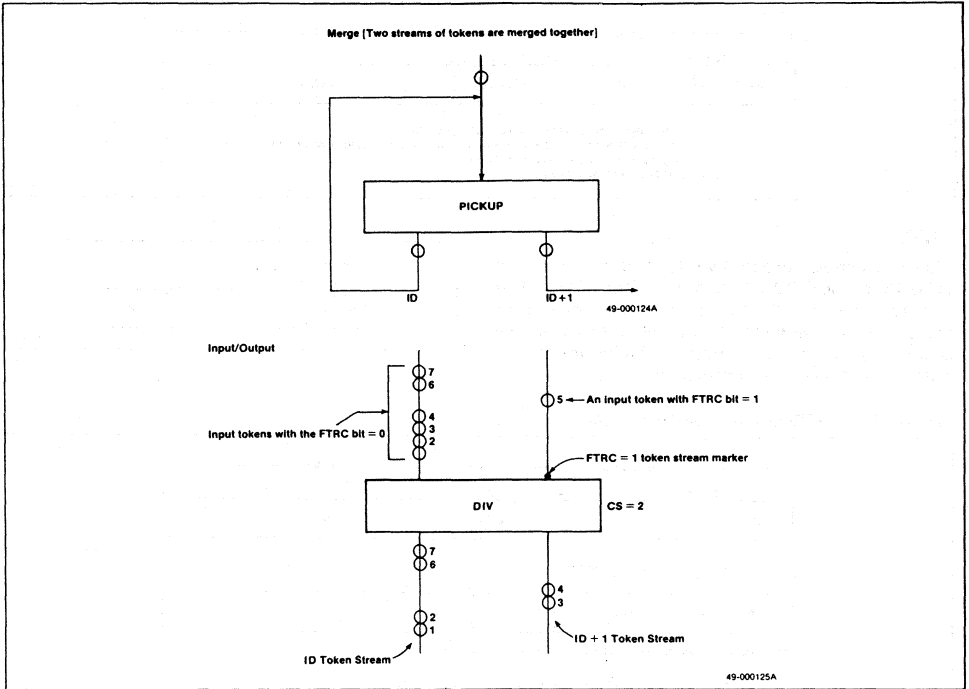
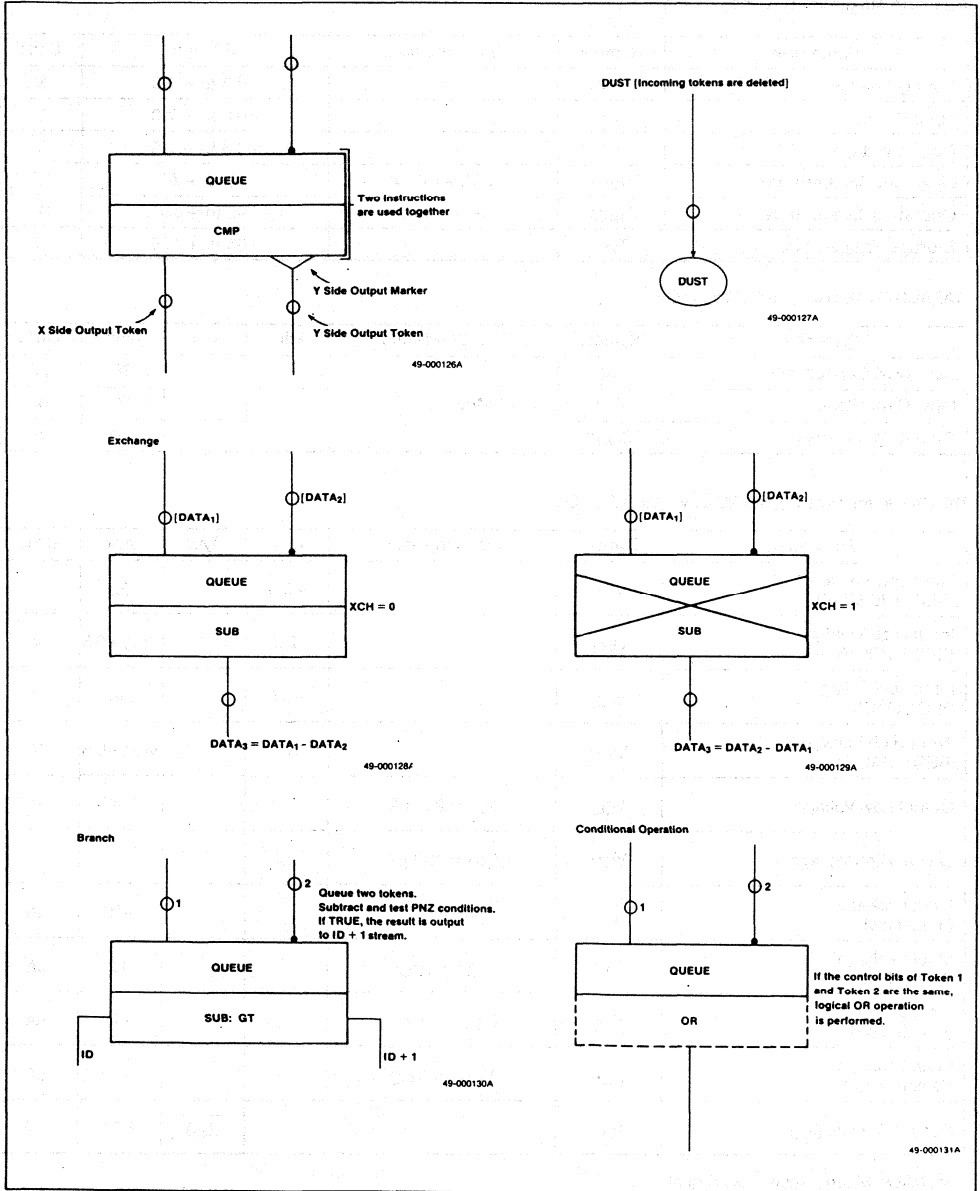


Figure 27. Data-Flow Graph Explanation (cont)



## μPD7281

### ELECTRICAL SPECIFICATIONS μPD7281D

Absolute Maximum Ratings ( $T_a = 25^\circ\text{C}$ )

Parameter	Symbol	Test Conditions	Ratings	Units
Supply Voltage	VCC		-0.5 to + 7.0	V
Input Voltage	V <sub>I</sub>		-0.5 to + 7.0	V
Output Voltage	V <sub>O</sub>		-0.5 to + 7.0	V
Operating Temperature	T <sub>opt1</sub>	with air flow	-8 to + 70	°C
Operating Temperature	T <sub>opt2</sub>	no air flow	-8 to + 45	°C
Storage Temperature	T <sub>stg</sub>		-65 to + 150	°C

CAPACITANCE ( $T_a = 25^\circ\text{C}$ , VCC = 0V)

Parameter	Symbol	Test Conditions	Min.	Typ.	Max.	Units
CLK, SCK Capacitance	C <sub>∅</sub>	Fc = 1 MHz			20	pF
Input Capacitance	C <sub>IN</sub>				10	pF
Output Capacitance	C <sub>OUT</sub>				20	pF

DC Characteristics ( $T_a = +25^\circ\text{C}$ , VCC = +5 ± 10%)

Parameter	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Input Low Voltage 1 (RESET/IDB15-0)	V <sub>IL1</sub>		-0.5		0.7	V
Input High Voltage 1 (RESET/IDB15-0)	V <sub>IH1</sub>		2.2		VCC+0.5	V
Input Low Voltage 2 REQ/, OACK/	V <sub>IL2</sub>		-0.5		0.45	V
Input High Voltage 2 IREQ/, OACK/	V <sub>IH2</sub>		3.5		VCC+0.5	V
Output Low Voltage	V <sub>OL</sub>	I <sub>OL</sub> = 2.0 mA			0.45	V
Output High Voltage	V <sub>OH</sub>	I <sub>OH</sub> = 400 μA	2.4			V
Input Leakage (Input Low)	I <sub>LIL</sub>	V <sub>IN</sub> = 0V			-10	μA
Input Leakage (Input High)	I <sub>LIH</sub>	V <sub>IN</sub> = VCC			10	μA
Input Leakage (Output Low)	I <sub>LOL</sub>	V <sub>OUT</sub> = 0.47V			-10	μA
Input Leakage (Output High)	I <sub>LOH</sub>	V <sub>OUT</sub> = VCC			-10	μA
Supply Current (VCC)	I <sub>CC</sub>			320	500	mA

"SIGNAL/" Means Active Low Signal

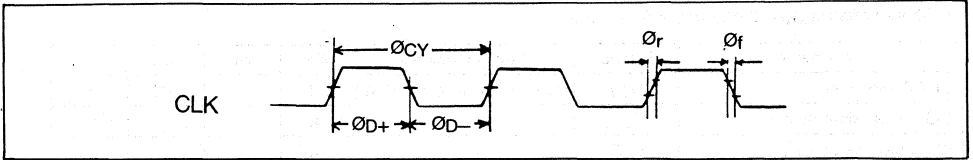
## ELECTRICAL SPECIFICATIONS μPD7281D

AC Characteristics (T<sub>a</sub> = 25°C, V<sub>CC</sub> = 5V ± 10%)

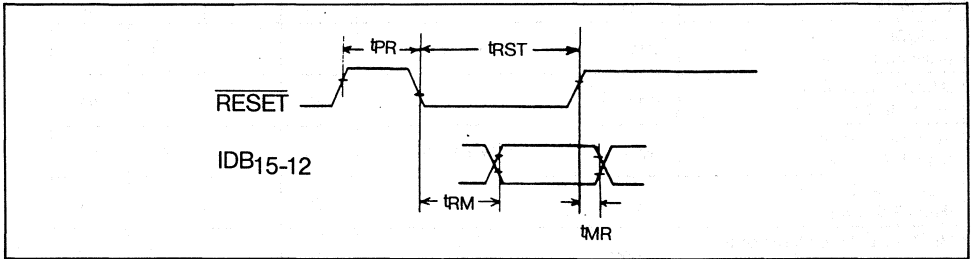
Parameter	Symbol	Test Conditions	Min.	Typ	Max.	Units
CLK Cycle Time	∅CY	Measured Between 1V&3V	100		500	ns
CLK Pulse width high	∅D+		40			ns
CLK Pulse width low	∅D-		40			ns
CLK Rise time	∅r				10	ns
CLK Fall time	∅f				10	ns
IACK/Delay time -1 (from IREQ/down)	tWA -1	Higher 16 bit	20		50	ns
IACK/Delay time -2 (from IREQ/up)	tWA -2		20		55	ns
IACK/Delay Time -3 (from IREQ/down)	tWA -3	Lower 16 bit	20		70	ns
IACK/Delay time -4 (From IREQ/up)	tWA -4		20		70	ns
Min Time between transitions on IREQ/and IACK/	tAW		15			ns
IREQ/rise time	t <sub>r</sub>				10	ns
IREQ/fall time	t <sub>f</sub>				10	ns
Data set up time (before IREQ/up)	tDW		40			ns
Data Hold time (after IREQ/up)	tWD		0			ns
OREQ/Delay time -1 (from OACK/down)	tAR -1		15		35	ns
OREQ/Delay time -2 (from OACK/up)	tAR -2		15		45	ns
OREQ/Delay time -3 (from OACK/down)	tAR -3		15		35	ns
OREQ/Delay time -4 (from OACK/up)	tAR -4		15		45	ns
Min time between transitions on OREQ/ and OACK/	tRA		15			ns
OACK/rise time	t <sub>r</sub>				10	ns
OACK/fall time Data	t <sub>f</sub>				10	ns
Access Time (after OREQ/down)	tRD				25	ns
Data Float Time (after OREQ/up)	tDF		10		35	ns
Pre-RESET/high time	tPR		2			OCY
RESER/low time	tRST		6			OCY
Module number data setup time (after RESET/down)	tRM				2	OCY
Module number data hold time (after RESET/up)	tMR		0			ns

"Up" Means On Rise Edge, "Down" Means On Falling Edge, "Signal/" Means Active Low Signal

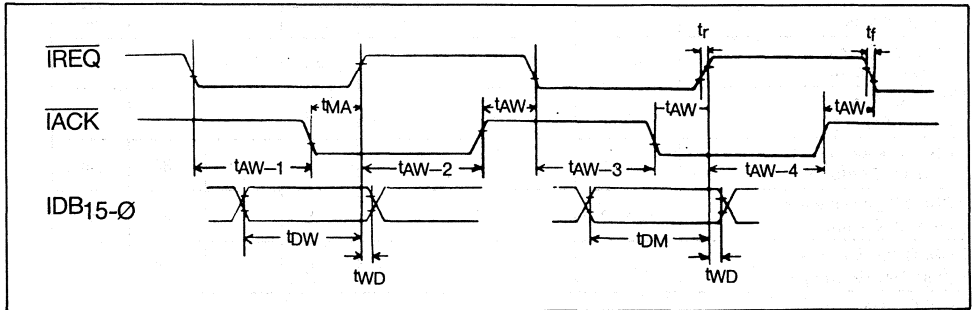
**Preliminary Electrical Specification**



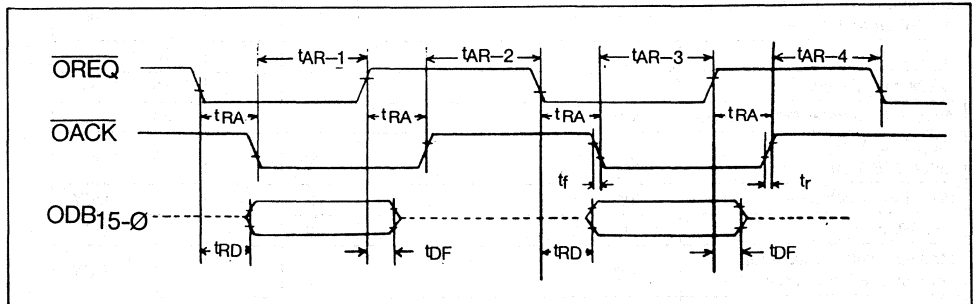
**Module Number Setting**



**Input Handshake timing**



**Output Handshake timing**



### Description

The NEC  $\mu$ PD9305 memory access and general bus interface chip (MAGIC) is a peripheral LSI support device for the  $\mu$ PD7281 image pipelined processor (ImPP). The  $\mu$ PD7281 is a data flow architecture processor that supports high speed image and signal processing applications. The  $\mu$ PD9305 chip can support from one to eight  $\mu$ PD7281s and also interfaces to both 8-bit and 16-bit host processors.

The  $\mu$ PD9305's powerful interface capabilities allow it to support basic interface operations, object program load, read/write/modify operations on image memory, and multiple  $\mu$ PD7281 image memory accesses.

Since the  $\mu$ PD7281 ImPP does not use direct addressing, the memories in a  $\mu$ PD7281 processor system can be seen as processing modules with unique module numbers. These separate modules must output memory access tokens containing their own unique address, data, and control signals. The modules must perform the necessary processing, and then output the result of the access as another memory access token. To do this, the multiple  $\mu$ PD7281 modules require external circuitry to process the memory access tokens that they output. In addition, this same circuitry is required to organize the data output from the memory into token format.

Circuitry is also needed between the host processor and the  $\mu$ PD7281s to organize the data from the host into token format and to return the data output from the  $\mu$ PD7281s into the form required by the host processor. Finally, tokens may have to be returned to other  $\mu$ PD7281s in token form for further processing.

The  $\mu$ PD9305 simplifies the above operations by keeping the data in the most convenient form. The  $\mu$ PD9305 replaces approximately 80 medium/small scale integrated devices with a single integrated circuit.

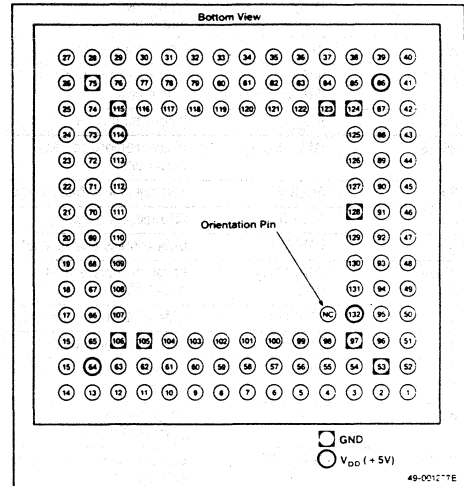
### Features

- High performance image memory interface
- Reduces external circuits required for ImPP system
- Simplifies host interface
- Up to 24-bit image memory addressing
- Up to 18-bit image memory data
- Register file for memory access
- Refresh control of image memory
- Functions with separate DMA controller
- Single +5 V power supply
- CMOS technology for lower power consumption

### Ordering Information

Part Number	Package Type
$\mu$ PD9305R	132-pin ceramic grid array

### Pin Configuration



### Pin Identification

No.	Symbol	Function
1	CLK	Clock input
2-4	D <sub>10</sub> , D <sub>12</sub> , D <sub>15</sub>	Bidirectional data bus bits
5	$\bar{O}ACK$	Output acknowledge input
6	$\bar{O}REQ$	Output request output
7	IDB <sub>14</sub>	Input data bus bit
8	ODB <sub>14</sub>	Output data bus bit
9	IDB <sub>11</sub>	Input data bus bit
10, 11	ODB <sub>11</sub> , ODB <sub>8</sub>	Output data bus bits
12	IDB <sub>9</sub>	Input data bus bit
13	ODB <sub>5</sub>	Output data bus bit
14	IDB <sub>8</sub>	Input data bus bit
15	ODB <sub>2</sub>	Output data bus bit
16	IDB <sub>7</sub>	Input data bus bit
17	ODB <sub>2</sub>	Output data bus bit

## Pin Identification (Cont)

No.	Symbol	Function
18	IDB <sub>6</sub>	Input data bus bit
19	MN <sub>2</sub>	Module number output
20	IDB <sub>4</sub>	Input data bus bit
21	IMA <sub>22</sub>	Image memory address output bit
22	IDB <sub>2</sub>	Input data bus bit
23, 24	IMA <sub>18</sub> , IMA <sub>15</sub>	Image memory address output bits
25	IDB <sub>0</sub>	Input data bus bit
26-28	IMA <sub>12</sub> -IMA <sub>10</sub>	Image memory address output bits
29	SOLBSY	Self object load busy output
30	CPURQ	CPU request output
31	DMAAEN	DMA address enable input
32-34	IMA <sub>5</sub> , IMA <sub>2</sub> , IMA <sub>0</sub>	Image memory address output bits
35	DMAAKT	DMA / 1 acknowledge input
36	DMARQ1	DMA / 1 request output
37	IMD <sub>13</sub>	Bidirectional image memory data bus bit
38	IMAK	Image memory acknowledge input
39-42	IMD <sub>10</sub> -IMD <sub>7</sub>	Bidirectional image memory data bus bits
43	A <sub>0</sub>	Address select input
44, 45	IMD <sub>3</sub> , IMD <sub>1</sub>	Bidirectional image memory data bus bits
46	IMWR	Image memory write output
47	WR	Write input
48, 49	D <sub>2</sub> , D <sub>5</sub>	Bidirectional data bus bits
50	CS	Chip select input
51, 52	D <sub>8</sub> , D <sub>9</sub>	Bidirectional data bus bits
53	GND	Ground
54, 55	D <sub>11</sub> , D <sub>14</sub>	Bidirectional data bus bits
56	IREQ	Input request input
57	IACK	Input acknowledge output
58	IDB <sub>13</sub>	Input data bus bit
59	ODB <sub>13</sub>	Output data bus bit
60	IDB <sub>10</sub>	Input data bus bit
61-63	ODB <sub>10</sub> , ODB <sub>7</sub> , ODB <sub>6</sub>	Output data bus bits
64	V <sub>DD</sub>	+5 V power supply
65, 66	ODB <sub>3</sub> , ODB <sub>1</sub>	Output data bus bits
67	IDB <sub>5</sub>	Input data bus bit
68	MN <sub>1</sub>	Module number output bit

## Pin Identification (Cont)

No.	Symbol	Function
69, 70	IMA <sub>23</sub> , IMA <sub>21</sub>	Image memory address output bits
71	IDB <sub>1</sub>	Input data bus bit
72-74	IMA <sub>17</sub> , IMA <sub>14</sub> , IMA <sub>13</sub>	Image memory address output bits
75	GND	Ground
76, 77	IMA <sub>9</sub> , IMA <sub>8</sub>	Image memory address output bits
78	INBUSY	Input to ImPP busy output
79, 80	IMA <sub>4</sub> , IMA <sub>1</sub>	Image memory address output bits
81	IMD <sub>17</sub>	Bidirectional image memory data bus bit
82	DMAAK2	DMA / 2 acknowledge input
83	DMARQ2	DMA / 2 request output
84, 85	IMD <sub>12</sub> , IMD <sub>11</sub>	Bidirectional image memory data bus bits
86	V <sub>DD</sub>	+5 V power supply
87, 88	IMD <sub>6</sub> , IMD <sub>5</sub>	Bidirectional image memory data bus bits
89	A <sub>1</sub>	Address select input
90	IMD <sub>0</sub>	Bidirectional image memory data bus bit
91	IMRF	Image memory refresh output
92	D <sub>0</sub>	Bidirectional data bus bit
93	RD	Read input
94-96	D <sub>4</sub> , D <sub>6</sub> , D <sub>7</sub>	Bidirectional data bus bits
97	GND	Ground
98	D <sub>13</sub>	Bidirectional data bus bit
99	IPPRST	Image pipelined processor reset output
100	IDB <sub>15</sub>	Input data bus bit
101	ODB <sub>15</sub>	Output data bus bit
102	IDB <sub>12</sub>	Input data bus bit
103, 104	ODB <sub>12</sub> , ODB <sub>9</sub>	Output data bus bits
105, 106	GND	Ground
107	ODB <sub>0</sub>	Output data bus bit
108, 109	MN <sub>3</sub> , MN <sub>0</sub>	Module number output bits
110	IDB <sub>3</sub>	Input data bus bit
111-113	IMA <sub>20</sub> , IMA <sub>19</sub> , IMA <sub>16</sub>	Image memory address outputs
114	V <sub>DD</sub>	+5 V power supply
115	GND	Ground
116-118	IMA <sub>7</sub> , IMA <sub>6</sub> , IMA <sub>3</sub>	Image memory address outputs

## Pin Identification (Cont)

No.	Symbol	Function
119	RESET	Reset input
120-122	IMD <sub>16</sub> -IMD <sub>14</sub>	Bidirectional image memory data bus bits
123,124	GND	Ground
125,126	IMD <sub>4</sub> ,IMD <sub>2</sub>	Bidirectional image memory data bus bits
127	IMRD	Image memory read output
128	GND	Ground
129	ERR	Error output
130,131	D <sub>1</sub> ,D <sub>3</sub>	Bidirectional data bus bits
132	V <sub>DD</sub>	+5 V power supply

## Pin Functions

Table 1 shows the μPD9305 pins in their particular functional groups. The paragraphs that follow table 1 describe the operation of the pins in each group.

All unused input or output pins should be pulled up to V<sub>DD</sub> or down to GND through a 2K-3K ohm resistor.

Table 1. μPD9305 Pins by Function

I/O	Signal	No.
I	CLK	1
	RESET	119
<b>Status</b>		
0	ERR	129
	SOLBSY	29
	CPURQ	30
	INBUSY	78
<b>Host Interface</b>		
I/O	WR	47
	RD	93
	CS	50
	A <sub>0</sub>	43
	A <sub>1</sub>	89
	D <sub>0</sub>	92
	D <sub>1</sub>	130
	D <sub>2</sub>	48
	D <sub>3</sub>	131
	D <sub>4</sub>	94
	D <sub>5</sub>	49
	D <sub>6</sub>	95
	D <sub>7</sub>	96
	D <sub>8</sub>	51
	D <sub>9</sub>	52
D <sub>10</sub>	2	
D <sub>11</sub>	54	
D <sub>12</sub>	3	
D <sub>13</sub>	98	
D <sub>14</sub>	55	
D <sub>15</sub>	4	
<b>DMA</b>		
0	DMARQ1	36
	DMARQ2	83
I	DMAAK1	35
	DMAAK2	82
	DMAAEN	31



Table 1. μPD9305 Pins by Function (Cont)

I/O	Signal	No.
μPD7281 Interface		
	MN <sub>0</sub>	109
0	MN <sub>1</sub>	68
	MN <sub>2</sub>	19
	MN <sub>3</sub>	108
0	OREQ	6
	OACK	5
1	IREQ	56
0	IACK	57
	IPPRST	99
	ODB <sub>0</sub>	107
	ODB <sub>1</sub>	66
	ODB <sub>2</sub>	17
	ODB <sub>3</sub>	65
	ODB <sub>4</sub>	15
	ODB <sub>5</sub>	13
	ODB <sub>6</sub>	63
0	ODB <sub>7</sub>	62
	ODB <sub>8</sub>	11
	ODB <sub>9</sub>	104
	ODB <sub>10</sub>	61
	ODB <sub>11</sub>	10
	ODB <sub>12</sub>	103
	ODB <sub>13</sub>	59
	ODB <sub>14</sub>	8
	ODB <sub>15</sub>	101
	IDB <sub>0</sub>	25
	IDB <sub>1</sub>	71
	IDB <sub>2</sub>	22
	IDB <sub>3</sub>	110
	IDB <sub>4</sub>	20
	IDB <sub>5</sub>	67
	IDB <sub>6</sub>	18
1	IDB <sub>7</sub>	16
	IDB <sub>8</sub>	11
	IDB <sub>9</sub>	12
	IDB <sub>10</sub>	60
	IDB <sub>11</sub>	9
	IDB <sub>12</sub>	102
	IDB <sub>14</sub>	7
	IDB <sub>15</sub>	100

Table 1. μPD9305 Pins by Function (Cont)

I/O	Signal	No.
Image Memory Interface		
1	IMAK	38
	IMRD	127
0	IMWR	46
	IMRF	91
	IMD <sub>0</sub>	90
	IMD <sub>1</sub>	45
	IMD <sub>2</sub>	126
	IMD <sub>3</sub>	44
	IMD <sub>4</sub>	125
	IMD <sub>5</sub>	88
	IMD <sub>6</sub>	87
	IMD <sub>7</sub>	42
I/O	IMD <sub>8</sub>	41
	IMD <sub>9</sub>	40
	IMD <sub>10</sub>	39
	IMD <sub>11</sub>	85
	IMD <sub>12</sub>	84
	IMD <sub>13</sub>	37
	IMD <sub>14</sub>	122
	IMD <sub>15</sub>	121
	IMD <sub>16</sub>	120
	IMD <sub>17</sub>	81

**Table 1. μPD9305 Pins by Function (Cont)**

I/O	Signal	No.
Image Memory Interface		
	IMA <sub>0</sub>	34
	IMA <sub>1</sub>	80
	IMA <sub>2</sub>	33
	IMA <sub>3</sub>	118
	IMA <sub>4</sub>	79
	IMA <sub>5</sub>	32
	IMA <sub>6</sub>	117
	IMA <sub>7</sub>	116
	IMA <sub>8</sub>	77
	IMA <sub>9</sub>	76
	IMA <sub>10</sub>	28
	IMA <sub>11</sub>	27
	IMA <sub>12</sub>	26
	IMA <sub>13</sub>	74
	IMA <sub>14</sub>	73
	IMA <sub>15</sub>	24
	IMA <sub>16</sub>	113
	IMA <sub>17</sub>	72
	IMA <sub>18</sub>	23
	IMA <sub>19</sub>	112
	IMA <sub>20</sub>	111
	IMA <sub>21</sub>	70
	IMA <sub>22</sub>	21
	IMA <sub>23</sub>	69

### CLK (Clock)

CLK is the single phase master clock input. The μPD9305 clock frequency can be independent of ImPP clock frequency.

### RESET (Reset)

RESET initializes the μPD9305. A reset places OREQ, IACK, the token I/O flip-flop, and IM access request signals at an inactive level. RESET resets the refresh address counter, refresh timer counter, and mode register to 0. RESET must be held low for a minimum of four μPD9305 or μPD7281 clock cycles, whichever is slower.

### V<sub>DD</sub> (Power)

V<sub>DD</sub> is the single +5 volt power supply.

### GND (Ground)

GND is the ground signal.

## Status Signal Pin Functions

### CPURQ (CPU Request)

CPURQ indicates to the host processor that the μPD9305 is ready to transfer a token to the host.

### INBUSY (Input Busy)

INBUSY indicates that tokens are being input to the first ImPP from the μPD9305.

### SOLBSY (Self Object Load Busy)

SOLBSY indicates that a self object load is being executed.

### ERR (Error)

ERROR indicates that an error was output from the ImPPs, the host has read an invalid output token, or that the host has input a token while INBUSY was active.

## Host Interface Signal Pin Functions

### RD (Read)

RD reads the contents of the internal registers specified by A<sub>1</sub> and A<sub>0</sub>.

### WR (Write)

WR writes an input from the data bus to the internal register specified by A<sub>1</sub> and A<sub>0</sub>.

### CS (Chip Select)

CS enables the RD or WR control signals.

### A<sub>0</sub>, A<sub>1</sub> (Address)

A<sub>0</sub> and A<sub>1</sub> select the internal register for a read or write operation.

### D<sub>0</sub>-D<sub>15</sub> (Data Bus)

The contents of the internal registers are read from or written to via data bus bits D<sub>0</sub>-D<sub>15</sub>.

## DMA Signal Pin Functions

### DMAAEN (Direct Memory Access Address Enable)

DMAAEN is used to indicate to the μPD9305 that an external DMA controller is putting DMA addresses on the address bus. During a DMA operation, DMA addresses (system memory addresses) are input to A<sub>0</sub> and A<sub>1</sub>. However, these addresses have no meaning for the μPD9305 and might alter register contents. For this reason, the μPD9305 operates as if A<sub>0</sub> and A<sub>1</sub> are both reset to 0 when DMAAEN is active (high).

**$\overline{\text{DMARQ1}}$  (Direct Memory Access Request 1)**

$\overline{\text{DMARQ1}}$  issues a request to an external DMA controller to transfer data from the host system memory to the  $\mu\text{PD9305}$ .

 **$\overline{\text{DMARQ2}}$  (Direct Memory Access Request 2)**

$\overline{\text{DMARQ2}}$  issues a request to an external DMA controller to transfer data from the  $\mu\text{PD9305}$  to the host system memory.

 **$\overline{\text{DMAAK1}}$  (Direct Memory Access Acknowledge 1)**

$\overline{\text{DMAAK1}}$  is issued by the external DMA controller to indicate to the  $\mu\text{PD9305}$  that  $\overline{\text{DMARQ1}}$  has been received.

 **$\overline{\text{DMAAK2}}$  (Direct Memory Access Acknowledge 2)**

$\overline{\text{DMAAK2}}$  is issued by the external DMA controller to indicate to the  $\mu\text{PD9305}$  that  $\overline{\text{DMARQ2}}$  has been received.

 **$\mu\text{PD7281}$  Interface Signal Pin Functions** **$\text{MN}_0\text{-MN}_3$  (Module Number)**

$\text{MN}_0\text{-MN}_3$  specify the module number of one ImPP. During a reset, one module number is output via  $\text{MN}_0\text{-MN}_3$ , the other via  $\text{IDB}_{12}\text{-IDB}_{15}$ .  $\text{MN}_0\text{-MN}_3$  are three-state pins.

 **$\overline{\text{OREQ}}$  (Output Request)**

$\overline{\text{OREQ}}$  signals to the first ImPP that the  $\mu\text{PD9305}$  is ready to transfer half a token.

 **$\overline{\text{OACK}}$  (Output Acknowledge)**

$\overline{\text{OACK}}$  signals to the  $\mu\text{PD9305}$  that a half token has been accepted by the first ImPP.

 **$\overline{\text{IREQ}}$  (Input Request)**

$\overline{\text{IREQ}}$  signals from the last ImPP that a half token is ready to be transferred from the ImPP to the  $\mu\text{PD9305}$ .

 **$\overline{\text{IACK}}$  (Input Acknowledge)**

$\overline{\text{IACK}}$  indicates to the last ImPP that the  $\mu\text{PD9305}$  has accepted the half token.

 **$\overline{\text{IPPRST}}$  (Image Pipelined Processor Reset)**

$\overline{\text{IPPRST}}$  resets the ImPPs during RESET or a command reset.

 **$\text{ODB}_0\text{-ODB}_{15}$  (Output Data Bus)**

$\text{ODB}_0\text{-ODB}_{15}$  transfer tokens from the  $\mu\text{PD9305}$  to the first ImPP.

 **$\text{IDB}_0\text{-IDB}_{15}$  (Input Data Bus)**

$\text{IDB}_0\text{-IDB}_{15}$  transfer tokens between the output of the last ImPP and the  $\mu\text{PD9305}$ .

**Image Memory Interface Signal Pin Functions****IMRD (Image Memory Read)**

IMRD requests a read of the contents of the image memory addressed by  $\text{IMA}_0\text{-IMA}_{23}$ .

**IMWR (Image Memory Write)**

IMWR requests a write to the image memory location addressed by  $\text{IMA}_0\text{-IMA}_{23}$ .

**IMRF (Image Memory Refresh)**

IMRF indicates an image memory refresh cycle.

**IMAK (Image Memory Acknowledge)**

IMAK indicates to the  $\mu\text{PD9305}$  that an image memory read, write or refresh has been completed.

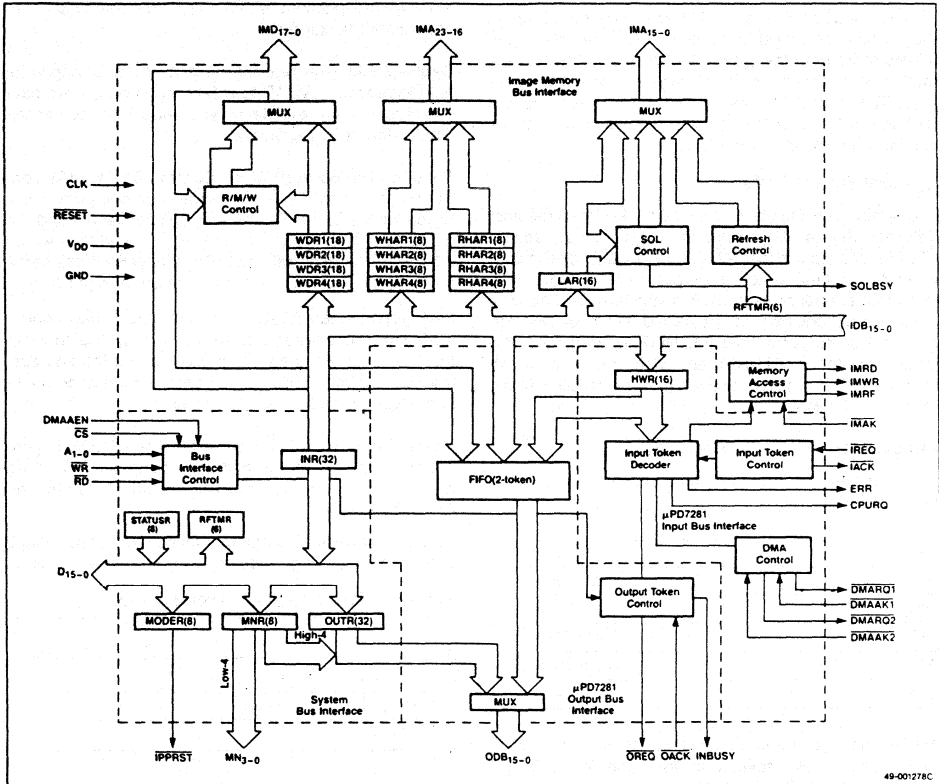
 **$\text{IMA}_0\text{-IMA}_{23}$  (Image Memory Address)**

$\text{IMA}_0\text{-IMA}_{23}$  supplies the image memory address for a read or write operation or for DRAM refresh ( $\text{IMA}_0\text{-IMA}_9$  only).

 **$\text{IMD}_0\text{-IMD}_{17}$  (Image Memory Data)**

$\text{IMD}_0\text{-IMD}_{17}$  is the bidirectional data bus for transferring data to and from the image memory.

## μPD9305 Block Diagram



### Functional Description

The μPD9305 has the following functional units:

- μPD7281 input bus interface
- μPD7281 output bus interface
- System bus interface
- Image memory bus interface
  - Register file
  - R/M/W control
  - Self object load control
  - Image memory refresh control

### μPD7281 Input Bus Interface

After the last ImPP outputs a token, the input bus interface determines whether the token should be an output token to the host CPU, to the image memory, or to the output bus interface block. The high order 16 bits of the token output from the last ImPP are latched into the high word register (HWR) and then decoded by the input token decoder to determine the token type.

### μPD7281 Output Bus Interface

The output bus interface logic transmits tokens through the multiplexer (MUX) to the first ImPP. The transmitted tokens come from the system bus interface, the μPD7281 input bus interface, or the image memory bus interface. The output bus interface uses a priority control mechanism to prevent collisions between the tokens coming from the different blocks.

### System Bus Interface

The system bus interface receives a token from the host CPU for the ImPPs, sends it to the output register (OUTR), and signals the output bus interface. Conversely, it sends a token, which is output from the last ImPP, through the input register (INR) to the host CPU according to instructions from the host CPU. The host CPU can set input or output modes (MODER register), read the status register (STATUSR), set image memory refresh timing (RFTMR register), and set module numbers (MNR) for two μPD7281s.

### Image Memory Bus Interface

The image memory bus interface accepts the following five types of tokens:

Token	Description
WHA	Write high address
WLA	Write low address
WD	Write data
RHA	Read high address
RLA	Read low address

Tokens have a 16-bit data value, so the address is transferred in two tokens to form the 24-bit image memory address. The lower 16-bits of the image memory address are latched in the lower address register.

The image memory bus interface also performs read/modify/write functions with the R/M/W control logic and provides a register file.

**Register File.** The register file is used for storing write high addresses (WHAR/four 8-bit registers), write data (WDR/four 18-bit registers), and read high addresses (RHAR/four 8-bit registers).

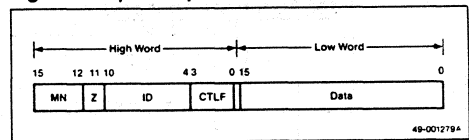
**Read/Modify/Write (R/M/W) Control.** The R/M/W control reads a word from the image memory, performs a logical operation (AND, OR, or XOR) between it and the contents of a write data register (WDR), and then writes it back to a location referenced by the WHAR (the same lower 16-bit address, but a different upper eight bits).

**Self Object Load (SOL).** The self object load control loads ImPP object programs stored in image memory into the ImPPs. When the SOL is given a starting address, the SOL control automatically generates the appropriate addresses to read the image memory.

**Image Memory Refresh Control.** The μPD9305 generates a 10-bit address and the timing for refreshing dynamic image memories. The timing is set by the RFTMR register.

Figure 1 shows the input/output token format and table 2 shows how the image memory access tokens function.

**Figure 1. Input/output Token Format**



**Table 2. Image Memory Access Tokens<sup>(1)</sup>**

MN	Z	ID	CTLF	Data	Function	Operation	
0001	-	MN'	ID'	----	Image memory read low address	Image memory read (RHAR1 reference)	R
		1 1 1	----	-----	Image memory read high address	Read high address register (RHAR1) set (Note 2)	S
0010	-	MN'	ID'	----	Image memory read low address	Image memory read (RHAR2 reference)	R
		1 1 1	----	-----	Image memory read high address	Read high address register (RHAR2) set (Note 2)	S
0011	-	MN'	ID'	----	Image memory read low address	Image memory read (RHAR3 reference)	R
		1 1 1	----	-----	Image memory read high address	Read high address register (RHAR3) set (Note 2)	S
0100	-	MN'	ID'	----	Image memory read low address	Image memory read (RHAR4 reference)	R
		1 1 1	----	-----	Image memory read high address	Read high address register (RHAR4) set (Note 2)	S
		0 0 0 0	DIR	----	Image memory write low address	Image memory write (referencing WHAR and WDR selected by DIR)	W
		0 0 1 --	DIR	----	Image memory write high address	Set write high address register (WHAR) selected by DIR	S
0101	-	0 1 0 --	DIR	-- C,S	Image memory write data register	Set write data register (WDR) selected by DIR	S
		0 1 1 --	DIR	----	Image memory read high address	Set read high address register (RHAR) selected by DIR	S
		1 0 0 MASK	DIR	----	Read/write low address	Read/modify/write	RW
		1 0 1 --	DIR	----	Read/write low address	Read / modify / write (write CS bits selects mask)	RW
		0 0 ----	DIR	----	Load starting low address	Self object load	R
0110	-	0 1 ----	DIR	----	Load starting low address	Self object load MN of output token is SOLMN)	R
		1 ----	--	-----	SOLMN	Set SOLMN for self object load	S

**Notes:**

(1) The following definitions refer to the above table:

- MN: Module number
- Z: Always 0
- ID: Identifier
- CTLF: Control field
- ID': ID used for next circulation
- MN': MN used for next circulation (MN ≠ 111)
- DIR: Specifies registers for memory image access
- MASK: Specifies the modify mode
- : Do not care
- S: Set
- R: Read
- W: Write

(2) When RHASEL of the mode register is 1, the tokens become image memory read (request) tokens

Table 3 shows module number (MN) values and the five token types (refer to figure 12).

The five token types are:

- (1) Output request data to the host
- (2) Image memory access data
- (3) DMA request data
- (4) Pass data
- (5) Delete data

**Table 3. MN Values and Token Types**

Token Type	MN	ID	Function	Abbreviation
(1)	0 0 0 0	x x x x x x x	μPD7281 output data to host	CPU
(2)	0 0 0 1	MN' ID'	Image memory read1 (RHAR1 select)	IMR
		x x x x x x x	RHAR1 set (Note 2)	
	0 0 1 0	MN' ID'	Image memory read2 (RHAR2 select)	
		x x x x x x x	RHAR2 set (Note 2)	
	0 0 1 1	MN' ID'	Image memory read3 (RHAR3 select)	
		— —	RHAR3 set (Note 2)	
	0 1 0 0	MN' ID'	Image memory read4 (RHAR4 select)	
		— —	RHAR4 set (Note 2)	
(2)	0 1 0 1	0 0 0 0 0 DIR	Image memory write	IMW
		—		
		0 0 1 x x DIR	High address set for write (selected register file is DIR +1)	IMWHA
		0 1 0 x x DIR	Write data set (selected register file is DIR +1)	IMWD
		—		
		0 1 1 x x DIR	High address set for read (selected register file is DIR +1)	IMREA
		—		
		1 0 0 Mask DIR	Read/modify/write1	RMW1
		— —		
		1 0 1 x x DIR	Read / modify / write2 (mask selected by CS bits of image memory write data)	RMW2
		—		
(3)	0 1 0 1	1 1 0 x x x x	DMA1 (host — μPD7281)	DMA1
		1 1 1 x x x x	DMA2 (μPD7281 — host)	DMA2
(2)	0 1 1 0	0 0 x x x DIR	Self object load1	SOL1
		—		
		0 1 x x x DIR	Self object load2 (rewrite MN)	SOL2
		—		
		1 x x x x x x	MN set for self object load	SOLMN
(4)	0 1 1 1		μPD7281 module number (when RHASEL=1)	PASS
		1 0 0 0		
		1 0 0 1		
		1 0 1 0		
		1 1 0 0		μPD7281 module numbers
		1 1 0 1		
		1 1 1 0		
(5)	1 1 1 1		Deleted	VANISH

**Notes:**

(1) The following definitions refer to the above table:

MN: Module number

ID: Identifier

MN': MN used for next circulation (MN ≠ 111)

ID': ID used for next circulation

(2) When RHASEL of the mode register is 1, the tokens become image memory read tokens.

## μPD9305 Operation

Table 4 shows how the μPD9305 uses signals  $\overline{CS}$ ,  $\overline{RD}$ ,  $\overline{WR}$ , and  $A_1, A_0$  to read or write to I/O ports.

**Table 4. I/O Ports**

CS	$\overline{RD}$	$\overline{WR}$	A <sub>1</sub>	A <sub>0</sub>	Internal I/O Ports
0	0	1	0	0	Read ImPP input data register (from ImPP)
0	0	1	0	1	Read status register
0	0	1	1	0	Command RESET, data read has no meaning
0	0	1	1	1	Not used
0	1	0	0	0	Write ImPP output data register (to ImPP)
0	1	0	0	1	Write mode register
0	1	0	1	0	Write module number register
0	1	0	1	1	Write refresh timing register

Figure 2 shows the status register format.

**Figure 2. Status Register Format**

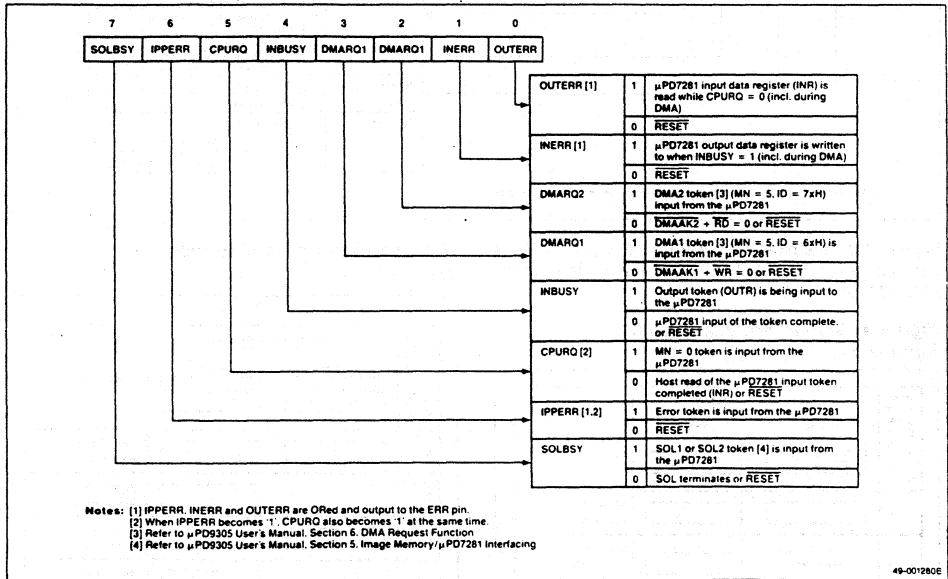
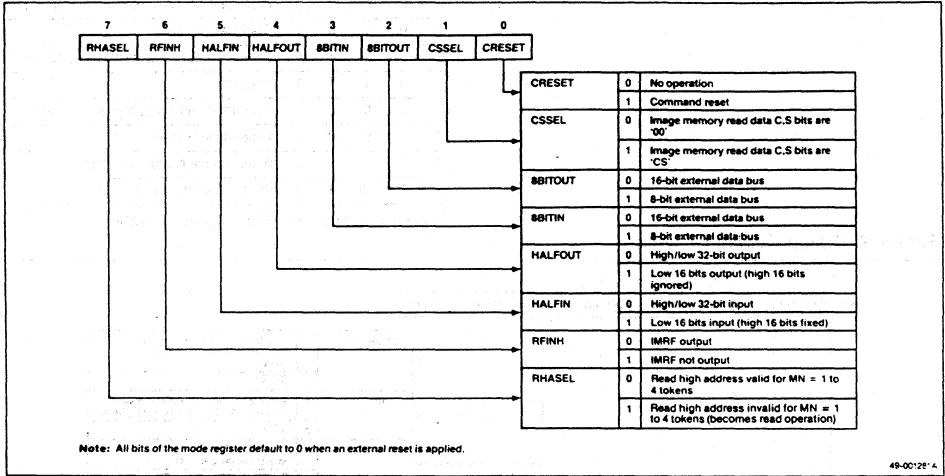




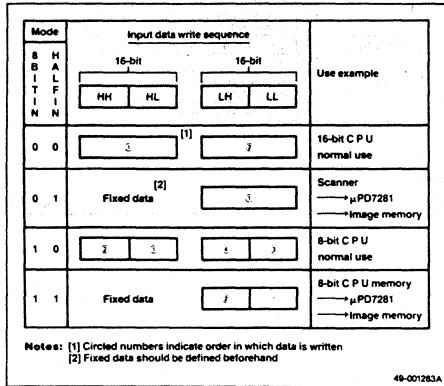
Figure 3 shows the mode register format.

**Figure 3. Mode Register Format**

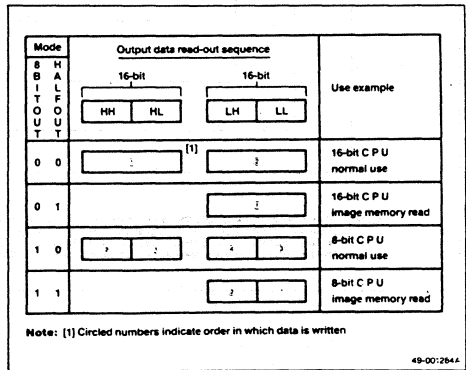


Figures 4-20 graphically show μPD9305 operation. For a detailed description of μPD9305 operation, refer to the μPD9305 User's Manual.

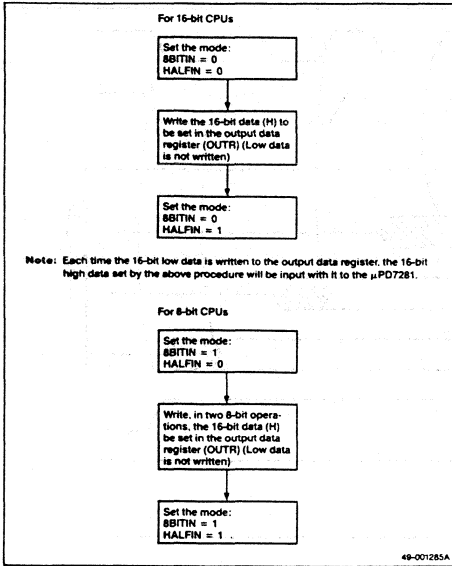
**Figure 4. Setting Write Method for Input Data**



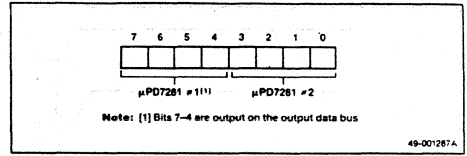
**Figure 5. Setting Read Method for Output Data**



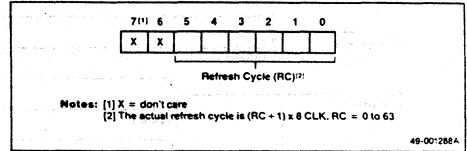
**Figure 6. Setting Fixed (16-Bit) Data**



**Figure 7. MN Register**



**Figure 8. Refresh Timing Register**



**Figure 9. Input Timing (Host to μPD9305 to μPD7281)**

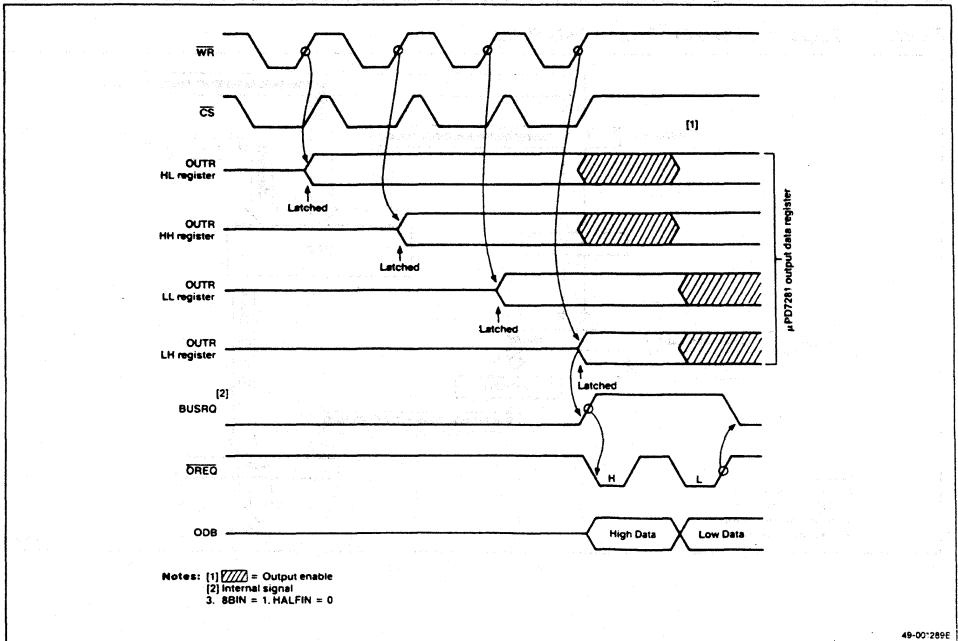


Figure 10. Output Timing (μPD7281 to μPD9305 to Host)

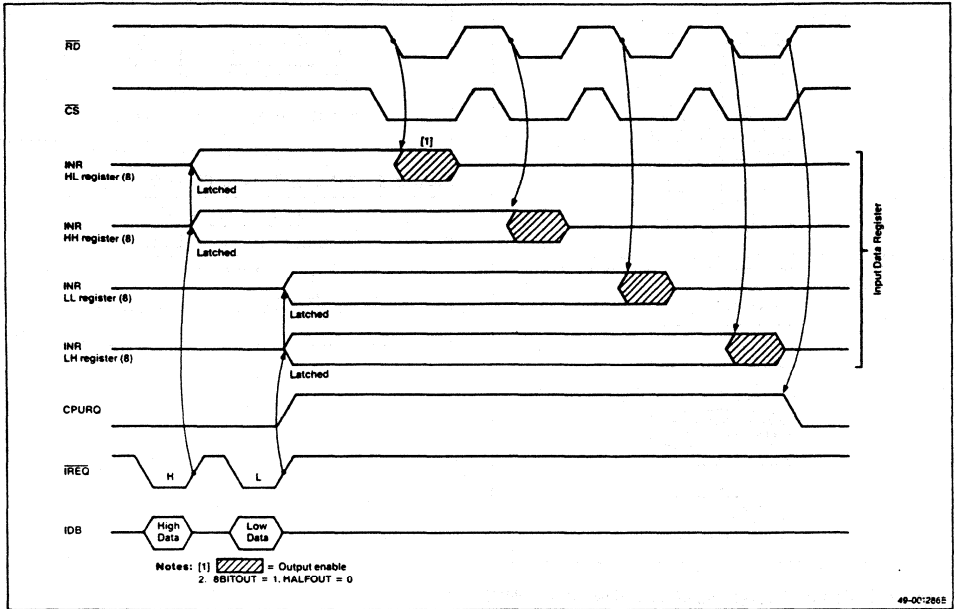


Figure 11. Output to μPD7281, Control Data Paths

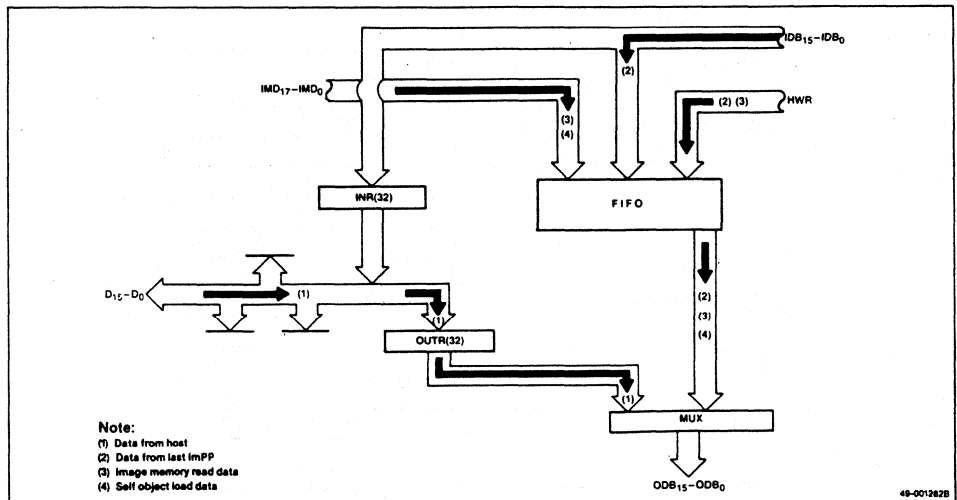


Figure 12. μPD7281, Input Control Data Flow

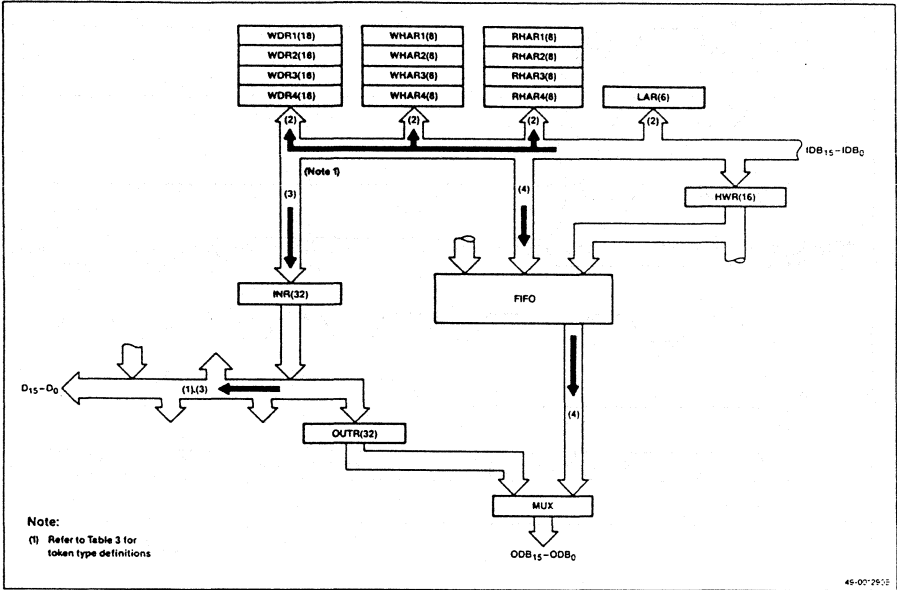


Figure 13. Image Memory Read Timing (Without Refresh Request)

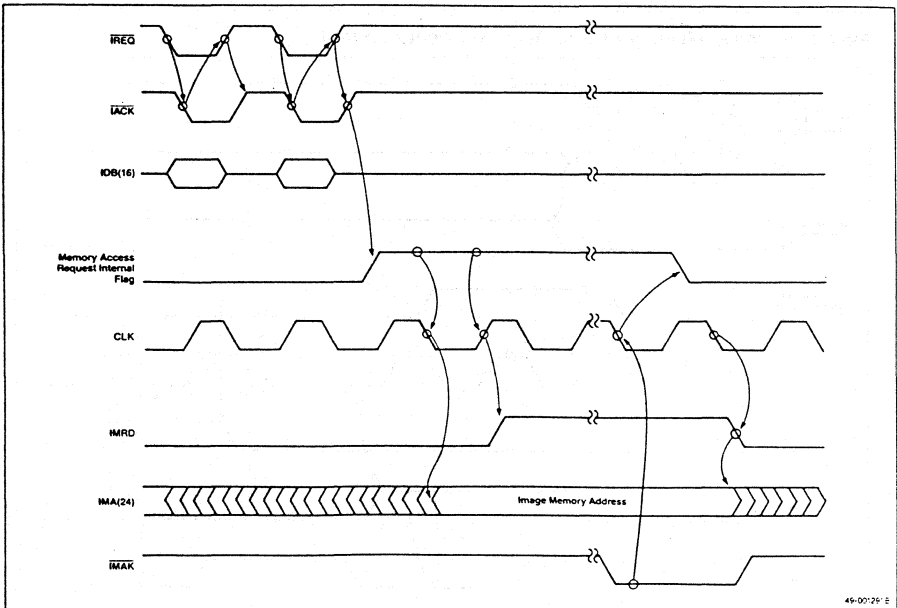


Figure 14. Image Memory Write Timing (Without Refresh Request)

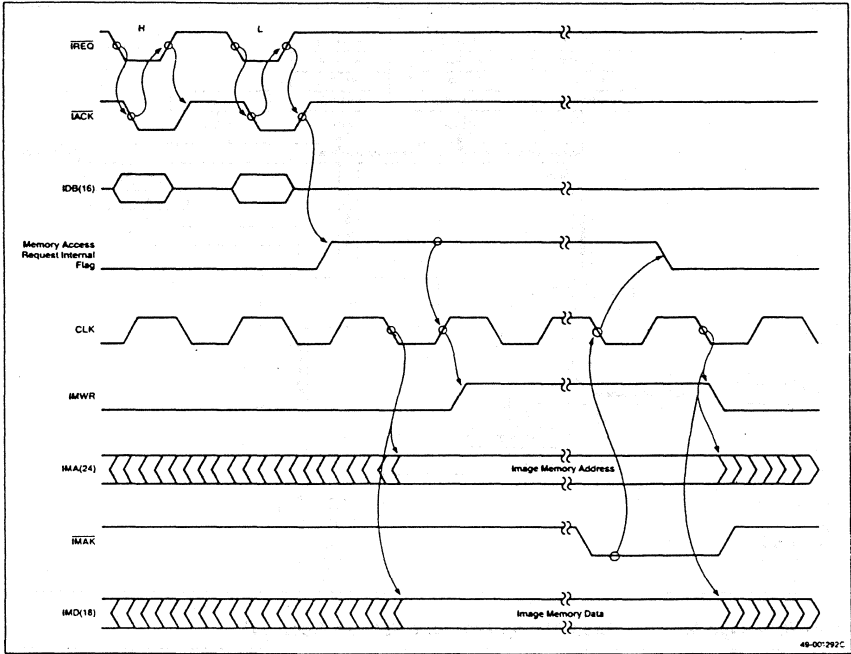


Figure 15. Image Memory Access Request Priority Control

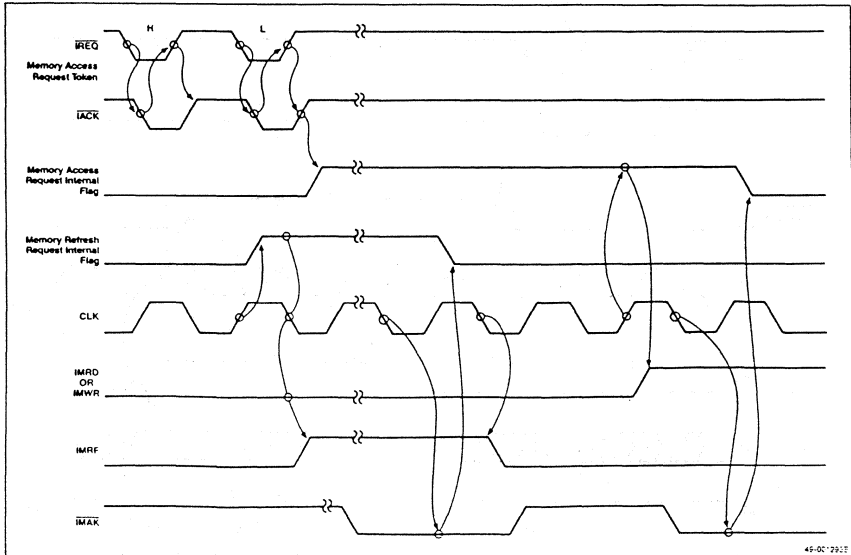


Figure 16. Read Data -- μPD7281 Output Timing (Single Output)

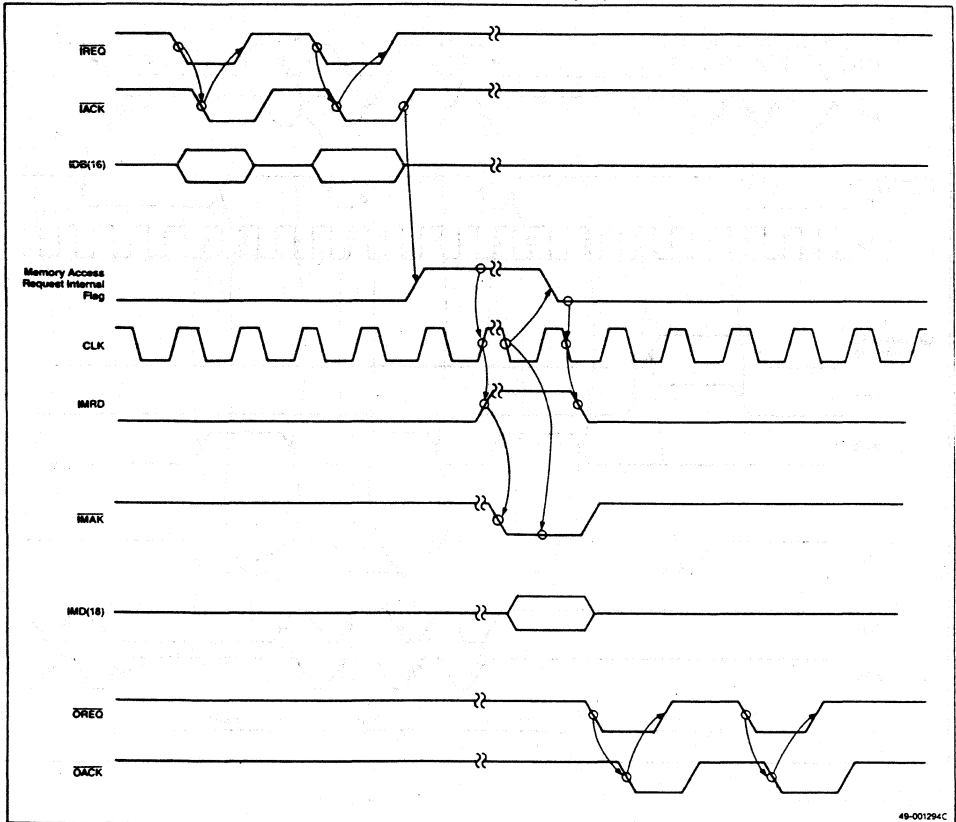
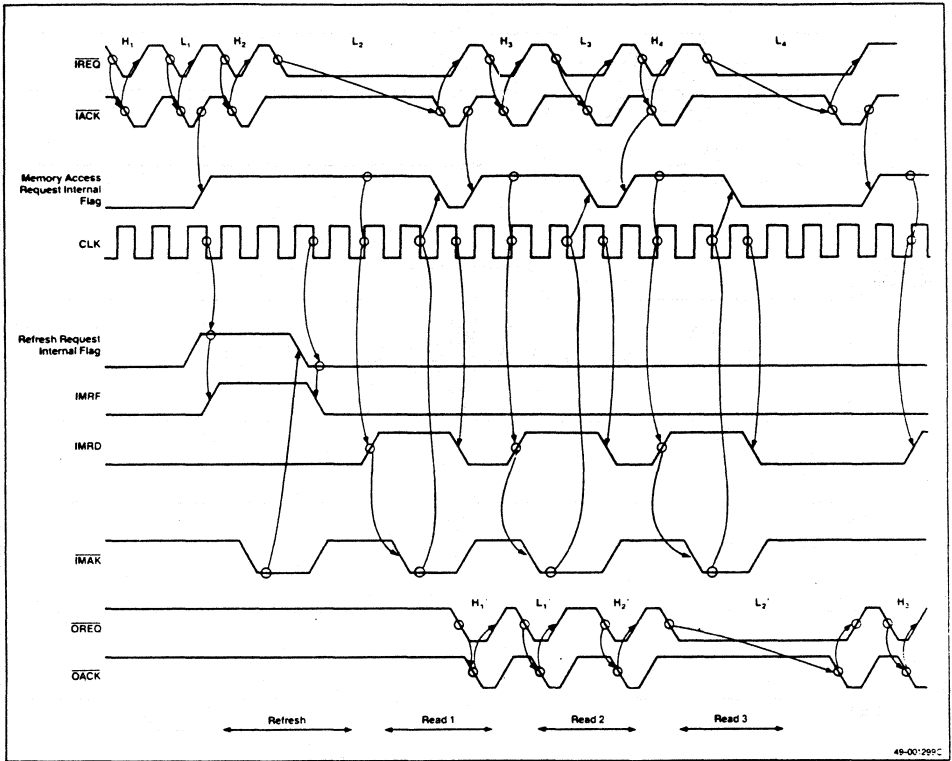
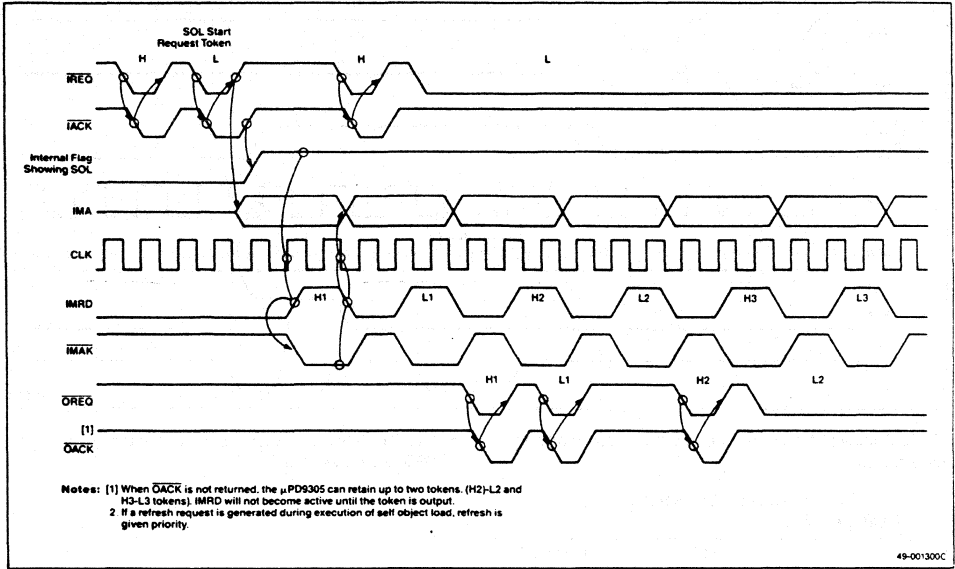


Figure 17. Read Data — μPD7281 Output Timing (Continuous Output)

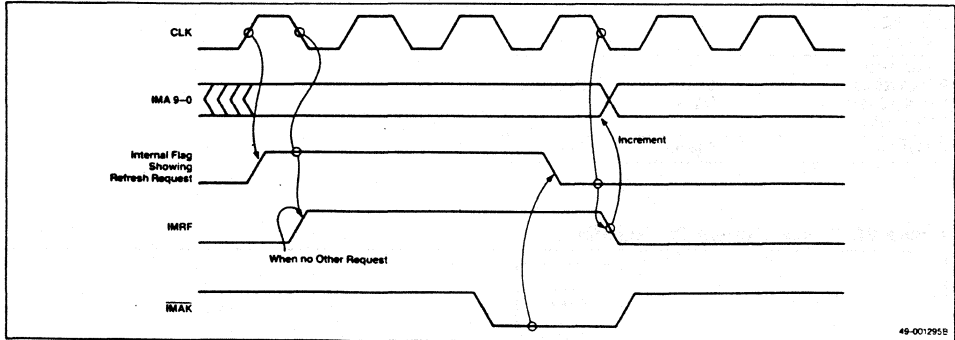


**Figure 18. Self Object Load Timing**



5

**Figure 19. Refresh Timing**





**Figure 20. Read/Modify/Write Timing**

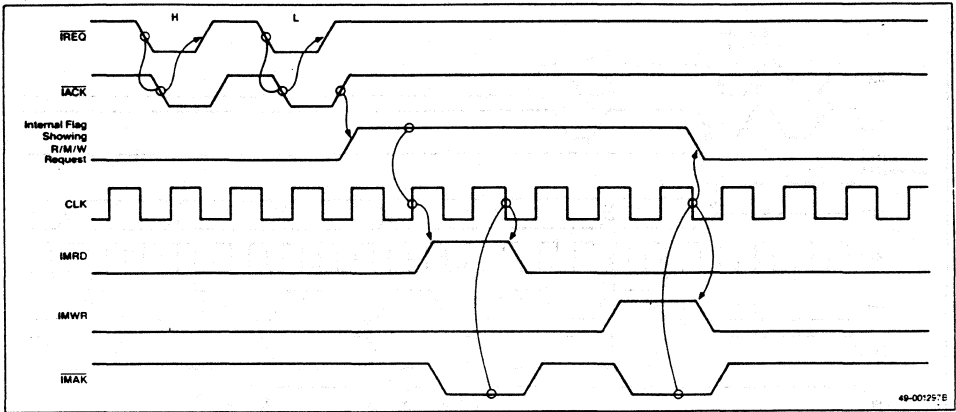


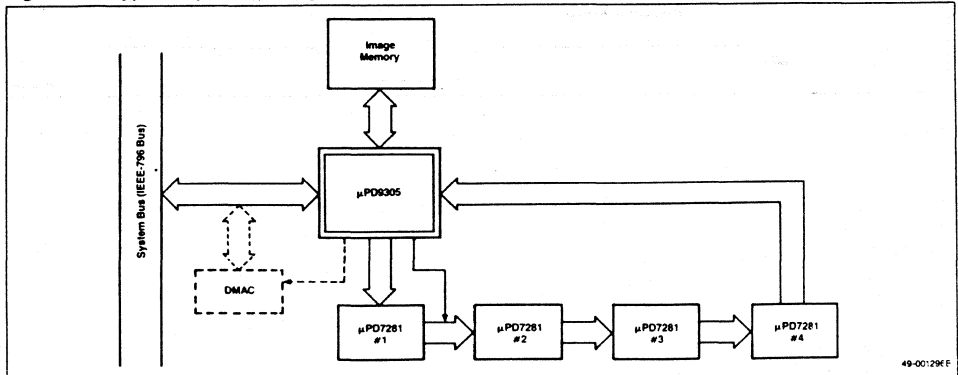
Table 5 shows the differences between command and external resets.

Figure 21 shows a typical system configuration using the μPD9305 with several ImPPs.

**Table 5. Command and External Reset Differences**

Item	RESET	Command Reset
I/O data counter		
Tokens in the μPD9305		
Image memory access requests (except refresh)	Cleared	Cleared
OREQ, IACK		
DMA request		
Refresh timer		
Refresh request	Default values	No change
Refresh address		
Mode register		
IPPRST pin	0 (active)	0 (active)

**Figure 21. Typical System Configuration**



## μPD9305 Preliminary Electrical Specification

### Absolute Maximum Ratings (T<sub>a</sub> = 25°C)

Parameter	Symbol	Test Conditions	Ratings	Units
Supply Voltage	V <sub>DD</sub>		-0.5 to 7.0	V
Input Voltage	V <sub>I</sub>		-0.5 to 7.0	V
Output Current	I <sub>O</sub>		10	mA
Operating Temperature	T <sub>opt</sub>		0 to 70	°C
Storage Temperature	T <sub>stg</sub>		-65 to 150	°C

### DC Characteristics (T<sub>a</sub> = 0 to + 70°C, V<sub>DD</sub> = 5V ± 10%)

Parameter	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Input Low Voltage	V <sub>IL</sub>		-0.5		0.8	V
Input High Voltage	V <sub>IH</sub>		2.0		V <sub>DD</sub> +0.5	V
Output Low Voltage	V <sub>OL</sub>	I <sub>OL</sub> = 2mA			0.4	V
Output High Voltage	V <sub>OH</sub>	I <sub>OL</sub> = -400 μA	V <sub>DD</sub> -0.4			V
Input Leakage Current	I <sub>LI</sub>	0 ≤ V <sub>I</sub> ≤ V <sub>DD</sub>			± 10	μA
Output Leakage Current	I <sub>LO</sub>	0 ≤ V <sub>I</sub> ≤ V <sub>DD</sub>			± 10	μA
Supply Current	I <sub>DD</sub>	10 MHz		10	100	mA

### Capacitance (T<sub>a</sub> = 25°C)

Parameter	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Input Capacitance	C <sub>I</sub>	f <sub>c</sub> = 1 MHz			10	p F
Output Capacitance	C <sub>O</sub>	Unmeasured Pins return to 0V			15	p F
I/O Capacitance	C <sub>IO</sub>				15	p F

### AC Characteristics (T<sub>a</sub> = 0 to + 70°C, V<sub>DD</sub> = 5V ± 10%)

#### Clock Timing

Parameter	Symbol	Test Conditions	Min.	Max.	Units
CLK Cycle Time	t <sub>CYK</sub>		80		ns
Clock Pulse width high	t <sub>WKH</sub>		30		ns
Clock Pulse width low	t <sub>WKL</sub>		30		ns
Clock Rise Time	t <sub>KR</sub>			10	ns
Clock Fall Time	t <sub>KF</sub>			10	ns

**Input Timing**

Parameter	Symbol	Test Conditions	Min.	Max.	Units
Input Rise Time	t <sub>IR</sub>		0	10	μs
Input Fall Time	t <sub>IF</sub>		0	10	μs

**RESET Timing**

Parameter	Symbol	Test Conditions	Min.	Max.	Units
RESET/Pulse Width	t <sub>RST</sub>	μPD9305 (only)	t <sub>CYK</sub>		ns
RESET/Setup time to IPPRST/	t <sub>DRSPRL</sub>			40	ns
IPPRST/Hold Time after RESET/up	t <sub>DRSPRH</sub>			50	ns
IPPRST/Setup to MN0-3	t <sub>DMN</sub>			60	ns
MN0-3 Float time after IPPRST/up	t <sub>FMN</sub>			50	ns
IPPRST/Low Till ODB15-12 Active	t <sub>PROD</sub>			60	ns
ODB15-12 Float time after IPPRST/up	t <sub>FPROD</sub>			50	ns

**Host «----» μPD9305 Read/Write (Timing)**

Parameter	Symbol	Test Conditions	Min.	Max.	Units
Address Setup to WR/down, RD/down	t <sub>SARW</sub>		20		ns
Address hold time after WR/up, RD/up	t <sub>HRWA</sub>		20		ns
CS/setup to WR/down, RD/down	t <sub>SCRW</sub>		0		ns
CS/Hold time after WR/up, RD/up	t <sub>HRWC</sub>		0		ns
WR/, RD/Pulse Width	t <sub>WRWL</sub>		100		ns
RD/Setup to Data	t <sub>DRD</sub>			80	ns
Data Float time after RD/up	t <sub>FRD</sub>			30	ns
Data setup to WR/	t <sub>SDW</sub>		20		ns
Data Hold after WR/up	t <sub>HWD</sub>		20		ns

### I/O Request/Acknowledge Timing

Parameter	Symbol	Test Conditions	Min.	Max.	Units
IREQ/setup time to IACK/	t <sub>DIQIAL</sub>		15	60	ns
1st IACK/setup time to IREQ/up	t <sub>DIAIQH</sub>		10		ns
1st IREQ/up setup time to IACK/up	t <sub>DIQIAH</sub>		20	70	ns
1st IACK/up setup to IREQ/low	t <sub>DIAIQL</sub>		10		ns
ID Bus setup time to IREQ/up	t <sub>SIDIQ</sub>		20		ns
ID Bus Hold time from IREQ/up	t <sub>HIQID</sub>		10		ns
OREQ/setup time to OACK/	t <sub>DOQOAL</sub>		10		ns
OACK/setup time to OREQ/up	t <sub>DOAOQH</sub>		20	70	ns
OREQ/up setup time to OACK/up	t <sub>DOQOAH</sub>		10		ns
OACK/up setup time to OREQ/low	t <sub>DOAOQL</sub>		15	60	ns
OREQ/setup time to ODB Valid	t <sub>DOQOD</sub>			10	ns
ODB Float time after OREQ/up	t <sub>FOQOD</sub>		10		ns

Pull-up resistors required on μPD9305 IDB<sub>15</sub>-IDB<sub>0</sub> to meet t<sub>HIQID</sub> timing.

### DMA Transfer Timing

Parameter	Symbol	Test Conditions	Min.	Max.	Units
DMARQ/low setup time to DMAAK/low	t <sub>DDQDA</sub>		20		ns
DMARQ/Hold time from DMAAK/low	t <sub>DDADQ</sub>			50	ns
DMARQ/Recovery Time DMAAK/high	t <sub>RVDQ</sub>		50		ns
DMAAEN up setup time to (RD/, WR/) low	t <sub>SDERW</sub>		30		ns
DMAAEN low Hold time after (RD/, WR/) high	t <sub>HRWDE</sub>		30		ns
DMAAK/low setup time to (RD/, WR/) low	t <sub>S DARW</sub>		0		ns
DMAAK hold time after (RD/, WR/) high	t <sub>HRWDA</sub>		0		ns
DMAAK/pulse width (low to high)	t <sub>WDAL</sub>		t <sub>CYK</sub>		ns

DMARQ/ = DMARQ1/, DMARQ2/  
DMAAK/ = DMAAK1/, DMAAK2/

**Image Memory Read, Write, Refresh Timing**

Parameter	Symbol	Test Conditions	Min.	Max.	Units
IMA Active time from (1) CLK ↓	t <sub>DKMARF</sub>	IM Refresh		100	ns
IMA Active time from CLK ↓	t <sub>DKMAMC</sub>	IM Read/IM Write		60	ns
IMA Float time from IMWR/IMRD low	t <sub>FMOMA</sub>		10		ns
IMWR/IMRD Recovery time	t <sub>RVMC</sub>		1.5 t <sub>CYK</sub>		ns
IMWR/IMRD/IMWR high delay time from CLK high	t <sub>DKMCH</sub>			35	ns
IMWR/IMRD/IMWR low delay time from CLK low	t <sub>DKMCL</sub>			40	ns
IMAK/setup time to CLK down	t <sub>SMKK</sub>		10		ns
IMD setup time to CLK high	t <sub>SMDK</sub>	Image Memory Read Timing	20		ns
IMD hold time from IMRD low	t <sub>HMRMD</sub>	Image Memory Read Timing	0		ns
IMD delay time from CLK low	t <sub>DKMD</sub>	Image Memory Write Timing		30	ns
IMD Float time from IMWR low	t <sub>FMWMD</sub>	Image Memory Write Timing	20		ns
IMAK/Recovery time	t <sub>RVMK</sub>		1.5 t <sub>CYK</sub>		ns
IMAK/hold time from IMRD/IMWR low	t <sub>HCMCK</sub>		0		ns

**Note:**

(1) IMA = IMA<sub>23</sub>-IMA<sub>0</sub>

(2) IMC + IMRD, IMWR or IMRF

(3) To maximize IM access time use  $\overline{IMAK} = \overline{IMC}$ . Then IM cycle time will be 3.t<sub>CYK</sub>

### SOLBSY Timing

Parameter	Symbol	Test Conditions	Min.	Max.	Units
SOLBSY Delay time from IACK/up	tDIASB			30	ns
SOLBSY delay time from CLK low	tDKSB			60	ns

### CPURQ Timing

Parameter	Symbol	Test Conditions	Min.	Max.	Units
CPURQ delay time from IACK/high	tDIAPQ			30	ns
CPURQ delay time from RD/high	tDPRQ			60	ns

### INBUSY Timing

Parameter	Symbol	Test Conditions	Min.	Max.	Units
INBUSY delay time from WR/high	tDWIB			70	ns
INBUSY delay time from OREQ/high	tDOQIB			40	ns

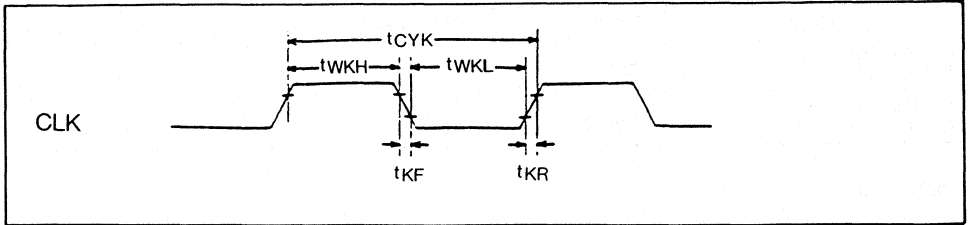
### ERR Timing

Parameter	Symbol	Test Conditions	Min.	Max.	Units
ERR delay time from IACK/high	tDIAE			30	ns
ERR delay time from WR/low	tDWE	INBUSY = 1		60	ns
ERR delay time RD/low	tDRE	CPURQ = 0		60	ns
INBUSY hold time from WR/low	tHWIB			10	ns
CPURQ setup time to RD/low	tSPQR			10	ns

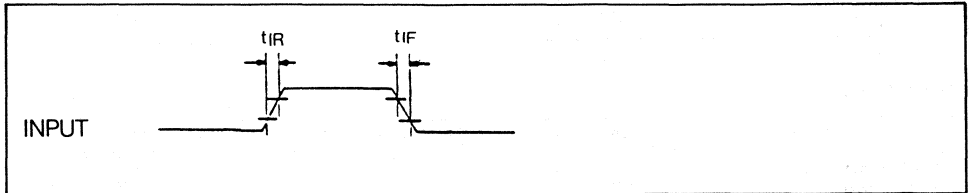
**Note:**

All unused input or output pins should be pulled to V<sub>DD</sub> or down to GND through a 2K-3K ohm resistor.

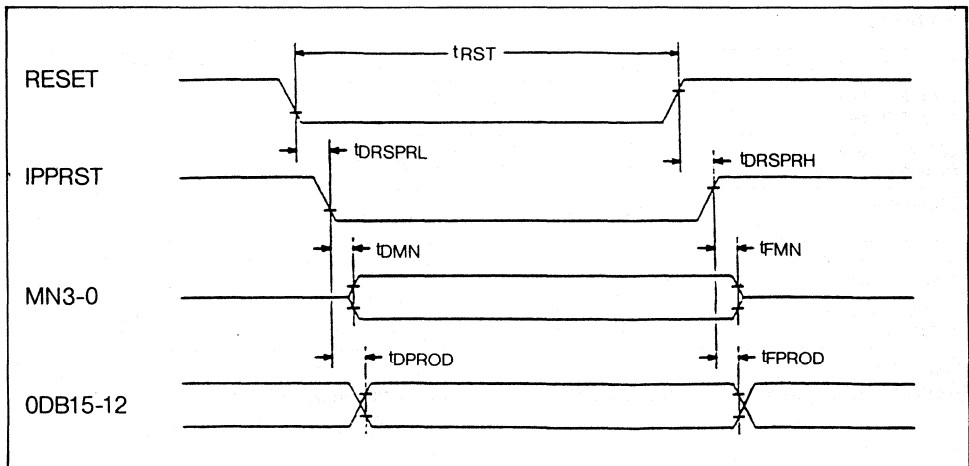
**Clock timing**



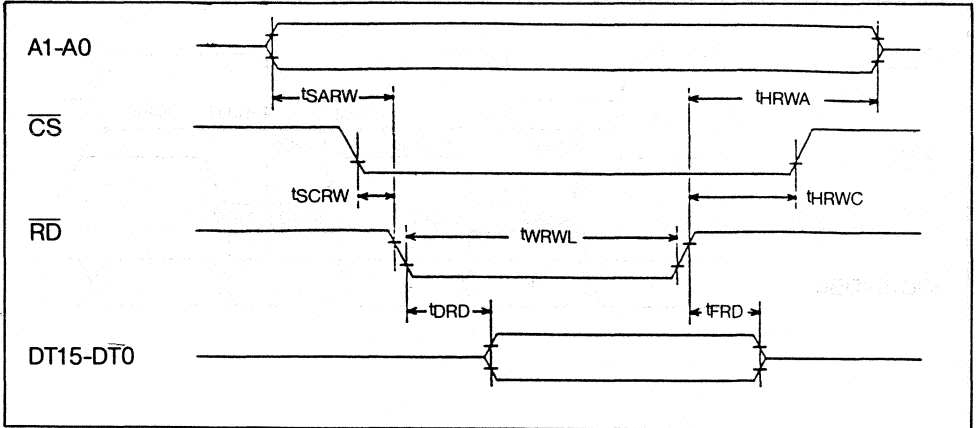
**Input timing**



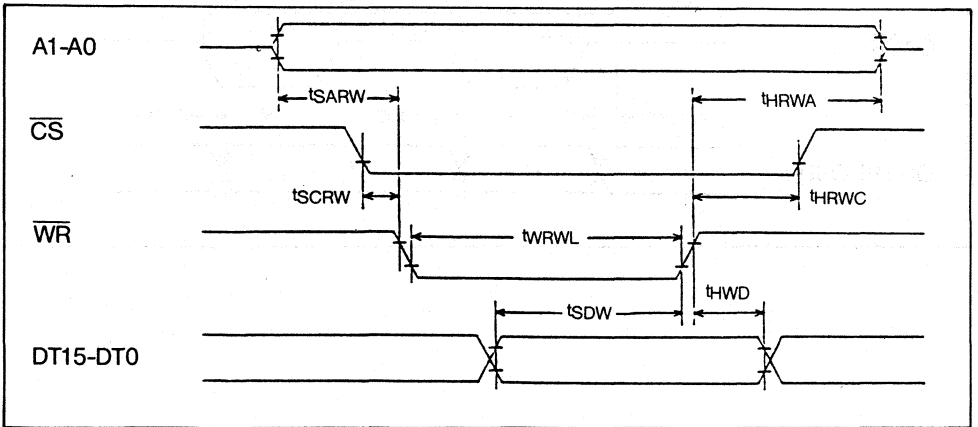
**Reset timing**



### Host CPU Read timing



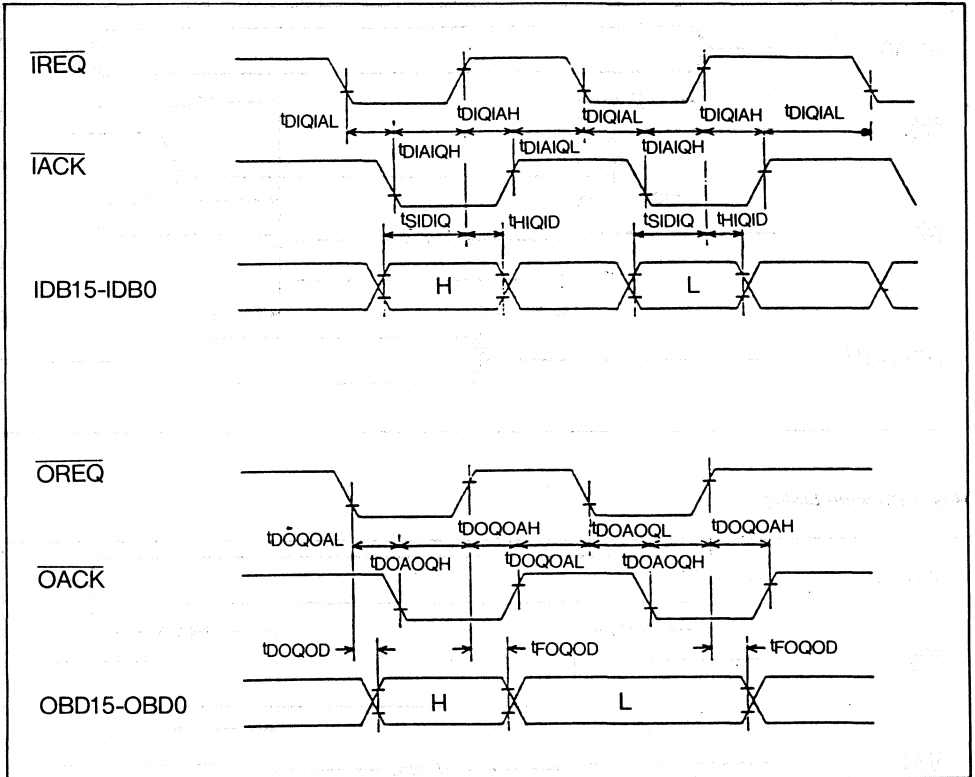
### Host CPU write timing



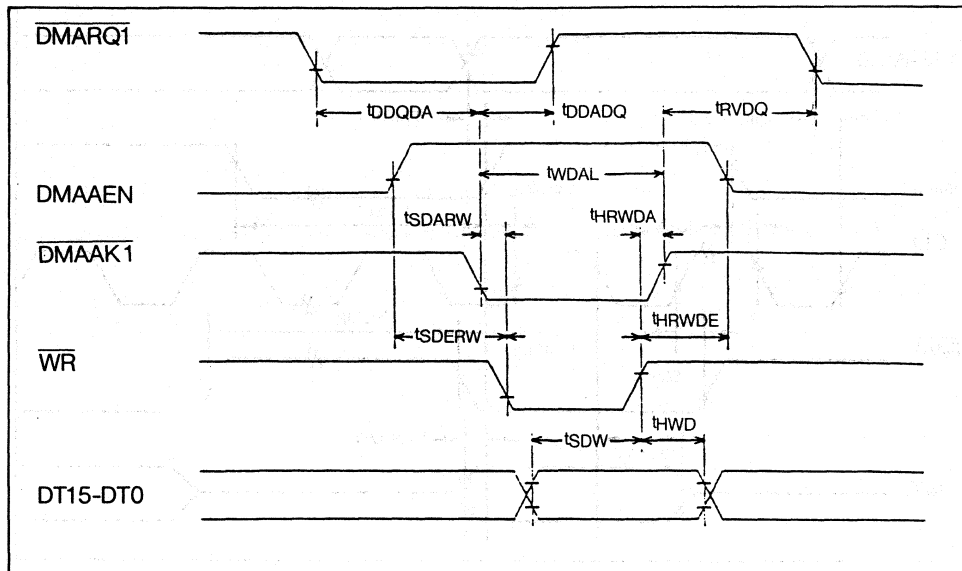


I/O Data Bus Handshake timing

Control Signals Timing Diagram



**DMA1 timing**



5

**DMA2 timing**

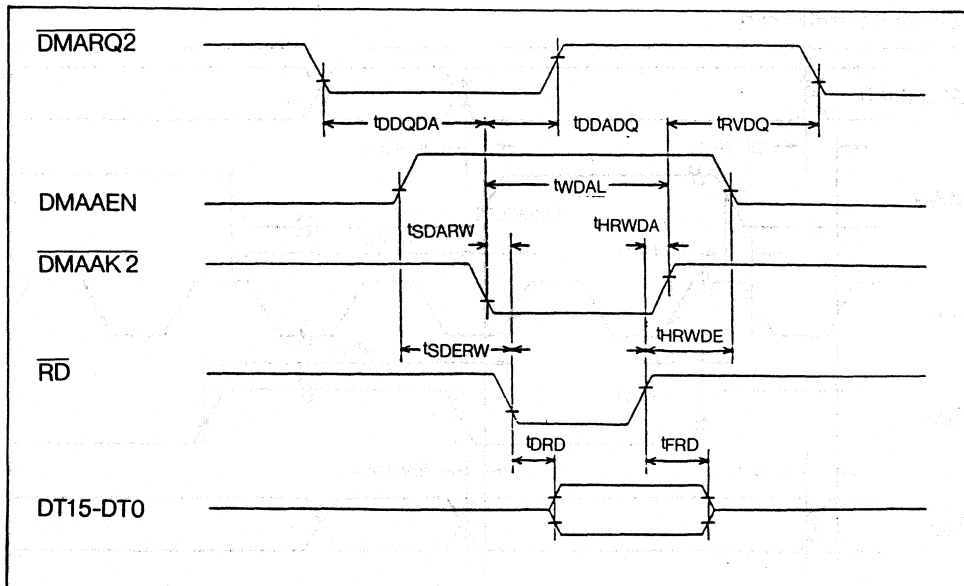


Image Memory Read timing

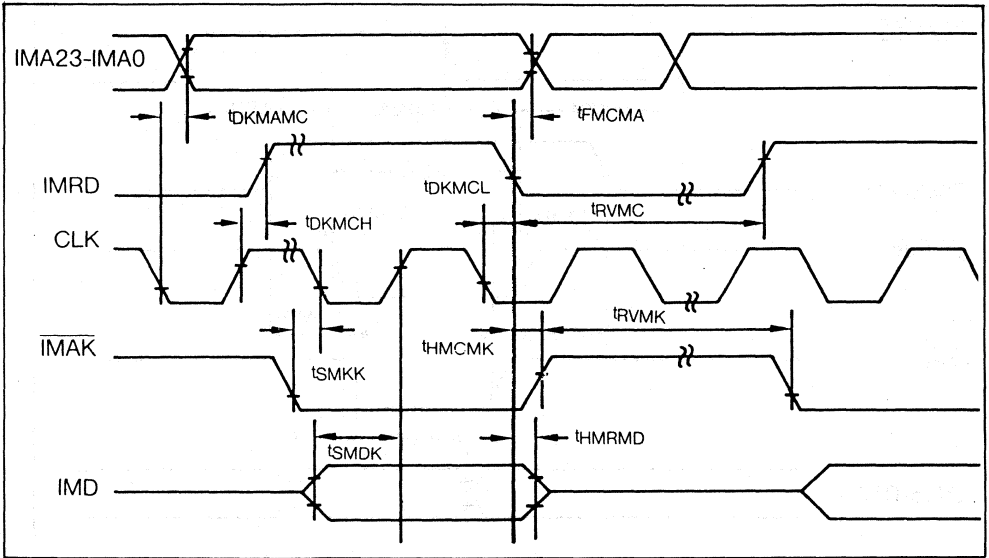
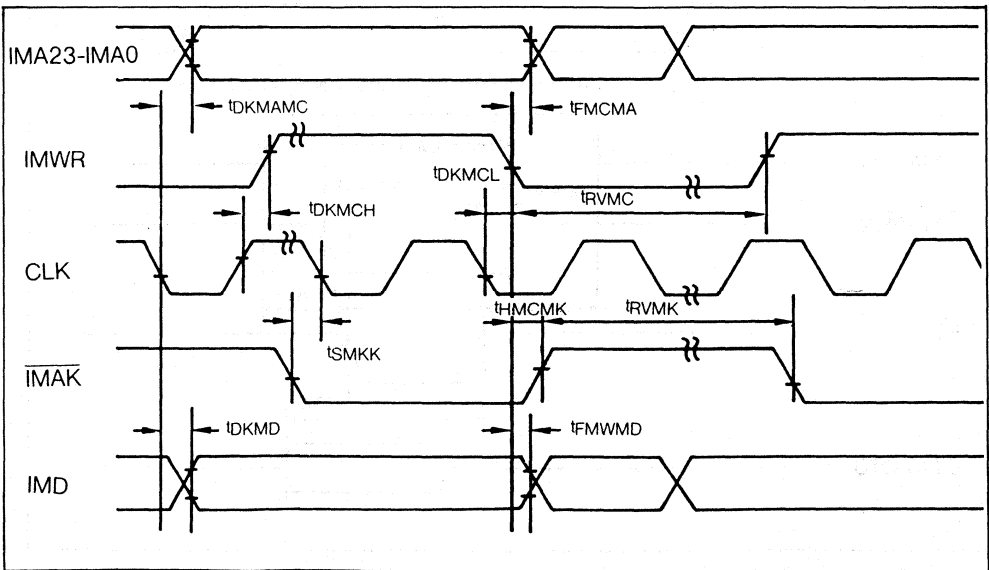
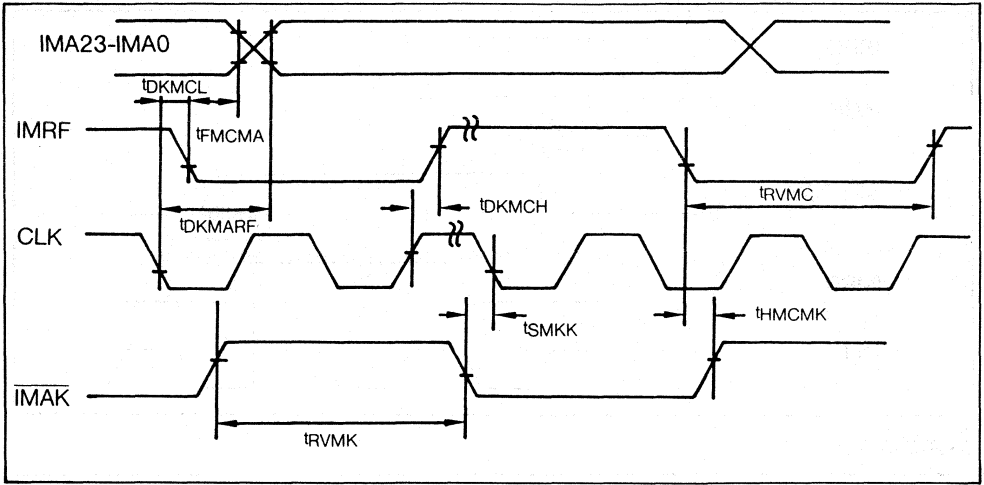


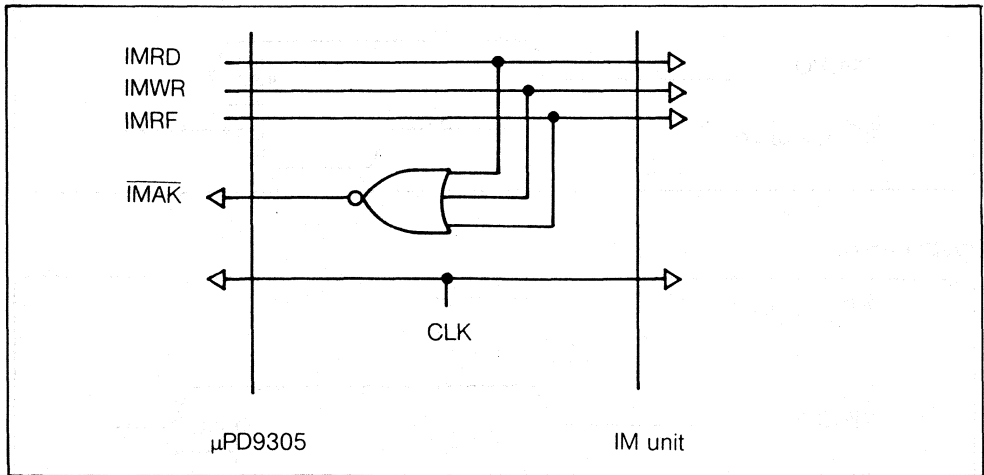
Image Memory Write timing



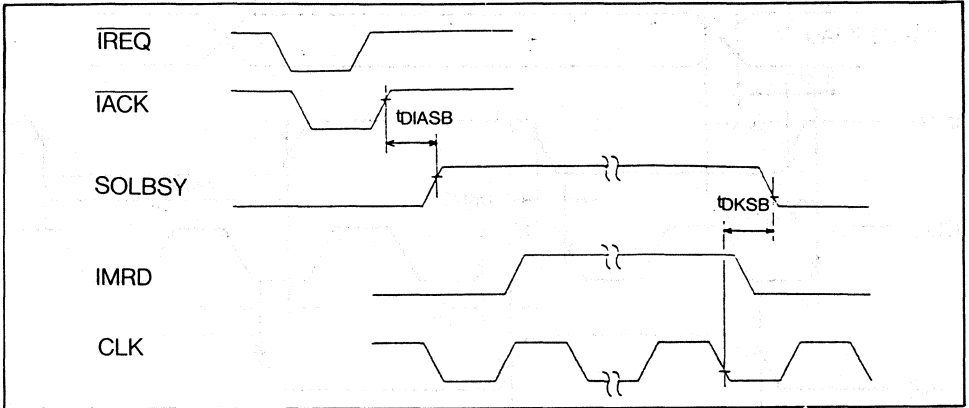
**Image Memory Refresh timing**



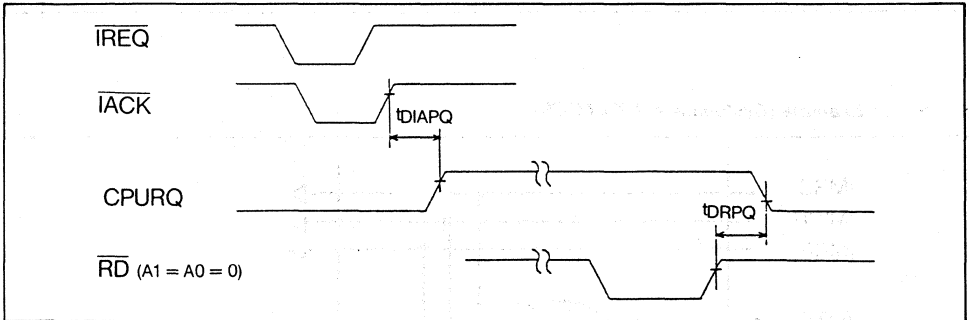
**Generation Example (Cycletime = 3 X CLOCK)**



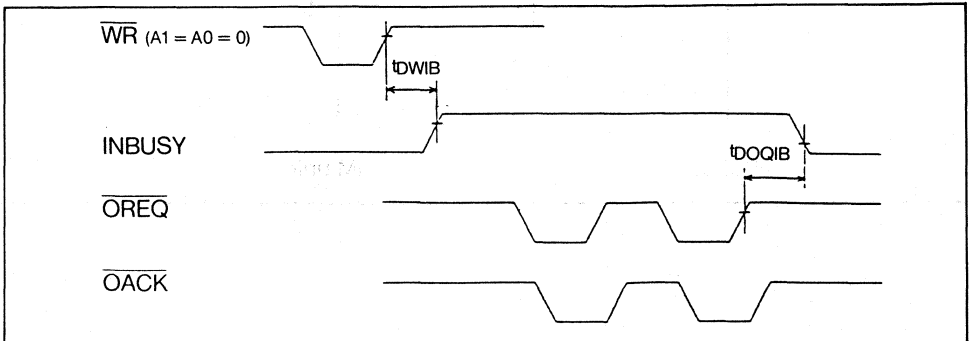
### SOLBSY timing



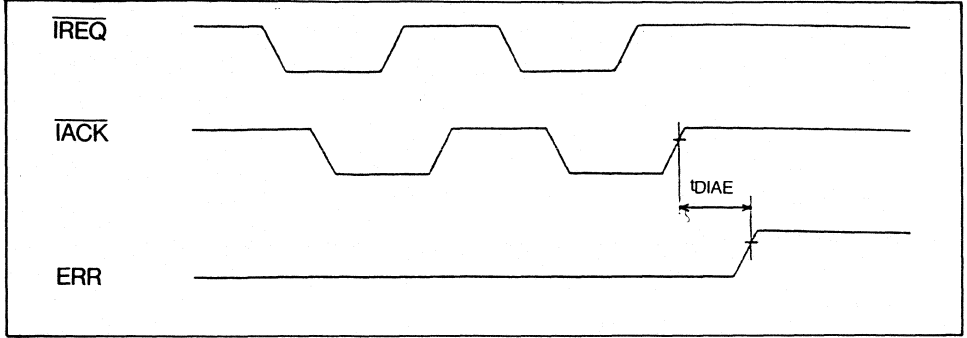
### CPURQ timing



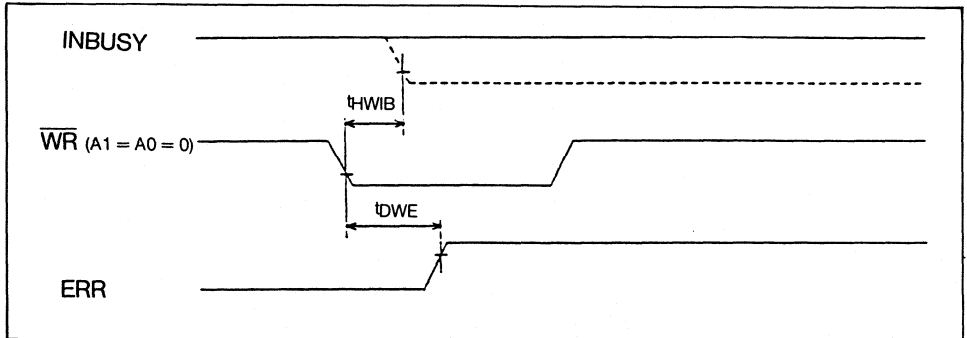
### INBUSY timing



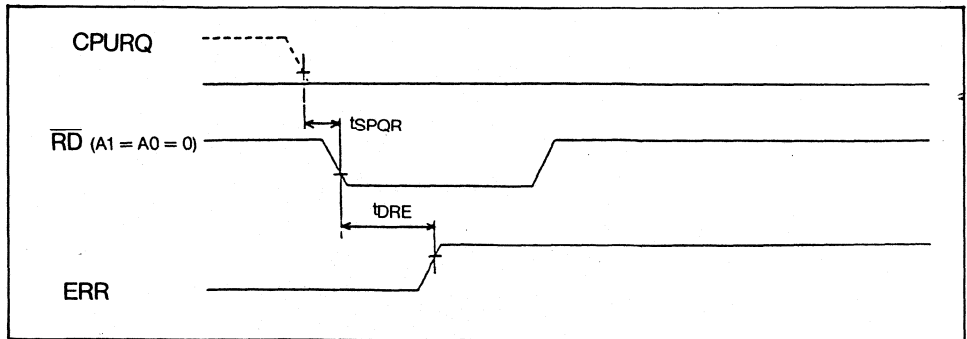
### ERR timing (1)



### ERR timing (2)



### ERR timing (3)









## Section 6 – CMOS System Support Products

μPD71011	Clock pulse Generator/Driver .....	6.3
μPD71051	Serial Control Unit .....	6.13
μPD71054	Programmable Timer/Counter .....	6.33
μPD71055	Parallel Interface Unit .....	6.49
μPD71059	Interrupt Control Unit .....	6.71
μPD71071	DMA Control Unit .....	6.97
μPD71082/83	8-Bit Latches .....	6.135
μPD71084	Clock Pulse Generator/Driver .....	6.139
μPD71086/87	8-Bit Bus Buffer/Drivers .....	6.147
μPD71088	System Bus Controller .....	6.151
μPD71611	Clock Pulse Generator/Driver for V60 .....	6.159
μPD71613	System Bus Controller for V60 .....	6.169
μPD82C43	CMOS Input/Output Expander for μPD8048/C48 Family .....	6.177

## Description

The μPD71011 is a clock pulse generator/driver for the V20™/V30™ microprocessors and their peripherals using NEC's high-speed CMOS technology.

## Features

- CMOS technology
- Clock pulse generator/driver for μPD70108/70116 or other CMOS or NMOS CPUs and their peripherals
- 50% duty cycle
- Frequency source can be crystal or external clock input
- Reset signal with Schmitt-trigger circuit for CPU or peripherals
- Bus ready signal with two-bus system synchronization
- Clock synchronization with other μPD71011s
- Single +5-volt ±10% power supply
- Industrial temperature range: -40 to +85°C

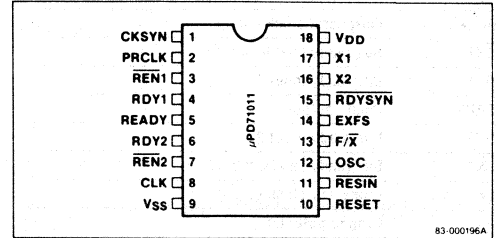
## Ordering Information

Part Number	Package	Max Clock Frequency
μPD71011C	18-pin plastic DIP	8 MHz
C-10	18-pin plastic DIP	10 MHz
G	20-pin plastic SO	8 MHz

V20 and V30 are trademarks of NEC Corporation.

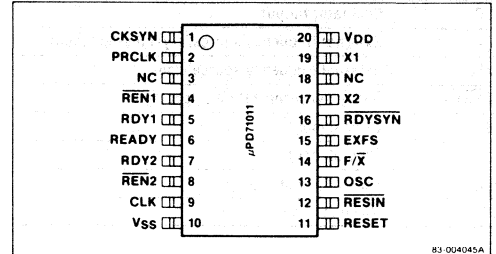
## Pin Configurations

### 18-Pin Plastic DIP



83-000196A

### 20-Pin Plastic SO (Small Outline)



83-000452A

**Pin Identification**

Symbol	Function
CKSYN	Clock synchronization input
PRCLK	Peripheral clock output
REN1	Bus ready enable input 1
RDY1	Bus ready input 1
READY	Ready output
RDY2	Bus ready input 2
REN2	Bus ready enable input 2
CLK	Processor clock output
V <sub>SS</sub>	Ground potential
RESET	Reset output
RESIN	Reset input
OSC	Oscillator output
F $\bar{X}$	External frequency source/crystal select input
EXFS	External frequency source input
RDYSYN	Ready synchronization select input
X2	Crystal input
X1	Crystal input
V <sub>DD</sub>	+5-volt power supply
NC	No connection

**Pin Functions****X1, X2 [Crystal]**

When F $\bar{X}$  is low, a crystal connected to X1 and X2 will be the frequency source for a CPU and its peripherals. The crystal frequency should be two times the frequency of CLK.

**EXFS [External Frequency Source]**

EXFS input is the external frequency input in the external TTL-frequency source mode (F $\bar{X}$  high). A square TTL-level clock signal two times the frequency of CLK's output should be used for the source.

**F $\bar{X}$  [Frequency/Crystal Select]**

F $\bar{X}$  input selects whether an external TTL-type input or an external crystal input is the frequency source of the CLK output. When F $\bar{X}$  is low, CLK is generated from the crystal connected to X1 and X2. When F $\bar{X}$  is high, CLK is generated from an external TTL-level frequency input on the EXFS pin. At the same time, the internal oscillator circuit will go into stop mode and the OSC output will be high.

**CLK [Processor Clock]**

The CLK output supplies the CPU and its local bus peripherals' clocks. CLK is a 50-percent duty cycle clock of one-half the frequency of the external frequency source. The CLK output is +0.4 V higher than the other outputs.

**PRCLK [Peripheral Clock]**

The PRCLK output supplies a 50-percent duty cycle clock at one-half the frequency of CLK to drive peripheral devices.

**OSC [Oscillator]**

OSC outputs a signal at the same frequency as the crystal input. When EXFS is selected, the OSC output is powered down, and its output will be a high.

**CKSYN [Clock Synchronization]**

CKSYN synchronizes one μPD71011 to other μPD71011s. A high level at CKSYN resets the internal counter, and a low level enables it to count.

**RESIN [Reset]**

This Schmitt-trigger input generates the RESET output. It is used as a power-on reset.

**RESET [Reset]**

This output is a reset signal for the CPU. Reset timing is provided by the RESIN input to a Schmitt-trigger input gate and a flip-flop which will synchronize the reset timing to the falling edge of CLK. Power-on reset can be provided by a simple RC circuit on the RESIN input.

**RDY1, RDY2 [Bus Ready]**

A peripheral device sends RDY1 or RDY2 to signal that the data on the system bus has been received or is ready to be sent. REN1 and REN2 enable the RDY1 or RDY2 signals.

**REN1, REN2 [Bus Ready Enable]**

REN1 and REN2 qualify their respective RDY inputs.

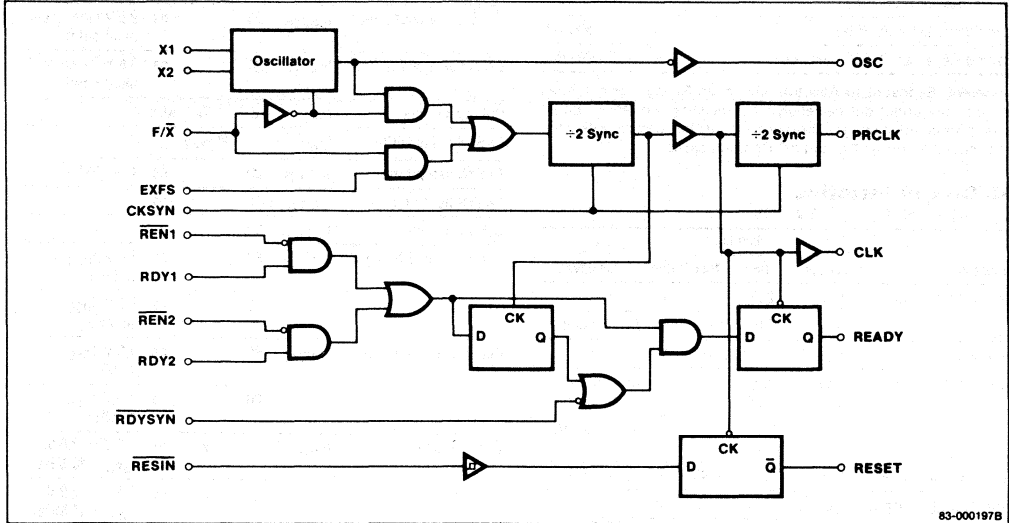
**RDYSYN [Ready Synchronization Select]**

RDYSYN selects the mode of READY signal synchronization. A low-level signal makes the synchronization a two-step process. This is used when RDY1 and RDY2 inputs are not synchronized to CLK. A high-level signal makes synchronization a one-step process. This is used when RDY1 and RDY2 are synchronized to CLK. See Block Diagram.

## READY [Ready]

The READY signal to the processor is synchronized by the RDY inputs to the processor CLK. READY is cleared after the RDY signal goes low and the guaranteed hold time of the processor has been met.

## Block Diagram



83-000197B

## Crystal

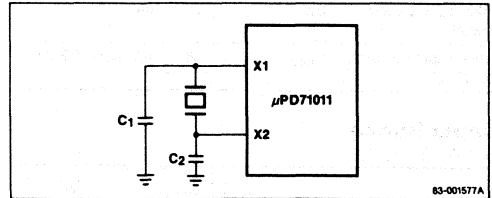
The oscillator circuit of the μPD71011 works with a parallel-resonant, fundamental mode, "AT cut" crystal connected to pins X1 and X2.

Figure 1 shows the recommended circuit configuration. Capacitors C1 and C2 are required for frequency stability. The values of C1 and C2 (C1 = C2) can be calculated from the load capacitance (C<sub>L</sub>) specified by the crystal manufacturer.

$$C_L = \frac{C_1 \times C_2}{C_1 + C_2} + C_s$$

Where C<sub>s</sub> is any stray capacitance in parallel with the crystal, such as the μPD71011 input capacitance C<sub>i</sub>.

Figure 1. Crystal Configuration Circuit



83-001577A

**Absolute Maximum Ratings**

T<sub>A</sub> = 25°C; V<sub>SS</sub> = 0 V

Power supply voltage, V <sub>DD</sub>	- 0.5 to + 7.0 V
Input voltage, V <sub>I</sub>	- 1.0 V to V <sub>DD</sub> + 1.0 V
Output voltage, V <sub>O</sub>	- 0.5 V to V <sub>DD</sub> + 0.5 V
Operating temperature, T <sub>OPT</sub>	- 40 to +85°C
Storage temperature, T <sub>STG</sub>	- 65 to +150°C
Power dissipation, P <sub>D</sub> (DIP)	500 mW
Power dissipation, P <sub>D</sub> (SO package)	200 mW

**Comment:** Exposure to Absolute Maximum Ratings for extended periods may affect device reliability; exceeding the ratings could cause permanent damage. The device should be operated within the limits specified under DC and AC Characteristics.

**DC Characteristics**

T<sub>A</sub> = -40 to +85°C; V<sub>DD</sub> = 5 V ±10%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input voltage, high	V <sub>IH</sub>	2.2		V	
			2.6	V	RESIN input
Input voltage, low	V <sub>IL</sub>		0.8	V	
Output voltage, high	V <sub>OH</sub>	V <sub>DD</sub>		V	CLK output, I <sub>OH</sub> = -4 mA
		- 0.4		V	I <sub>OH</sub> = -4 mA
Output voltage, low	V <sub>OL</sub>		0.45	V	I <sub>OL</sub> = 4 mA
Input leakage current	I <sub>IN</sub>	-1.0	1.0	μA	
		-400	1.0	μA	RDYSYN input
RESIN hysteresis	V <sub>H</sub>	0.25		V	
Power supply current (static)	I <sub>DD</sub>		200	μA	
Power supply current (dynamic)	I <sub>DDdyn</sub>		30	mA	f <sub>in</sub> = 20 MHz

**Capacitance**

T<sub>A</sub> = 25°C; V<sub>DD</sub> = +5 V

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input capacitance	C <sub>I</sub>		12	pF	f <sub>C</sub> = 1 MHz

**AC Characteristics, μPD71011**

(1) f<sub>OSC</sub> = 10 MHz; T<sub>A</sub> = -40 to +85°C; V<sub>DD</sub> = +5 V ±10%

(2) f<sub>OSC</sub> = 16 MHz; T<sub>A</sub> = -10 to +70°C; V<sub>DD</sub> = +5 V ±5%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Clock Timing</b>					
EXFS input cycle time	t <sub>CYFS</sub>	50		ns	
EXFS pulse width, high	t <sub>PWFSH</sub>	20		ns	2.2 V measurement point
EXFS pulse width, low	t <sub>PWFSL</sub>	20		ns	0.8 V measurement point
OSC cycle time	t <sub>OSC</sub>	8	16	MHz	
CKSYN pulse width	t <sub>PWCT</sub>		2t <sub>CYFS</sub>	ns	
CKSYN setup time	t <sub>HFSTCK</sub>	20		ns	from EXFS
CKSYN hold time	t <sub>CTFS</sub>	20		ns	
CLK cycle time	t <sub>CYCK</sub>		125	ns	
CLK pulse width, high	t <sub>PWCKH</sub>	80		ns	3.0 V, f <sub>OSC</sub> = 10 MHz
			50	ns	3.0 V, f <sub>OSC</sub> = 16 MHz
CLK pulse width, low	t <sub>PWCKL</sub>	90		ns	1.5 V, f <sub>OSC</sub> = 10 MHz
			60	ns	1.5 V, f <sub>OSC</sub> = 16 MHz
CLK rise time	t <sub>LHCK</sub>	10		ns	1.5 V → 3.0 V, f <sub>OSC</sub> = 10 MHz
			8	ns	1.5 V → 3.0 V, f <sub>OSC</sub> = 16 MHz
CLK fall time	t <sub>HLCK</sub>	10		ns	3.0 V → 1.5 V, f <sub>OSC</sub> = 10 MHz
			7	ns	3.0 V → 1.5 V, f <sub>OSC</sub> = 16 MHz
CLK delay from OSC	t <sub>DCK</sub>	2	30	ns	CLK↑
			-6	28	ns
PRCLK cycle time	t <sub>CYPRK</sub>	250		ns	
PRCLK pulse width, high	t <sub>PWCKH</sub>	t <sub>CYCK</sub> - 20		ns	
PRCLK pulse width, low	t <sub>PWCKL</sub>	t <sub>CYCK</sub> - 20		ns	
PRCLK↑ delay from CLK↓	t <sub>DPK↑H</sub>		22	ns	
PRCLK↓ delay from CLK↓	t <sub>DPK↓L</sub>		22	ns	
<b>Reset Timing</b>					
RESIN setup to CLK↓	t <sub>SRICK</sub>	65		ns	
RESIN hold from CLK↓	t <sub>HCKR↓</sub>	20		ns	
RESET delay from CLK↓	t <sub>DCKRS</sub>		40	ns	

## AC Characteristics, μPD71011 (cont)

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Ready Timing (RDYSYN = 'H')</b>					
REN1, 2 setup to RDY1, 2	t <sub>SRERY</sub>	15		ns	
REN1, 2 hold from CLK↓	t <sub>HCKRE</sub>	0		ns	
RDY1, 2 setup to CLK↓	t <sub>SRYSCK</sub>	35		ns	
RDY1, 2 hold from CLK↓	t <sub>HCKRY</sub>	0		ns	
RDYSYN setup to CLK↓	t <sub>SRYSCK</sub>	50		ns	
RDYSYN hold from CLK↓	t <sub>HCKRYS</sub>	0		ns	
READY output delay from CLK↓	t <sub>DCKRDY</sub>	8		ns	READY↑
		8		ns	READY↓

### Ready Timing (RDYSYN = 'L')

REN1, 2 setup to RDY1, 2	t <sub>SRERY</sub>	15		ns	
REN1, 2 hold from CLK↓	t <sub>HCKRE</sub>	0		ns	
RDY1, 2 setup to CLK↓	t <sub>SRYSCK</sub>	35		ns	
RDY1, 2 hold from CLK↓	t <sub>HCKRY</sub>	0		ns	
RDYSYN setup to CLK↓	t <sub>SRYSCK</sub>	50		ns	
RDYSYN hold from CLK↓	t <sub>HCKRYS</sub>	0		ns	
READY output delay from CLK↓	t <sub>DCKRDY</sub>	8		ns	READY↑
		8		ns	READY↓

### Output Pin Timing

Rise time	t <sub>LH</sub>	20		ns	0.8 V → 2.0 V
Fall time	t <sub>HL</sub>	12		ns	2.0 V → 0.8 V

## AC Characteristics, μPD71011-10

f<sub>OSC</sub> = 20 MHz; T<sub>A</sub> = -10 to +70°C; V<sub>DD</sub> = +5 V ±5%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Clock Timing</b>					
EXFS input cycle time	t <sub>CYFS</sub>	50		ns	
EXFS pulse width, high	t <sub>PWFSH</sub>	20		ns	2.2 V measurement point
EXFS pulse width, low	t <sub>PWFSL</sub>	20		ns	0.8 V measurement point
OSC cycle time	t <sub>OSC</sub>	8	20	MHz	
CKSYN pulse width	t <sub>PWCT</sub>	2t <sub>CYFS</sub>		ns	
CKSYN setup time	t <sub>HFSTK</sub>	20		ns	from EXFS
CKSYN hold time	t <sub>SCTFS</sub>	20		ns	
CLK cycle time	t <sub>CYCK</sub>	100		ns	
CLK pulse width, high	t <sub>PWCKH</sub>	41		ns	3.0 V, f <sub>OSC</sub> = 20 MHz
CLK pulse width, low	t <sub>PWCKL</sub>	49		ns	1.5 V, f <sub>OSC</sub> = 20 MHz

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
CLK rise time	t <sub>LHCK</sub>	5		ns	1.5 V → 3.0 V, f <sub>OSC</sub> = 20 MHz
CLK fall time	t <sub>HLCK</sub>	5		ns	3.0 V → 1.5 V, f <sub>OSC</sub> = 20 MHz
CLK delay from OSC	t <sub>DCK</sub>	2	30	ns	CLK↑
		-6	28	ns	CLK↓
PRCLK cycle time	t <sub>CYPRK</sub>	200		ns	
PRCLK pulse width, high	t <sub>PWCKH</sub>	t <sub>CYCK</sub> - 20		ns	
PRCLK pulse width, low	t <sub>PWCKL</sub>	t <sub>CYCK</sub> - 20		ns	
PRCLK↑ delay from CLK↓	t <sub>DPRKH</sub>	22		ns	
PRCLK↓ delay from CLK↓	t <sub>DPRKL</sub>	22		ns	

### Reset Timing

RESIN setup to CLK↓	t <sub>SRIK</sub>	65		ns	
RESIN hold from CLK↓	t <sub>HCKRI</sub>	20		ns	
RESET delay from CLK↓	t <sub>DCKRS</sub>	20		ns	

### Ready Timing (RDYSYN = 'H')

REN1,2 setup to RDY1,2	t <sub>SRERY</sub>	15		ns	
REN1, 2 hold from CLK↓	t <sub>HCKRE</sub>	0		ns	
RDY1, 2 setup to CLK↓	t <sub>SRYSCK</sub>	35		ns	
RDY1, 2 hold from CLK↓	t <sub>HCKRY</sub>	0		ns	
RDYSYN setup to CLK↓	t <sub>SRYSCK</sub>	50		ns	
RDYSYN hold from CLK↓	t <sub>HCKRYS</sub>	0		ns	
READY output delay from CLK↓	t <sub>DCKRDY</sub>	8		ns	READY↑
		8		ns	READY↓

### Ready Timing (RDYSYN = 'L')

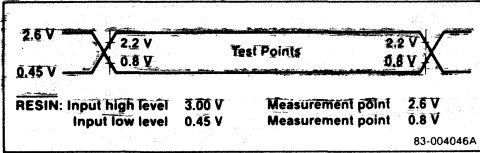
REN1,2 setup to RDY1,2	t <sub>SRERY</sub>	15		ns	
REN1, 2 hold from CLK↓	t <sub>HCKRE</sub>	0		ns	
RDY1, 2 setup to CLK↓	t <sub>SRYSCK</sub>	35		ns	
RDY1, 2 hold from CLK↓	t <sub>HCKRY</sub>	0		ns	
RDYSYN setup to CLK↓	t <sub>SRYSCK</sub>	50		ns	
RDYSYN hold from CLK↓	t <sub>HCKRYS</sub>	0		ns	
READY output delay from CLK↓	t <sub>DCKRDY</sub>	8		ns	READY↑
		8		ns	READY↓

### Output Pin Timing

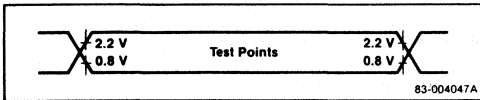
Rise time	t <sub>LH</sub>	20		ns	0.8 V → 2.0 V
Fall time	t <sub>HL</sub>	12		ns	2.0 V → 0.8 V

**Timing Waveforms**

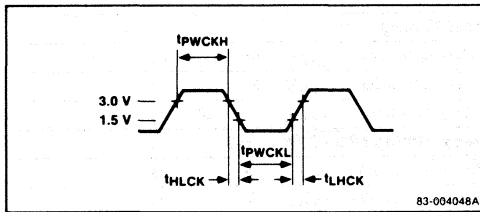
**AC Test Input (except RESIN)**



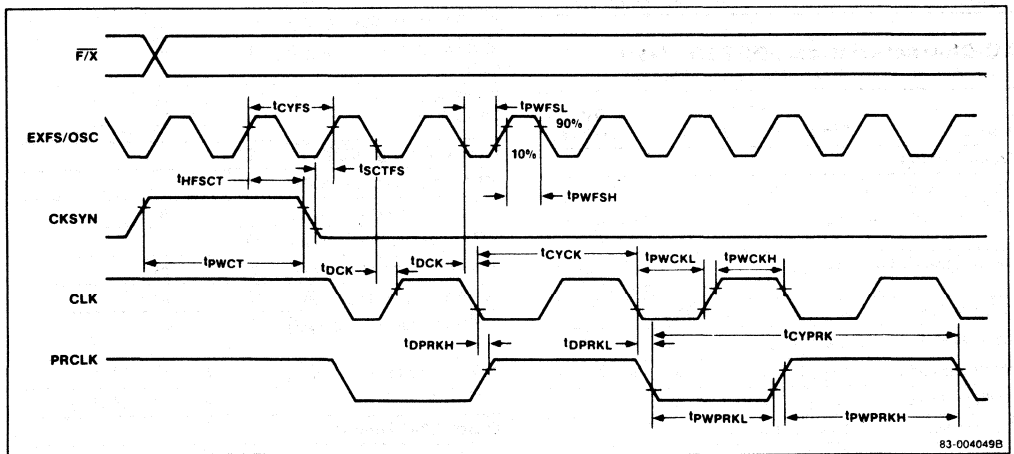
**AC Test Output (except CLK)**



**AC Test Input (CLK)**

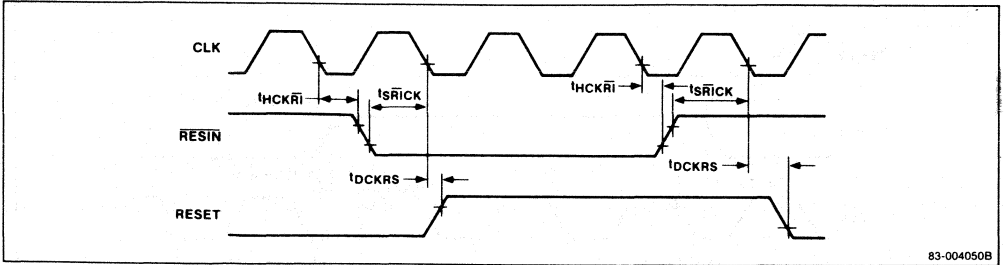


**CLK Pin**

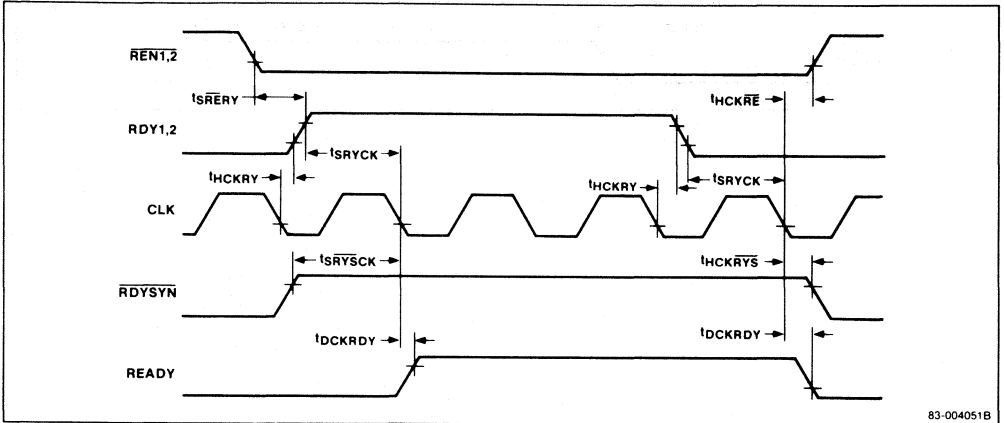


## Timing Waveforms (cont)

### RESET Pin



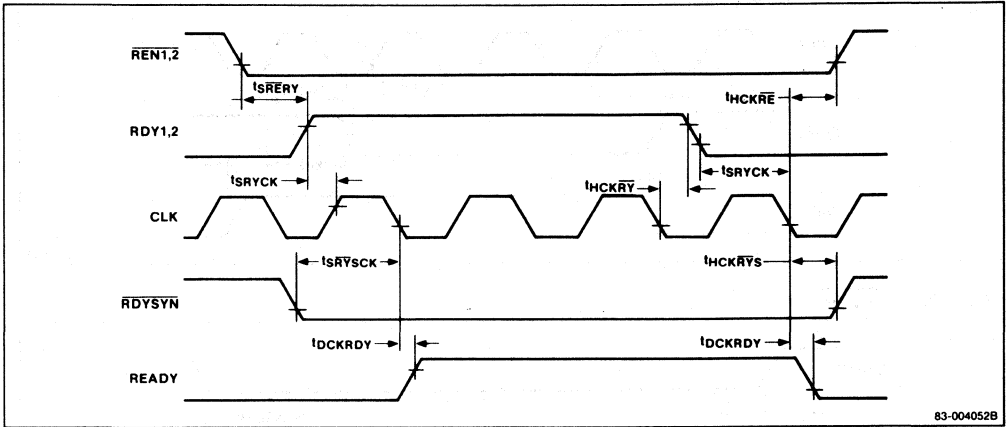
### READY Pin ( $\overline{RDYSYN} = 'H'$ )



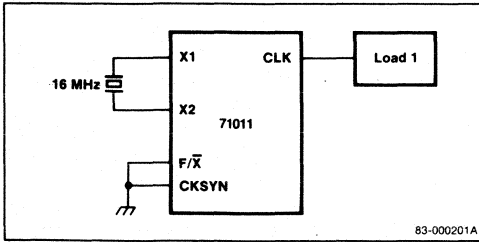


Timing Waveforms (cont)

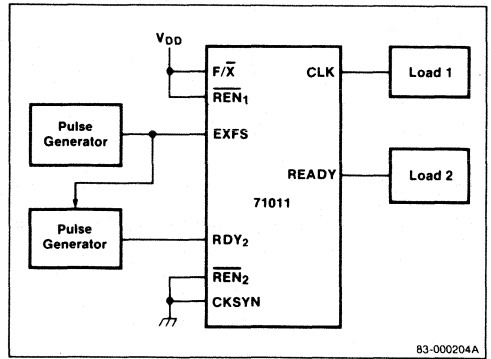
READY Pin ( $\overline{RDYSYN} = 'L'$ )



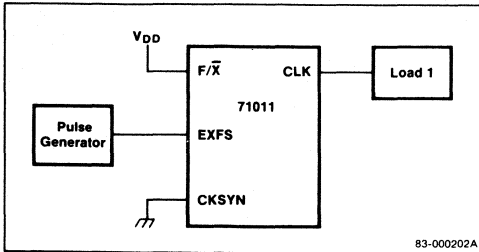
**Test Circuit for CLK High or Low Time (in Crystal Oscillation Mode)**



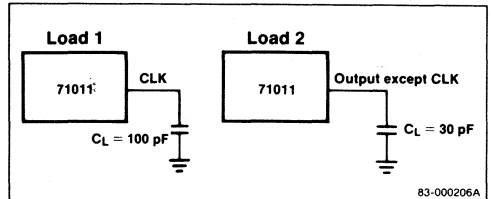
**Test Circuit for CLK to READY (in EXFS Oscillation Mode)**



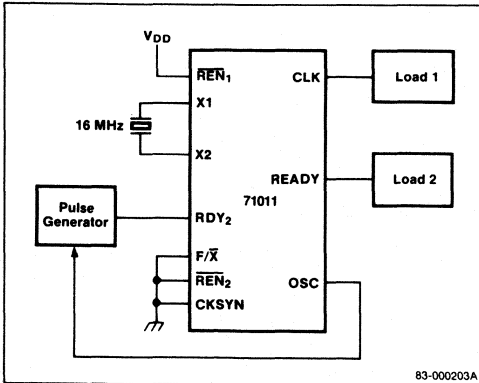
**Test Circuit for CLK High or Low Time (in EXFS Oscillation Mode)**



**Loading Circuits**



**Test Circuit for CLK to READY (in Crystal Oscillation Mode)**





## Description

The μPD71051 serial control unit is a CMOS USART designed to provide serial data communications in microcomputer systems. The CPU uses it as a peripheral I/O device and programs it to communicate in synchronous or asynchronous serial data transmission protocols, including IBM bisync.

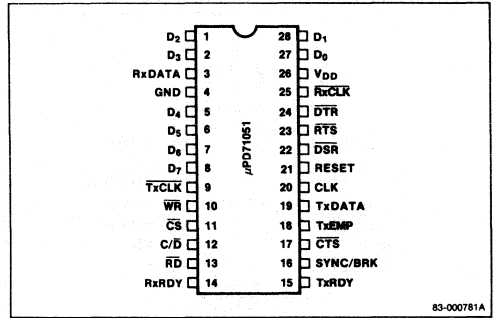
The USART receives serial data streams and converts them into parallel data characters for the CPU. While receiving serial data, the USART can also accept parallel data from the CPU, convert it to serial, and transmit the data. The USART signals the CPU when it has received or transmitted a character and requires service. The CPU may read complete USART status data at any time.

## Features

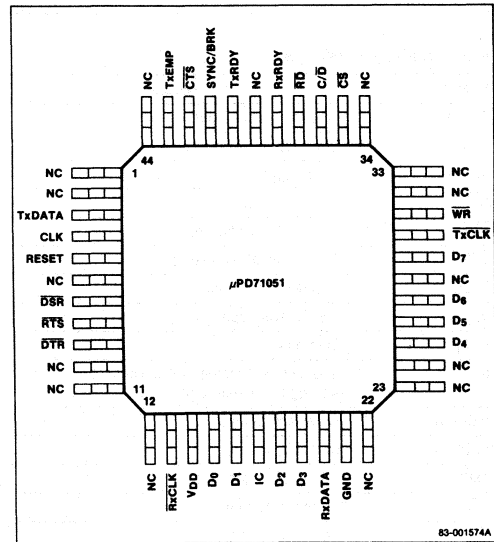
- Synchronous operation
  - One or two SYNC characters
  - Internal/external synchronization
  - Automatic SYNC character insertion
- Asynchronous operation
  - Clock rate: (baud rate)
  - x1, x16, or x64
  - Send stop bits: 1, 1.5, or 2 bits
  - Break transmission
  - Automatic break detection
  - Valid start bit detection
- Baud rate: DC - 240 kbit/s at x1 clock
- Full duplex, double-buffered transmitter/receiver
- Error detection: parity, overrun, and framing
- Five- to eight-bit characters
- Low-power standby mode
- Compatible with standard microcomputers
- Functionally equivalent to (except standby mode) and can replace the μPD8251AF
- CMOS technology
- Single +5 V ± 10% power supply
- Industrial temperature range -40 to +85°C
- 28-pin plastic DIP or PLCC or 44-pin plastic miniflat

## Pin Configurations

### 28-Pin Plastic DIP



### 44-Pin Plastic Flat Pack



## Ordering Information

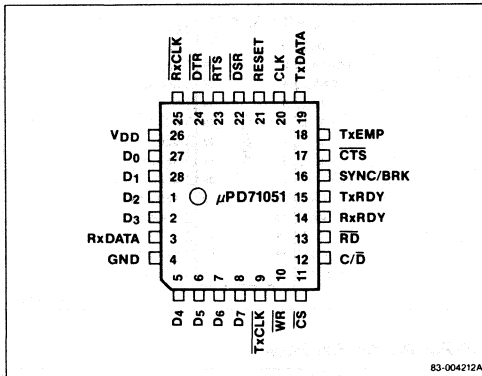
Part Number	Package Type	Max. Frequency of Operation
μPD71051C	28-pin plastic DIP	8 MHz
μPD71051GB	44-pin plastic flat pack	8 MHz
μPD71051L	28-pin PLCC	8 MHz

Part Number	Package Type	Max Frequency of Operation
μPD71051C-10	28-pin plastic DIP	10 MHz
μPD71051GB-10	44-pin plastic flat pack	10 MHz
μPD71051L-10	28-pin plastic PLCC	10 MHz

## μPD71051

### Pin Configurations (cont)

#### 28-Pin Plastic Leaded Chip Carrier (PLCC)



### Pin Identification

Symbol	Function
TxDATA	Transmit data output
CLK	Clock input
RESET	Reset input
DSR	Data set ready input
RTS	Request to send output
DTR	Data terminal ready output
RxCLK	Receiver clock input
V <sub>DD</sub>	+5 V power supply
D <sub>7-D<sub>0</sub></sub>	Data bus
IC	Internally connected (Do not connect any signal to an IC pin)
RxDATA	Receive data input
GND	Ground
TxCLK	Transmitter clock input
WR	Write strobe input
C <sub>S</sub>	Chip select input
C/ <sub>D</sub>	Control or data input
RD	Read strobe input
RxRDY	Receiver ready output
TxRDY	Transmitter ready output
SYNC/BRK	Synchronization/Break input/output
CTS	Clear to send input
TxEMP	Transmitter empty output
NC	Not connected

### Pin Functions

#### D<sub>7-D<sub>0</sub></sub> [Data Bus]

D<sub>7-D<sub>0</sub></sub> are an 8-bit, 3-state, bidirectional data bus. The bus transfers data by connecting to the CPU data bus.

#### RESET [Reset]

A high level to the RESET input resets the μPD71051 and puts it in an idle state. It performs no operations in the idle state. The μPD71051 enters standby mode when this signal falls from a high level to a low level. Standby mode is released when the CPU writes a mode byte to the μPD71051. The reset pulse width must be at least 6 t<sub>CYK</sub> cycles and the clock must be enabled.

#### CLK [Clock]

This clock input produces internal timing for the μPD71051. The clock frequency should be at least 30 times the transmitter or receiver clock input frequency (TxCLK, RxCLK) in sync or async mode with the X1 clock. This assures stable operation. The clock frequency must be more than 4.5 times the TxCLK or RxCLK in async mode using x16 or x64 clock mode.

#### C<sub>S</sub> [Chip Select]

The C<sub>S</sub> input selects the μPD71051. The μPD71051 is selected by setting C<sub>S</sub> = 0. When C<sub>S</sub> = 1, the μPD71051 is not selected, the data bus (D<sub>7-D<sub>0</sub></sub>) is in the high impedance state, and the RD and WR signals are ignored.

#### RD [Read Strobe]

The RD input is low when reading data or status information from the μPD71051.

#### WR [Write Strobe]

The WR input is low when writing data or a control byte to the μPD71051.

#### C/<sub>D</sub> [Control or Data]

The C/<sub>D</sub> input determines the data type when accessing the μPD71051. When C/<sub>D</sub> = 1, the data is a control byte (table 1) or status. When C/<sub>D</sub> = 0, the data is character data. This pin is normally connected to the least significant bit (A<sub>0</sub>) of the CPU address bus.

#### DSR [Data Set Ready]

DSR is a general-purpose input pin that can be used for modem control. The status of this pin can be determined by reading bit 7 of the status byte.

## **$\overline{\text{DTR}}$ [Data Terminal Ready]**

$\overline{\text{DTR}}$  is a general-purpose output pin that can be used for modem control. The state of this pin can be controlled by writing bit 1 of the command byte. If bit 1 = 0, then  $\overline{\text{DTR}} = 1$ . If bit 1 = 1, then  $\overline{\text{DTR}} = 0$ .

## **$\overline{\text{RTS}}$ [Request to Send]**

$\overline{\text{RTS}}$  is a general-purpose output pin that can be used for modem control. The status of this pin can be controlled by writing bit 5 of the command byte. If bit 5 = 1, then  $\overline{\text{RTS}} = 0$ . If bit 5 = 0, then  $\overline{\text{RTS}} = 1$ .

## **$\overline{\text{CTS}}$ [Clear to Send]**

The  $\overline{\text{CTS}}$  input controls data transmission. The  $\mu\text{PD71051}$  is able to transmit serial data when  $\overline{\text{CTS}} = 0$  and the command byte sets  $\text{TxEN} = 1$ . If  $\overline{\text{CTS}}$  is set equal to 1 during transmission, the sending operation stops after sending all currently written data and the  $\text{TxDATA}$  pin goes high.

## **$\text{TxDATA}$ [Transmit Data]**

The  $\mu\text{PD71051}$  sends serial data over the  $\text{TxDATA}$  output.

## **$\text{TxRDY}$ [Transmitter Ready]**

The  $\text{TxRDY}$  output tells the CPU that the transmit data buffer in the  $\mu\text{PD71051}$  is empty; that is, that new transmit data can be written. This signal is masked by the  $\text{TxEN}$  bit of the command byte and by the  $\overline{\text{CTS}}$  input. It can be used as an interrupt signal to request data from the CPU.

The status of  $\text{TxRDY}$  can be determined by reading bit 0 of the status byte. This allows the  $\mu\text{PD71051}$  to be polled. Note that  $\text{TxRDY}$  of the status byte is not masked by  $\overline{\text{CTS}}$  or  $\text{TxEN}$ .

$\text{TxRDY}$  is cleared to 0 by the falling edge of  $\overline{\text{WR}}$  when the CPU writes transmit data to the  $\mu\text{PD71051}$ . Data in the transmit data buffer that has not been sent is destroyed if transmit data is written while  $\text{TxRDY} = 0$ .

## **$\text{TxEMP}$ [Transmitter Empty]**

The  $\mu\text{PD71051}$  reduces CPU overhead by using a double buffer; the transmit data buffer (second buffer) and the transmit buffer (first buffer) in the transmitter. When the CPU writes transmit data to the transmit data buffer (second buffer), the  $\mu\text{PD71051}$  sends data by transferring the contents of the second buffer to the first buffer, after transmitting the contents of the first buffer.

This empties the second buffer and  $\text{TxRDY}$  is set to 1. The  $\text{TxEMP}$  output becomes 1 when the contents of the first buffer are sent and the second buffer is empty. Thus,  $\text{TxEMP} = 1$  shows that both buffers are empty. In half-duplex operation, you can determine when to change from sending to receiving by testing  $\text{TxEMP} = 1$ .

When  $\text{TxEMP} = 1$  occurs in async mode, the  $\text{TxDATA}$  pin goes high. When the CPU writes transmit data,  $\text{TxEMP}$  is set to 0 and data transmission resumes.

When  $\text{TxEMP} = 1$  occurs in sync mode, the  $\mu\text{PD71051}$  loads SYNC characters from the SYNC character register and sends them through the  $\text{TxDATA}$  pin.  $\text{TxEMP}$  is set to 0 and resumes sending data after sending (one or two) SYNC characters and the CPU writes new transmit data to the  $\mu\text{PD71051}$ .

## **$\overline{\text{TxCLK}}$ [Transmitter Clock]**

The  $\overline{\text{TxCLK}}$  input is the reference clock input that determines the transmission rate. Data is transmitted at the same rate as  $\overline{\text{TxCLK}}$  in sync mode. In async mode, set  $\overline{\text{TxCLK}}$  to 1, 16, or 64 times the transmission rate. Serial data from  $\text{TxDATA}$  is sent at the falling edge of  $\overline{\text{TxCLK}}$ .

For example, a rate of 19200 baud in sync mode means that  $\overline{\text{TxCLK}}$  is 19.2 kHz. A rate of 2400 baud in async mode can represent a  $\overline{\text{TxCLK}}$  of:

- x1 clock = 2.4 kHz
- x16 clock = 38.4 kHz
- x64 clock = 153.6 kHz

## **$\text{RxDATA}$ [Receive Data]**

The  $\mu\text{PD71051}$  receives serial data through the  $\text{RxDATA}$  input.

## **$\text{RxRDY}$ [Receiver Ready]**

The  $\text{RxRDY}$  output becomes 1 when the  $\mu\text{PD71051}$  receives one character of data and transfers that data to the receive data buffer; that is, when the receive data can be read. This signal can be used as an interrupt signal for a data read request to the CPU. You can determine the status of  $\text{RxRDY}$  by reading bit 1 of the status byte and use the  $\mu\text{PD71051}$  in a polling application.  $\text{RxRDY}$  becomes 0 when the CPU reads the receive data.

Unless the CPU reads the receive data (after  $\text{RxRDY} = 1$  is set) before the next single character is received and transferred to the receive buffer, an overrun error occurs, and the  $\text{OVE}$  status bit is set. The unread data in the receive data buffer is overwritten by newly transferred data and lost.

RxRDY is set to 0 in the receive disable state. This state is set by changing the RxEN bit to 0 through the command byte. After RxEN is set to 1 (making receiving possible), RxRDY becomes 1 whenever new characters are received and transferred to the receive data buffer.

**SYNC/BRK [Synchronization/Break]**

The SYNC pin detects synchronization characters in sync mode. The SYNC mode byte selects internal or external SYNC detection. The SYNC pin becomes an output when internal synchronization is set, and an input when external synchronization is set.

The SYNC output goes high when the μPD71051 detects a SYNC character in internal synchronization. When two SYNC characters are used, SYNC goes high when the last bit of the two consecutive SYNC characters is detected. You can read the status of the SYNC signal in bit 6 of the status byte. Both the SYNC pin and status are set to 0 by a read status operation.

In external synchronization, in order for the external circuit to detect synchronization, a high level of at least one period of RxCLK must be input to the SYNC pin. When the μPD71051 detects the high level, it begins to receive data, starting at the rising edge of the next RxCLK. The high level input may be removed when synchronization is released.

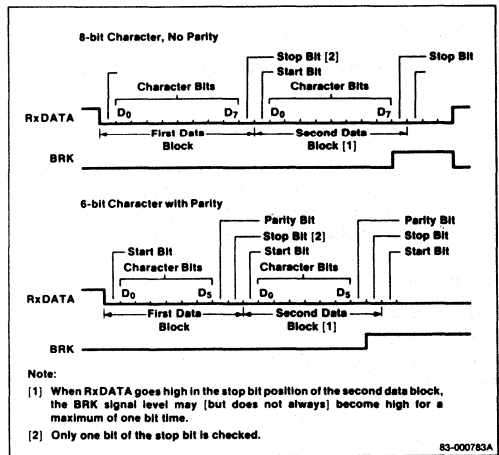
The BRK output is used only in async mode and shows the detection of a break state. BRK goes high when a low level signal is input to the RxDATA pin for two character bit lengths (including the start, stop, and parity bits). As with SYNC, you can read the status of BRK in bit 6 of the status byte. BRK is not cleared by the read operation.

The set BRK signal is cleared when the RxDATA pin returns to high level, or when the μPD71051 is reset by hardware or software. The SYNC/BRK pin goes low on reset, regardless of previous mode. Figure 1 shows the break state and BRK signal.

**Table 1. Control Signals and Operations**

CS	RD	WR	C/D	μPD71051	CPU Operation
0	0	1	0	Receive data buffer ↓ Data bus	Read receive data
0	0	1	1	Status register ↓ Data bus	Read status
0	1	0	0	Data bus ↓ Transmit data buffer	Write transmit data
0	1	0	1	Data bus ↓ Control byte register	Write control byte
0	1	1	x	Data bus High impedance	None
1	x	x	x	Data bus High impedance	None

**Figure 1. Break Status and Break Signal**



**RxCLK [Receiver Clock]**

RxCLK is a reference clock input that controls the receive data rate. In sync mode, the receiving rate is the same as RxCLK. In async mode, RxCLK can be 1, 16, or 64 times the receive rate. Serial data from RxDATA is input by the rising edge of RxCLK.

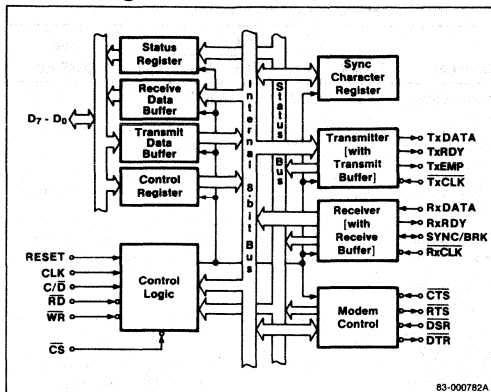
**V<sub>DD</sub> [Power]**

+5 V power supply.

**GND [Ground]**

Ground.

## Block Diagram



## μPD71051 Functions

The μPD71051 is a CMOS serial control (USART) unit that provides serial communications in microcomputer systems. The CPU handles the μPD71051 as an ordinary I/O device.

The μPD71051 can operate in synchronous or asynchronous systems. In sync mode, the character bit length, number of sync characters, and sync detection mode must be designated. In async mode, the communication rate, character bit length, stop bit length, etc., must be designated. The parity bit may be designated in either mode.

The μPD71051 converts parallel data received from the CPU into serial transmitted data (from the TxDATA pin), and converts serial input data (from the RxDATA pin) into parallel data so that the CPU can read it (receiving operation).

The CPU can read the current status of the μPD71051 and can process data after checking the status, after checking for transfer errors, and μPD71051 data buffer status.

The μPD71051 can be reset under hardware or software control to a standby mode that consumes less power and removes the device from system operation. In this mode, the μPD71051's previous operating mode is released and it waits for a mode byte to set the mode. The μPD71051 leaves standby mode and shifts to a designated operating mode when the CPU writes a mode byte to it.

## Status Register

The status register allows the CPU to read the status of the μPD71051 except in standby mode. This register indicates status and allows the CPU to manage data reading, writing, and error handling during operations.

## Receive Data Buffer

When the receiver has converted the serial data input from the RxDATA pin into parallel data, the converted data is stored in the receive data buffer. The CPU can then read it. Data for one character entering the receive buffer is transferred to the receive data buffer and RxRDY becomes 1, requesting that the CPU read the data.

## Transmit Data Buffer

The transmit data buffer holds the parallel data from the CPU that the transmitter will convert to serial data and output from the TxDATA pin. When the CPU writes transmit data to the μPD71051, the μPD71051 stores data in the transmit data buffer. The transmit data buffer transfers the data to the transmitter, which sends the data from the TxDATA pin.

## Control Register

This register stores the mode and the command bytes.

## Control Logic

The control logic sends control signals to the internal blocks and controls the operation of the μPD71051 based on internal and external signals.

## Synchronous Character Register

This register stores one or two SYNC characters used in sync mode. During transmission, the SYNC characters stored in this register are output from the TxDATA pin when the CPU does not send a new character and TxEMP status is set. During receiving, synchronization is established when the characters received and the SYNC characters stored in this register are the same.

## Transmitter

The contents of the transmit data buffer are transferred to the transmitter, converted from parallel to serial, and output from the TxDATA pin. The transmitter adds start, stop, and parity bits.



### Receiver

The receiver converts serial data input from the RxDATA pin into parallel data and transfers the parallel data to the receive data buffer, allowing the CPU to read it.

The receiver detects SYNC characters and checks parity bits in sync mode. It detects the start and stop bits, and checks parity in the async mode.

In async mode, receiving does not begin (the start bit is not detected) until one effective stop bit (high level) is input to the RxDATA pin and Receive Enable (RxEN = 1) is set after setting up the mode.

### Modem Control

This block controls the  $\overline{\text{CTS}}$ ,  $\overline{\text{RTS}}$ ,  $\overline{\text{DSR}}$ , and  $\overline{\text{DTR}}$  modem interface pins. The RTS, DSR, and DTR pins can also be used as general-purpose I/O pins.

### Absolute Maximum Ratings

$T_A = +25^\circ\text{C}$

Power supply voltage, $V_{DD}$	-0.5 to +7.0 V
Input voltage, $V_I$	-0.5 to $V_{DD} + 0.3$ V
Output voltage, $V_O$	-0.5 to $V_{DD} + 0.3$ V
Operating temperature, $T_{OPT}$	-40°C to +85°C
Storage temperature, $T_{STG}$	-65°C to +150°C
Power dissipation, $P_{D_{MAX}}$	1.0 W

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### Capacitance

$T_A = +25^\circ\text{C}$ ,  $V_{DD} = 0$  V

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input capacitance	$C_I$	10		pF	$f_c = 1\text{MHz}$ Unmeasured pins returned to 0 V
I/O capacitance	$C_{IO}$	20		pF	

### DC Characteristics

$T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ ,  $V_{DD} = +5\text{V} \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage high	$V_{IH}$	2.2		$V_{DD} + 0.3$	V	
Input voltage low	$V_{IL}$	-0.5		0.8	V	
Output voltage high	$V_{OH}$	$0.7 \times V_{DD}$			V	$I_{OH} = -400 \mu\text{A}$
Output voltage low	$V_{OL}$			0.4	V	$I_{OL} = 2.5 \text{mA}$
Input leakage current high	$I_{LIH}$			10	$\mu\text{A}$	$V_I = V_{DD}$
Input leakage current low	$I_{LIL}$			-10	$\mu\text{A}$	$V_I = 0$ V
Output leakage current high	$I_{LOH}$			10	$\mu\text{A}$	$V_O = V_{DD}$
Output leakage current low	$I_{LOL}$			-10	$\mu\text{A}$	$V_O = 0$ V
Supply current	$I_{DD1}$			10	mA	8/10 MHz operation
		$I_{DD2}$	50	100	$\mu\text{A}$	Stand-by mode $V_I = V_{DD} - 0.1$ V or 0.1 V 8 MHz clock Input

## AC Characteristics, μPD71051

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V, ±10%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Read Cycle</b>					
Address set-up to RD ↓	t <sub>SAR</sub>	0		ns	$\overline{CS}$ , C/ $\overline{D}$
Address hold from RD ↑	t <sub>HRA</sub>	0		ns	$\overline{CS}$ , C/ $\overline{D}$
R $\overline{D}$ low level width	t <sub>RRL</sub>	150		ns	
Data delay from RD ↓	t <sub>DRD</sub>		120	ns	C <sub>L</sub> = 150 pF
Data float from RD ↑	t <sub>FRD</sub>	10	80	ns	
Port ( $\overline{DSR}$ , $\overline{CTS}$ ) set-up to RD ↓	t <sub>SPR</sub>	20		t <sub>CYK</sub>	
<b>Write Cycle</b>					
Address set-up to WR ↓	t <sub>SAW</sub>	0		ns	$\overline{CS}$ , C/ $\overline{D}$
Address hold from WR ↑	t <sub>HWA</sub>	0		ns	$\overline{CS}$ , C/ $\overline{D}$
WR low level width	t <sub>WWL</sub>	150		ns	
Data set-up to WR ↑	t <sub>SDW</sub>	80		ns	
Data hold from WR ↑	t <sub>HWD</sub>	0		t <sub>CYK</sub>	
Port ( $\overline{DTR}$ , $\overline{RTS}$ ) delay from WR ↑	t <sub>DWP</sub>		8	t <sub>CYK</sub>	
Write recovery time	t <sub>RV</sub>	6		t <sub>CYK</sub>	Mode Initialize
		8		t <sub>CYK</sub>	Async Mode
		16		t <sub>CYK</sub>	Sync Mode
<b>Serial Transfer Timing</b>					
CLK cycle time	t <sub>CYK</sub>	125	DC	ns	
CLK high level width	t <sub>KKH</sub>	50		ns	
CLK low level width	t <sub>KKL</sub>	35		ns	
CLK rise time	t <sub>KR</sub>	5	20	ns	
CLK fall time	t <sub>KF</sub>	5	20	ns	
TxDATA delay from TxCLK	t <sub>DTKTD</sub>		0.5	μs	

## AC Characteristics, μPD71051

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V, ±10%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Serial Transfer Timing (cont)</b>					
Transmitter input clock pulse width low level	t <sub>TKTL</sub>	12		t <sub>CYK</sub>	1xBR (1)
		1		t <sub>CYK</sub>	16x, 64xBR
Transmitter input clock pulse width high level	t <sub>TKTH</sub>	15		t <sub>CYK</sub>	1xBR
		3		t <sub>CYK</sub>	16x, 64xBR
Transmitter input clock frequency	f <sub>TK(2)</sub>	DC	240	kHz	1xBR
		DC	1536	kHz	16xBR
		DC	1536	kHz	64xBR
Receiver input clock pulse width low level	t <sub>RKRL</sub>	12		t <sub>CYK</sub>	1xBR
		1		t <sub>CYK</sub>	16x, 64xBR
Receiver input clock pulse width high level	t <sub>RKRH</sub>	15		t <sub>CYK</sub>	1xBR
		3		t <sub>CYK</sub>	16x, 64xBR
Receiver input clock frequency	f <sub>RK(2)</sub>	DC	240	kHz	1xBR
		DC	1536	kHz	16xBR
		DC	1536	kHz	64xBR
RxDATA set-up to sampling pulse	t <sub>SRDSP</sub>	1		μs	
RxDATA hold from sampling pulse	t <sub>HSPRD</sub>	1		μs	
TxEMP delay time (TxDATA)	t <sub>DTXEP</sub>		20	t <sub>CYK</sub>	
TxRDY delay time (TxRDY↑)	t <sub>DTXR</sub>		8	t <sub>CYK</sub>	
TxRDY delay time (TxRDY↓)	t <sub>DWTXR</sub>		200	ns	
RxRDY delay time (RxRDY↑)	t <sub>DRXR</sub>		26	t <sub>CYK</sub>	
RxRDY delay time (RxRDY↓)	t <sub>DRRXR</sub>		200	ns	
SYNC output delay time (for internal sync)	t <sub>DRKSY</sub>		26	t <sub>CYK</sub>	
SYNC input set-up time (for external sync)	t <sub>SSYRK</sub>	18		t <sub>CYK</sub>	
RESET pulse width		6		t <sub>CYK</sub>	

### Note:

(1) BR = Baud rate

(2) 1xBR: f<sub>TK</sub> or f<sub>RK</sub> ≤ 1/30 t<sub>CLK</sub>, 16x, 64xBR: f<sub>TK</sub> or f<sub>RK</sub> ≤ 1/4.5 t<sub>CLK</sub>

(3) System CLK is needed during reset operation

(4) Status update can have a maximum delay of 28 t<sub>CYK</sub> from the event effecting the status.

**AC Characteristics, μPD71051-10**

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V, ±10%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Read Cycle</b>					
Address set-up to RD ↓	t <sub>SAR</sub>	0		ns	CS, C/ $\bar{D}$
Address hold from RD ↑	t <sub>HRA</sub>	0		ns	CS, C/ $\bar{D}$
R $\bar{D}$ low level width	t <sub>RRL</sub>	95		ns	
Data delay from RD ↓	t <sub>DRD</sub>		85	ns	C <sub>L</sub> = 150 pF
Data float from RD ↑	t <sub>FRD</sub>	10	65	ns	
Port ( $\overline{DSR}$ , CTS) set-up to RD ↓	t <sub>SPR</sub>	20		t <sub>CYK</sub>	
<b>Write Cycle</b>					
Address set-up to WR ↓	t <sub>SAW</sub>	0		ns	CS, C/ $\bar{D}$
Address hold from WR ↑	t <sub>HWA</sub>	0		ns	CS, C/ $\bar{D}$
WR low level width	t <sub>WWL</sub>	95		ns	
Data set-up to WR ↑	t <sub>SDW</sub>	95		ns	
Data hold from WR ↑	t <sub>HWD</sub>	0		t <sub>CYK</sub>	
Port ( $\overline{DTR}$ , RTS) delay from WR ↑	t <sub>DWP</sub>		8	t <sub>CYK</sub>	
Write recovery time	t <sub>RV</sub>	6		t <sub>CYK</sub>	Mode Initialize
		8		t <sub>CYK</sub>	Async Mode
		16		t <sub>CYK</sub>	Sync Mode
<b>Serial Transfer Timing</b>					
CLK cycle time	t <sub>CYK</sub>	100	DC	ns	
CLK high level width	t <sub>KKH</sub>	35		ns	
CLK low level width	t <sub>KKL</sub>	25		ns	
CLK rise time	t <sub>KR</sub>	5	20	ns	
CLK fall time	t <sub>KF</sub>	5	20	ns	
TxDATA delay from TxCLK	t <sub>DTKTD</sub>		0.5	μs	

**AC Characteristics, μPD71051-10**

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V, ±10%

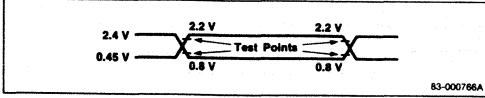
Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Serial Transfer Timing (cont)</b>					
Transmitter input clock pulse width low level	t <sub>TKTL</sub>	12		t <sub>CYK</sub>	1xBR (1)
		1		t <sub>CYK</sub>	16x, 64xBR
Transmitter input clock pulse width high level	t <sub>TKTH</sub>	15		t <sub>CYK</sub>	1xBR
		3		t <sub>CYK</sub>	16x, 64xBR
Transmitter input clock frequency	f <sub>TK(2)</sub>	DC	300	kHZ	1xBR
		DC	1920	kHZ	16xBR
		DC	1920	kHZ	64xBR
Receiver input clock pulse width low level	t <sub>RKRL</sub>	12		t <sub>CYK</sub>	1xBR
		1		t <sub>CYK</sub>	16x, 64xBR
Receiver input clock pulse width high level	t <sub>RKRH</sub>	15		t <sub>CYK</sub>	1xBR
		3		t <sub>CYK</sub>	16x, 64xBR
Receiver input clock frequency	f <sub>RK(2)</sub>	DC	300	kHZ	1xBR
		DC	1920	kHZ	16xBR
		DC	1920	kHZ	64xBR
RxDATA set-up to sampling pulse	t <sub>SRDSP</sub>	1		μs	
RxDATA hold from sampling pulse	t <sub>HSPRD</sub>	1		μs	
TxE <sub>MP</sub> delay time (TxDATA)	t <sub>DTXEP</sub>		20	t <sub>CYK</sub>	
TxR <sub>DY</sub> delay time (TxRDY↑)	t <sub>DTXR</sub>		8	t <sub>CYK</sub>	
TxR <sub>DY</sub> delay time (TxRDY↓)	t <sub>DWTXR</sub>		100	ns	
RxR <sub>DY</sub> delay time (RxRDY↑)	t <sub>DRXR</sub>		26	t <sub>CYK</sub>	
RxR <sub>DY</sub> delay time (RxRDY↓)	t <sub>DRRXR</sub>		100	ns	
SYNC output delay time (for internal sync)	t <sub>DRKSY</sub>		26	t <sub>CYK</sub>	
SYNC input set-up time (for external sync)	t <sub>SSYRK</sub>	18		t <sub>CYK</sub>	
RESET pulse width		6		t <sub>CYK</sub>	

**Note:**

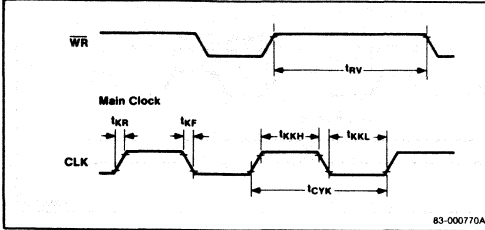
- (1) BR = Baud rate
- (2) 1xBR: f<sub>TK</sub> or f<sub>RK</sub> ≤ 1/30 t<sub>CLK</sub>, 16x, 64xBR: f<sub>TK</sub> or f<sub>RK</sub> ≤ 1/4.5 t<sub>CLK</sub>
- (3) System CLK is needed during reset operation
- (4) Status update can have a maximum delay of 28 t<sub>CYK</sub> from the event effecting the status.

## Timing Waveforms

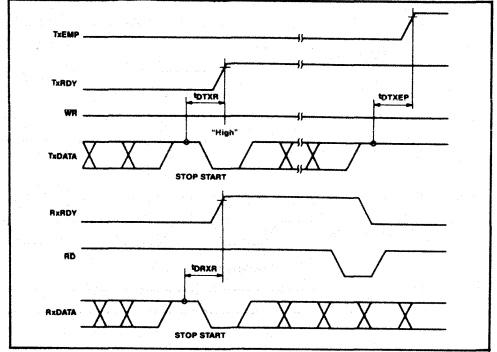
### AC Test Input



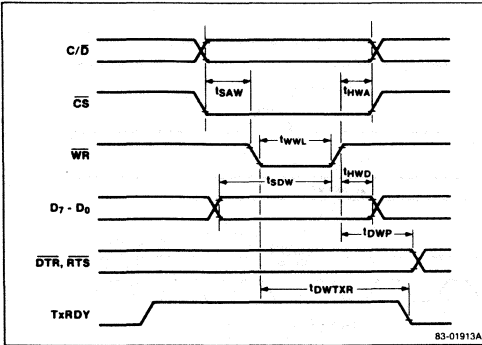
### Write Recovery Time



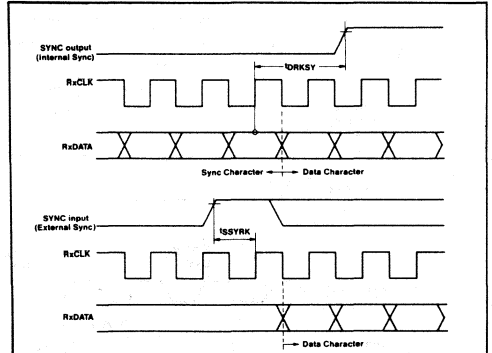
### Flag Timings



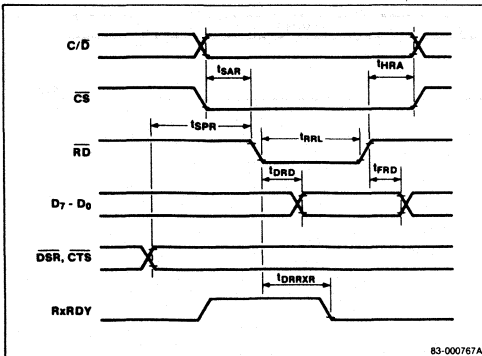
### Write Data Cycle



### Sync Timing

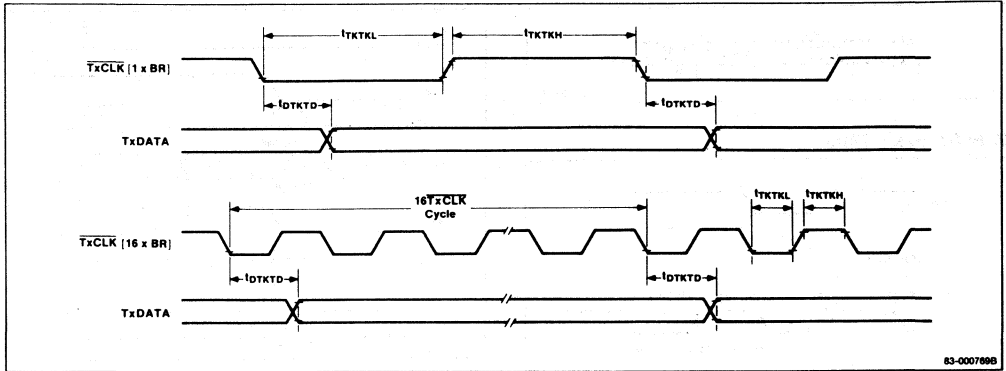


### Read Data Cycle



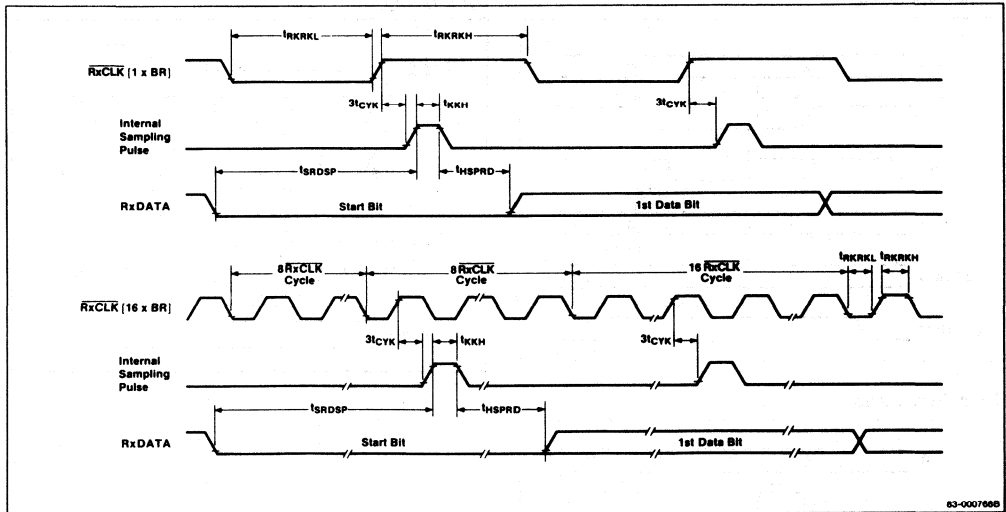
Timing Waveforms (cont)

Transmitter Clock and TxDATA



63-0007669

Receiver Clock and RxDATA Timing



63-0007669

## Connecting the μPD71051 to the System

The CPU uses the μPD71051 as an I/O device by allocating two I/O addresses, set by the value of C/D. One I/O address is allocated when the level of C/D is low and becomes a port to the transmit and receive data register. The other I/O address is allocated when C/D is high and becomes a port to the mode, command, and status registers. Generally, the least significant bit (A<sub>0</sub>) of the CPU address bus is connected to C/D to get a continuous I/O address. This is shown in figure 2.

Pins TxRDY and RxRDY are connected to the CPU or, when interrupts are used, to the interrupt pin of the interrupt controller.

## Operating the μPD71051

Start with a hardware reset (set the RESET pin high) after powering on the μPD71051. This puts the μPD71051 into standby mode and it waits for a mode byte. In async mode, the μPD71051 is ready for a command byte after the mode byte; the mode byte sets the communication protocol to the async mode. In sync mode, the μPD71051 waits for one or two SYNC characters to be sent after the mode byte; set C/D = 1. A command byte may be sent after the SYNC characters are written. Figure 3 shows this operation sequence.

In both modes, it is possible to write transmit data, read receive data, read status, and write more command bytes after the first command byte is written. The μPD71051 performs a reset, enters standby mode, and returns to a state where it waits for a mode byte when the command byte performs a software reset.

Figure 2. System Connection

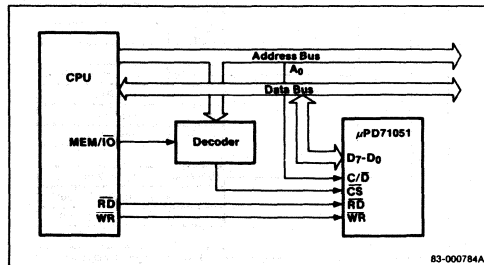
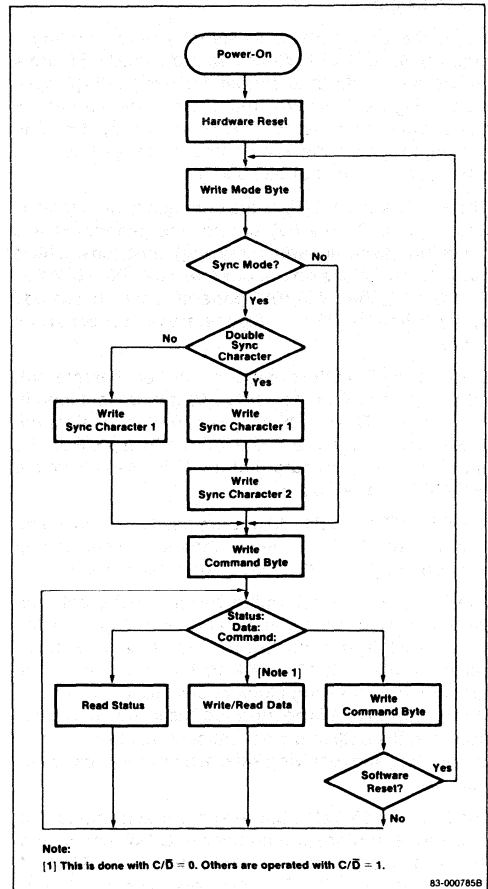


Figure 3. μPD71051 Operating Procedure



**Mode Register**

When the μPD71051 is in standby mode, writing a mode byte to it will release standby mode. Figure 4 shows the mode byte format for designating async mode. Figure 5 shows the mode byte format for designating sync mode. Bits 0 and 1 must be 00 to designate sync mode. Async mode is designated by all other combinations of bits 0 and 1.

The P1, P0 and L1, L0 bits are common to both modes. Bits P1 and P0 (parity) control the generation and checking (sending and receiving) functions. These parity bit functions do not operate when P0 = 0. When P1, P0 = 01, the μPD71051 generates and checks odd parity. When P1, P0 = 11, it generates and checks even parity.

Bits L1 and L0 set the number of bits per character (n). Additional bits such as parity bits are not included in this number. Given n bits, the μPD71051 receives the lower n bits of the 8-bit data written by the CPU. The upper bits (8 - n) of data that the CPU reads from the μPD71051 are set to zero.

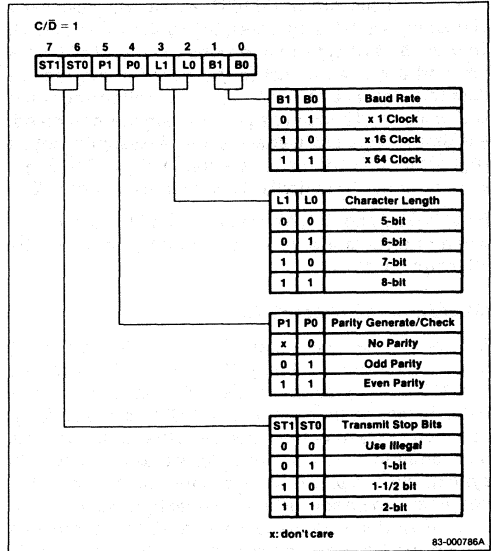
The ST1, ST0 and B1, B0 bits are used in async mode. The ST1 and ST0 bits determine the number of stop bits added by the μPD71051 during transmission.

The B1 and B0 bits determine the relationship between the baud rates for sending and receiving, and the clocks TxCLK and RxCLK. B1 and B0 select a multiplication rate of 1, 16, or 64 for the frequency of the sending and receiving clock relative to the baud rate. Multiplication by 1 is not normally used in async mode. Note that the data and clock must be synchronized on the sending and receiving sides when multiplication by 1 is used.

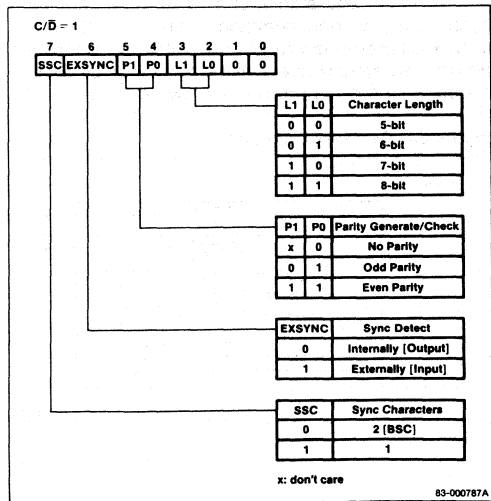
The SSC and EXSYNC bits are used in sync mode. The SSC bit determines the number of SYNC characters. SSC = 1 designates one SYNC character. SSC = 0 designates two SYNC characters. The number of SYNC characters determined by the SSC bit are written to the μPD71051 immediately after writing the mode byte.

The EXSYNC bit determines whether sync detection during receiving operations is internal or external. EXSYNC = 1 selects external sync detection and EXSYNC = 0 selects internal sync detection.

**Figure 4. Mode Byte for Setting Asynchronous Mode**



**Figure 5. Mode byte for Setting Synchronous Mode**



## Command Register

Commands are issued to the μPD71051 by the CPU by command bytes that control the sending and receiving operations of the μPD71051. A command byte is sent after the mode byte (in sync mode, a command byte may only be sent after writing SYNC characters) and the CPU must set  $C/\bar{D} = 1$ . Figure 6 shows the command byte format.

Bit EH is set to 1 when entering hunt phase to synchronize in sync mode. Bit RxEN should also be set to 1 at that time. Data reception begins when SYNC characters are detected and synchronization is achieved, thus releasing hunt phase.

When bit SRES is set to 1, a software reset is executed, and the μPD71051 goes into standby mode and waits for a mode byte.

Bit RTS controls the  $\overline{RTS}$  output pin.  $\overline{RTS}$  is low when the RTS bit = 1, and goes high when RTS = 0.

Setting bit ECL to 1 clears the error flags (PE, OVE, and FE) in the status register. Set ECL to 1 when entering the hunt phase or enabling the receiver.

Bit SBRK sends a break. When SBRK = 1, the data currently being sent is destroyed and the TxDATA pin goes low. Set SBRK = 0 to release a break. Break also works when TxEN = 0 (send disable).

Bit RxEN enables and disables the receiver. RxEN = 1 enables the receiver and RxEN = 0 disables the receiver. Synchronization is lost if RxEN = 0 during sync mode.

Bit DTR controls the  $\overline{DTR}$  output pin.  $\overline{DTR}$  goes low when the DTR bit = 1 and goes high when the DTR bit = 0.

The TxEN bit enables and disables the transmitter. TxEN = 1 enables the transmitter and TxEN = 0 disables the transmitter. When TxEN = 0, sending stops and the TxDATA pin goes high (mark status) after all the currently written data is sent.

## Status Register

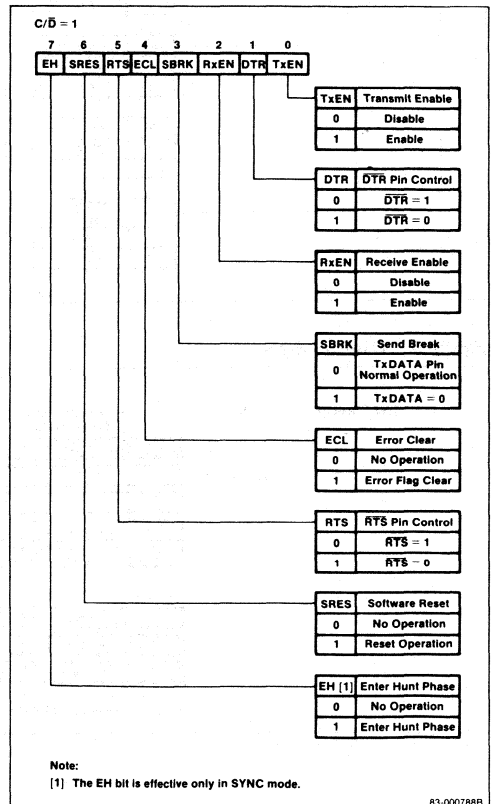
The CPU can read the status of the μPD71051 at any time except when the μPD71051 is in standby mode. Status can be read after setting  $C/\bar{D} = 1$  and  $\bar{RD} = 0$ . Status is not updated while being read. Status updating is delayed at least 28 clock periods after an event that affects the status. Figure 7 shows the format of the status register.

The TxEMP and RxRDY bits have the same meaning as the pins of the same name. The SYNC/BRK bit generally has the same meaning as the SYNC/BRK pin. In external synchronization mode, the status of this bit does not always coincide with the pin. In this case, the SYNC pin becomes an input and the status bit goes to 1 when a rising edge is detected at the input. The status bit remains at 1 until it is read, even when the input level at the SYNC pin goes low. The status bit becomes 1 when a SYNC character is input with the RxDATA input, even when the pin is at a low level.

The DSR bit shows the status of the  $\overline{DSR}$  input pin. The status bit is 1 when the  $\overline{DSR}$  pin is low.

The FE bit (framing error) becomes 1 when less than one stop bit is detected at the end of each data block during asynchronous receiving. Figure 8 shows how a framing error can happen.

Figure 6. Command Byte Format



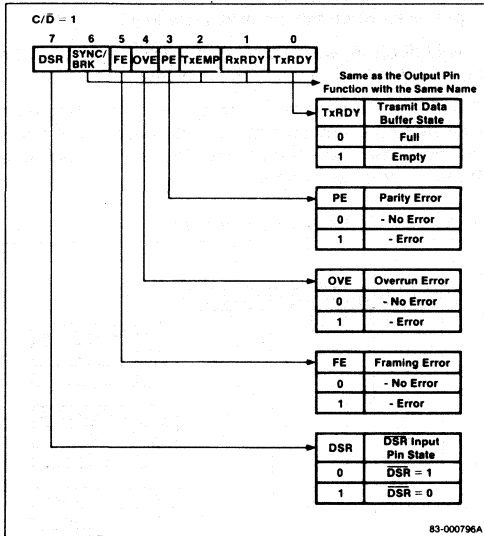
6



The OVE bit (overrun error) becomes 1 when the CPU delays reading the received data and two new data bytes have been received. In this case, the first data byte received is overwritten and lost in the receive data buffer. Figure 9 shows how an overrun can happen.

The PE bit (parity error) becomes 1 when a parity error occurs in a receive state.

Figure 7. Status Register Format



Framing, overrun, and parity errors do not disable the μPD71051's operations. All three error flags are cleared to 0 by a command byte that sets the ECL bit to 1.

The TxRDY bit becomes 1 when the transmit data buffer is empty. The TxRDY output pin becomes 1 when the transmit data buffer is empty, the CTS pin is low, and TxEN = 1. That is, bit TxRDY = Transmit Data Buffer Empty) • (CTS = 0) • (TxEN = 1).

Figure 8. Framing Error

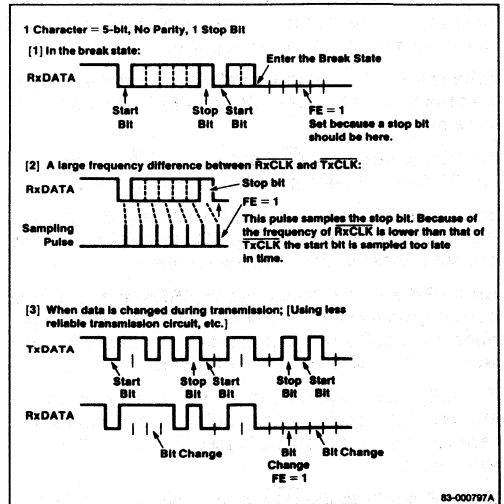
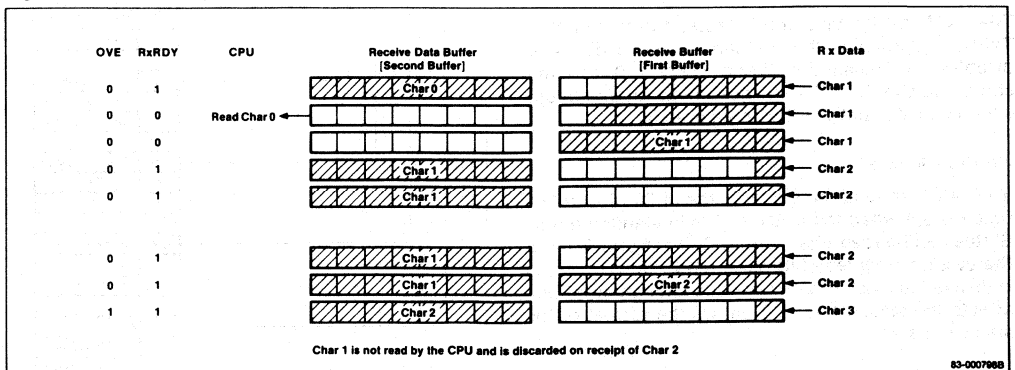


Figure 9. Overrun Error

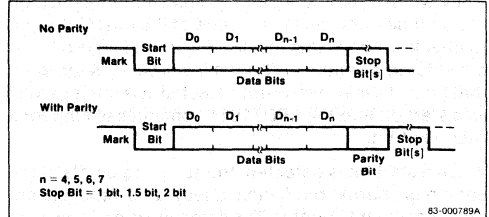


## Sending in Asynchronous Mode

The TxDATA pin is typically in the high state (marking) when data is not being sent. When the CPU writes transmit data to the μPD71051, the μPD71051 transfers the transmit data from the transmit data buffer to the send buffer and sends the data from the TxDATA pin after adding one start bit (low level) and a programmed stop bit. If parity is used, a parity bit is inserted between the character and the stop bit. Figure 10 shows the data format for async mode characters. Serial data is sent by the falling edge of the signal that divided TxCLK (1/1, 1/16, or 1/64).

When bit SBRK is set to 1, the TxDATA pin goes low (break status), regardless of whether data is being sent. Figure 11 is a fragment of a typical program to send data in the async mode. Figure 12 shows the output from pin TxDATA.

**Figure 10. Asynchronous Mode Data Format**

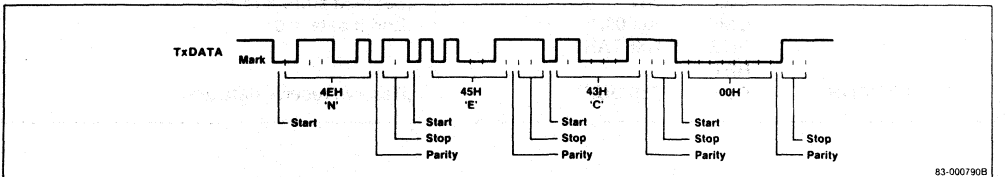


**Figure 11. Asynchronous Transmitter Example**

```

ASYNTAX :   CALL    ASYNMOD           ;Set async mode
            MOV     AL,00010001B      ;Command: clear error flag, transmit enable
            OUT    PCTRL,AL
            MOV    BW,OFFSET TXDADR   ;Transmit data area
TXSTART :   IN     AL,PCTRL
            TEST1  AL,0                ;Read status
            BNE   TXSTART             ;Wait until TxRDY = 1
            MOV    AL,[BW]            ;Write transmit data
            OUT    PDATA,AL
            INC   BW
            CMP   AL,00H              ;Set next data address
            BNE   TXSTART             ;End if data = 0
            RET
TXDADR     DB    'NEC'                ;Transmit data 4EH, 45H, 43H, 00
            DB    0
ASYNMOD :   MOV    AL,0                ;Writes control bytes three times
            OUT    PCTRL,AL           ;with 00H to unconditionally
            OUT    PCTRL,AL           ;accept the new command byte
            OUT    PCTRL,AL
            MOV    AL,01000000B      ;Software reset
            OUT    PCTRL,AL
            MOV    AL,11111010B     ;Write mode byte
            OUT    PCTRL,AL           ;Stop bit = 2 bits, even parity
            RET                       ;7 bits/character, x16 clock
    
```

**Figure 12. TxDATA Pin Output**



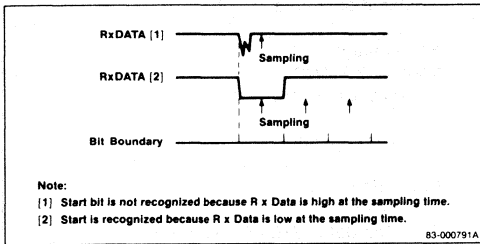
**Receiving in Asynchronous Mode**

The RxDATA pin is normally in the high state when data is not being received, as shown in figure 13. The μPD71051 detects the falling edge of a low level signal when a low level signal enters it.

The μPD71051 samples the level of the RxDATA input (only when x16 or x64 clock is selected) in a position 1/2 bit time after the falling edge of the RxDATA input to check whether this low level is a valid start bit. It is considered a valid start bit if a low level is detected at that time. If a low level is not detected, it is not regarded as a start bit and the μPD71051 continues testing for a valid start bit.

When a start bit is detected, the sampling points of the data bits, parity bit (when used), and stop bit are decided by a bit counter. The sampling is performed by the rising edge of the RxCLK when an X1 clock is used. When a x16 or x64 clock is used, it is sampled at the nominal middle of RxCLK.

**Figure 13. Start Bit Detection**



Data for one character entering the receive buffer is transferred to the receive data buffer and causes RxRDY = 1, requesting that the CPU read the data. When the CPU reads the data, RxRDY becomes 0.

When a valid stop bit is detected, the μPD71051 waits for the start bit of the next data. If a low level is detected in the stop bit, a framing error flag is set; however, the receiving operation continues as if the correct high level had been detected. A parity error flag is set if a parity error is detected. An overrun error flag is set when the CPU does not read the data in time, and the next receiving data is transferred to the receive data buffer, overwriting the unread data. The μPD71051's sending and receiving operations are not affected by these errors.

If a low level is input to the RxDATA pin for more than two data blocks during a receive operation, the μPD71051 considers it a break state and the SYNC/BRK pin status becomes 1.

In async mode, the start bit is not detected until a high level of more than one bit is input to the RxDATA pin and the receiver is enabled. Figure 14 is a fragment of a typical program to receive the data sent in the previous async transmit example.

**Figure 14. Asynchronous Receiver Example**

```

ASYNRX :   CALL   ASYNMOD           ;Set ASYNC mode
           MOV    AL, 00010100B     ;Command: clear error flag, receive
                                           ;enable

           OUT   PCTRL,AL
           MOV   BW, OFFSET RXDADR   ;Data store area
RXSTART :   IN    AL, PCTRL
           TEST1 AL, 1               ;Read status
           BNE   RXSTART             ;Wait until RxRDY = 1
           IN    AL, PDATA           ;Read and store the receive data
           MOV   [BW], AL
           INC   BW                   ;Set next store address
           CMP   AL, 00H             ;End if data = 0
           BNE   RXSTART
           RET
RXDADR     DB    256 DUP             ;Reserve receive data area
    
```

## Sending in Synchronous Mode

Following the establishment of sync mode and the enabling of the transmitter, the TxDATA pin stays high until the CPU writes the first character (normally, SYNC characters). When data is written, the TxDATA pin sends one bit for each falling edge of TxCLK if the CTS pin is low. Unlike async mode, start and stop bits are not used. However, a parity bit may be set. Figure 15 shows these data formats.

Once sending begins, the CPU must write data to the μPD71051 at the same rate as that of TxCLK. If TxEMP goes to 1 because of a delay in writing by the CPU, the μPD71051 sends SYNC characters until the CPU writes data. TxEMP goes to 0 when data is written, and the data is sent as soon as transmission of SYNC characters stops.

Figure 15. Synchronous Mode Data Format

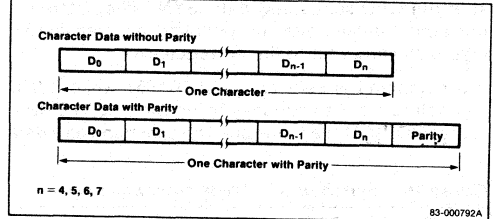


Figure 16. Synchronous Mode Transmit Timing

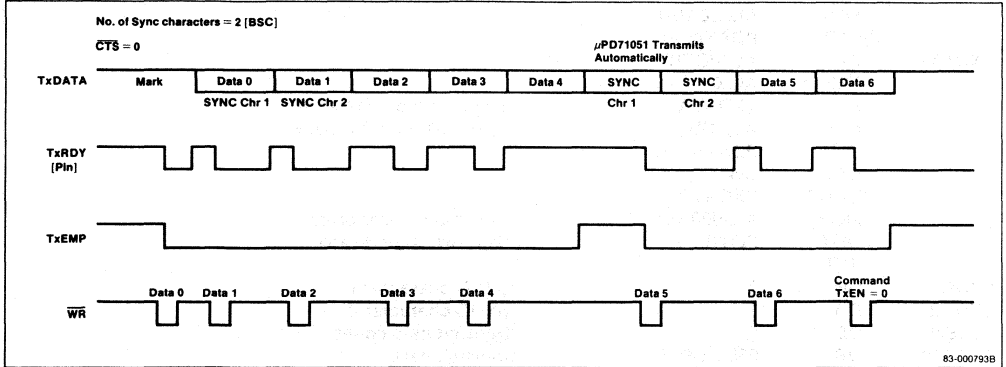
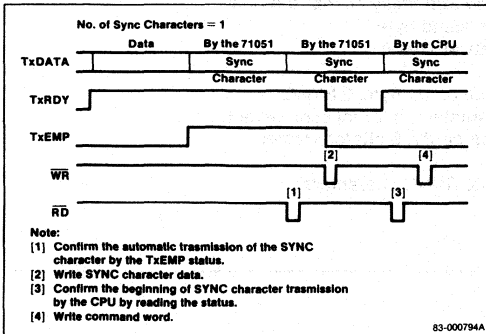


Figure 17. Issuing a Command During SYNC Character Transmission



Automatic transmission of SYNC characters begins after the CPU sends new data. SYNC characters are not automatically sent by enabling the transmitter. Figure 16 shows these timing sequences.

If a command is sent to the μPD71051 while SYNC characters are automatically being sent and TxEMP = 1, the μPD71051 may interpret the command as a data

byte and transmit it as data. If a command must be sent under these conditions, the CPU should send a SYNC character to the μPD71051 and send the command while the SYNC character is being transmitted. This is shown in figure 17.

Figure 18 is a fragment of a typical program for sending in sync mode.

Figure 18. Synchronous Transmitter Example

```

SYNTAX :   CALL    SYNMOD                ;Set sync mode
           MOV     AL, 00010001B         ;Command; clear error
           OUT     PCTRL, AL             ;flags, transmit enable
           MOV     BW, OFFSET TXDADR     ;Start location of data area TxDADR
           MOV     CL, LDLEN             ;Set number of bytes (LDLEN) to be transmitted
           MOV     CH, 00H
TXLEN :    IN      AL, PCTRL              ;Transmit the length byte
           TEST1   AL, 0
           BZ      TXLEN
           MOV     AL, LDLEN
           OUT     PDATA, AL
TXDATA :   IN      AL, PCTRL
           TEST1   AL, 0
           BZ      TXDATA                ;Transmit the number of
           MOV     AL, (BW)              ;bytes specified by LDLEN
           OUT     PDATA, AL
           INC     BW
           DBNZ    TXDATA
           MOV     AL, 00010000B         ;Command; clear error
           OUT     PCTRL, AL             ;flags, transmit disable
           RET
SYNC1     DB      ?                      ;SYNC character 1
SYNC2     DB      ?                      ;SYNC character 2
LDLEN     DB      ?                      ;transmit data count
TXDADR    DB      255 DUP (?)           ;transmit data
SYNMOD :   MOV     AL, 00H
           OUT     PCTRL, AL             ;Write control bytes
           OUT     PCTRL, AL             ;three times with 00H to
           OUT     PCTRL, AL             ;unconditionally accept the new
           ;command byte
           MOV     AL, 01000000B         ;Software reset
           OUT     PCTRL, AL
           MOV     AL, 00111100B         ;Write mode byte: 2 SYNC
           OUT     PCTRL, AL             ;characters, internal sync detect,
           ;even parity, 8 bits/character
           MOV     AL, SYNC1
           OUT     PCTRL, AL             ;Write SYNC characters
           MOV     AL, SYNC2
           OUT     PCTRL, AL
           RET

```

## Receiving in Synchronous Mode

In order to receive in sync mode, synchronization must be established with the sending side. The first command after setting sync mode and writing the SYNC character must be EH = 1, ECL = 1, and RxEN = 1. When hunt phase is entered all the bits in the receive buffer are set to 1. In internal synchronization, data on the RxDATA pin is input to the receive buffer for each rising edge of RxCLK and is compared with the SYNC character at the same time. Figure 19 shows this internal sync detection.

When the receive buffer and the SYNC character coincide, and parity is not used, the μPD71051 ends hunt phase and SYNC is set to 1 in the center of the last SYNC bit. When parity is used, SYNC becomes 1 in the center of the parity bit. Receiving starts with the bit which follows the bit when SYNC is set to 1.

In external sync detection, synchronization is achieved by setting the SYNC pin high from an external circuit for at least one period of RxCLK. Hunt phase ends, and data reception can start. At this time, the SYNC status

bit becomes 1, and goes to 0 when the status is read. The SYNC status bit is set to 1 when the SYNC input has a rising edge followed by a high level of more than one period of RxCLK, even after synchronization is achieved.

The μPD71051 can regain lost synchronization anytime by issuing an enter hunt phase command.

After synchronization, the SYNC character is compared with each character regardless of whether internal or external synchronization is used. When the characters coincide, SYNC becomes 1, indicating that a SYNC character has been received. SYNC (SYNC status bit only in external detection) becomes 0 when the status is read.

Overflow and parity errors are checked the same way as in async mode, affecting only the status flag. Parity checking is not performed in the hunt phase. Figure 20 is a fragment of a typical program that receives the data sent by the previous sync transmit program example. Note that the frequencies of TxCLK on the transmitter and RxCLK on the receiver must be the same.

**Figure 19. Internal Sync Detection Example**

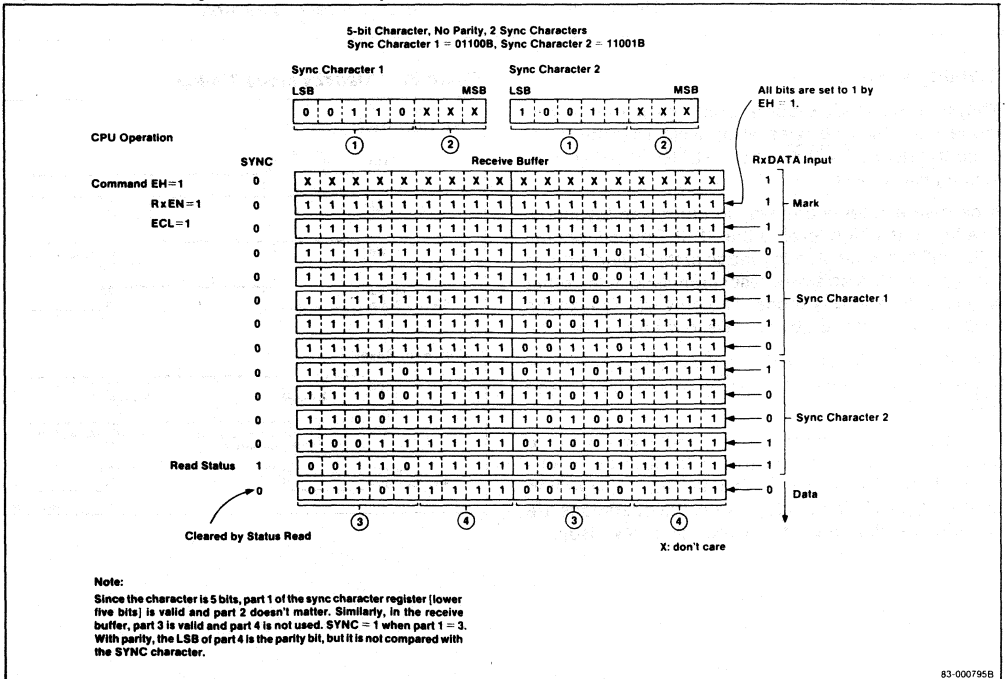


Figure 20. Synchronous Receiver Example

```

SYNRX :   CALL     SYNMOD           ;Set sync mode
          MOV      AL, 10010100B    ;Command: enter hunt
          OUT      PCTRL, AL        ;phase, clear error flags, receive enable
          MOV      BW, OFFSET RXDADR ;Set receive data store address
RXLEN :   IN       AL, PCTRL
          TEST1    AL, 1
          BZ       RXLEN           ;Receive the number of
          IN       AL, DATA        ;receive data
          MOV      STLEN, AL        ;Set the number of
          MOV      CL, AL           ;receive data to both variable and
                                   ;counter
          MOV      CH, 00H
RXDATA :  IN       AL, PCTRL
          TEST1    AL, 1
          BZ       RXDATA         ;Receive and store the
          IN       AL, PDATA        ;number of data bytes
          MOV      [BW],AL         ;stated by the counter
          INC      BW
          DBNZ     RXDATA
          MOV      AL, 00000000B    ;Command: receive disable
          OUT      PCTRL, AL
          RET
STLEN    DB      ?                ;Set number of receiver data
RXDADR   DB      256 DUP (0)      ;Reserve receive data area
    
```

**Standby Mode**

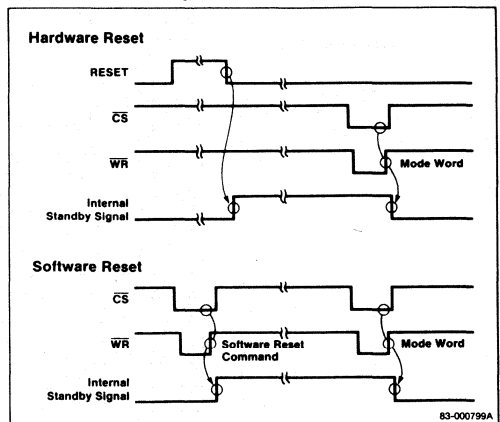
The μPD71051 is a low-power CMOS device. In standby mode, it disables the external input clocks to the inside circuitry (CLK, TxCLK, and RxCLK), thereby consuming less power.

A hardware reset is one way to enter standby mode. The input of a high level to the RESET pin causes the μPD71051 to enter standby mode at the falling edge of the high level. A software reset command is the other way to enter standby mode. The only way to take the μPD71051 out of standby mode is to write a mode byte.

In standby mode, the TxRDY, RxRDY, and SYNC/BRK pins are at low level and the TxDATA, DTS, TxEMP, and RTS pins are at high level.

Figure 21 shows the timing for standby mode. While the internal standby signal is high, the external clocks to the μPD71051 are ignored. If data (C/D = 0) is written to the μPD71051 in standby mode, the operations are undefined and unpredictable operation may result.

Figure 21. Standby Mode Timing



## Description

The  $\mu$ PD71054 is a high-performance, programmable counter for microcomputer system timing control. Three 16-bit counters, each with its own clock input, gate input, and OUT pin, can be clocked from DC to 8 MHz. Under software control, the  $\mu$ PD71054 can generate accurate time delays. Initialize the counter, and the  $\mu$ PD71054 counts the delay, and interrupts the CPU when the task is complete. This eliminates the need for software timing loops.

The  $\mu$ PD71054 contains three counters capable of binary or BCD operation. There are six programmable count modes. The counters operate independently and each can be set to a different mode. Use address lines A<sub>1</sub>, A<sub>0</sub> to select a counter and perform a read/write operation.

## Features

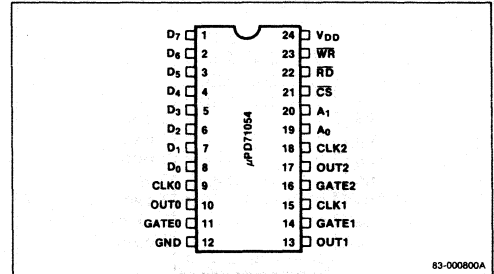
- Three independent 16-bit counters
- Six programmable counter modes
- Binary or BCD count
- Multiple latch command
- Clock rate DC (standby mode) to 8 MHz ( $\mu$ PD71054)  
(10 MHz for  $\mu$ PD71054-10)
- Low-power standby mode
- CMOS technology
- Single power supply, 5 V  $\pm$ 10%
- Industrial temperature range -40 to +85°C

## Ordering Information

Part Number	Package Type	Max. Frequency of Operation
$\mu$ PD71054C	28-pin plastic DIP	8 MHz
$\mu$ PD71054GB	44-pin plastic flat pack	8 MHz
$\mu$ PD71054L	28-pin PLCC	8 MHz
$\mu$ PD71054C-10	24-pin plastic DIP	10 MHz
$\mu$ PD71054GB-10	44-pin plastic flat pack	10 MHz
$\mu$ PD71054L-10	28-pin plastic PLCC	10 MHz

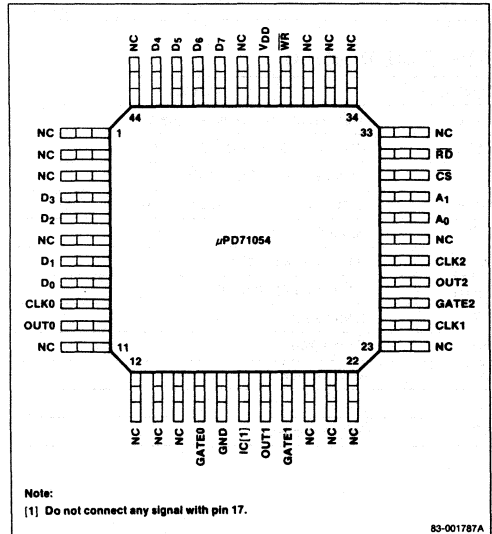
## Pin Configurations

### 24-Pin Plastic DIP



83-000800A

### 44-Pin Plastic Flat Pack



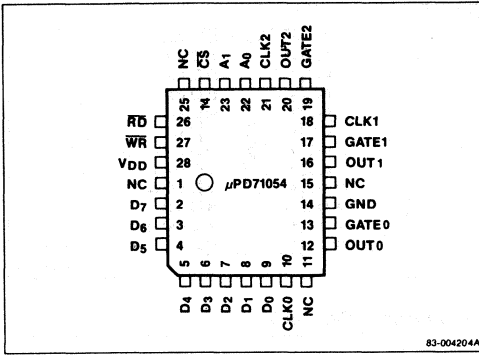
Note:  
[1] Do not connect any signal with pin 17.

83-001787A



**Pin Configurations (cont)**

**28-Pin PLCC (Plastic Leaded Chip Carrier)**



83-004204A

**Pin Identification**

Symbol	Function
D7-D0	Three-state, bidirectional data bus
CLKn	Counter n clock output (n = 0-2)
OUTn	Counter n output (n = 0-2)
GATEn	Output to inhibit or trigger counter n (n = 0-2)
GND	Ground
IC	Internally connected
A0-A1	Select counter input 0, 1, or 2
CS	Chip select
RD	Read strobe
WR	Write strobe
VDD	+5 V
NC	Not connected

**Pin Functions**

**D7-D0 [Data Bus]**

These pins are an 8-bit three-state bidirectional data bus. This bus is used to program counter modes and to read status and count values. The data bus is active when  $\overline{CS} = 0$ , and is high impedance when  $\overline{CS} = 1$ .

**CLKn [Counter Clock, n = 0-2]**

These pins are the clock input that determine the count rate for counter n. The clock rate may be DC (standby mode) to 8 MHz.

**OUTn [Counter Output, n = 0-2]**

These are the output pins for counter n. A variety of outputs is available depending on the count mode. When the μPD71054 is used as an interrupt source, these pins can output an interrupt request signal.

**GATEn [Counter Gate, n = 0-2]**

These output pins inhibit or trigger counter n according to the mode selected.

**A1, A0 [Address]**

These input pins select the counter. A1, A0 equal to 00, 01, or 10 selects counter 0, 1, or 2, respectively. The control register is selected when A1, A0 equals 11. These pins are normally connected to the address bus.

**CS [Chip Select]**

When the  $\overline{CS}$  input = 1, all the bits of the data bus become high impedance.  $\overline{CS}$  must be low to access the μPD71054.

**RD [Read Strobe]**

The  $\overline{RD}$  input must be low to read data from the μPD71054.

**WR [Write Strobe]**

The  $\overline{WR}$  input must be low to write data to the μPD71054. The contents of the data bus are written to the μPD71054 at the rising edge of  $\overline{WR}$ .

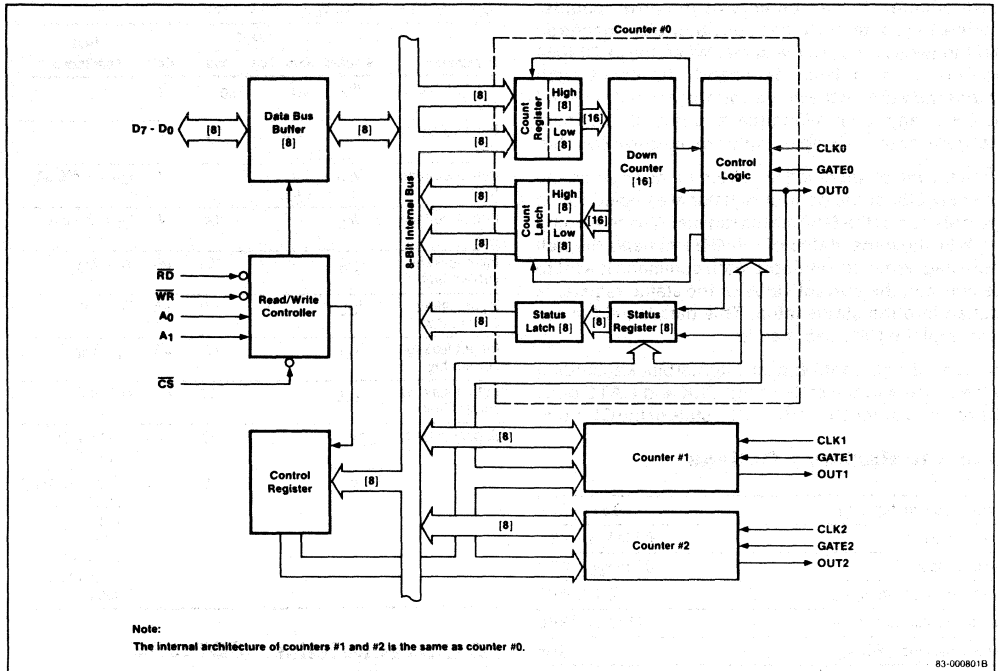
**VDD [Power]**

+5 V.

**GND [Ground]**

Ground.

## Block Diagram



## Block Functions

### Data Bus Buffer

This is an 8-bit three-state bidirectional buffer that acts as an interface between the μPD71054 and the system data bus. The data bus buffer handles control commands, the count to be written to the count register, count data read from the count latch, and status data read from the status latch.

### Read/Write Control

This circuit decodes signals from the system bus and sends control signals to other blocks of the μPD71054. A<sub>1</sub> and A<sub>0</sub> select one of the counters or the control register. A low signal on  $\overline{RD}$  or  $\overline{WR}$  selects a read or write operation. CS must be low to enable these operations.

### Control Register

This is an 8-bit register into which is written the control command that determines the operating mode of the counter. Data is written to this register when the CPU

executes an OUT command when A<sub>1</sub>, A<sub>0</sub> = 11. The contents of this register cannot be read if the CPU executes an IN command when A<sub>1</sub>, A<sub>0</sub> = 11. However, the multiple latch command allows you to read the mode and status of each counter.

### Counter n [n = 0-2]

A 16-bit synchronous down counter performs the actual count operation within the counter. You can preset this counter and select binary or BCD operation.

The count register is a 16-bit register that stores the count when it is first written to the counter. The count is transferred to the down counter and a count operation for a specified number of counts begins.

The 8-bit width of the internal data bus permits the transfer of only eight bits at a time when the count is written to the count register. However, when data is written from the count register to the down counter, all 16 bits can be written at once. When the count is written to the count register while the counter is in read/write one byte mode, a 00H is written to the remaining byte of the register.

The count latch normally holds the current value of the down counter. If the contents of the down counter change, the contents of the count latch also change so that the two values are the same. When the μPD71054 receives a count latch command, the count latch latches the value of the down counter and holds it until the CPU can read it. When the data is read, the count latch returns to tracking the value of the down counter.

When the mode specified is written to the counter, the lower six bits of the control register are copied to the lower six bits of the 8-bit status register. The remaining two bits show the status of the OUT pin and the null count flag. When the multiple latch command is sent to the counter, the current value of the status register is latched into the status latch. This data is held in the latch until the CPU can read it.

The control logic controls each internal block according to the mode and the state of the CLK and GATE pins. The result is output to and sets the state of the OUT pin.

**Absolute Maximum Ratings**

$T_A = +25^\circ\text{C}$

Power supply voltage, $V_{DD}$	-0.5 to +7.0 V
Input voltage, $V_I$	-0.5 to $V_{DD} + 0.3$ V
Output voltage, $V_O$	-0.5 to $V_{DD} + 0.3$ V
Operating temperature, $T_{OPT}$	-40°C to 85°C
Storage temperature, $T_{STG}$	-65°C to +150°C
Power dissipation, $P_{DMAX}$	1.0 W

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Capacitance**

$T_A = +25^\circ\text{C}, V_{DD} = 0\text{ V}$

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input capacitance	$C_{IN}$		10	pF	$f_c = 1\text{ MHz}$
I/O capacitance	$C_{I/O}$		20	pF	Unmeasured pins returned to 0 V

**DC Characteristics**

$T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ ,  $V_{DD} = +5\text{ V} \pm 10\%$

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input voltage high	$V_{IH}$	2.2		V	$V_{DD} + 0.3$
Input voltage low	$V_{IL}$	-0.5	0.8	V	
Output voltage high	$V_{OH}$	0.7 $\times V_{DD}$		V	$I_{OH} = -400\ \mu\text{A}$
Output voltage low	$V_{OL}$		0.4	V	$I_{OL} = 2.5\ \text{mA}$
Input leakage current high	$I_{LIH}$		10	$\mu\text{A}$	$V_I = V_{DD}$
Input leakage current low	$I_{LIL}$		-10	$\mu\text{A}$	$V_I = 0\text{ V}$
Output leakage current high	$I_{LOH}$		10	$\mu\text{A}$	$V_O = V_{DD}$
Output leakage current low	$I_{LOL}$		-10	$\mu\text{A}$	$V_O = 0\text{ V}$
Supply current	$I_{DD1}$	10	20	mA	8/10 MHz operation
	$I_{DD2}$	2	50	$\mu\text{A}$	Stand-by mode Input pins: $V_{IH} = V_{DD}$ $-0.1\text{ V}$ $V_{IL} = 0.1\text{ V}$ Output pins: open

**AC Characteristics, μPD71054**

$T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ ,  $V_{DD} = 5\text{ V} \pm 10\%$

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Read Cycle</b>					
Address set-up to $\overline{\text{RD}} \downarrow$	$t_{SAR}$	30		ns	
Address hold from $\overline{\text{RD}} \uparrow$	$t_{HRA}$	10		ns	
$\overline{\text{CS}}$ set-up to $\overline{\text{RD}} \downarrow$	$t_{SCR}$	0		ns	
$\overline{\text{RD}}$ low level width	$t_{RRL}$	150		ns	
Data delay from $\overline{\text{RD}} \downarrow$	$t_{DRD}$		120	ns	$C_L = 150\ \text{pF}$
Data float from $\overline{\text{RD}} \uparrow$	$t_{FRD}$	10	85	ns	$C_L = 20\ \text{pF}$ $R_L = 2\ \text{k}\Omega$
Data delay from address	$t_{DAD}$		220	ns	$C_L = 150\ \text{pF}$
Read recovery time	$t_{RV}$	200		ns	

## AC Characteristics, μPD71054 (cont)

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = 5 V ±10%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Write Cycle</b>					
Address set-up to WR ↓	t <sub>SAW</sub>	0		ns	
Address hold from WR ↑	t <sub>HWA</sub>	0		ns	
CS set-up to WR ↓	t <sub>SCW</sub>	0		ns	
WR low level width	t <sub>WWL</sub>	160		ns	
Data hold to WR ↑	t <sub>SDW</sub>	120		ns	
Data hold from WR ↑	t <sub>HWD</sub>	0		ns	
Write recovery time	t <sub>RV</sub>	200		ns	
<b>CLK and Gate Timing</b>					
CLK cycle time	t <sub>CYK</sub>	125	DC	ns	
CLK high level width	t <sub>KKH</sub>	60		ns	
CLK low level width	t <sub>KKL</sub>	60		ns	
CLK rise time	t <sub>KR</sub>		25	ns	
CLK fall time	t <sub>KF</sub>		25	ns	
GATE high level width	t <sub>GGH</sub>	50		ns	
GATE low level width	t <sub>GGL</sub>	50		ns	
GATE set-up to CLK ↑	t <sub>SGK</sub>	50		ns	
GATE hold from CLK ↑	t <sub>HKG</sub>	50		ns	
Clock delay from WR ↑ (count transfer)	t <sub>DWK</sub>	100		ns	t <sub>KKH</sub> ≥ 125 ns
		225 - t <sub>KKH</sub>		ns	t <sub>KKH</sub> < 125 ns
Clock set-up to WR ↑ (latch)	t <sub>SKW</sub>	85		ns	
GATE delay from WR ↑	t <sub>DWG</sub>	0		ns	
OUT delay from GATE ↓	t <sub>DGO</sub>		120	ns	C <sub>L</sub> = 150 pF
OUT delay from CLK ↓	t <sub>DKO</sub>		150	ns	C <sub>L</sub> = 150 pF
OUT delay from WR ↑ (initial out)	t <sub>DWO</sub>		295	ns	C <sub>L</sub> = 150 pF

**Note:**

AC timing test points for output V<sub>OH</sub> = 2.2 V, V<sub>OL</sub> = 0.8 V

## AC Characteristics, μPD71054-10

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = 5 V ±10%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Read Cycle</b>					
Address set-up to RD ↓	t <sub>SAR</sub>	20		ns	
Address hold from RD ↑	t <sub>HRA</sub>	0		ns	
CS set-up to RD ↓	t <sub>SCR</sub>	0		ns	
RD low level width	t <sub>RRL</sub>	95		ns	
Data delay from RD ↓	t <sub>DRD</sub>		85	ns	C <sub>L</sub> = 150 pF
Data float from RD ↑	t <sub>FRD</sub>	10	65	ns	C <sub>L</sub> = 20 pF R <sub>L</sub> = 2 kΩ
Data delay from address	t <sub>DAD</sub>		185	ns	C <sub>L</sub> = 150 pF
Read recovery time	t <sub>RV</sub>	165		ns	
<b>Write Cycle</b>					
Address set-up to WR ↓	t <sub>SAW</sub>	0		ns	
Address hold from WR ↑	t <sub>HWA</sub>	0		ns	
CS set-up to WR ↓	t <sub>SCW</sub>	0		ns	
WR low level width	t <sub>WWL</sub>	95		ns	
Data hold to WR ↑	t <sub>SDW</sub>	95		ns	
Data hold from WR ↑	t <sub>HWD</sub>	0		ns	
Write recovery time	t <sub>RV</sub>	165		ns	

**AC Characteristics, μPD71054-10 (cont)**

$T_A = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{DD} = 5\text{ V} \pm 10\%$

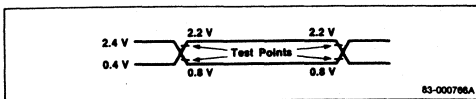
Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>CLK and Gate Timing</b>					
CLK cycle time	$t_{CYK}$	100	DC	ns	
CLK high level width	$t_{KKH}$	30		ns	
CLK low level width	$t_{KKL}$	45		ns	
CLK rise time	$t_{KR}$		25	ns	
CLK fall time	$t_{KF}$		25	ns	
GATE high level width	$t_{GGH}$	50		ns	
GATE low level width	$t_{GGL}$	50		ns	
GATE set-up to CLK ↑	$t_{SGK}$	40		ns	
GATE hold from CLK ↑	$t_{HKG}$	50		ns	
Clock delay from WR ↑ (count transfer)	$t_{DWK}$	40		ns	
Clock set-up to WR ↑ (latch)	$t_{SKW}$	60		ns	
GATE delay from WR ↑	$t_{DWG}$	0		ns	
OUT delay from GATE ↓	$t_{DGO}$		100	ns	$C_L = 150\text{ pF}$
OUT delay from CLK ↓	$t_{DKO}$		100	ns	$C_L = 150\text{ pF}$
OUT delay from WR ↑ (initial out)	$t_{DWO}$		240	ns	$C_L = 150\text{ pF}$

**Note:**

AC timing test points for output  $V_{OH} = 2.2\text{ V}$ ,  $V_{OL} = 0.8\text{ V}$

**Timing Waveforms**

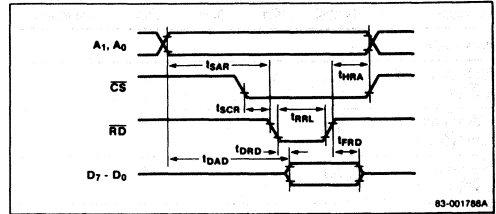
**AC Test Input**



83-000768A

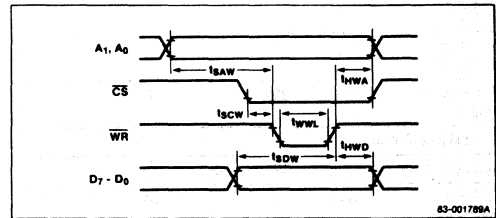
**Timing Waveforms (cont)**

**Read Cycle**



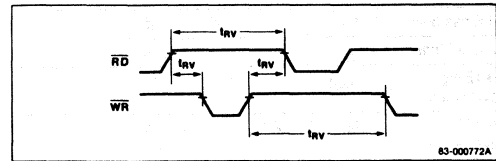
83-001786A

**Write Cycle**



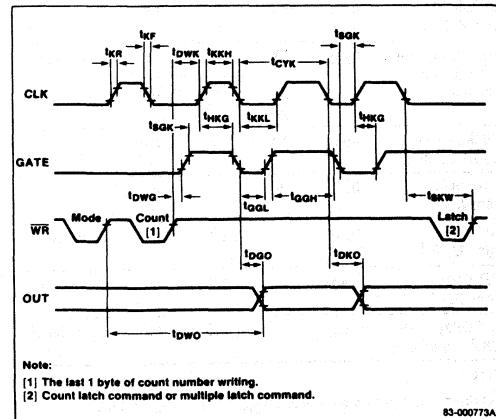
83-000772A

**Read/Write Recovery**



83-000772A

**CLK and GATE Timing**



**Note:**

- [1] The last 1 byte of count number writing.
- [2] Count latch command or multiple latch command.

83-000772A

## Functional Description

### μPD71054 System Configuration Example

The CPU views the three counters and the control register as four I/O ports. A<sub>1</sub> and A<sub>0</sub> are connected to the A<sub>1</sub> and A<sub>0</sub> pins of the system address bus.  $\overline{CS}$  is generated by decoding the address and  $\overline{IO}/MEM$  signals so that  $\overline{CS}$  goes low when the address bus is set to the target I/O address and I/O is selected. These connections are shown in figure 1.

You can use the μPD71054 in memory-mapped I/O configurations. However, the decoding should be such that  $\overline{CS}$  goes low when memory is selected.

### Programming and Reading the Counter

The counter must be programmed and the operating mode specified before you can use the μPD71054. Once a mode has been selected for a counter, it operates in that mode until another mode is set. The count is written to the count register and when that data is transferred to the down counter, a new count operation begins. The current count and status can be read while the counter is in operation. Figure 2 outlines the steps of operation.

### Programming the Counter

The μPD71054 is controlled by a microcomputer program. The program must write a control command to set the counter mode and write the count data that determines the length of the count operation. Table 1 shows the values for A<sub>1</sub> and A<sub>0</sub> that determine the target counter for write operations.

**Table 1. Write Operations ( $\overline{CS} = 0, \overline{RD} = 1, \overline{WR} = 0$ )**

A <sub>1</sub>	A <sub>0</sub>	Write Target
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control word register

### Control and Mode Setting

The control command must be written to set the counter mode before operating the counter. If a write operation is performed when A<sub>1</sub>, A<sub>0</sub> = 11, a control command is written to the control register. Figure 3 shows the format of the 8-bit control command.

Bits SC1 and SC0 specify a counter or the multiple latch command. When a counter is chosen, the specifications described below apply to the counter.

Bits RMW1 and RMW2 specify the read/write operation to the counter or select the count latch command.

Bits CM2, CM1, and CM0 set the counter mode (0 to 5).

Bit BCD selects binary or BCD operation. The count may be 0 to FFFFH in binary mode or 0 to 9999 in BCD.

If a control command written to the counter specifies a mode, the lower six bits of the control command are copied to the lower six bits of the status register of the counter selected by SC1 and SC0. The mode selected remains in effect until a new mode is set. This is not true if the control command specifies the count latch or multiple latch command.

### Writing the Count

The count is written to the counter after the mode is set. Set A<sub>1</sub>, A<sub>0</sub> to specify the target counter as shown in table 1. A new count can be written to a counter at any time, but the read/write mode selected (when the mode was written) must be used when writing the count.

In high 1-byte and low 1-byte modes only, the higher or lower byte of the count register is written by the first write. The write operation ends and 00H is automatically written to the remaining byte by the μPD71054. In the 2-byte modes, the lower byte is written by the first write and the higher byte by the second.

For example, if the 2-byte count 8801H is written to a counter set in lower 1-byte mode, the lower byte (01H) is written first, followed by the higher byte (88H). Therefore, the data written to the count register is 0001H for the first write and 0088H for the second. This is shown in Table 2.

**Table 2. Read/Write Mode and Count Write**

Read/Write Mode	No. of Writes	Count Register	
		Higher Byte	Lower Byte
Low 1-byte	1	00H	nnH
High 1-byte	1	nnH	00H
Low/High 2-byte	2	nnH (2nd write)	nnH (1st write)

nnH = Two-digit hexadecimal value

### Reading the Counter

The following three methods allow you to read the contents of the down counter during operation. In particular, the multiple latch command reads the current count data and the counter mode or the state of the OUT pin. Table 3 shows the values of A<sub>1</sub>, A<sub>0</sub> used to select the counter to be read.

Figure 1. Typical System Configuration

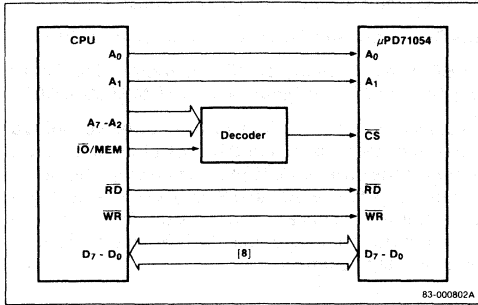


Figure 2. Basic Operating Procedure

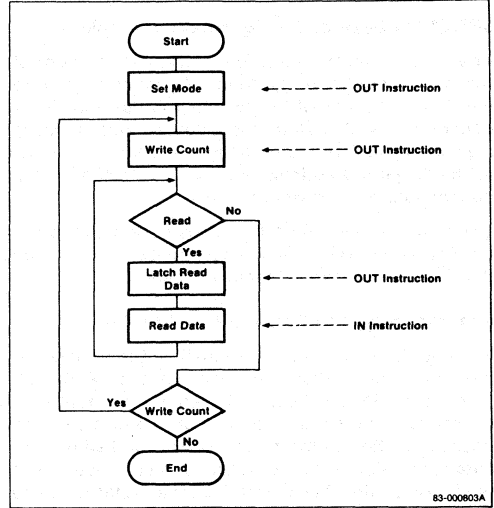
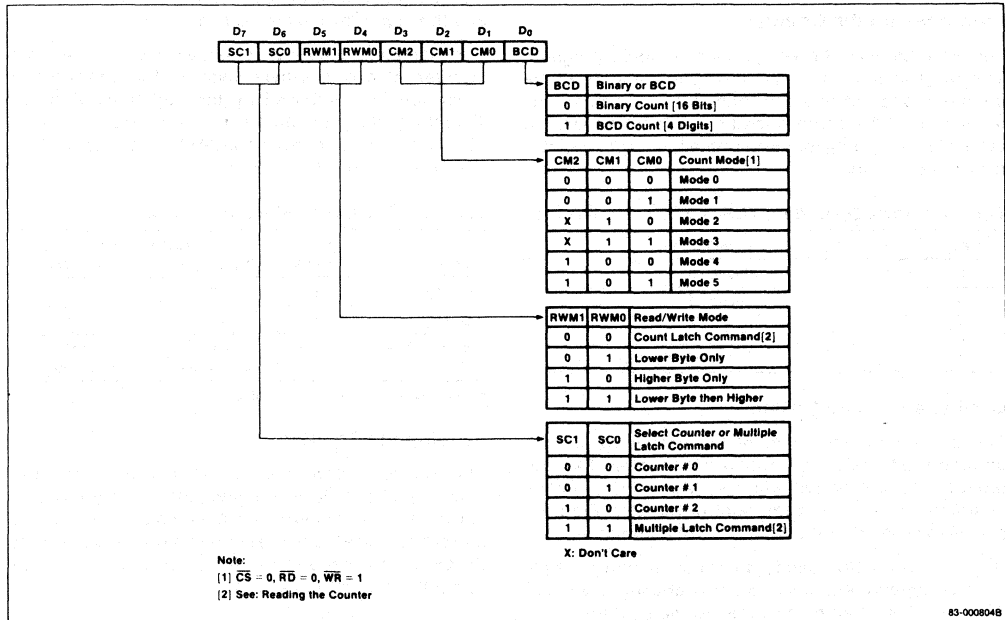


Figure 3. Control Register Format



**Table 3. Read Operations ( $\overline{CS} = 0, \overline{RD} = 0, \overline{WR} = 1$ )**

A <sub>1</sub>	A <sub>0</sub>	Read Target
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2

### Directly Reading the Counter

You can read the current value of the counter by reading the counter selected by A<sub>1</sub>, A<sub>0</sub> as shown in table 3. This involves reading the count latch; since the value of the down counter may change while the the count latch is read, this method may not provide an accurate reading. You must control the CLK or GATE input to stop the counter and read it for a correct reading.

### Using the Count Latch Command

When the count latch command is executed, the current counter value is latched into the counter latch. This value is held by the latch until it is read or until a new mode is set. This provides an accurate reading of the counter value when the command is executed without affecting counter operation. Figure 4 shows the format for the count latch command.

If the counter value that was latched into the count latch is not read before a second count latch command is executed, the second command is ignored. This is because the counter value latched by the first command is held until it is read or until a new mode is set. When the data in the count latch is read, the latch is released and continues tracking the value of the down counter.

### Using the Multiple Latch Command

When the multiple latch command is received, the counter value and status register for any counter may be selectively latched into the count latch and status latch. Bits D<sub>1</sub>-D<sub>5</sub> of the multiple latch command specify the counter latching. The CPU can then read the status and counter value for the selected counter. Figure 5 shows the format for this command.

Bits CNT2, CNT1, and CNT0 correspond to counters 2, 1, and 0. The command is executed for all counters whose corresponding bit is 1. This allows the data for more than one counter to be latched by a single count latch command.

When the count bit is 0, the counter value of the selected counters is latched into the count latches.

When the status bit is 0, the status of the selected counters is latched into the status latches. Bits D<sub>5</sub>-D<sub>0</sub> of the status register show the mode status of the counter. The output bit (D<sub>7</sub>) shows the state of the OUT pin of that counter. These bits are shown in figure 6. The null count bit (D<sub>6</sub>) indicates whether the count data is valid. When the count is transferred from the count register to the down counter, this bit changes to 0 to show that the data is valid. Table 4 shows how the null count flag operates.

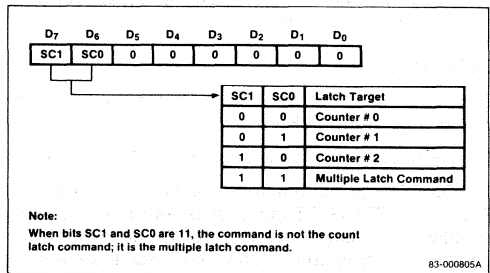
**Table 4. Null Count Flag Operation**

Operation	Null Count Flag
Write control word for mode set	1
Write count to count register(1)	1
Transfer count from count register to down counter	0

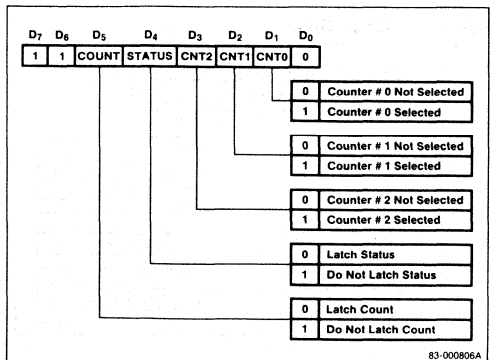
**Note:**

(1) When 2-byte mode is selected, the flag becomes 1 when the second byte is written.

**Figure 4. Control Register Format for Count Latch Command**



**Figure 5. Control Register Format for Multiple Latch Command**





If the data that was latched is not read before a second multiple latch command is executed, the second command is ignored for those latches whose contents have not been read. This is because the data latched by the first command is held until it is read or until a new mode is set. When the data in the latch is read, the latch is released. See figure 7.

It is possible to latch both the count and status using two multiple latch commands. However, regardless of which data is latched first, the status is always read first. The count data is read by the next read operation (1- or 2-step read as determined by read/write mode). If additional read commands are received, the count data that has not been latched (the contents of the down counter as reflected by the current counter value) is read.

Read operations must be performed in accordance with read/write mode. In 2-byte mode, two bytes of data must always be read. This does not imply that the second byte must be read immediately after the first; other counter operations may be performed between the two reads. For example, you could read the lower byte, write a new lower byte, read the higher byte, and write a new higher byte.

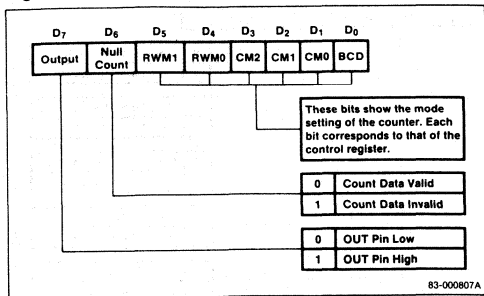
**Definitions**

CLK pulse refers to the time from the rising to the falling edge of the CLK<sub>n</sub> input.

Trigger refers to the rising edge of the GATEn input.

The GATEn input is sampled at each rising edge of the CLK<sub>n</sub> input. The GATE input can be level or rising edge sensitive. In the latter case, counter n's internal flip-flop is set at the rising edge of the GATE signal, sensed at the rising edge of the next CLK pulse, and reset immediately. This allows edge-triggering to be sensed whenever it occurs.

**Figure 6. Status Data**



Initial OUT refers to the state of the OUT pin immediately after the mode is set.

Count transfer refers to the transfer from the count register to the down counter. The down counter is decremented at the falling edge of the CLK pulse.

Count zero is the state of the down counter when the counter is decremented to zero.

PCNT0, PCNT1, and PCNT2 are the I/O ports for counters 0, 1, and 2, respectively. PCTRL is the I/O port for the control command.

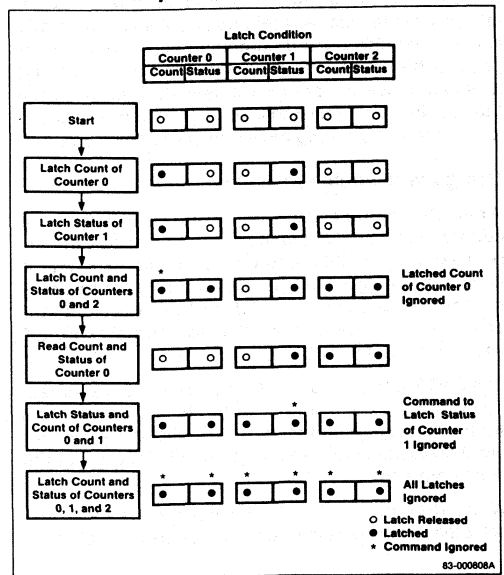
CW is the control command.

HB is the higher byte of the count.

LB is the lower byte of the count.

In the timing charts for each counter mode, counter 0 is in the read/write 1-byte and binary count mode. When no GATE signal appears in the charts, assume a high level signal. The value shown below the OUT signal is the counter value. The maximum value that can be set for the count in each mode is 0. When this value is set, a maximum value of 10000H (hexadecimal count) or 10000 (BCD count) is obtained.

**Figure 7. Multiple Latch Command Execution Example**



## Counter Modes

**Mode 0: Interrupt on End of Count.** In this mode, the OUT output changes from low to high level when the end of the specified count is reached. See table 5 and figure 8.

**Table 5. Mode 0 Operation**

Function	Result
Initial OUT	Low level
GATE High	Count enable
GATE Low	Count disable
Count Write	The OUT pin goes low independent of the CLK pulse. In 2-byte mode, the count is disabled when the first byte is written. The OUT pin goes low. OUT goes low when a new mode or new count is written.
Count Transfer and Operation	When the count is written with GATE high: Transfer is performed at the first CLK pulse after the count value is written. The down counter is decremented beginning at the first CLK pulse after data transfer. If a count of n is set, the OUT pin goes high after n + 1 CLK pulses. When the count is written with GATE low: Transfer is performed at the first CLK pulse after the count is written. The down counter is decremented beginning at the first CLK pulse after the GATE signal goes high. If a count of n is set, OUT is low for a period of n CLK pulses after GATE goes high.
Count Zero	The signal at the OUT pin goes high. The count operation does not stop and counts down to FFFFH (hexadecimal) or 9999 (BCD) and continues to count down.
Minimum Count	1

**Mode 0 Program Example.** This subroutine causes a delay of 10004 (decimal, or 2710H) CLK pulses. In this program, counter 2 is set to 2-byte mode and binary count. See figure 9.

SUBR0:	MOV	AL,10110000B	;set mode: counter 2, ;2-byte mode,
	OUT	PCTRL,AL	;count mode 0, binary
	MOV	AL,10H	
	OUT	PCNT2,AL	
	MOV	AL,27H	;write count 10000 (decimal)
	OUT	PCNT2,AL	
	RET		

**Mode 1: GATE Retriggerable One-Shot.** In mode 1, the μPD71054 functions as a retriggerable one-shot. A low-level pulse triggered by the GATE input is output from the OUT pin. See table 6 and figure 10.

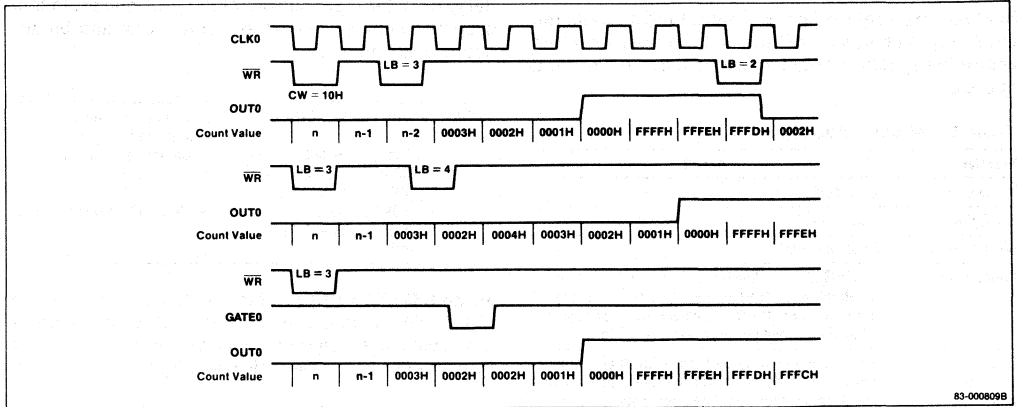
**Table 6. Mode 1 Operation**

Function	Result
Initial OUT	High level
GATE Trigger(1)	Count data is transferred at the CLK pulse after the trigger.
Count Write	The count is written without affecting the current operation.
Count Transfer and Operation	Transfer is performed at the first CLK pulse after the trigger. At the same time, the signal at the OUT pin goes low to start the one-shot pulse operation. The count is decremented beginning at the next CLK pulse. If a count of n is set, the one-shot output from the OUT pin continues for n CLK pulses.
Count Zero	The signal at the OUT pin becomes high. Count operation does not stop and wraps to FFFFH (hexadecimal) or 9999 (BCD) and continues to count.
Minimum Count	1

**Note:**

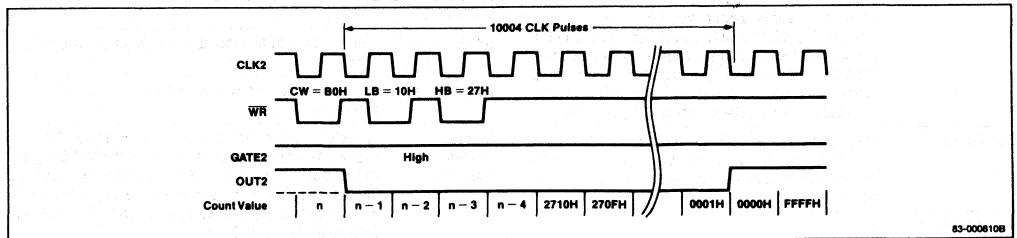
(1) The trigger is ignored when the count has not been written after the mode is set, or when only one byte of the count has been written in 2-byte count mode.

Figure 8. Mode 0 Timing Chart



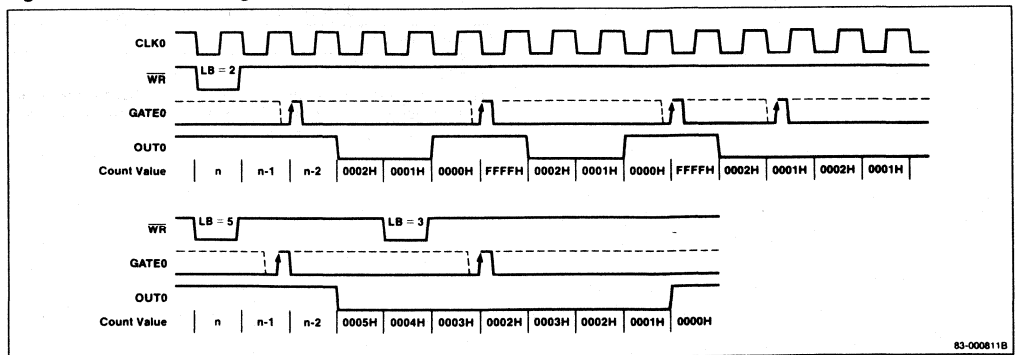
83-000809B

Figure 9. Mode 0 Program Example Timing Chart



83-000810B

Figure 10. Mode 1 Timing Chart



83-000811B

**Mode 1 Program Example.** This subroutine waits until no trigger is generated for an interval of 200 or more CLK pulses after the first gate trigger and returns to the main program. Counter 1 is set to low-byte read/write mode and binary count. See figure 11.

```

SUBR1:  MOV  AL,01010010B ;set mode: counter 1, low-byte
        OUT  PCTRL,AL    ;read/write mode, count mode 1,
        MOV  AL,200      ;binary
        OUT  PCNT1,AL    ;write low byte of count
        ;
FSTTRG: MOV  AL,11100100B ;multiple latch command:
        OUT  PCNT1,AL    ;counter 1,
        IN   AL,PCNT1    ;status
        TEST1 AL,7        ;wait for first trigger
        BNZ  FSTTRG
        ;
WAIT:   MOV  AL,11100100B ;multiple latch command:
        OUT  PCTRL,AL    ;counter 1,
        IN   AL,PCNT1    ;status
        TEST1 AL,7        ;wait until output goes high
        BZ   WAIT
        RET
    
```

**Table 7. Mode 2 Operation**

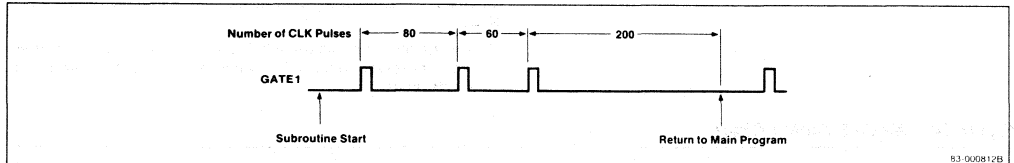
Function	Result
Initial OUT	High level
GATE High	Count enable
GATE Low	Count disabled. If GATE goes low when OUT is low, OUT will go high (independent of the CLK pulse).
GATE Trigger(1)	Transfer is performed at the first CLK pulse after the trigger.
Count Write	Count is written without affecting the current operation.
Count Transfer and Operation	Transfer is performed at the CLK pulse after the count is written following the mode setting. The counter is then decremented. Transfer is again performed at the first CLK pulse after the count becomes 1. When the trigger is used, transfer is performed at the next CLK pulse. When the contents of the down counter becomes 1, OUT goes low for one CLK pulse and returns to high. If a count of n is set, OUT repeats this sequence every n CLK pulses.
Count Zero	Never occurs in this mode.
Minimum Count	2

**Note:**

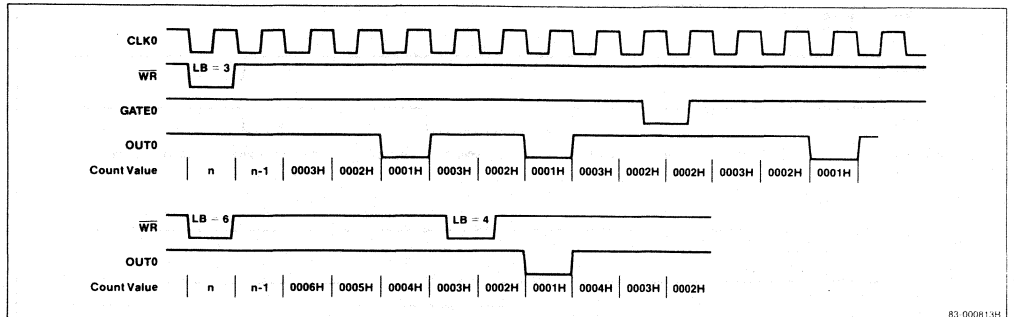
(1) The trigger is ignored when the count has not been written or when only one byte of the count has been written in 2-byte mode.

**Mode 2: Rate Generator.** In mode 2, the signal from the OUT pin cyclically goes low for one clock period when the counter reaches 0001H. The counter operates as a frequency divider. See table 7 and figure 12.

**Figure 11. Mode 1 Program Example Timing Chart**



**Figure 12. Mode 2 Operation Timing Chart**



**Mode 2 Program Example.** This subroutine generates an interrupt to the CPU each time 10000 (decimal) clock pulses elapse. Counter 0 is in 2-byte mode and binary counting. See figure 13.

```

SUBR3:  MOV  AL,00110100B    ;mode setting: counter 0, 2-byte
        OUT  PCTRL,AL       ;mode, count mode 2, binary
        MOV  AL,10H         ;mode, count mode 2, binary
        OUT  PCNT0,AL
        MOV  AL,27H         ;write count 10000 (decimal)
        OUT  PCNT0,AL
        RET
    
```

**Mode 3: Square Wave Generator.** Mode 3 is a frequency divider similar to mode 2, but with a different duty cycle. See table 8 and figure 14.

Figure 13. Mode 2 Configuration

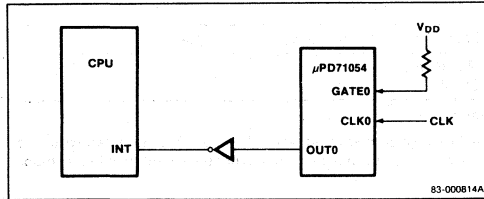


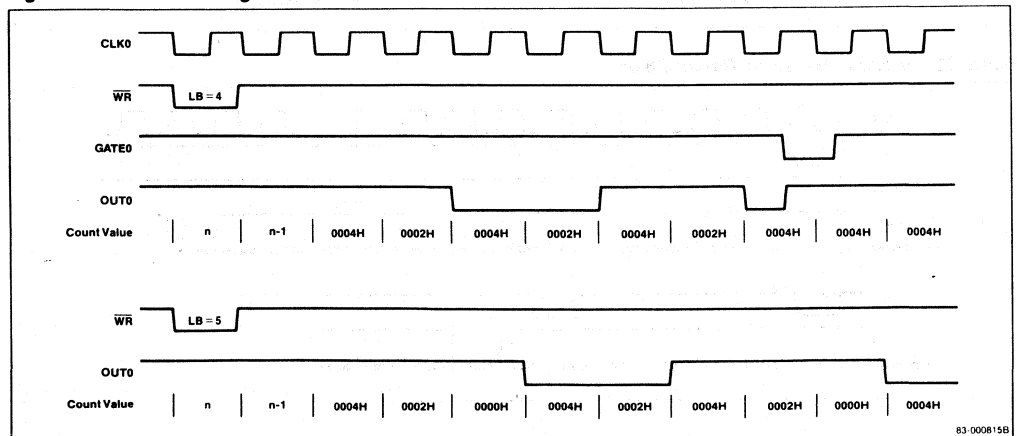
Table 8. Mode 3 Operation

Function	Result
Initial OUT	High level
GATE High	Count enable
GATE Low	Count disable. If GATE goes low when OUT is low, OUT will go high (independent of the CLK pulse).
GATE Trigger(1)	Transfer is performed at the first CLK pulse after the trigger.
Count Write	Current operation is not affected. The count is transferred at the end of the half-period of the current square wave and the OUT pin goes high.
Count Transfer and Operation	Count data is transferred at the first CLK pulse after the count write following the mode setting. Transfer is performed at the end of the current half-cycle and the OUT pin is inverted. Transfer is also performed at the CLK pulse after the trigger. The operation performed depends on whether count n is even or odd. When n is even, the count is decremented by two on each following clock pulse. At the end of the count of two, the count is again transferred and the OUT pin is inverted. This is taken as a half-cycle and repeated. When n is odd, n - 1 is transferred and the count is decremented by two on each following clock pulse. The half-cycle when the OUT pin is high continues until the end of count 0 and n - 1 is transferred again at the next CLK pulse. The half-cycle while OUT is low continues until the end of count 2. Thus, the half-cycle while OUT is high is one CLK longer than the half-cycle while OUT is low.
Count Zero	Occurs only when the count is odd.
Minimum Count	3

**Note:**

- (1) The trigger is ignored when the count has not been written after the mode is set or when only one byte of count has been written in 2-byte mode.

Figure 14. Mode 3 Timing Chart

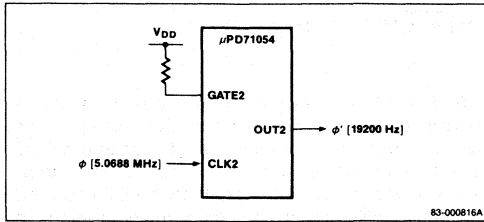


**Mode 3 Program Example.** This subroutine divides the input CLK frequency (5.0688 MHz) by 264 to get a 19,200 Hz clock. Counter 2 is in 2-byte binary mode. See figure 15.

```

SUBR4:  MOV    AL,10110110B    ;mode setting: counter 2, 2-byte
        OUT    PCTRL,AL      ;mode, count mode 3, binary
        MOV    AL,08H
        OUT    PCNT2,AL
        MOV    A,01H         ;264 frequency division
        OUT    PCNT2,AL
        RET
    
```

**Figure 15. Frequency Division**



**Mode 4: Software-Triggered Strobe.** In mode 4, when the specified count is reached, OUT goes low for one CLK pulse. See table 9 and figure 16.

**Table 9. Mode 4 Operation**

Function	Result
Initial OUT	High level
GATE High	Count enable
GATE Low	Count disable
Count Write	Count is transferred at the next CLK pulse when the count is written. In 2-byte mode, data is transferred after the second byte is written.
Count Transfer and Operation	Count is transferred at the first CLK following the count write. If GATE is high, the down counter begins to decrement from the next CLK. If GATE is low, decrement begins at the first CLK after GATE goes high.
Count Zero	OUT is low for one CLK pulse and returns to high. The down counter wraps to FFFFH (hexadecimal) or 9999 (BCD) without stopping counter operation.
Minimum Count	1

**Figure 16. Mode 4 Timing Chart**

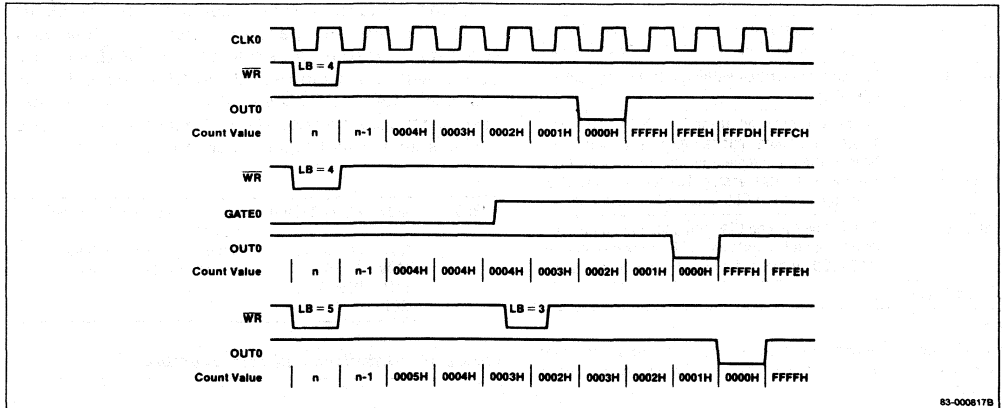
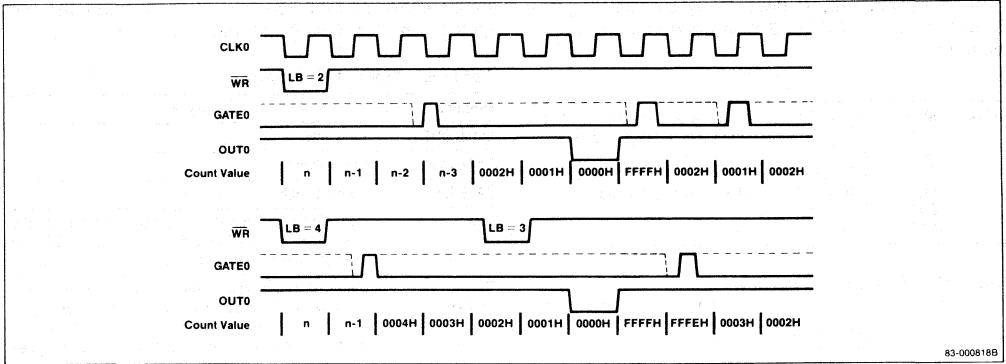


Figure 17. Mode 5 Timing Chart



**Mode 5: Hardware-Triggered Strobe [Retriggerable].** Mode 5 is similar to mode 4 except that operation is triggered by the GATE input and can be retriggered. See table 10 and figure 17.

Table 10. Mode 5 Operation

Function	Result
Initial OUT	High level
GATE Trigger(1)	The count is transferred at the CLK pulse after the trigger. The GATE has no effect on the OUT signal.
Count Write	The count is written without affecting the current operation.
Count Transfer and Operation	Count is transferred at the first CLK pulse after a trigger, providing that the mode and count have been written. Decrement begins from the first CLK pulse after a data transfer. If a count of n is set, OUT goes low for n + 1 CLK pulses after the trigger.
Count Zero	OUT is low for one CLK and goes high again. The down counter counts to FFFFH (hexadecimal) or 9999 (BCD) without stopping the counter operation.
Minimum Count	1

**Note:**

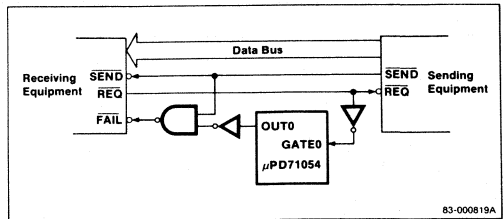
- (1) The trigger is ignored when the count has not been written after the mode is set or when only one byte has been written in 2-byte mode.

**Mode 5 Program Example.** Use mode 5 to add a fail-safe function to an interface. For example, the receiving equipment requests data by issuing a REQ signal to the sending equipment. The sending equipment responds by outputting data to the data bus and returning a SEND signal to the receiving equipment. In this type of system, if a malfunction exists in the sending equipment and no SEND signal is sent, the receiving equipment waits indefinitely for the SEND signal and system operation stops. The following subroutine remedies this situation. If no SEND signal is output within a given period (50 CLK cycles in this example) after the REQ signal is output, the system assumes the sending equipment is malfunctioning and a FAIL signal is sent to the receiving equipment.

```

SUBR5:  MOV  AL,00011010B  ;mode setting: counter 0, low
        ;1-byte
        OUT  PCTRL,AL    ;mode, count mode 5, binary
        MOV  AL,50      ;set interval: 50 CLK pulses
        OUT  PCNT0,AL
        RET
    
```

Figure 18. Interface Fail-safe Example



## Description

The μPD71055 is a low-power CMOS programmable parallel interface unit for use in microcomputer systems. The μPD71055 has three I/O ports and is typically used to interface peripheral devices to a microcomputer system bus.

## Features

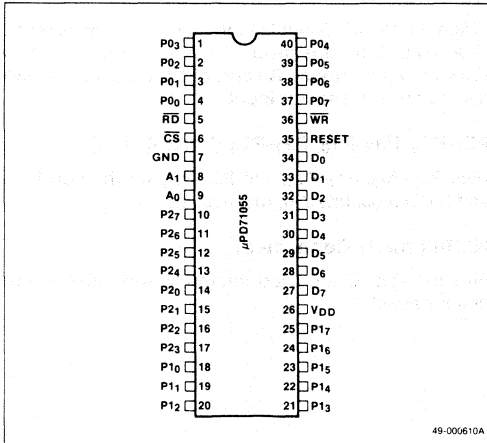
- Three 8-bit I/O ports
- Three programmable operation modes
- Bit manipulation command
- Microcomputer compatible
- 8 MHz operation
- CMOS technology
- Single +5 V ±10% power supply
- Industrial temperature range: -40 to +85°C

## Ordering Information

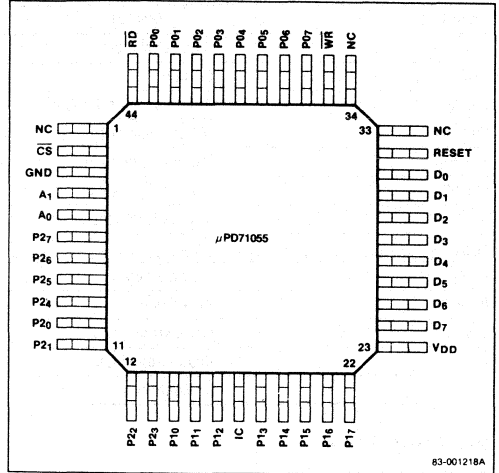
Part Number	Package Type	Max. Frequency of Operation
μPD71055C	40-pin plastic DIP	8 MHz
μPD71055GB	44-pin plastic flat pack	8 MHz
μPD71055L	44-pin PLCC	8 MHz
μPD71055C-10	40-pin plastic DIP	10 MHz
μPD71055GB-10	44-pin plastic flat pack	10 MHz
μPD71055L-10	44-pin plastic PLCC	10 MHz

## Pin Configuration

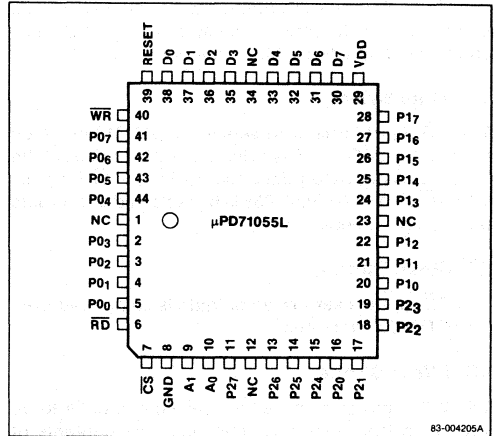
### 40-Pin Plastic DIP



### 44-Pin Plastic Flat Pack



### 44-Pin Plastic Leaded Chip Carrier (PLCC)





**Pin Identification**

Symbol	Function
$\overline{CS}$	Chip select input
GND	Ground
A <sub>1</sub> , A <sub>0</sub>	Address inputs 1 and 0
P0 <sub>7</sub> -P0 <sub>0</sub>	I/O port 0, bits 7-0
P1 <sub>7</sub> -P1 <sub>0</sub>	I/O port 1, bits 7-0
P2 <sub>7</sub> -P2 <sub>0</sub>	I/O port 2, bits 7-0
IC	Internally connected
V <sub>DD</sub>	+5 V
D <sub>7</sub> -D <sub>0</sub>	I/O data bus
RESET	Reset input
$\overline{WR}$	Write strobe input
$\overline{RD}$	Read strobe input
NC	No connection

**Pin Functions**

**D<sub>7</sub>-D<sub>0</sub> [Data Bus]**

D<sub>7</sub>-D<sub>0</sub> make up an 8-bit, three-state, bidirectional data bus. The bus is connected to the system data bus. It is used to send commands to the μPD71055 and to send data to and from the μPD71055.

**$\overline{CS}$  [Chip Select]**

The  $\overline{CS}$  input is used to select the μPD71055. When  $\overline{CS} = 0$ , the μPD71055 is selected and the states of the D<sub>7</sub>-D<sub>0</sub> pins are determined by the  $\overline{RD}$  and  $\overline{WR}$  inputs. When  $\overline{CS} = 1$ , the μPD71055 is not selected and its data bus is high-impedance.

**$\overline{RD}$  [Read Strobe]**

The  $\overline{RD}$  input is set low when data is being read from the μPD71055 data bus.

**$\overline{WR}$  [Write Strobe]**

The  $\overline{WR}$  input should be set low when data is to be written to the μPD71055 data bus. The contents of the data bus are written to the μPD71055 at the rising edge (low to high) of the  $\overline{WR}$  signal.

**A<sub>1</sub>, A<sub>0</sub> [Address]**

The A<sub>1</sub> and A<sub>0</sub> inputs are used in combination with the  $\overline{RD}$  and  $\overline{WR}$  signals to select one of the three ports or the command register. A<sub>1</sub> and A<sub>0</sub> are usually connected to the lower two bits of the system address bus (table 1).

**$\overline{WR}$  [Write Strobe]**

The  $\overline{WR}$  input should be set low when data is to be written to the μPD71055 data bus. The contents of the data bus are written to the μPD71055 at the rising edge (low to high) of the  $\overline{WR}$  signal.

**A<sub>1</sub>, A<sub>0</sub> [Address]**

The A<sub>1</sub> and A<sub>0</sub> inputs are used in combination with the  $\overline{RD}$  and  $\overline{WR}$  signals to select one of the three ports or the command register. A<sub>1</sub> and A<sub>0</sub> are usually connected to the lower two bits of the system address bus (table 1).

**Table 1. Control Signals and Operation**

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A <sub>1</sub>	A <sub>0</sub>	Operation	μPD71055 Operation
0	0	1	0	0	Port 0 to data bus	Input
0	0	1	0	1	Port 1 to data bus	Input
0	0	1	1	0	Port 2 to data bus	Input
0	0	1	1	1	Use prohibited	
0	0	0	x	x		
0	1	0	0	0	Data bus to port 0	Output
0	1	0	0	1	Data bus to port 1	Output
0	1	0	1	0	Data bus to port 2	Output
0	1	0	1	1	Data bus to command register	Output
0	1	1	x	x	Data bus high impedance	
1	x	x	x	x		

**RESET [Reset]**

When the RESET input is high, the μPD71055 is reset. The group 0 and the group 1 ports are set to mode 0 (basic I/O port mode). All port bits are cleared to zero and all ports are set for input.

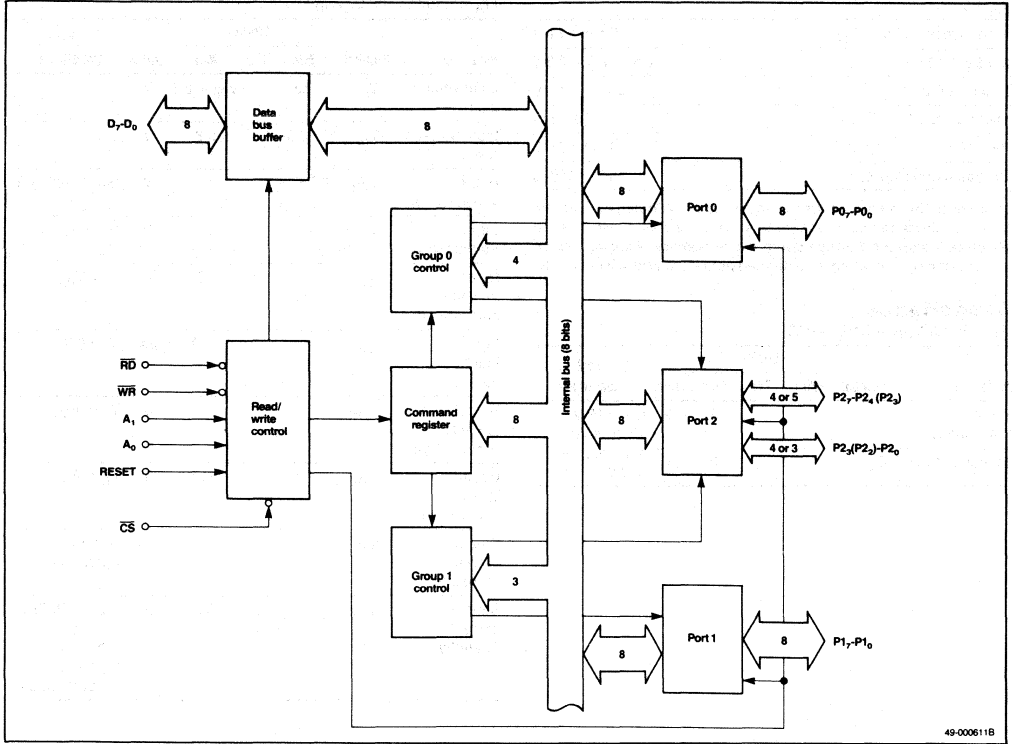
**P0<sub>7</sub>-P0<sub>0</sub>, P1<sub>7</sub>-P1<sub>0</sub>, P2<sub>7</sub>-P2<sub>0</sub> [Ports 0, 1, 2]**

Pins P0<sub>7</sub>-P0<sub>0</sub>, P1<sub>7</sub>-P1<sub>0</sub>, and P2<sub>7</sub>-P2<sub>0</sub> are the port 0, 1, and 2 I/O pins, bits 7-0, respectively.

**IC [Internally Connected]**

Pins marked IC are used internally and must be left unconnected.

## Block Diagram



## Functional Description

### Ports 0, 1, 2

The μPD71055 has three 8-bit I/O ports, referred to as port 0, port 1, and port 2. These ports are divided into two groups, group 0 and group 1. The groups can be in one of three modes, mode 0, mode 1, and mode 2. Modes can be set independently for each group.

When port 0 is in mode 0, port 0 and the four upper bits of port 2 belong to group 0, and port 1 and the four lower bits of port 2 belong to group 1. When port 0 is in mode 1 or 2, port 0 and the 5 upper bits of port 2 belong to group 0 and port 1 and the three lower bits of port 2 belong to group 1.

### Command Register

The host writes command words to the μPD71055 in this register. These commands control group 0 and group 1. Note that the contents of this register cannot be read.

### Group 0 Control and Group 1 Control

These blocks control the operation of group 0 and group 1.

### Read/Write Control

The read/write control controls the read/write operations for the ports and the data bus in response to the  $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{CS}$ , and address signals. It also handles RESET signals and the  $A_0$ ,  $A_1$  address inputs.

### Data Bus Buffer

The data bus buffer latches information going to or from the system data bus.

### Absolute Maximum Ratings

(T<sub>A</sub> = 25 °C)

Power supply voltage, V <sub>DD</sub>	-0.5 to +7.0 V
Input voltage, V <sub>I</sub>	-0.5 to V <sub>DD</sub> + 0.3 V
Output voltage, V <sub>O</sub>	-0.5 to V <sub>DD</sub> + 0.3 V
Power dissipation, P <sub>D</sub> MAX	500 mW
Operating temperature, T <sub>opt</sub>	-40 to +85 °C
Storage temperature, T <sub>stg</sub>	-65 to +150 °C

**Comment:** These devices are not meant to be operated outside the limits specified above. Exposure to stresses beyond those listed in Absolute Maximum Ratings could cause damage. Exposure to an absolute maximum rating for extended periods may affect reliability.

### Capacitance

(T<sub>A</sub> = 25 °C, V<sub>DD</sub> = GND = 0 V)

Parameter	Symbol	Limits			Units	Test Conditions
		Min	Typ	Max		
Input capacitance	C <sub>I</sub>		10		pF	f <sub>c</sub> = 1 MHz Unmeasured pins returned to 0 V
I/O capacitance	C <sub>I0</sub>		20		pF	

### DC Characteristics

(T<sub>A</sub> = -40 to +85 °C, V<sub>DD</sub> = 5 V ±10%)

Parameter	Symbol	Limits			Units	Test Conditions
		Min	Typ	Max		
Input voltage high	V <sub>IH</sub>	2.2		V <sub>DD</sub> + 0.3	V	
Input voltage low	V <sub>IL</sub>	-0.5		0.8	V	
Output voltage high	V <sub>OH</sub>	0.7 V <sub>DD</sub>			V	I <sub>OH</sub> = -400 μA
Output voltage low	V <sub>OL</sub>		0.4		V	I <sub>OL</sub> = 2.5 mA
Input leakage current high	I <sub>LIH</sub>		10		μA	V <sub>I</sub> = V <sub>DD</sub>
Input leakage current low	I <sub>LIL</sub>		-10		μA	V <sub>I</sub> = 0 V
Output leakage current high	I <sub>LOH</sub>		10		μA	V <sub>O</sub> = V <sub>DD</sub>
Output leakage current low	I <sub>LOL</sub>		-10		μA	V <sub>O</sub> = 0 V
Supply current (dynamic)	I <sub>DD1</sub>		5	10	mA	Normal operation
Supply current (standby)	I <sub>DD2</sub>		2	50	μA	Inputs: RESET = 0.1 V, others = V <sub>DD</sub> - 0.1 V Outputs: Open

## AC Characteristics, μPD71055

(T<sub>A</sub> = -40 to +85°C, V<sub>DD</sub> = 5 V ±10%)

Parameter	Symbol	Limits			Units	Test Conditions
		Min	Typ	Max		
<b>Read Timing</b>						
A <sub>1</sub> , A <sub>0</sub> , $\overline{CS}$ set-up to RD ↓	t <sub>SAR</sub>	0			ns	
A <sub>1</sub> , A <sub>0</sub> , $\overline{CS}$ hold from RD ↑	t <sub>HRA</sub>	0			ns	
$\overline{RD}$ pulse width	t <sub>RRL</sub>	160			ns	
Data delay from RD ↓	t <sub>DRD</sub>			120	ns	C <sub>L</sub> = 150 pF
Data float from $\overline{RD}$ ↑	t <sub>FRD</sub>	10		85	ns	C <sub>L</sub> = 20 pF R <sub>L</sub> = 2 kΩ
Read recovery time	t <sub>RV</sub>	200			ns	
<b>Write Timing</b>						
A <sub>1</sub> , A <sub>0</sub> , $\overline{CS}$ set-up to WR ↓	t <sub>SAW</sub>	0			ns	
A <sub>1</sub> , A <sub>0</sub> , $\overline{CS}$ hold from WR ↑	t <sub>HWA</sub>	0			ns	
WR pulse width	t <sub>WWL</sub>	120			ns	
Data set-up to $\overline{WR}$ ↑	t <sub>SDW</sub>	100			ns	
Data hold from $\overline{WR}$ ↑	t <sub>HWD</sub>	0			ns	
Write recovery time	t <sub>RV</sub>	200			ns	
<b>Other Timing</b>						
Port set-up time to RD ↓	t <sub>SPR</sub>	0			ns	
Port hold time from RD ↑	t <sub>HRP</sub>	0			ns	
Port set-up time to STB ↓	t <sub>SPS</sub>	0			ns	
Port hold time from STB ↑	t <sub>HSP</sub>	150			ns	

## AC Characteristics, μPD71055 (cont)

(T<sub>A</sub> = -40 to +85°C, V<sub>DD</sub> = 5 V ±10%)

Parameter	Symbol	Limits			Units	Test Conditions
		Min	Typ	Max		
<b>Other Timing (cont)</b>						
Port delay time from WR ↑	t <sub>DWP</sub>			350	ns	C <sub>L</sub> = 150 pF
STB pulse width	t <sub>SSL</sub>	350			ns	
DAK pulse width	t <sub>DADAL</sub>	300			ns	
Port delay time from DAK ↓ (mode 2)	t <sub>DDAP</sub>			300	ns	C <sub>L</sub> = 150 pF
Port float time from DAK ↑ (mode 2)	t <sub>FDAP</sub>	20		250	ns	C <sub>L</sub> = 20 pF R <sub>L</sub> = 2 kΩ
$\overline{OBF}$ set delay from WR ↑	t <sub>DWOB</sub>			300	ns	C <sub>L</sub> = 150 pF
$\overline{OBF}$ clear delay from DAK ↓	t <sub>DDAOB</sub>			350	ns	
IBF set delay from STB ↓	t <sub>DSIB</sub>			300	ns	
IBF clear delay from RD ↑	t <sub>DRIB</sub>			300	ns	
INT set delay from DAK ↑	t <sub>DDAI</sub>			350	ns	
INT clear delay from WR ↓	t <sub>DWI</sub>			450	ns	
INT set delay from STB ↑	t <sub>DSI</sub>			300	ns	
INT clear delay from RD ↓	t <sub>DRI</sub>			400	ns	
RESET pulse width	t <sub>RESET1</sub>	50			μs	During or right after power-on
	t <sub>RESET2</sub>	500			ns	During operation

**AC Characteristics, μPD71055-10**

(T<sub>A</sub> = -40 to +85°C, V<sub>DD</sub> = 5 V ±10%)

Parameter	Symbol	Limits			Units	Test Conditions
		Min	Typ	Max		
<b>Read Timing</b>						
A <sub>1</sub> , A <sub>0</sub> , $\overline{CS}$ set-up to RD ↓	t <sub>SAR</sub>	0			ns	
A <sub>1</sub> , A <sub>0</sub> , $\overline{CS}$ hold from RD ↑	t <sub>HRA</sub>	0			ns	
$\overline{RD}$ pulse width	t <sub>RRL</sub>	150			ns	
Data delay from RD ↓	t <sub>DRD</sub>			100	ns	C <sub>L</sub> = 150 pF
Data float from RD ↑	t <sub>FRD</sub>	10		60	ns	C <sub>L</sub> = 20 pF R <sub>L</sub> = 2 kΩ
Read recovery time	t <sub>RV</sub>	150			ns	
<b>Write Timing</b>						
A <sub>1</sub> , A <sub>0</sub> , $\overline{CS}$ set-up to WR ↓	t <sub>SAW</sub>	0			ns	
A <sub>1</sub> , A <sub>0</sub> , $\overline{CS}$ hold from WR ↑	t <sub>HWA</sub>	0			ns	
WR pulse width	t <sub>WWL</sub>	100			ns	
Data set-up to WR ↑	t <sub>SDW</sub>	100			ns	
Data hold from WR ↑	t <sub>HWD</sub>	0			ns	
Write recovery time	t <sub>RV</sub>	150			ns	
<b>Other Timing</b>						
Port set-up time to RD ↓	t <sub>SPR</sub>	0			ns	
Port hold time from RD ↑	t <sub>HRP</sub>	0			ns	
Port set-up time to STB ↓	t <sub>SPS</sub>	0			ns	
Port hold time from STB ↑	t <sub>HSP</sub>	150			ns	

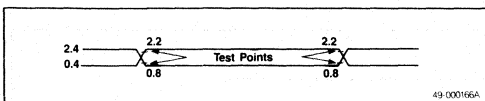
**AC Characteristics, μPD71055-10 (cont)**

(T<sub>A</sub> = -40 to +85°C, V<sub>DD</sub> = 5 V ±10%)

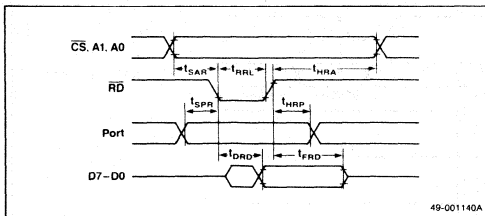
Parameter	Symbol	Limits			Units	Test Conditions
		Min	Typ	Max		
<b>Other Timing (cont)</b>						
Port delay time from WR ↑	t <sub>DWP</sub>			200	ns	C <sub>L</sub> = 150 pF
STB pulse width	t <sub>SSL</sub>	100			ns	
DAK pulse width	t <sub>DADAL</sub>	100			ns	
Port delay time from DAK ↓ (mode 2)	t <sub>DDAP</sub>			150	ns	C <sub>L</sub> = 150 pF
Port float time from DAK ↑ (mode 2)	t <sub>FDAP</sub>	20		250	ns	C <sub>L</sub> = 20 pF R <sub>L</sub> = 2 kΩ
$\overline{OBF}$ set delay from WR ↑	t <sub>DWOB</sub>			150	ns	C <sub>L</sub> = 150 pF
$\overline{OBF}$ clear delay from DAK ↓	t <sub>DDAOB</sub>			150	ns	
IBF set delay from STB ↓	t <sub>DSIB</sub>			150	ns	
IBF clear delay from RD ↑	t <sub>DRIB</sub>			150	ns	
INT set delay from DAK ↑	t <sub>DDAI</sub>			150	ns	
INT clear delay from WR ↓	t <sub>DWI</sub>			200	ns	
INT set delay from STB ↑	t <sub>DSI</sub>			150	ns	
INT clear delay from RD ↓	t <sub>DRI</sub>			200	ns	
RESET pulse width	t <sub>RESET1</sub>	50			μs	During or right after power-on
	t <sub>RESET2</sub>	500			ns	During operation

## Timing Waveforms

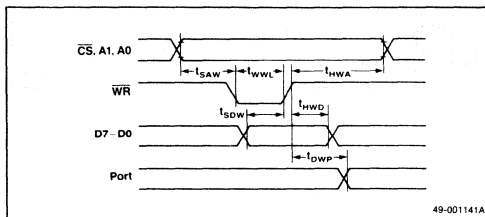
### AC Test Waveform



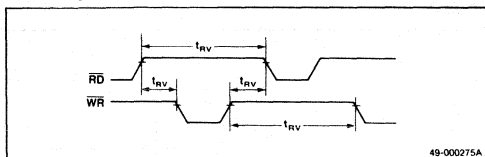
### Timing Mode 0: Input



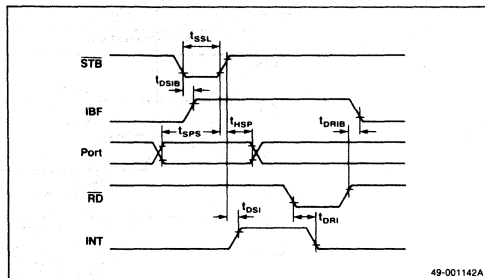
### Mode 0: Output



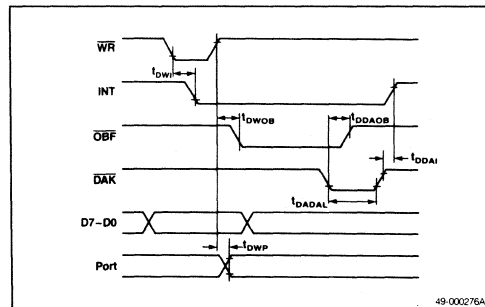
### Recovery Time



### Mode 1: Input

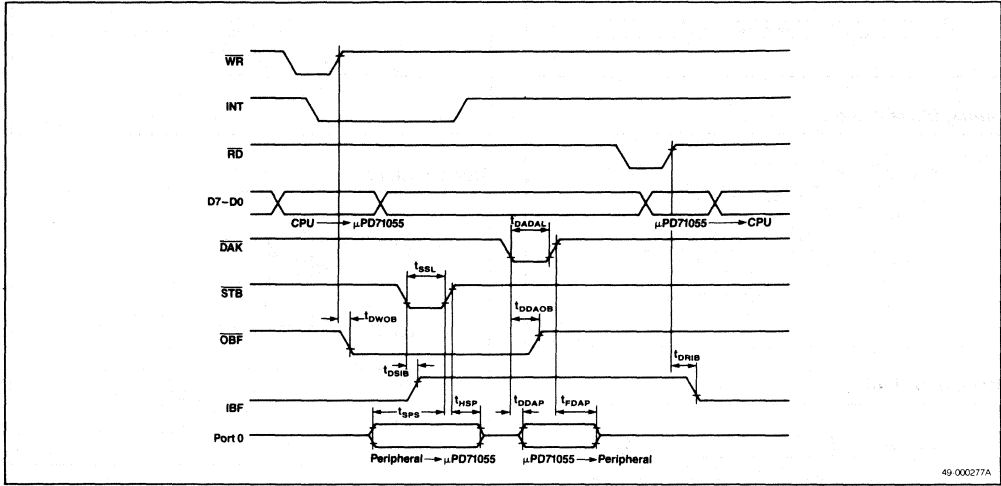


### Mode 1: Output



Timing Waveforms (cont)

Mode 2



49 000277A

## μPD71055 Commands

Two commands control μPD71055 operation. The mode select command determines the operation of group 0 and group 1 ports. The bit manipulation command sets or resets the bits of port 2. These commands are executed by writing an 8-bit command word to the command register ( $A_1A_0 = 11$ ).

### Mode Select

The μPD71055 port groups have three modes. Modes 0 and 1 can be specified for groups 0 and 1, but mode 2 can only be specified for group 0. The bits of all ports are cleared when a mode is selected or when the μPD71055 is reset.

**Mode 0.** Basic input/output port operation.

**Mode 1.** Strobed input/output operation controlled by three or four bits of port 2 used as control/status signals.

**Mode 2.** (Only available for group 0). Port 0 is the bidirectional I/O port and the higher 5 bits of port 2 are used for status and control signals.

To specify the mode, set the command word as shown in figure 1 and write it to the command register.

### Bit Manipulation Command

This command (figure 2) affects only port 2. It is mainly used in mode 1 and mode 2 to control the port 2 bits which are used as control/status signals. It is also used to enable and disable μPD71055-generated interrupts and to set and reset port 2 general input/output pins.

For example, to set bit 2 of port 2 to 1 ( $P_{22} = 1$ ), set the command word as shown in figure 3 (05H) in the command register.

### Operation in Each Mode

The operation mode for each group in the μPD71055 can be set according to the application. Group 0 can be in modes 0, 1, or 2, while group 1 is in mode 0 or 1. Group 1 cannot be used in mode 2.

The  $\overline{RD}$  and  $\overline{WR}$  signals that appear in the descriptions of each mode refer to the port in question as addressed by  $A_1$  and  $A_0$ . These signals only affect the port addressed by  $A_1$  and  $A_0$ .

Where the port addressed may not be clear, 0 or 1 is appended to the signal name to indicate the port.

### Mode 0

In this mode the ports of the μPD71055 are used to perform basic I/O operations. Each port operates with a buffered input and a buffered latched output. See figure 4.

Depending on the control word sent to the μPD71055 from the system bus, ports 0, 1, and 2 can be independently specified for input or output.

### Input Port Operation

While the  $\overline{RD}$  signal is low, data from the port selected by the  $A_1A_0$  signals is put on the data bus. See figure 5.

### Output Port Operation

When the μPD71055 is written to ( $\overline{WR} = 0$ ), the data on the data bus will be latched in the port selected by the  $A_1A_0$  signals at the rising edge of  $\overline{WR}$  and output to the port pins (figure 6). Following the programming of mode 0, all outputs are at a low level.

By reading a port which is set for output, the output value of the port can be obtained.

Note: When group 0 is in mode 1 or mode 2, only bits  $P_{22}$ - $P_{20}$  of port 2 can be used by group 1. Bit  $P_{23}$  belongs to group 0.

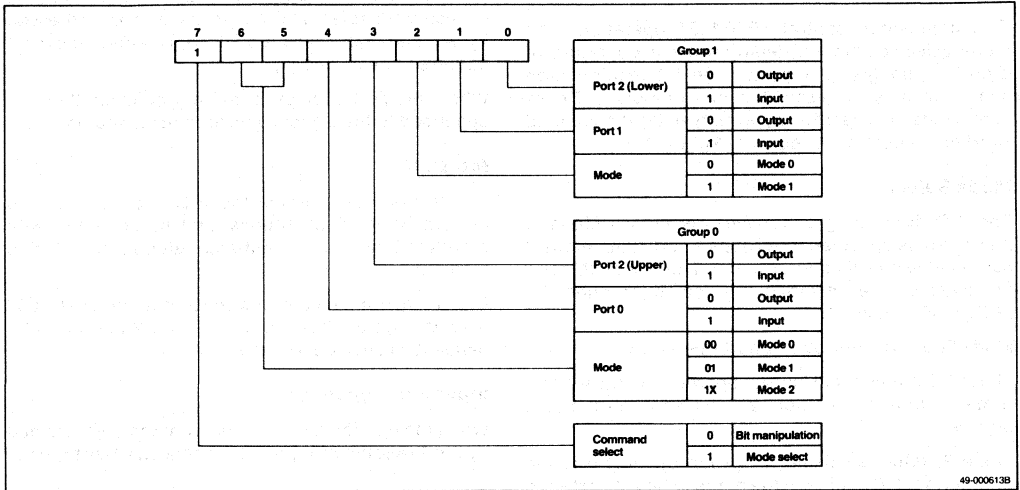
### Mode 0 Example

This is an example of a CPU connected to an A/D converter via a μPD71055 (figure 7). Here both group 0 and group 1 are set to mode 0 and port 2 is used to start conversion and detect the end of the conversion process.

Figure 8 is a subroutine that reads the converted data from an A/D converter.

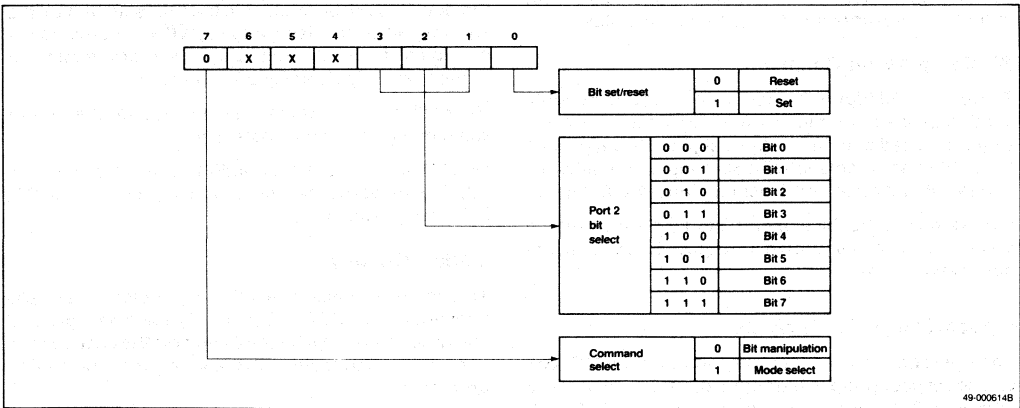


Figure 1. Mode Select Command Word



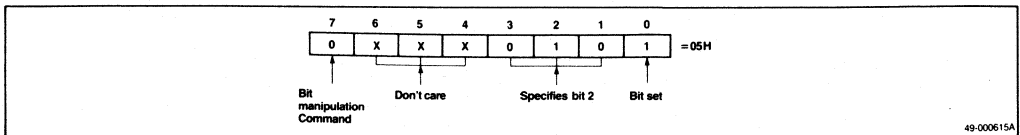
49-000613B

Figure 2. Bit Manipulation Command Word



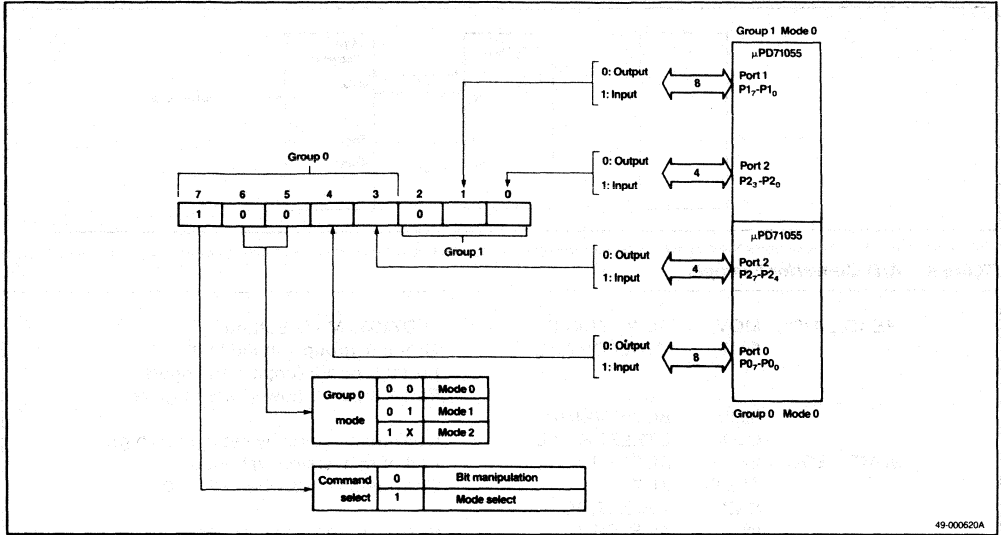
49-000614B

Figure 3. Bit Manipulation Command Example

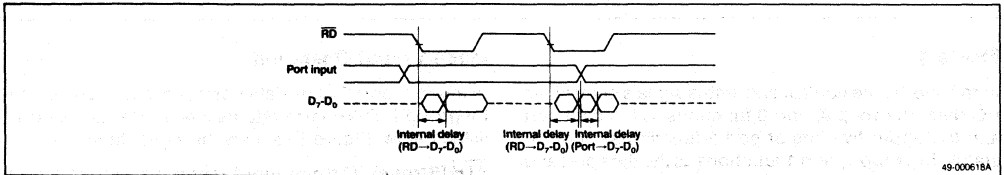


49-000615A

**Figure 4. Mode 0**



**Figure 5. Mode 0 Input Timing**



**Figure 6. Mode 0 Output Timing**

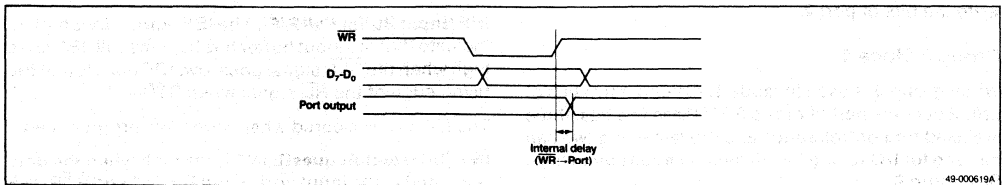


Figure 7. A/D Converter Connection Example

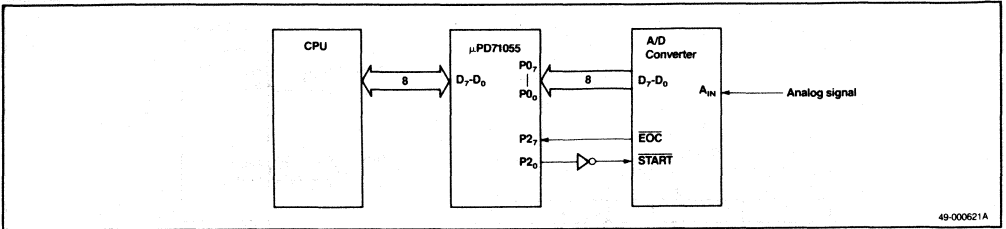


Figure 8. A/D Converter Example

```

READ_A/D:  MOV     AL,10011000B           ;μPD71055 Mode Setting:
           OUT     CTRLPORT,AL          ;Group 0, group 1 in mode 0
                                           ;Port 0 & port 2 (upper) are inputs
                                           ;Port 1 & port 2 (lower) are outputs

           MOV     AL,00000001B
           OUT     CTRLPORT,AL          ;Conversion starts by setting P2<sub>0</sub> high
WAIT_EOC:  IN      AL,PORT2              ;End of conversion wait loop
           TEST1   AL,7                  ;Conversion ends when P2<sub>7</sub> = 0
           BNZ     WAIT_EOC
           IN      AL,PORT0              ;Read A/D converted values
           RET
    
```

**Mode 1**

In this mode, the control and status signals control the I/O data. In group 0, port 0 functions as the data port and the upper five bits of port 2 function as control/status. In group 1, port 1 functions as the data port and the lower three bits of port 2 function as control/status.

In mode 1, the bit manipulation command is used to write the bits of port 2.

**Group 0 Mode 1**

When group 0 is used in mode 1, the upper five bits of port 2 become part of group 0. Of these five bits, three are used for control/status and the remaining two can be used for I/O (using the bit manipulation command). See figure 9.

**Group 1 Mode 1**

When group 1 is used in mode 1, the lower three or four bits of port 2 become part of group 1. Of these four bits, three are used for control/status. The remaining bit, P2<sub>3</sub>, can be used for I/O only if group 0 is in mode 0. Otherwise, P2<sub>3</sub> belongs to group 0 as a control/status bit. See figure 9 and table 4.

**Mode 1 Input Operation**

In mode 1, port 0 is the data port for group 0, and port 1 for group 1. The control/status bits (port 2) are used as listed below. Figure 10 shows the signal timing.

**STB [Strobe]**. The data input at port 0 is latched in port 0 when the STB<sub>0</sub> input is brought low. The data input at port 1 is latched in port 1 by STB<sub>1</sub>.

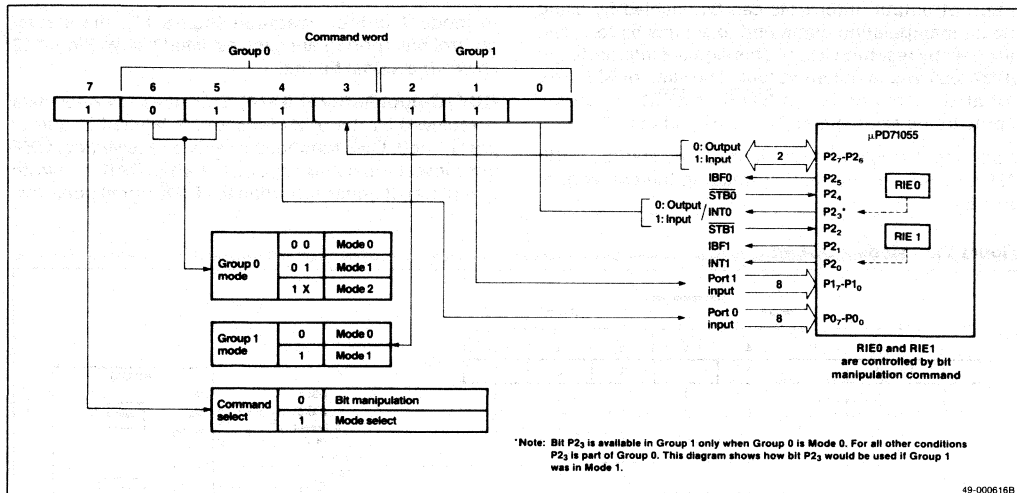
**IBF [Input Buffer Full F/F]**. The IBF output goes high to indicate that the input buffer has become full. IBF goes high when the STB signal goes low. IBF goes low at the rising edge of the RD signal when STB = 1.

The IBF F/F is cleared when mode 1 is programmed.

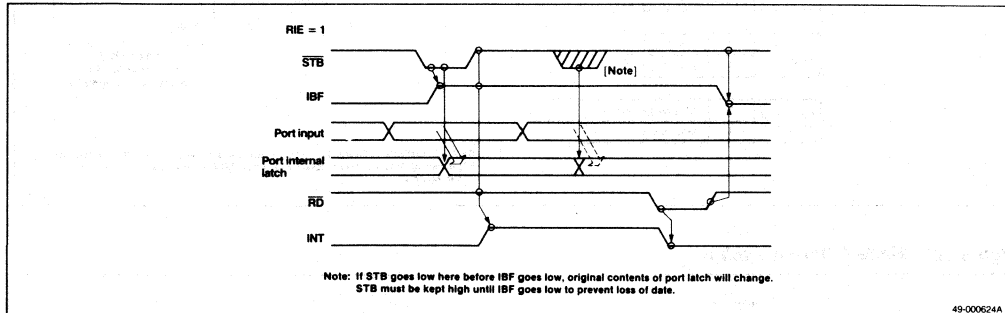
**INT [Interrupt Request]**. INT goes high when the data is latched in the input port, when RIE is 1 and STB, IBF and RD are all high. INT goes low at the falling edge of the RD signal. It can function as a data read request interrupt signal to a CPU.

INT is cleared when mode 1 is programmed.

**Figure 9. Mode 1 Input**



**Figure 10. Mode 1 Input Timing**



**RIE [Read Interrupt Enable Flag].** RIE controls the interrupt output. Interrupts can be enabled by using the bit manipulation command to set this bit to 1, and disabled by resetting it to 0. This signal is internal to the μPD71055 and is not an output. The state of RIE does not affect the function of STB0 or STB1, which are inputs to the same bits (P2<sub>4</sub> and P2<sub>2</sub>) of port 2.

When input is specified in mode 1, the status of IBF, INT and RIE can be read by reading the contents of port 2.

**Mode 1 Output Operation**

In mode 1 output operation (figure 11), the status/control bits (port 2) are used as listed below. Figure 12 shows the signal timing.

**ÖBF [Output Buffer Full F/F].** ÖBF goes low when data is received by the μPD71055 and is latched in output ports 1 or 0. ÖBF functions as a data receive flag. ÖBF goes low at the rising edge of WR when DAK = 1 (write complete). It goes high when the DAK signal goes low.

Figure 11. Mode 1 Output

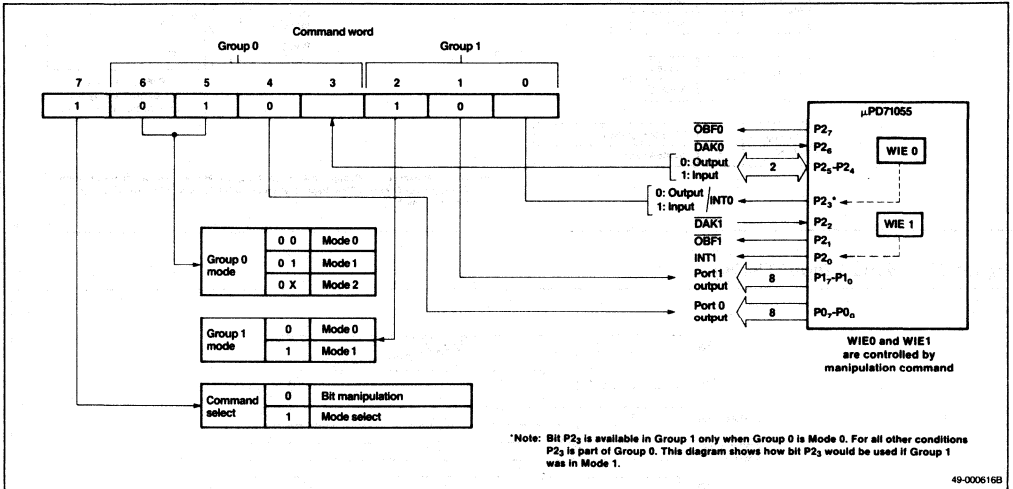
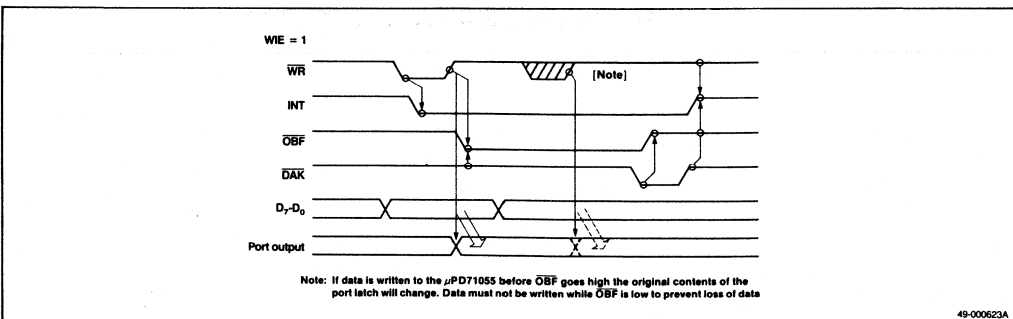


Figure 12. Mode 1 Output Timing



**DAK [Data Acknowledge].** When this input is low, it signals the μPD71055 that output port data has been taken from the 71055.

**INT [Interrupt Request].** INT goes high when the output data is taken when WIE is set to 1 and WR, OBF and DAK are all high. It goes low at the falling edge of the WR signal. INT therefore functions as a write request signal, indicating that new data should be sent to the μPD71055.

**WIE [Write Interrupt Enable Flag].** WIE controls the interrupt output. Interrupts can be enabled by using the bit manipulation command to set this bit to 1 and disabled by resetting it to 0. This signal is internal to the μPD71055 and is not an output. The state of WIE does not affect the function of DAK addressed to the same bits of port 2.

When output is specified in mode 1, the status of OBF, INT and WIE can be obtained by reading the contents of port 2.

Table 2 shows a summary of these signals.

**Table 2. Functions of Port 2 Bits in Mode 1**

Group Bit	Data Input	Data Output
1	P2 <sub>0</sub> INT1 (Interrupt request)	INT1 (Interrupt request)
	P2 <sub>1</sub> IBF1 (Input buffer full f/f)	OBF1 (Output buffer full f/f)
	P2 <sub>2</sub> STB1 (Strobe input)	DAK1 (Data acknowledge input)
	RIE1 (Read interrupt enable flag)	WIE1 (Write interrupt enable flag)
P2 <sub>3</sub>	I/O (Note)	I/O (Note)
0	P2 <sub>3</sub> INT0 (Interrupt request)	INT0 (Interrupt request)
	P2 <sub>4</sub> STB0 (Strobe input)	I/O
		RIE0 (Read interrupt enable flag)
	P2 <sub>5</sub> IBF0 (Input buffer full f/f)	I/O
	P2 <sub>6</sub> I/O	DAK0 (Data acknowledge input)
		WIE0 (Write interrupt enable flag)
P2 <sub>7</sub>	I/O	OBF0 (Output buffer full f/f)

**Note:** Can be used with group 1 only when group 0 is set to mode 0. In other modes, P2<sub>3</sub> belongs to group 0.

### Mode 1 Example

This example (figure 13) demonstrates connecting a printer to the μPD71055. Group 0 is used in mode 1 output. Group 1 can operate in mode 0 or 1; in this example it is set to mode 0.

**Figure 13. Connection to Printer**

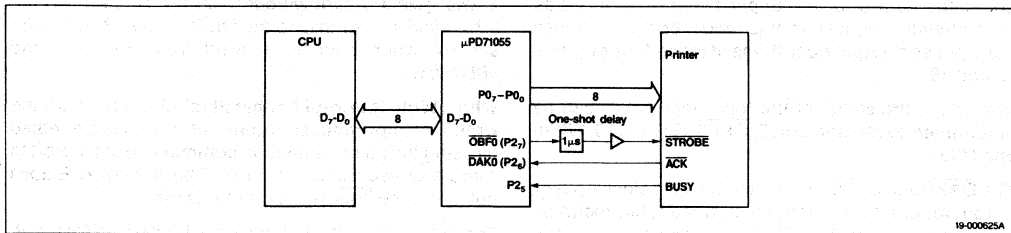


Figure 14. Printer Example Subroutine

```

;This subroutine sends character strings to the printer
INIT:      MOV      AL,10101000B      ; $\mu$ PD71055 Mode Setting:
;Group 0: mode 1 output
;Group 1: mode 0

          OUT      CTRLPORT,AL

SENDPRN:   MOV      BW,DATA          ;Output data address
PRNLOOP:   MOV      AL,[BW]
          CMP      AL,0FFH          ;End if data = 0FFH
          BNZ      WAIT
          RET

WAIT:      IN      AL,PORT2
          TEST1    AL,7              ;Wait until output buffer is empty
          BZ      WAIT
          TEST1    AL,5              ;Wait until printer can accept data
          BNZ      WAIT
          MOV      AL,[BW]          ;Send data to printer
          OUT      PORT0,AL
          INC      BW
          BR      PRNLOOP

```

## Mode 2

Mode 2 can only be used by group 0. In this mode, port 0 functions as a bidirectional 8-bit data port operating under the control of the upper five bits of port 2 as control/status signals. In this mode, port 0 combines the input and output operations of mode 1. See figures 15 and 16.

In mode 2, the status of the following signals can be determined by reading port 2:  $\overline{\text{OBF0}}$ ,  $\text{IBF0}$ ,  $\text{INT0}$ ,  $\text{WIE0}$ , and  $\text{RIE0}$ .

The  $\overline{\text{DAK0}}$  and  $\overline{\text{STB0}}$  signals are used to select input or output for port 0. By using these signals, bidirectional operation between the  $\mu$ PD71055 and peripheral can be realized.

In mode 2, the bit manipulation command is used to write to port 2.

### Control/Status Port Operation

The following control/status signals are used for output:

**$\overline{\text{OBF0}}$  [Output Buffer Full].**  $\overline{\text{OBF0}}$  goes low when data is received from the  $\text{D}_0$ - $\text{D}_7$  data bus and is latched in the port 0 output buffer. It therefore functions as a receive request signal to the peripheral.  $\overline{\text{OBF0}}$  goes low

at the rising edge of the  $\overline{\text{WR0}}$  signal (end of data write). It goes high when  $\overline{\text{DAK0}}$  is low (output data from port 0 received).

**$\overline{\text{DAK0}}$  [Data Acknowledge].**  $\overline{\text{DAK0}}$  is sent to the  $\mu$ PD71055 in response to the  $\overline{\text{OBF0}}$  signal. It should be set low when data is received from port 0 of the  $\mu$ PD71055.

**$\text{WIE0}$  [Write Interrupt Enable Flag].**  $\text{WIE0}$  controls the write interrupt request output. Interrupts are enabled by using the bit manipulation command to set this bit to 1 and disabled by setting it to 0. The state of  $\text{WIE}$  does not affect the  $\overline{\text{DAK}}$  function of this pin.

The following control/status signals are used for input:

**$\overline{\text{STB0}}$  [Strobe Input].** When  $\overline{\text{STB0}}$  goes low, the data being sent to the  $\mu$ PD71055 is latched in port 0.

**$\text{IBF0}$  [Input Buffer Full F/F].** When  $\text{IBF0}$  goes high, it indicates that the input buffer is full. It functions as a signal which can be used to prohibit further data transfer.  $\text{IBF0}$  goes high when  $\overline{\text{STB0}}$  goes low. It goes low at the rising edge of  $\text{RD0}$  when  $\overline{\text{STB0}} = 1$  (read complete).

Figure 15. Mode 2

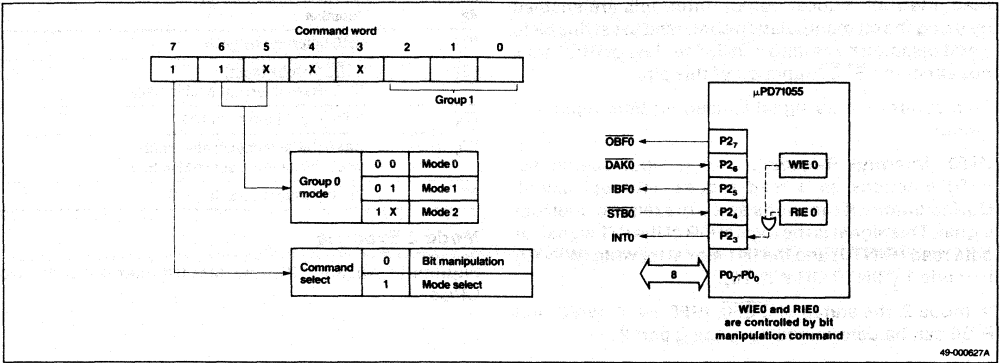
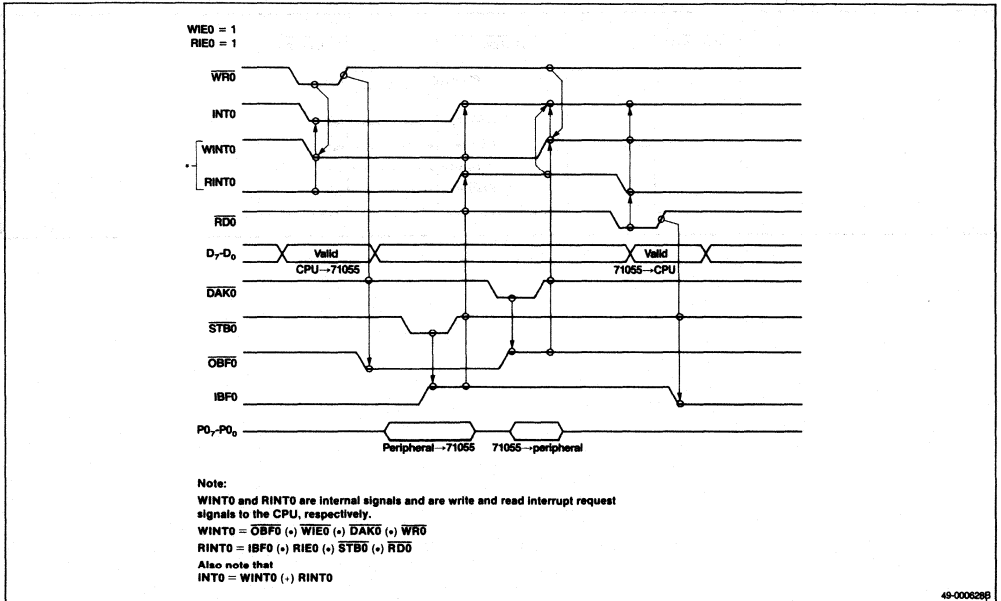


Figure 16. Mode 2 Timing





**RIE0 [Read Interrupt Enable Flag].** RIE0 controls the read interrupt request output. Interrupts are enabled by using the bit manipulation command to set this bit to 1 and disabled by setting it to 0. The state of RIE0 does not affect the STB0 function of this pin.

This control/status signal is used for both input and output:

**INT0 [Interrupt Request].** During input operations, INTO functions as a read request interrupt signal. During output, it functions as a write request interrupt signal. This signal is the logical OR of the INT signal for data read (RINT0) and the INT signal for write (WINT0) in mode 1 (RINT0 OR WINT0).

In mode 2, the status of  $\overline{\text{OBF0}}$ , IBF0, INTO, WIE0, and RIE0 can be determined by reading port 2.

Table 3 is a summary of these signals.

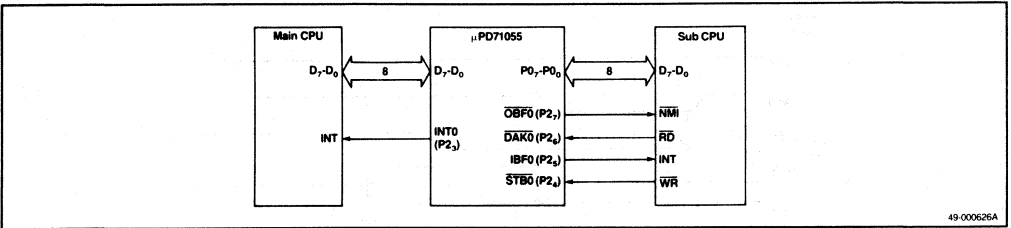
**Table 3. Functions of Port 2 in Mode 2**

BIT	Function
P2 <sub>3</sub>	INT0 (Interrupt request)
P2 <sub>4</sub>	STB0 (Strobe input) RIE0 (Read interrupt enable flag)
P2 <sub>5</sub>	IBF0 (Input buffer full f/f)
P2 <sub>6</sub>	DAK0 (Data acknowledge input) WIE0 (Write interrupt enable flag)
P2 <sub>7</sub>	$\overline{\text{OBF0}}$ (Output buffer full f/f)

**Mode 2 Example**

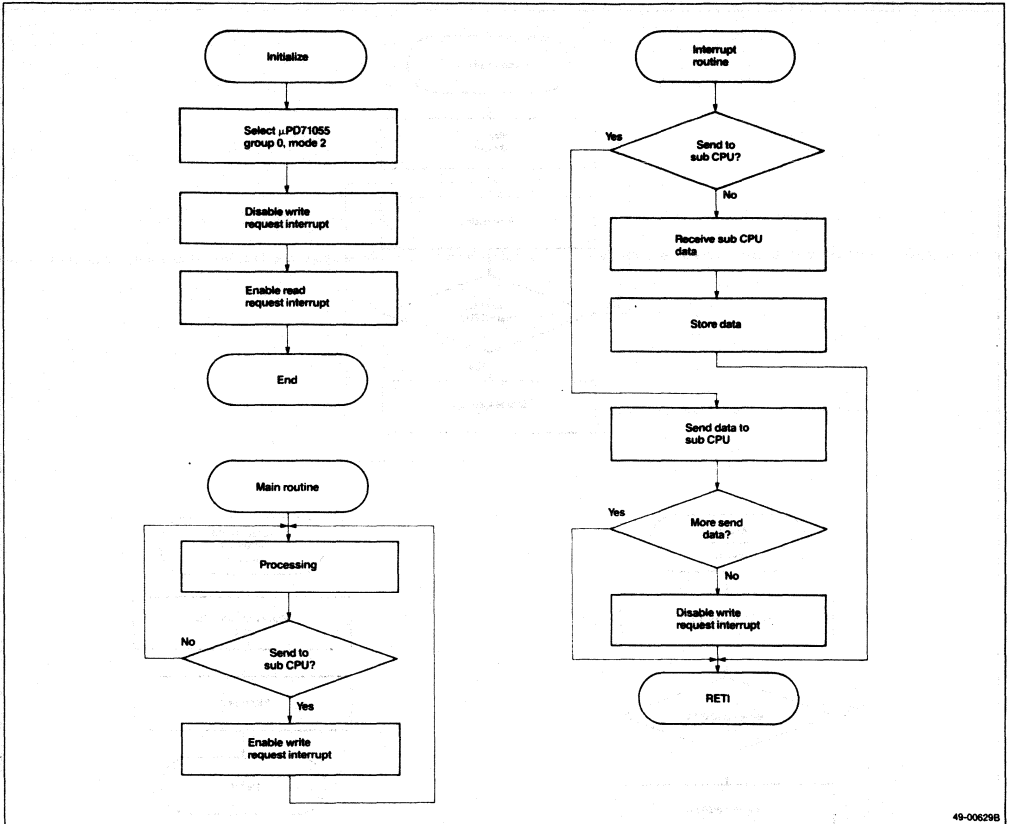
Figures 17, 18, and 19 show data transfer between two CPUs.

**Figure 17. Connecting Two CPUs**



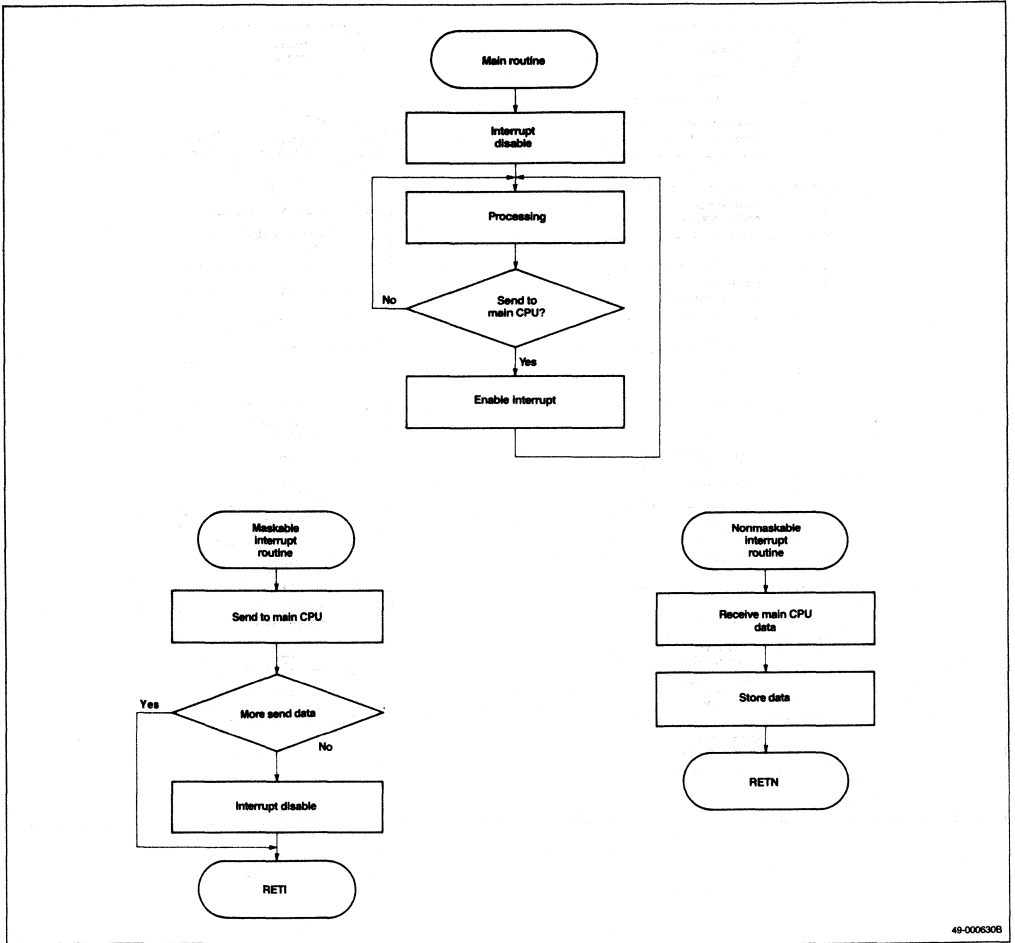
49-000626A

Figure 18. Main CPU Flowchart



49-006298

Figure 19. Sub CPU Flowchart



49-000630B

## Mode Combinations

Table 4 is a complete list of all the combinations of modes and groups, and the function of the port 2 bits in each mode.

**Table 4. Mode Combinations and Port 2 Bit Functions**

Mode	Group 0						Mode	Group 1				
	P0 <sub>7</sub> -P0 <sub>0</sub>	P2 <sub>7</sub>	P2 <sub>6</sub>	P2 <sub>5</sub>	P2 <sub>4</sub>	P2 <sub>3</sub>		P1 <sub>7</sub> -P1 <sub>0</sub>	P2 <sub>3</sub>	P2 <sub>2</sub>	P2 <sub>1</sub>	P2 <sub>0</sub>
0	In	D	D	D	D	NA	0	In	D	D	D	D
0	In	D	D	D	D	NA	0	Out	D	D	D	D
0	In	D	D	D	D	NA	1	In	B	STB1 (RIE1)	IBF1	INT1
0	In	D	D	D	D	NA	1	Out	B	DAK1 (WIE1)	OB $\overline$ F1	INT1
0	Out	D	D	D	D	NA	0	In	D	D	D	D
0	Out	D	D	D	D	NA	0	Out	D	D	D	D
0	Out	D	D	D	D	NA	1	In	B	STB1 (RIE1)	IBF1	INT1
0	Out	D	D	D	D	NA	1	Out	B	DAK1 (WIE1)	OB $\overline$ F1	INT1
1	In	B	B	IBF0	STB0 (RIE0)	INT0	0	In	NA	D	D	D
1	In	B	B	IBF0	STB0 (RIE0)	INT0	0	Out	NA	D	D	D
1	In	B	B	IBF0	STB0 (RIE0)	INT0	1	In	NA	STB1 (RIE1)	IBF1	INT1
1	In	B	B	IBF0	STB0 (RIE0)	INT0	1	Out	NA	DAK1 (WIE1)	OB $\overline$ F1	INT1
1	Out	OB $\overline$ F0	DAK0 (WIE0)	B	B	INT0	0	In	NA	D	D	D
1	Out	OB $\overline$ F0	DAK0 (WIE0)	B	B	INT0	0	Out	NA	D	D	D
1	Out	OB $\overline$ F0	DAK0 (WIE0)	B	B	INT0	1	In	NA	STB1 (RIE1)	IBF1	INT1
1	Out	OB $\overline$ F0	DAK0 (WIE0)	B	B	INT0	1	Out	NA	DAK1 (WIE1)	OB $\overline$ F1	INT1
2	I/O	OB $\overline$ F0	DAK0 (WIE0)	IBF0	STB0 (RIE0)	INT0	0	In	NA	D	D	D
2	I/O	OB $\overline$ F0	DAK0 (WIE0)	IBF0	STB0 (RIE0)	INT0	0	Out	NA	D	D	D
2	I/O	OB $\overline$ F0	DAK0 (WIE0)	IBF0	STB0 (RIE0)	INT0	1	In	NA	STB1 (RIE1)	IBF1	INT1
2	I/O	OB $\overline$ F0	DAK0 (WIE0)	IBF0	STB0 (RIE0)	INT0	1	Out	NA	DAK1 (WIE1)	OB $\overline$ F1	INT1

**Note:**

- (1) In this chart, "NA" indicates that the bit cannot be used by this group.
- (2) The symbol "B" indicates bits that can only be rewritten by the bit manipulation command.
- (3) In this chart, "D" indicates that is used by the user.
- (4) Symbols in parentheses are internal flags. They are not output to port 2 pins and they cannot be read by the host.
- (5) In indicates Input, Out indicates Output, and I/O indicates Input/Output.



## Description

The  $\mu$ PD71059 is a low-power CMOS programmable interrupt control unit for microcomputer systems. It can process eight interrupt request inputs, allocating a priority level to each one. It transfers the interrupt with the highest priority to the CPU, along with interrupt address information. By cascading up to eight slave  $\mu$ PD71059s to a master  $\mu$ PD71059, a system can process up to 64 interrupt requests. System scale, interrupt routine address, interrupt request priority and masking are all under complete program control.

## Features

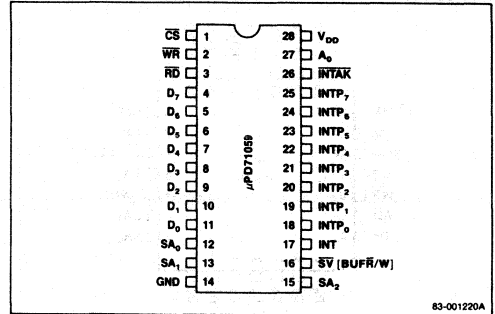
- $\mu$ PD8085A compatible (CALL mode)
- $\mu$ PD70108/70116 compatible (vector mode)
- Eight interrupt request inputs per chip
- Up to 64 interrupt requests inputs per system (extended mode)
- Edge- or level-triggered interrupt request inputs
- Each interrupt maskable
- Programmable priority level
- Polling operation
- Single +5 V  $\pm$ 10% power supply
- Industrial temperature range: -40 to +85 °C
- CMOS technology

## Ordering Information

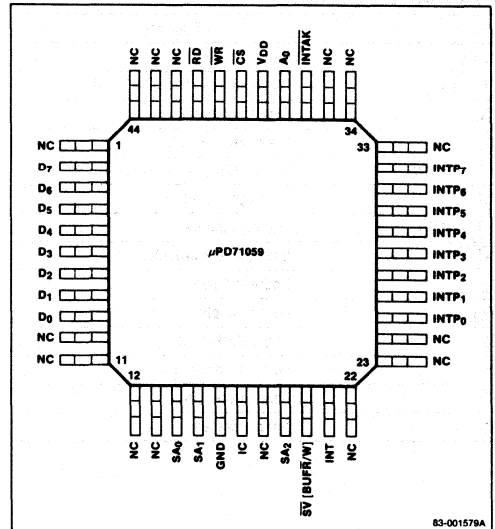
Part Number	Package Type	Max. Frequency of Operation
$\mu$ PD71059C	28-pin plastic DIP	8 MHz
$\mu$ PD71059GB	44-pin plastic flat pack	8 MHz
$\mu$ PD71059L	28-pin PLCC	8 MHz
$\mu$ PD71059C-10	28-pin plastic DIP*	10 MHz
$\mu$ PD71059GB-10	44-pin plastic flat pack	10 MHz
$\mu$ PD71059L-10	28-pin plastic PLCC	10 MHz

## Pin Configurations

### 28-Pin Plastic DIP

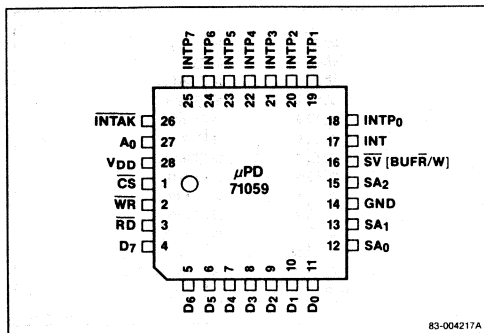


### 44-Pin Flat Pack



### Pin Configurations (cont)

#### 28-Pin Plastic Leaded Chip Carrier (PLCC)



83-004217A

### Pin Identification

Symbol	Function
D <sub>7</sub> -D <sub>0</sub>	Data bus I/O
SA <sub>2</sub> -SA <sub>0</sub>	Slave address I/O, bits 2, 1, 0
GND	Ground potential
IC	Internally connected
SV (BUFR/W)	Slave (Buffer read write) I/O
INT	Interrupt output
INTP <sub>0</sub> -INTP <sub>7</sub>	Interrupt inputs
INTAK	Interrupt acknowledge input
A <sub>0</sub>	Address input
V <sub>DD</sub>	Power supply
CS	Chip select input
WR	Write strobe input
RD	Read strobe input
NC	Not connected

### Pin Functions

#### D<sub>7</sub>-D<sub>0</sub> [Data Bus]

The 8-bit 3-state bidirectional bus transfers data to and from the CPU through the system bus. The data bus becomes active when data is sent to the CPU in the  $\overline{\text{INTAK}}$  sequence. Otherwise, the data bus is high impedance.

#### CS [Chip Select]

The CPU uses the μPD71059's  $\overline{\text{CS}}$  input to select a μPD71059 to read from (IN instructions) or write to (OUT instructions). The RD and WR signals to the μPD71059 are enabled when CS is low. CS is not used for the INTAK sequence.

#### RD [Read Strobe]

The CPU sets the RD input to 0 when reading the internal registers IMR, IRR and ISR, and during polling operations to read polling data.

#### WR [Write Strobe]

The CPU sets the WR input to 0 when writing initializing words IW1-IW4 and command words IMW, PFCW and MCW.

#### A<sub>0</sub> [Address]

The A<sub>0</sub> input is used with CS, RD, and WR to read or write to the μPD71059. Normally, A<sub>0</sub> is connected to A<sub>0</sub> of the address bus. Table 1 shows the relationship between read/write operations and the control signals (CS, WR, RD, and A<sub>0</sub>).

#### INTP<sub>7</sub>-INTP<sub>0</sub> [Interrupt Request from Peripheral]

INTP<sub>7</sub>-INTP<sub>0</sub> are eight asynchronous interrupt request inputs. They can be set to be either edge- or level-triggered. These pins are pulled up by an internal resistance. Their power consumption is lower at high-level input than at low-level input.

#### INT [Interrupt]

INT is the interrupt request output from a μPD71059 to the CPU or master μPD71059. When an interrupt from a peripheral is input to an INTP pin and acknowledged, the μPD71059 asserts INT high to generate an interrupt request at the CPU or master μPD71059.

## INTAK [Interrupt Acknowledge]

The INTAK input from the CPU acknowledges an interrupt from the μPD71059. After acknowledging the interrupt request, the CPU returns three low-level pulses (μPD8085) or two low-level pulses (μPD70108/70116). Synchronizing to these pulses, the μPD71059 sends a CALL instruction in three bytes, or an interrupt vector number in one byte through the data bus.

## SV [BUF $\bar{R}$ /W] [Slave, Buffer Read/Write]

This pin has two functions. When no external buffer is used in the data bus, it is the SV input. When SV is low, the μPD71059 acts as a slave. It operates as a master when SV is high. SV has no master/slave meaning when the μPD71059 is set to single mode.

As the BUF $\bar{R}$ /W output, this pin can allow a bus transceiver to be controlled by the μPD71059, if one is required. When the μPD71059 changes its data bus to output, it sets BUF $\bar{R}$ /W low. It sets BUF $\bar{R}$ /W high when the data bus changes to input.

## SA<sub>2</sub>-SA<sub>0</sub> [Slave Address]

These pins are only used in systems with cascaded μPD71059s. The master μPD71059 uses these pins to address up to eight slave μPD71059s. These pins are output pins for masters, and input pins for slaves.

**Note:** In the single mode, SA<sub>2</sub>-SA<sub>0</sub> are output pins, but the output data has no meaning.

## VDD [Power Supply]

This is the positive power supply.

## GND [Ground]

This is the ground potential.

## IC [Internally Connected]

This pin must be left unconnected.

**Table 1. Read/Write Operations**

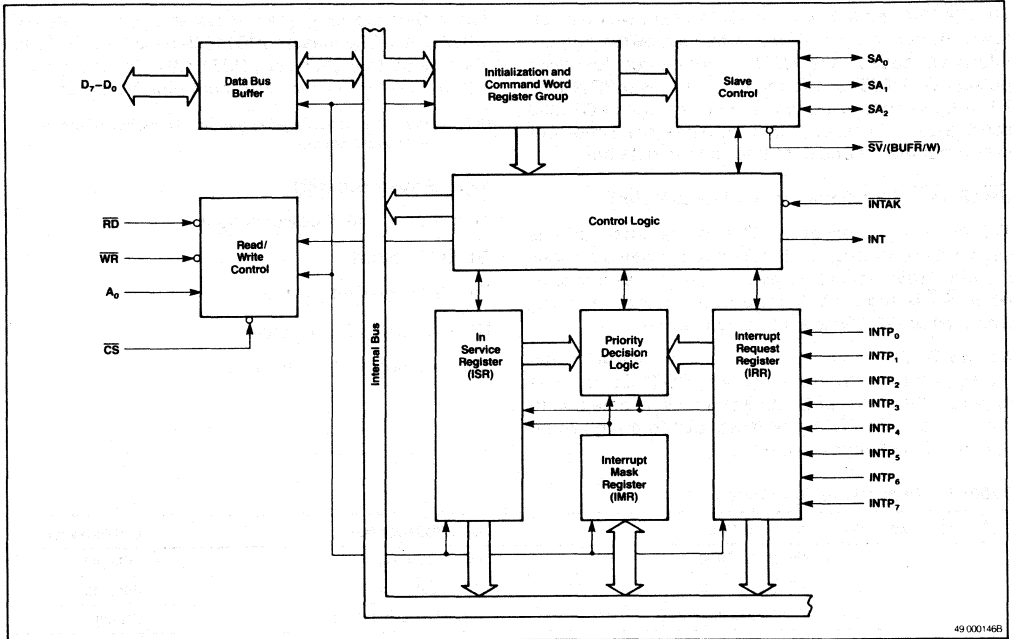
CS	RD	WR	A <sub>0</sub>	Other Conditions	μPD71059 Operation	CPU Operation
0	0	1	0	IRR set by MCW	IRR to Data bus	IRR read
				ISR set by MCW	ISR to Data bus	ISR read
				Polling phase (Note 1)	Polling data to Data bus	Polling
0	0	1	1		IMR to Data bus	IMR read
0	1	0	0	D <sub>4</sub> = 1	Data bus to IW1 register	IW1 write
				D <sub>4</sub> , D <sub>3</sub> = 0	Data bus to PFCW register	PFCW write
				D <sub>4</sub> = 0, D <sub>3</sub> = 1	Data bus to MCW register	MCW write
0	1	0	1	(Note 2)	Data bus to IW2 register	IW2 write
					Data bus to IW3 register	IW3 write
					Data bus to IW4 register	IW4 write
				After initializing	Data bus to IMR	IMW write
0	1	1	x		Data bus high impedance	
1	x	x	x			
0	0	0	x		Illegal	

**Note:**

- (1) In the polling phase, polling data is read instead of IRR and ISR.
- (2) Refer to Control Words section for IW2-IW4 writing procedure.



Block Diagram



49 0001466

Block Diagram Functions

Data Bus Buffer

The data bus buffer is a buffer between D<sub>7</sub>-D<sub>0</sub> and the μPD71059's internal bus.

Read/Write Control

The read/write control controls the CPU's reading and writing to and from the μPD71059 registers.

Initialization and Command Word Registers

These registers store initializing words IW1-IW4 and command words PFCW (priority and finish control word) and MCW (mode control word). The CPU cannot read these registers.

Interrupt Mask Register [IMR]

The interrupt mask register stores the interrupt mask word (IMW) command word. Each bit masks an interrupt. If bit n of this register is 1, the interrupt request INTP<sub>n</sub> is masked and cannot be accepted by the μPD71059. The CPU can read this register by performing an IN instruction with A<sub>0</sub> = 1.

Interrupt Request Register [IRR]

The interrupt request register shows which interrupt levels are currently being requested. If bit n of the IRR is 1, INTP<sub>n</sub> is requesting an interrupt. The CPU can read this register.

In-Service Register [ISR]

The in-service register shows all interrupt levels currently in service. If bit n of this register is 1, the interrupt routine corresponding to INTP<sub>n</sub> is currently being executed. The CPU can read this register.

Slave Control

Slave control is used in systems with cascaded μPD71059s. A master μPD71059 uses it to control slave μPD71059s, and a slave uses it to interface with the master μPD71059.

Control Logic

The control logic receives and generates the signals that control the sequence of events in an interrupt.

## Priority Decision Logic

The priority decision logic determines which interrupt request from the IRR will be serviced next. The decision is made based upon the current interrupt mask, interrupt service status, mode status, and current priority.

## Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$

Power supply voltage, $V_{DD}$	-0.5 to +7.0 V
Input voltage, $V_I$	-0.5 to $V_{DD} + 0.3$ V
Output voltage, $V_O$	-0.5 to $V_{DD} + 0.3$ V
Power dissipation, $P_{D_{MAX}}$	500 mW
Operating temperature, $T_{opt}$	-40 to +85°C
Storage temperature, $T_{stg}$	-65 to +150°C

**Comment:** Exposing the device to stresses above those listed in the absolute maximum ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational section of this specification. Exposure to absolute maximum ratings for extended periods may affect device reliability.

## Capacitance

$T_A = 25^\circ\text{C}; V_{DD} = \text{GND} = 0$  V

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	$C_I$		10		pF	$f_c = 1$ MHz
I/O capacitance	$C_{IO}$		20		pF	Unmeasured pins returned to 0 V

## DC Characteristics

$T_A = -40$  to  $+85^\circ\text{C}; V_{DD} = 5$  V  $\pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage, high	$V_{IH}$	2.2		$V_{DD} + 0.3$	V	
Input voltage, low	$V_{IL}$	-0.5		0.8	V	
Output voltage, high	$V_{OH}$	$0.7 \times V_{DD}$			V	$I_{OH} = -400$ μA
Output voltage, low	$V_{OL}$		0.4		V	$I_{OL} = 2.5$ mA
Input leakage current, high	$I_{LIH}$			10	μA	$V_I = V_{DD}$
Input leakage current, low	$I_{LIL}$			-10	μA	$V_I = 0$ V
Output leakage current, high	$I_{LOH}$			10	μA	$V_O = V_{DD}$
Output leakage current, low	$I_{LOL}$			-10	μA	$V_O = 0$ V
INTP input leakage current, high	$I_{LI PH}$			10	μA	$V_I = V_{DD}$
INTP input leakage current, low	$I_{LI PL}$			-300	μA	$V_I = 0$ V
Supply current (dynamic)	$I_{DD1}$		3.5	9	mA	
Supply current (power down mode)	$I_{DD2}$		2	50	μA	Input pins: $V_{IH} = V_{DD} - 0.1$ V $V_{IL} = 0.1$ V Output pins: open (Note 1)

### Note:

(1) In power down mode,  $\overline{\text{INTP}}_0$ - $\overline{\text{INTP}}_7$ ,  $\overline{\text{INTAK}}$ , and  $\overline{\text{CS}}$  must be at high level ( $V_{IH} = V_{DD} - 0.1$  V).

**AC Characteristics, μPD71059**

T<sub>A</sub> = -40 to +85°C; V<sub>DD</sub> ± 5 V + 10%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
<b>Read Timing</b>						
A <sub>0</sub> , $\overline{\text{CS}}$ set-up to RD ↓	t <sub>SAR</sub>	0			ns	
A <sub>0</sub> , $\overline{\text{CS}}$ hold from RD ↑	t <sub>HRA</sub>	0			ns	
$\overline{\text{RD}}$ pulse width low	t <sub>RRL</sub>	160			ns	
$\overline{\text{RD}}$ pulse width high	t <sub>RRH</sub>	120			ns	
Data delay from RD ↓	t <sub>DRD</sub>			120	ns	C <sub>L</sub> = 150 pF
Data float from RD ↑	t <sub>FRD</sub>	10		85	ns	C <sub>L</sub> = 100 pF
Data delay from A <sub>0</sub> , $\overline{\text{CS}}$	t <sub>DAD</sub>			200	ns	C <sub>L</sub> = 150 pF
BUFR/W delay from RD ↓	t <sub>DRBL</sub>			100	ns	
BUFR/W delay from RD ↑	t <sub>DRBH</sub>			150	ns	
<b>Write Timing</b>						
A <sub>0</sub> , $\overline{\text{CS}}$ set-up to WR ↓	t <sub>SAW</sub>	0			ns	
A <sub>0</sub> , $\overline{\text{CS}}$ hold from WR ↑	t <sub>HWA</sub>	0			ns	
$\overline{\text{WR}}$ pulse width low	t <sub>WWL</sub>	120			ns	
$\overline{\text{WR}}$ pulse width high	t <sub>WWH</sub>	120			ns	
Data set-up from WR ↑	t <sub>SDW</sub>	120			ns	
Data hold from WR ↑	t <sub>HWD</sub>	0			ns	

**Notes:**

- (1) The time to clear the input latch in edge-trigger mode.
- (2) The time to move from read to write operation.
- (3) The time to move to the next  $\overline{\text{INTAK}}$  operation.
- (4) The time to move  $\overline{\text{INTAK}}$  to/from command (read/write).

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
<b>Interrupt Timing</b>						
INTP pulse width	t <sub>PIPL</sub>	100			ns	(Note 1)
SA set-up to second, third $\overline{\text{INTAK}}$ ↓	t <sub>SSIA</sub>	40			ns	Slave
$\overline{\text{INTAK}}$ pulse width low	t <sub>IAlAL</sub>	160			ns	
$\overline{\text{INTAK}}$ pulse width high	t <sub>IAlAH</sub>	120			ns	$\overline{\text{INTAK}}$ Sequence
INT delay from INTP ↑	t <sub>DIPI</sub>			300	ns	C <sub>L</sub> = 150 pF
SA delay from first $\overline{\text{INTAK}}$ ↓	t <sub>DIAS</sub>			360	ns	Master, C <sub>L</sub> = 150 pF
Data delay from $\overline{\text{INTAK}}$ ↓	t <sub>DIAD</sub>			120	ns	C <sub>L</sub> = 150 pF
Data float from $\overline{\text{INTAK}}$ ↑	t <sub>FIAD</sub>	10		85	ns	
Data delay from SA	t <sub>DSD</sub>			200	ns	Slave, C <sub>L</sub> = 150 pF
BUFR/W delay from $\overline{\text{INTAK}}$ ↓	t <sub>DIABL</sub>			100	ns	C <sub>L</sub> = 150 pF
BUFR/W delay from $\overline{\text{INTAK}}$ ↑	t <sub>DIABH</sub>			150	ns	
<b>Other Timing</b>						
Command recovery time	t <sub>RV1</sub>	120			ns	(Note 2)
$\overline{\text{INTAK}}$ recovery time	t <sub>RV2</sub>	250			ns	(Note 3)
$\overline{\text{INTAK}}$ /command recovery time	t <sub>RV3</sub>	250			ns	(Note 4)

## AC Characteristics, μPD71059-10

T<sub>A</sub> = -40 to +85°C; V<sub>DD</sub> ± 5 V + 10%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
<b>Read Timing</b>						
A <sub>0</sub> CS set-up to RD ↓	t <sub>SAR</sub>	0			ns	
A <sub>0</sub> CS hold from RD ↑	t <sub>HRA</sub>	0			ns	
RD pulse width low	t <sub>RRL</sub>	120			ns	
RD pulse width high	t <sub>RRH</sub>	90			ns	
Data delay from RD ↓	t <sub>DRD</sub>			95	ns	C <sub>L</sub> = 150 pF
Data float from RD ↑	t <sub>FRD</sub>	10		60	ns	C <sub>L</sub> = 100 pF
Data delay from A <sub>0</sub> CS	t <sub>DAD</sub>			120	ns	C <sub>L</sub> = 150 pF
BUFR/W delay from RD ↓	t <sub>DRBL</sub>			80	ns	
BUFR/W delay from RD ↑	t <sub>DRBH</sub>			100	ns	
<b>Write Timing</b>						
A <sub>0</sub> CS set-up to WR ↓	t <sub>SAW</sub>	0			ns	
A <sub>0</sub> CS hold from WR ↑	t <sub>HWA</sub>	0			ns	
WR pulse width low	t <sub>WWL</sub>	100			ns	
WR pulse width high	t <sub>WWH</sub>	90			ns	
Data set-up from WR ↑	t <sub>SDW</sub>	100			ns	
Data hold from WR ↑	t <sub>HWD</sub>	0			ns	

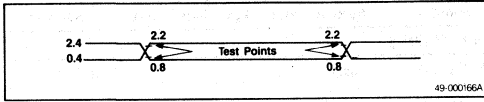
### Notes:

- (1) The time to clear the input latch in edge-trigger mode.
- (2) The time to move from read to write operation.
- (3) The time to move to the next INTAK operation.
- (4) The time to move INTAK to/from command (read/write).

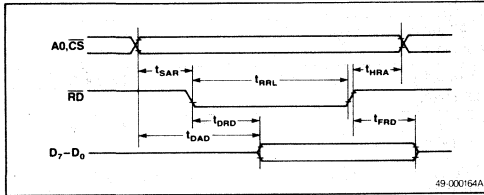
Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
<b>Interrupt Timing</b>						
INTP pulse width	t <sub>PIPL</sub>	80			ns	(Note 1)
SA set-up to second, third INTAK ↓	t <sub>SSIA</sub>	40			ns	Slave
INTAK pulse width low	t <sub>IAIL</sub>	120			ns	
INTAK pulse width high	t <sub>IAIH</sub>	90			ns	INTAK Sequence
INT delay from INTP ↑	t <sub>DIPI</sub>			200	ns	C <sub>L</sub> = 150 pF
SA delay from first INTAK ↓	t <sub>DIAS</sub>			250	ns	Master, C <sub>L</sub> = 150 pF
Data delay from INTAK ↓	t <sub>DIAD</sub>			95	ns	C <sub>L</sub> = 150 pF
Data float from INTAK ↑	t <sub>FIAD</sub>	10		60	ns	
Data delay from SA	t <sub>DSD</sub>			150	ns	Slave, C <sub>L</sub> = 150 pF
BUFR/W delay from INTAK ↓	t <sub>DIABL</sub>			80	ns	C <sub>L</sub> = 150 pF
BUFR/W delay from INTAK ↑	t <sub>DIABH</sub>			100	ns	
<b>Other Timing</b>						
Command recovery time	t <sub>RV1</sub>	90			ns	(Note 2)
INTAK recovery time	t <sub>RV2</sub>	90			ns	(Note 3)
INTAK/command recovery time	t <sub>RV3</sub>	90			ns	(Note 4)

**Timing Waveforms**

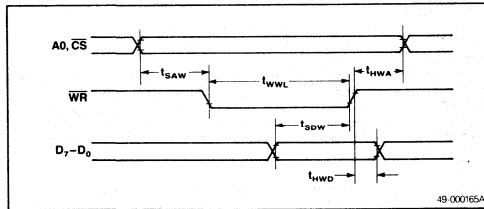
**AC Test Input/Output Waveform**



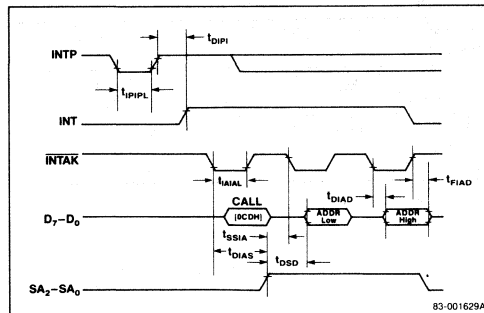
**Read Cycle**



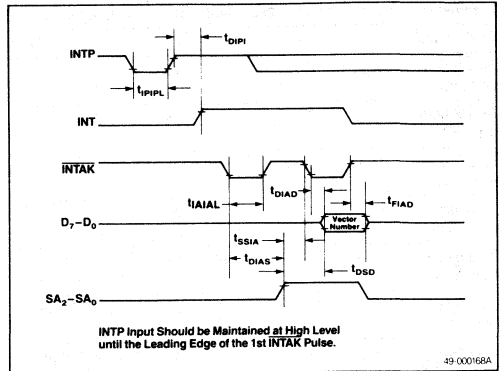
**Write Cycle**



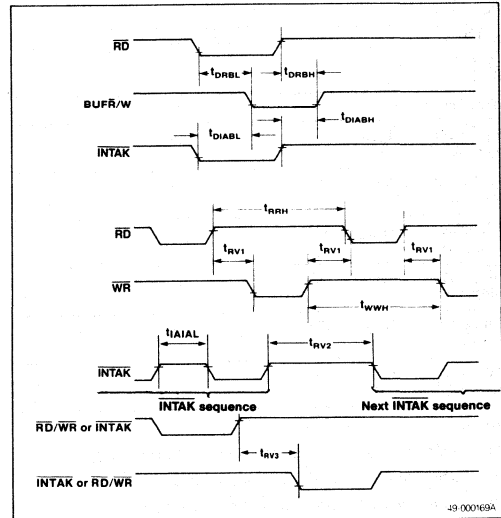
**INTAK Sequence (CALL Mode)**



**INTAK Sequence (Vector Mode)**



**Other Timing**



## Interrupt Operation

Almost all microcomputer systems use interrupts to reduce software overhead when controlling peripherals. However, the number of interrupt pins on a CPU is limited. When the number of interrupt lines increases beyond that limit, external circuits like the μPD71059 become necessary.

The μPD71059 can process eight interrupt request according to an allocated priority order and transmit the signal with the highest priority to the CPU. It also supplies the CPU with information to ascertain the interrupt routine start address. Cascading μPD71059s by connecting up to eight "slave" μPD71059s to a single "master" μPD71059 permits expansion to up to a maximum of 64 interrupt request signals.

Interrupt system scale (master/slave), interrupt routine addresses, interrupt request priority, and interrupt request masking are all programmable, and can be set by the CPU.

Normal interrupt operation for a single μPD71059 is as follows. First, the initialization registers are set with a sequence of initialization words. When the μPD71059 detects an interrupt request from a peripheral to an INT<sub>P</sub> pin it sets the corresponding bit of the interrupt request register (IRR). The interrupt is checked against the interrupt mask register (IMR) and the interrupt service register (ISR). If the interrupt is not masked and there is no other interrupt with a higher priority in service or requesting service, it generates an INT signal to the CPU.

The CPU acknowledges the interrupt by bringing the INTAK line low. The μPD71059 then outputs interrupt CALL or vector data onto the data bus in response to INTAK pulses. During the last INTAK pulse, the μPD71059 sets the corresponding bit in its ISR to indicate that this interrupt is in service and to disable interrupts with lower priority. It resets the bit in the IRR at this point. When the CPU has finished processing the interrupt, it will inform the μPD71059 by sending a finish interrupt (FI) command. This resets the bit in the ISR and allows the μPD71059 to accept interrupts with lower priorities. If the μPD71059 is in the self-FI mode, the ISR bit is reset automatically and this step is not necessary.

## Software Features

The μPD71059 has the following software features:

- Interrupt types: CALL/vector
- Interrupt masking: Normal/extended nesting
- End of interrupt: Self-FI/normal FI/  
specific FI
- Priority rotation: Normal nested/extended  
nested/exceptional nested  
Automatic priority rotation  
Rotate to specific priority
- Polled mode
- CPU-readable registers

## Hardware Configurations

The μPD71059 has the following hardware configurations:

- Interrupt input: Edge/level sensitive
- Cascading μPD71059s: Single/extended  
(master/slave)
- Output driver control: Buffered/non-buffered

## Mode Control

These features and configurations are selected and controlled by the four initialization words (IW1-IW4) and the three command words (IMW, PFCW, and MCW). The format of these words are shown in figures 2 and 3, respectively.

## Control Words

There are two types of μPD71059 control words: initialization words and command words.

There are four initialization words: IW1-IW4. These words must be written to the μPD71059 at least once to initialize it. They must be written in sequence.

There are three types of command words: interrupt mask word (IMW), priority and finish control word (PFCW), and the mode control word (MCW). These words can be written freely after initialization.

**Initialization Words**

**Initialization sequence.** When data is written to a μPD71059 after setting  $A_0 = 0$  and  $D_4 = 1$ , data is always accepted as IW1. This results in a default initialization as shown below. See figure 1.

- (1) The edge-trigger circuit of the INTP input is reset. IRR is cleared in the edge-trigger mode.
- (2) ISR and IMR are cleared.
- (3) INTP<sub>7</sub> receives the lowest priority; INTP<sub>0</sub> receives the highest.
- (4) The exceptional nesting mode is released. IRR is set as the register to be read.
- (5) Register IW4 is cleared. The normal nesting mode, non-buffer mode, FI command mode, and CALL mode are set.

**Initialization Words.** The initialization words are written consecutively, and in order. The first two, IW1 and IW2, set the interrupt address or vector. IW3 specifies which interrupts are slaves for master systems, and defines the slave number of a slave system. Therefore, IW3 is only required in extended systems. The μPD71059 will only expect it if bit D<sub>1</sub> of IW1, SNGL = 0. IW4 is only written if bit D<sub>0</sub> of IW1, I4 = 1. See figure 2 for the format of the initialization words.

**Command Words**

The command words give various commands to a μPD71059 during its operation to change interrupt masks and priorities, to end interrupt processing, etc. See figure 3.

**IMW [Interrupt Mask Word].** This word masks the IRR and disables the corresponding INTP interrupt requests. It also masks the ISR in the exceptional nesting mode. Bits M<sub>7</sub>-M<sub>0</sub> correspond to the interrupt levels of INTP<sub>7</sub>-INTP<sub>0</sub>, respectively.

In the exceptional nesting mode, interrupts corresponding to the bits of IRR and ISR are masked if the M<sub>n</sub> bit is set to 1.

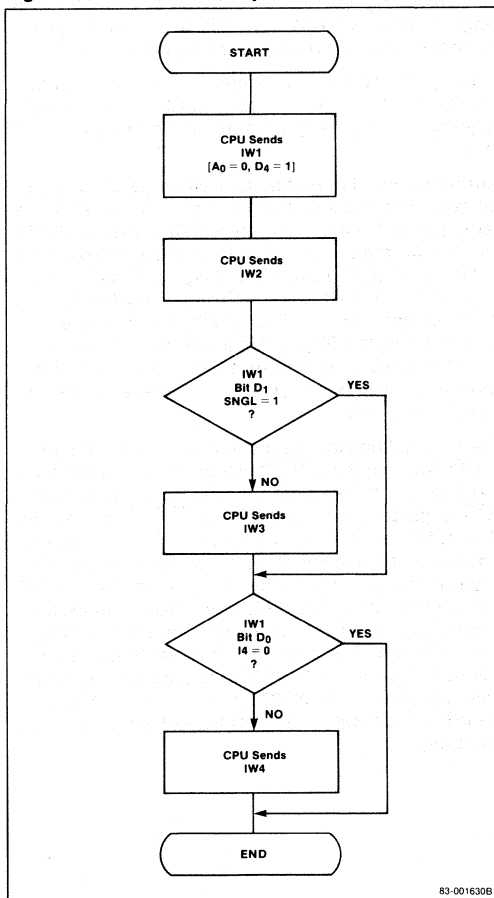
**PCW [Priority and Finish Control Word].** This word sets the FI (finish interrupt) command that defines the way that interrupts are ended, and the commands that change interrupt request priorities.

When RP (rotate priority) is set to 1, the priorities of the interrupt requests change (rotate). The priority order of the 8 INTP pins is as shown in figure 4. Setting a level as the lowest priority sets all the other levels correspondingly. For example, if INTP<sub>3</sub> is the lowest priority, INTP<sub>4</sub> will be the highest. (INTP<sub>7</sub> has the lowest priority after initialization).

SIL (specify interrupt level) is set to 1 to change the priority order or designate an interrupt level. It is used with the RP and FI bits (bits D<sub>7</sub> and D<sub>5</sub>). When SIL = 1 and RP or FI = 1, the level identified by IL<sub>2</sub>-IL<sub>0</sub> is designated as the lowest priority level. The other priorities will be set correspondingly. When used with FI = 1, it resets the ISR bit corresponding to the interrupt level IL<sub>2</sub>-IL<sub>0</sub>.

**MCW [Mode Control Word].** This word is used to set the exceptional nesting mode, to poll the μPD71059, and to read the ISR and IRR registers.

**Figure 1. Initialization Sequence**



83-001630B

Bits SR and IS/ $\bar{I}R$  are used to read the contents of the IRR and ISR registers. When SR = 0, no operation is performed. To read IRR or ISR, set A<sub>0</sub> = 0 and select the IRR or ISR register by writing to MCW. To select the IRR register, write MCW with SR = 1 and IS/ $\bar{I}R$  = 0. To select the ISR, write MCW with SR = 1 and IS/ $\bar{I}R$  = 1. The selection is retained, and MCW does not have to be rewritten to read the same register again. IRR and ISR are not masked by the IMR.

**Figure 2. Initialization Word Formats (Sheet 1 of 2)**

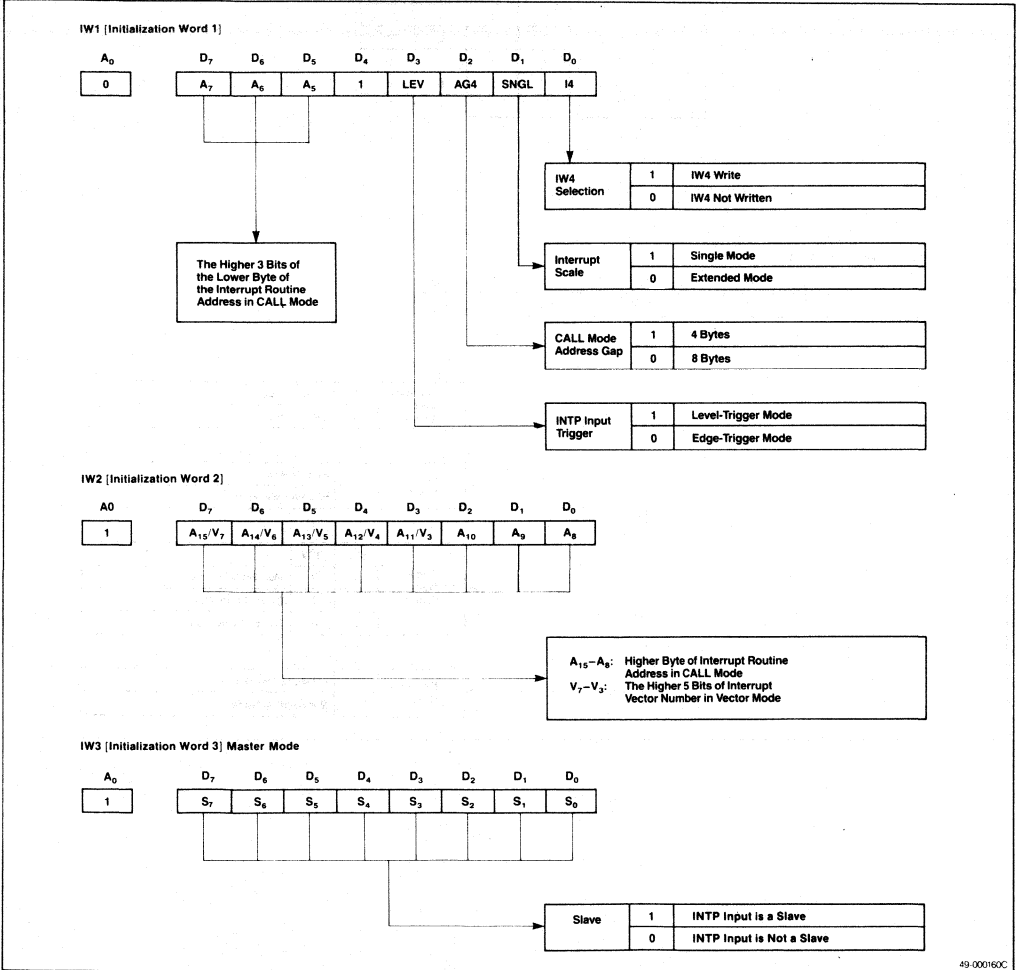




Figure 2. Initialization Word Formats (Sheet 2 of 2)

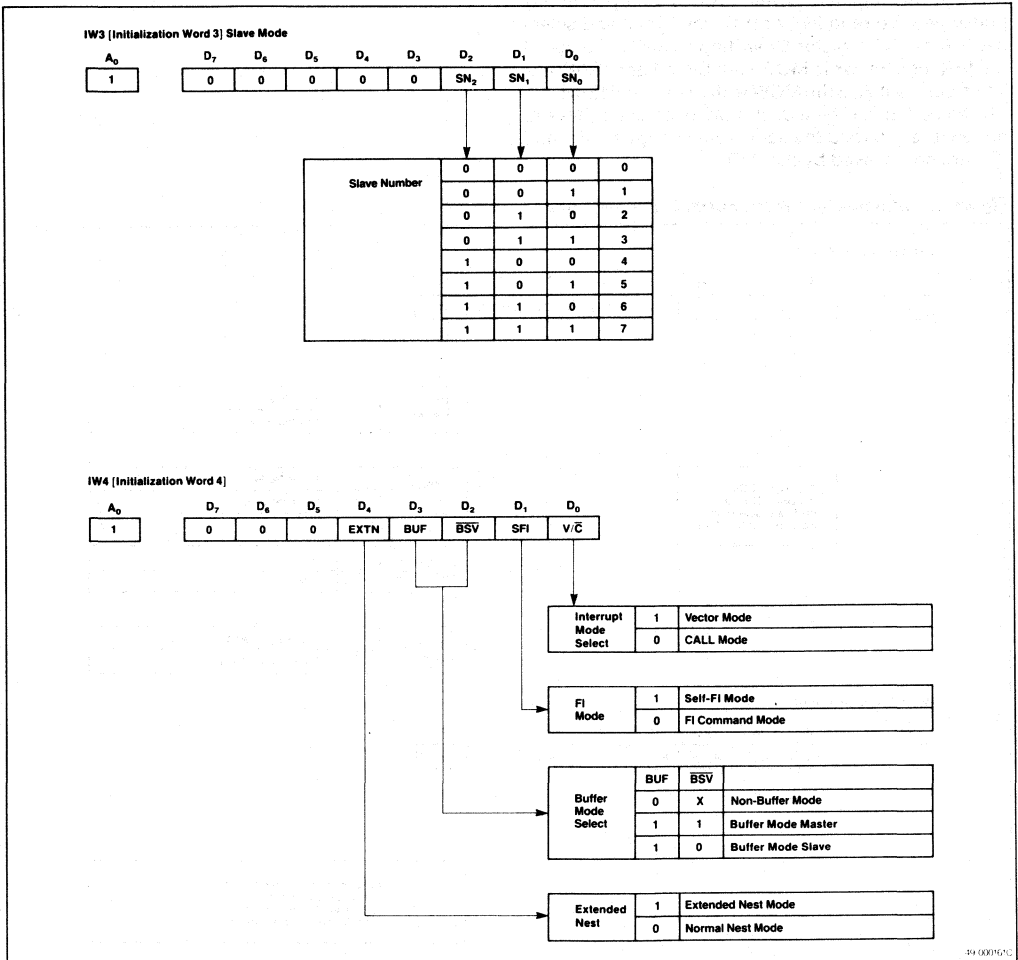
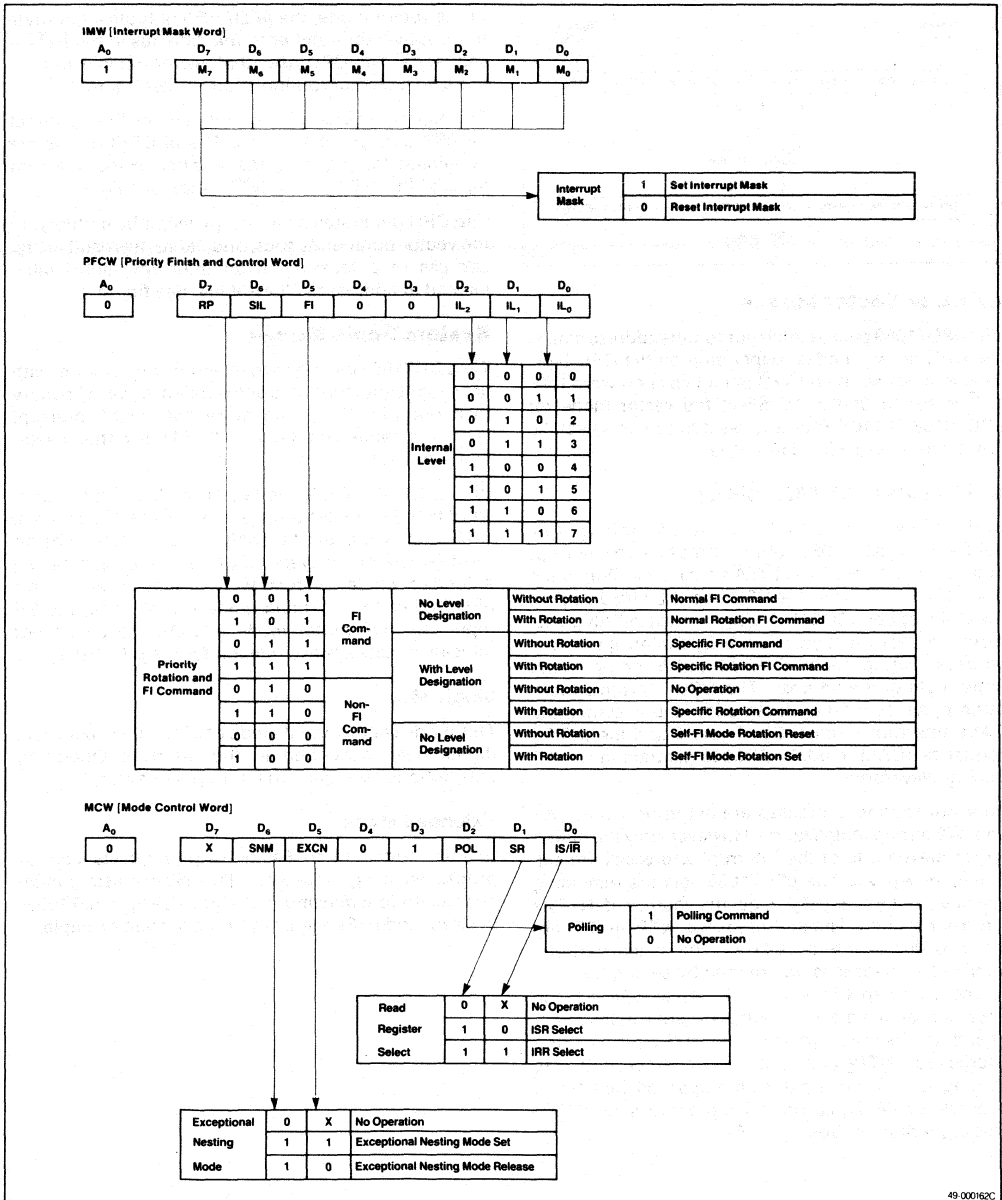
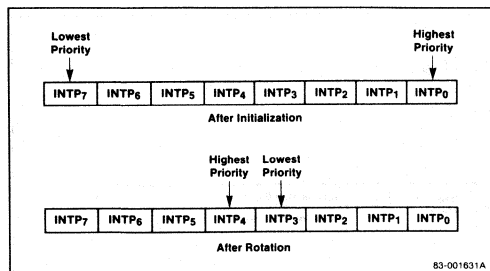


Figure 3. Command Word Format



6

Figure 4. INTP Priority Order



**CALL or Vector Modes**

The μPD71059 passes interrupt routine address data to the CPU in two modes, depending on the CPU type. This mode is set by bit V/C in initialization word IW4. V/C is set to one to select the vector mode for μPD70108/70116 CPUs, and reset to zero to select the CALL mode for μPD8085A CPUs.

**CALL Mode [μPD8085A CPUs]**

In this mode, when an interrupt is acknowledged by the CPU, the μPD71059 outputs three bytes of interrupt data to the data bus in its INTAK sequence. During the first INTAK pulse from the CPU, the μPD71059 outputs the CALL opcode 0CDH. During the next INTAK pulse, it outputs the lower byte of a two-byte interrupt routine address. During the third INTAK pulse, it outputs the upper byte of the address. The CPU interprets these three bytes as a CALL instruction and executes the CALL interrupt routine. See figure 5 and the INTAK sequence (CALL mode) μPD8085 diagram in the AC Timing Waveforms.

Interrupt routine addresses are set using words IW1 and IW2 during initialization. However, only the higher ten or eleven bits of the interrupt addresses are set, A<sub>15</sub>-A<sub>6</sub> or A<sub>15</sub>-A<sub>5</sub>. The μPD71059 sets the remaining low bits (D<sub>5</sub>-D<sub>0</sub> or D<sub>4</sub>-D<sub>0</sub>) to get the address of INTP<sub>n</sub>'s interrupt routine. The addresses for INTP<sub>1</sub>-INTP<sub>7</sub> are set in order of interrupt level. The space between interrupt addresses is determined by setting the AG4 bit (address gap 4 bytes) of IW1. When AG4 = 1, the interrupt routine starting addresses are 4 bytes apart. Therefore, the starting address for INTP<sub>n</sub> is the starting address for INTP<sub>0</sub> plus four times n. When AG4 = 0, starting addresses are eight bytes apart, so the starting address for INTP<sub>n</sub> is the starting address for INTP<sub>0</sub> plus eight times n. See figure 6.

**Vector Mode [μPD70108/70116 CPUs]**

In the vector mode, the μPD71059 outputs a one-byte interrupt vector number to the data bus in the INTAK sequence. The CPU uses that vector number to generate an interrupt routine address. See figure 7.

The higher five bits of the vector number, V<sub>7</sub>-V<sub>3</sub>, are set by IW2 during initialization. The μPD71059 sets the remaining three bits to the number 0 of the interrupt input (0 for INTP<sub>0</sub>, 1 for INTP<sub>1</sub>, etc). See figure 8.

The CPU generates an interrupt vector by multiplying the vector number by four, and using the result as the address of a location in an interrupt vector table located at addresses 000H-3FFH. See figure 9.

**System Scale Modes**

The μPD71059 can operate in either single mode, with up to eight interrupt lines or extended mode, with more than one μPD71059 and more than eight interrupt lines. In extended mode a μPD71059 is in either master or slave mode.

Bit D<sub>1</sub>, SNGL (single mode), of the first initialization word IW1 designates the scale of the interrupt system. SNGL = 1 designates that only one μPD71059 is being used (single mode system). SNGL = 0 designates an extended mode system with a master and slave μPD71059s. In the single mode (SNGL = 1), the SV input and IW4 buffer mode bits D<sub>3</sub> and D<sub>2</sub> do not indicate a master/slave relation for the μPD71059.

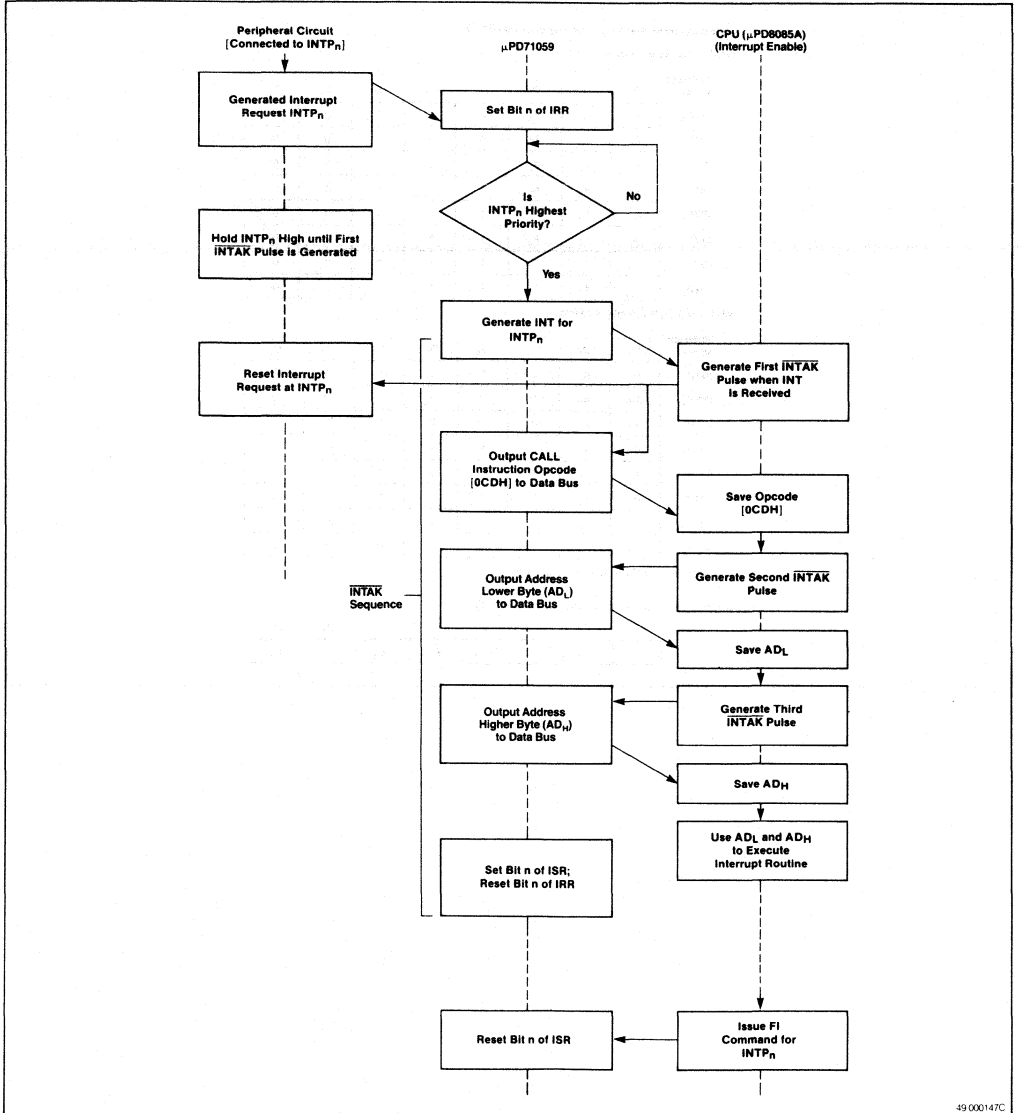
**Single Mode**

This mode is the normal mode of μPD71059 operation. It has been described in the Interrupt Operation description. See figure 10 for a system example.

**Extended Mode**

In this mode, up to 64 interrupt requests can be processed using a master (μPD71059 in master mode) connected to a maximum of eight slaves (μPD71059s in slave mode). See figure 11 for a system example.

Figure 5. CALL Mode Interrupt Sequence



49 000147C

Figure 6. CALL Mode Interrupt Address Sequence

• Address Lower Byte [AD<sub>L</sub>] During Second INTAK

AG4 = 1 (4-Byte Spacing Address)

Interrupt Level	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
INTP <sub>0</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	0	0	0	0	0
INTP <sub>1</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	0	0	1	0	0
INTP <sub>2</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	0	1	0	0	0
INTP <sub>3</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	0	1	1	0	0
INTP <sub>4</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	0	0	0	0
INTP <sub>5</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	0	1	0	0
INTP <sub>6</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	1	0	0	0
INTP <sub>7</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	1	1	0	0

AG4 = 0 (8-Byte Spacing Address)

Interrupt Level	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
INTP <sub>0</sub>	A <sub>7</sub>	A <sub>6</sub>	0	0	0	0	0	0
INTP <sub>1</sub>	A <sub>7</sub>	A <sub>6</sub>	0	0	1	0	0	0
INTP <sub>2</sub>	A <sub>7</sub>	A <sub>6</sub>	0	1	0	0	0	0
INTP <sub>3</sub>	A <sub>7</sub>	A <sub>6</sub>	0	1	1	0	0	0
INTP <sub>4</sub>	A <sub>7</sub>	A <sub>6</sub>	1	0	0	0	0	0
INTP <sub>5</sub>	A <sub>7</sub>	A <sub>6</sub>	1	0	1	0	0	0
INTP <sub>6</sub>	A <sub>7</sub>	A <sub>6</sub>	1	1	0	0	0	0
INTP <sub>7</sub>	A <sub>7</sub>	A <sub>6</sub>	1	1	1	0	0	0

Note: When AG4 = 0, bit A<sub>5</sub> is ignored.

• Address Higher Byte [AD<sub>H</sub>] During Third INTAK

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>

**Figure 7. Vector Mode Interrupt Sequence**

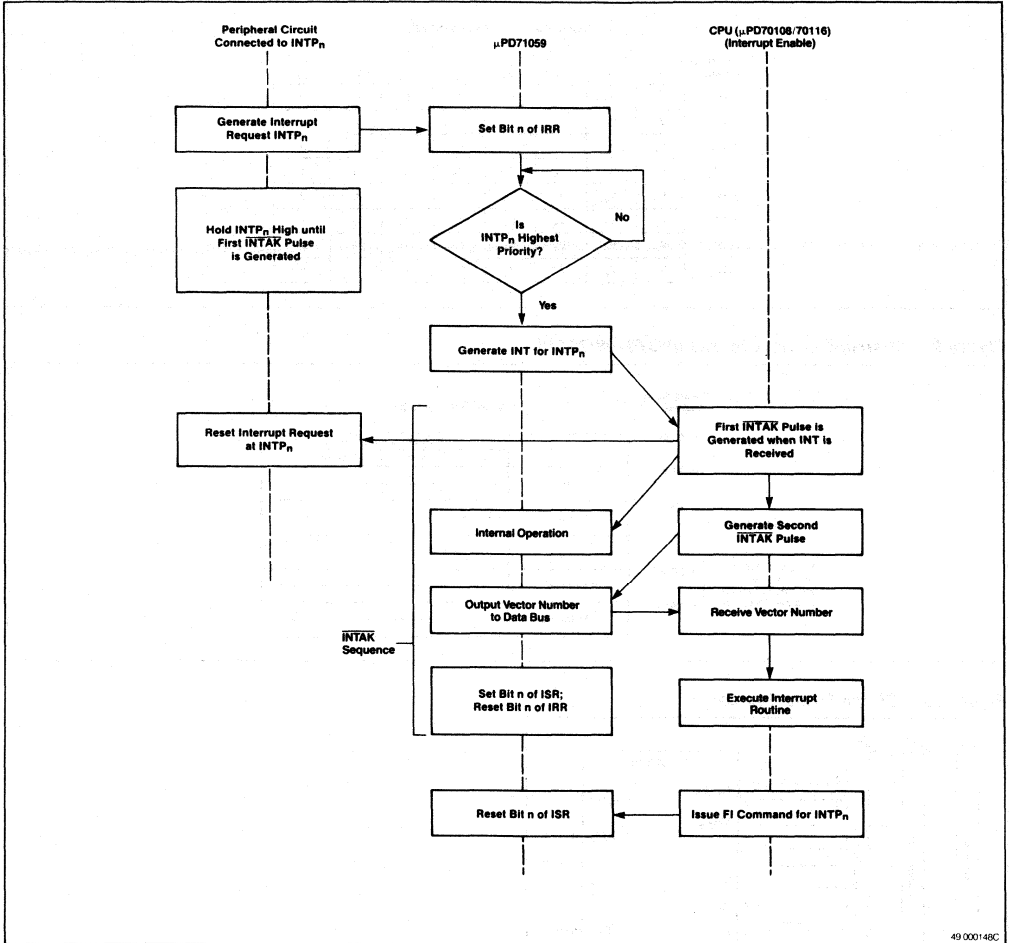


Figure 8. Vector Numbers Output in Vector Mode

Output During the Second INTAK

Interrupt Levels	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
INTP <sub>0</sub>	V <sub>7</sub>	V <sub>6</sub>	V <sub>5</sub>	V <sub>4</sub>	V <sub>3</sub>	0	0	0
INTP <sub>1</sub>	V <sub>7</sub>	V <sub>6</sub>	V <sub>5</sub>	V <sub>4</sub>	V <sub>3</sub>	0	0	1
INTP <sub>2</sub>	V <sub>7</sub>	V <sub>6</sub>	V <sub>5</sub>	V <sub>4</sub>	V <sub>3</sub>	0	1	0
INTP <sub>3</sub>	V <sub>7</sub>	V <sub>6</sub>	V <sub>5</sub>	V <sub>4</sub>	V <sub>3</sub>	0	1	1
INTP <sub>4</sub>	V <sub>7</sub>	V <sub>6</sub>	V <sub>5</sub>	V <sub>4</sub>	V <sub>3</sub>	1	0	0
INTP <sub>5</sub>	V <sub>7</sub>	V <sub>6</sub>	V <sub>5</sub>	V <sub>4</sub>	V <sub>3</sub>	1	0	1
INTP <sub>6</sub>	V <sub>7</sub>	V <sub>6</sub>	V <sub>5</sub>	V <sub>4</sub>	V <sub>3</sub>	1	1	0
INTP <sub>7</sub>	V <sub>7</sub>	V <sub>6</sub>	V <sub>5</sub>	V <sub>4</sub>	V <sub>3</sub>	1	1	1

83-001633B

Figure 9. Interrupt Vectors for the μPD70108/70116

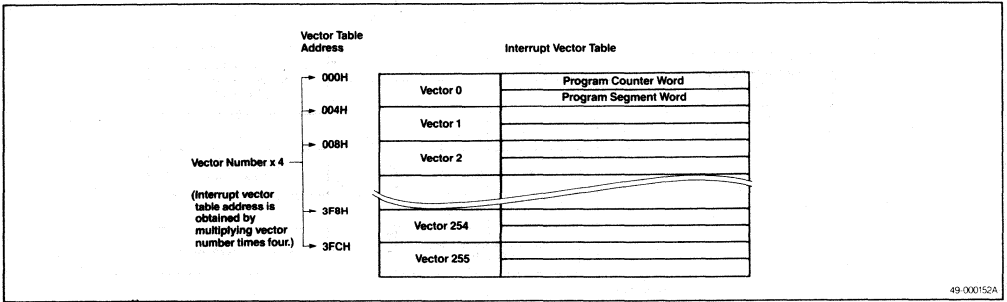


Figure 10. Single Mode System

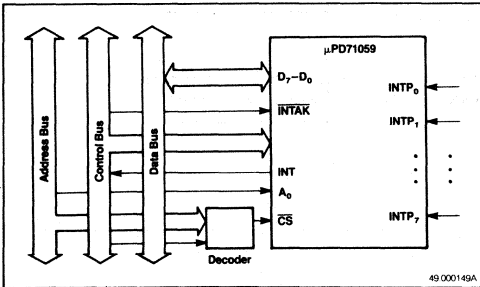
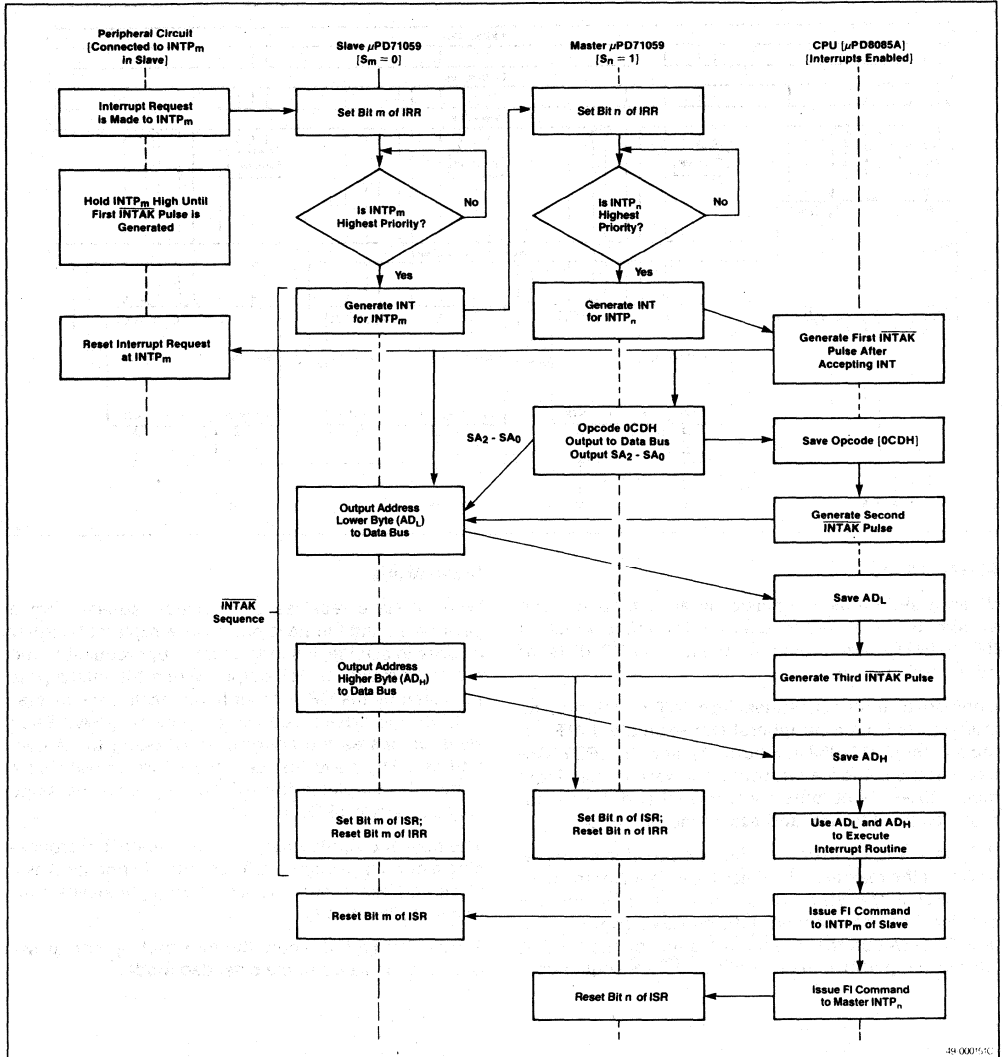




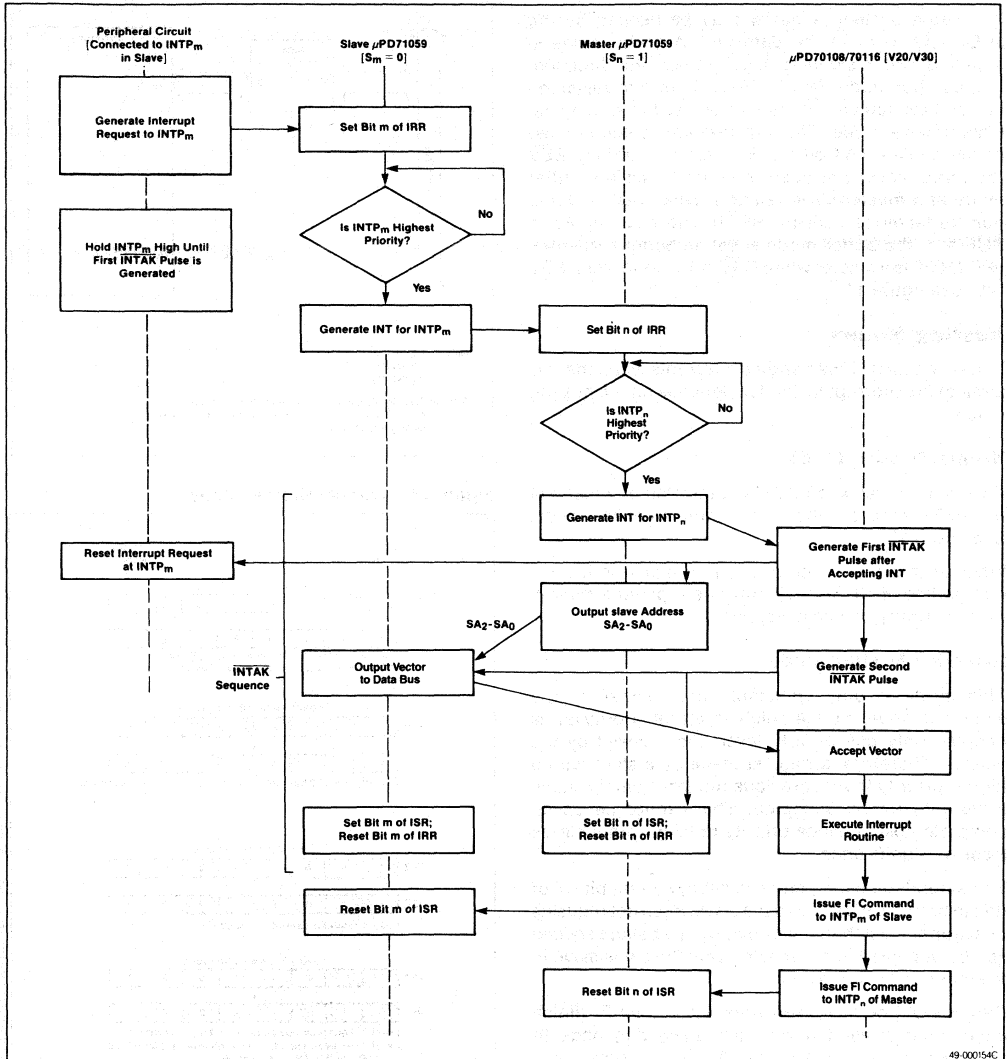


Figure 12. Interrupt from Slave (CALL Mode)



94-00079-01

Figure 13. Interrupt from Slave (Vector Mode)



49-000154C



## Exceptional Nesting Mode

A μPD71059 in the normal or extended nesting mode cannot accept interrupts of a lower priority than the interrupts in service. Sometimes, however, it is desirable that requests with lower priority be accepted while higher-priority interrupts are being serviced. Setting the exceptional nesting mode allows this. After releasing the exceptional mode, the previous mode is resumed.

The exceptional nesting mode is controlled by the SNM (set nesting mode) and EXCN (exceptional nesting mode) bits (D<sub>6</sub> and D<sub>5</sub>) of MCW. They set and release the exceptional nesting mode. The mode doesn't change when SNM = 0. Exceptional nesting is set if SNM and EXCN = 1 and released when SNM = 1 and EXCN = 0.

Setting a bit in the IMW in the exceptional nesting mode, inhibits interrupts of that level and allows unmasked interrupts to all other levels, higher or lower priority.

The procedure for setting the exceptional nesting (EN) mode is as follows:

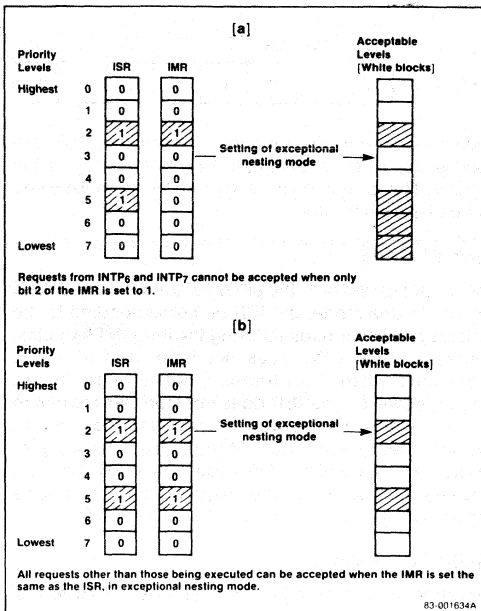
- (1) Read the ISR.
- (2) Write the ISR data to the IMR.
- (3) Set the exceptional nesting mode.

In this way, all interrupt requests not currently in service will be enabled.

Figure 16 (a) shows what happens if IMR is not set to ISR. When the exceptional nesting is set, bit 2 of ISR will be ignored, and bit 5 will be serviced. Servicing bit 5 will mask the lower priority interrupts 6 and 7. When the ISR is set equal to the IMR as in (b), all interrupts except 2 and 5 can be serviced when the exceptional nesting mode is set.

Issuing an FI command to a level masked by the exceptional nesting mode requires caution. Since the ISR bit is masked, the normal FI command will not work. For this reason, a specific FI command specifying the ISR bit must be issued. After the exceptional mode is released, the normal FI command may be used.

Figure 16. Exceptional Nesting Mode



## Finishing Interrupts (FI) and Changing the Priority Levels

The priority and finish control word (PFCW) issues FI commands and changes interrupt priorities.

### Normal FI Command

$$\text{PFCW} = \begin{matrix} D_7 & D_6 & D_5 & D_4 & D_3 & D_2 & D_1 & D_0 \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & X & X & X \end{bmatrix} \end{matrix}$$

When a normal FI command is issued, the μPD71059 resets the ISR bit corresponding to the highest priority level selected from the interrupts in service. This operation assumes that the interrupt accepted last has ended.

When an interrupt routine changes the priority level or the exceptional nesting mode is set, this command will not operate correctly because the highest priority interrupt is not necessarily the last interrupt in service.

### Specific FI Command

	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
PFCW =	0	1	1	0	0	IL <sub>2</sub>	IL <sub>1</sub>	IL <sub>0</sub>

When the specific FI command is issued, the μPD71059 resets the ISR bit designated by bits IL<sub>2</sub>-IL<sub>0</sub> of the PFCW. This command is used when the normal nesting mode isn't being used.

### Self-FI Mode

When SFI of IW4 = 1, the μPD71059 is set to the self-FI mode. In this mode, the ISR bit corresponding to the interrupt is set and reset during the third INTAK pulse. Therefore, the CPU does not have to issue an FI command when the interrupt routine ends. In this mode, however, the ISR does not store the routine in service. Unless interrupts are disabled by the interrupt routine, newly generated interrupt requests are generated without priority limitation by the ISR. This can cause a stack overflow when frequent interrupt requests occur, or when the interrupt is level triggered.

### Self-FI Rotation

Rotation of interrupt priorities can be added to the self-FI mode. In this case, the corresponding interrupt is set to the lowest priority level when a bit is reset in the ISR at the end of the INTAK sequence.

Self-FI Rotation Set:

	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
PFCW =	1	0	0	0	0	X	X	X

Self-FI Rotation Reset:

	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
PFCW =	0	0	0	0	0	X	X	X

### Normal Rotation FI Command

	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
PFCW =	1	0	1	0	0	X	X	X

When the normal rotation FI command is issued, the μPD71059 resets the ISR bit corresponding to the highest priority level selected from the interrupts in service, then rotates the priority levels so that the interrupt just completed has the lowest priority.

### Specific Rotation FI Command

	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
PFCW =	1	1	1	0	0	IL <sub>2</sub>	IL <sub>1</sub>	IL <sub>0</sub>

When the specific rotation FI command is issued, the μPD71059 resets the ISR bit designated by bits IL<sub>2</sub>-IL<sub>0</sub> of the PFCW and rotates the interrupt priorities so that the interrupt just reset becomes the lowest priority. This change in priority levels is different from the normal nesting mode, therefore, it is the user's responsibility to manage nesting.

### Specific Rotation Command

	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
PFCW =	1	1	0	0	0	IL <sub>2</sub>	IL <sub>1</sub>	IL <sub>0</sub>

When the specific rotation command is issued, the μPD71059 sets the interrupt priority specified by IL<sub>2</sub>-IL<sub>0</sub> to the lowest priority. In this case also, the user must manage nesting.

### Triggering Mode

Bit D<sub>3</sub> of the first initialization word, IW1, is LEV (level-trigger mode bit). LEV sets the trigger mode of the INTP inputs. The level-trigger mode is set when LEV = 1. The rising-edge-triggered mode is set when LEV = 0.

### Edge-Trigger Mode

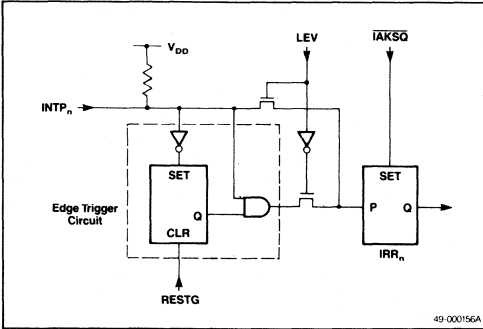
In the edge-trigger mode, an interrupt is detected by the rising edge of the signal on an INTP input. Although an IRR bit goes high when INTP is high, the IRR bit is not latched until the CPU returns an INTAK pulse. Therefore, the INTP input should be maintained high until INTAK is received. This filters out noise spikes on the INT lines. To send the next interrupt request, temporarily lower the INTP input, then raise it.

## Level-Trigger Mode

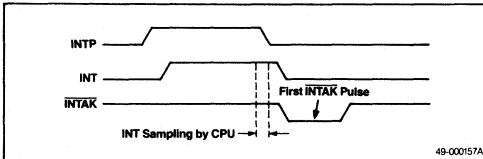
In the level-trigger mode, an IRR bit is set by the INTP input being at a high level. As in the edge-trigger mode, the INTP must be maintained high until the INTAK is received. Interrupts are requested as long as the INTP input remains high. Care should be taken so as not to cause a stack overflow in the CPU. See figure 17.

**Note:** The μPD71059 operates as if the INTP<sub>7</sub> interrupt had occurred if the INTAK pulse is sent to the μPD71059 by the CPU when the μPD71059 INT output level is low. Bit 7 of ISR is not set. Accordingly, if it is expected that this will occur, the INTP<sub>7</sub> interrupt should be reserved for servicing incomplete interrupts. The FI should not be issued for incomplete interrupts. See figure 18.

**Figure 17. INTP Input**



**Figure 18. Incomplete Interrupt Request**

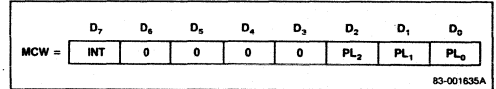


## Polling Operation

When polling, the CPU should disable its INT input. Next, it issues a polling command to the μPD71059 using MCW with POL = 1. This command sets the μPD71059 in polling mode until the CPU reads one of the μPD71059's registers.

When the CPU performs a read operation with A<sub>0</sub> = 0 in the polling mode, polling data as shown in figure 19 is read instead of ISR or IRR. The μPD71059 then ends the polling mode.

**Figure 19. Polling Data**



The INT bit has the same meaning as the INT pin. When it is set to 1, it means that the μPD71059 has accepted an INTP input.

The PL<sub>2</sub>-PL<sub>0</sub> (permitted level) bits show which INTP input requested an interrupt when INT = 1.

If INT in the polling data is 1, the μPD71059 sets the ISR bit corresponding to the interrupt level shown by bits PL<sub>2</sub>-PL<sub>0</sub> of the polling data and considers that interrupt as being executed. The CPU then processes the interrupt accordingly, based on the polling data read. An FI command should be issued when this processing ends.

**Note:** When a read is performed with A<sub>0</sub> = 1 after the polling command is sent to the μPD71059, the IMR will be read instead of polling data. However, when the polling command is sent, the μPD71059 operates in the same manner when A<sub>0</sub> = 0 as it does when A<sub>0</sub> = 1. This means that although A<sub>0</sub> was set to 1, the μPD71059 will send the contents of the IMR, but it will also set an ISR bit just as it would if A<sub>0</sub> had been set to zero. This may disturb the nesting. Therefore, performing a read operation with A<sub>0</sub> = 1 immediately after sending the polling command should be avoided.



## Description

The μPD71071 is a high-speed, high-performance direct memory access (DMA) controller that provides high-speed data transfers between peripheral devices and memory. A programmable bus width allows bidirectional data transfer in both 8- and 16-bit systems. In addition, the μPD71071 uses CMOS technology to reduce power consumption.

The μPD71071 can perform a variety of transfer functions including byte/word, memory-to-memory, and transfers between memory and I/O. The μPD71071 also utilizes single, demand, and block mode transfers; release and bus hold modes; and normal and compressed timing.

## Features

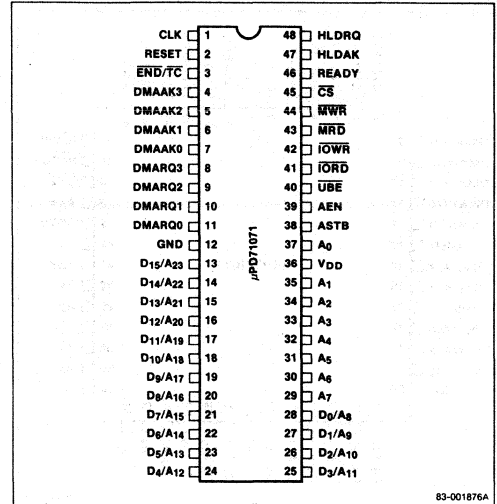
- Four independent DMA channels
- 16M-byte addressing
- 64K-byte/word transfer count
- 8- or 16-bit programmable data bus width
- Enable/disable of individual DMA requests
- Software DMA requests
- Enable/disable of autoinitialization
- Address increment/decrement
- Fixed/rotational DMA channel priority
- Terminal count output signal
- Forced transfer termination input
- Cascade capability
- Programmable DMA request and acknowledge signal polarities
- High performance: transfers up to
  - 5 MBytes/sec in normal mode
  - 6.6 MBytes/sec in compressed mode
- 10 MHz operation
- μPD70108/70116/70208/70216 compatible
- CMOS technology
- Low-power standby mode
- Single power supply, 5 V ±10%
- Industrial temperature range, -40 to +85°C

## Ordering Information

Part Number	Package Type	Max. Frequency of Operation
μPD71071D	48-pin ceramic DIP	8 MHz
μPD71071C	48-pin plastic DIP	10 MHz
μPD71071GC	52-pin plastic flat pack	10 MHz
μPD71071L	52-pin PLCC	10 MHz

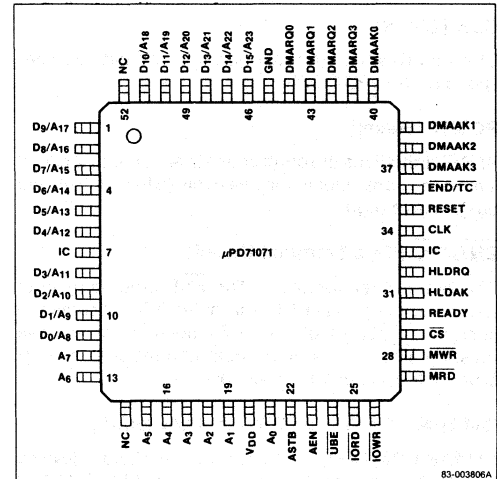
## Pin Configurations

### 48-Pin Plastic DIP, Ceramic DIP



83-001876A

### 52-Pin Flat Pack



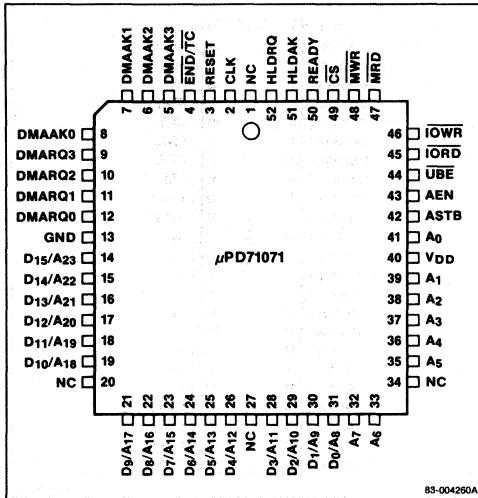
83-003806A



## μPD71071

### Pin Configurations (cont)

#### 52-Pin Plastic Leaded Chip Carrier (PLCC)



### Pin Functions

#### CLK [Clock]

CLK controls the internal operation and data transfer speed of the μPD71071.

#### RESET [Reset]

RESET initializes the controller's internal registers and leaves the controller in the idle cycle (CPU controls the bus). Active high.

#### END/TC [End/Terminal Count]

This is a bidirectional pin. The  $\overline{\text{END}}$  input is used to terminate the current DMA transfer.  $\overline{\text{TC}}$  indicates the designated cycles of the DMA count transfer have finished.  $\overline{\text{END/TC}}$  is open drain and requires an external pull-up resistor. Active low.

#### DMAA3-DMAA0 [DMA Acknowledge]

DMAA3-DMAA0 indicates to peripheral devices that DMA service has been granted. DMAA3-DMAA0 respond respectively to DMA channels 3-0 and the polarities are user programmable.

### Pin Identification

Symbol	Function
A <sub>23</sub> -A <sub>8</sub> / D <sub>15</sub> -D <sub>0</sub>	Bidirectional address/data bus
IC	Internally connected; leave open
A <sub>7</sub> -A <sub>4</sub>	Address bus output
NC	Not connected
A <sub>3</sub> -A <sub>0</sub>	Bidirectional address bus
V <sub>DD</sub>	Power supply
ASTB	Address strobe output
AEN	Address enable output
$\overline{\text{UBE}}$	Upper byte enable input/output
$\overline{\text{IORD}}$	I/O read input/output
$\overline{\text{IOWR}}$	I/O write input/output
$\overline{\text{MRD}}$	Memory read output
$\overline{\text{MWR}}$	Memory write output
CS	Chip select input
READY	Ready input
HLDK	Hold acknowledge input
HLDRO	Hold request output
CLK	Clock input
RESET	Reset input
END/TC	End DMA transfer input/terminal count output
DMAA3- DMAA0	DMA acknowledge output
DMARQ3- DMARQ0	DMA request input
GND	Ground

#### DMARQ3-DMARQ0 [DMA Request]

DMARQ3-DMARQ0 accept DMA service requests from peripheral devices. DMARQ3-DMARQ0 respond respectively to DMA channels 3-0 and the polarities are user programmable. DMARQ must remain asserted until DMAA3 is asserted.

#### GND [Ground]

GND connects to the power supply ground terminal.

#### A<sub>23</sub>-A<sub>8</sub>/D<sub>15</sub>-D<sub>0</sub> [Address/Data Bus]

A<sub>23</sub>-A<sub>8</sub>/D<sub>15</sub>-D<sub>0</sub> function as a 16-bit, multiplexed address/data bus when the μPD71071 is in the 16-bit data mode. In the 8-bit data mode, A<sub>23</sub>-A<sub>16</sub> (pins 13-20) become address bits only and A<sub>15</sub>-A<sub>8</sub>/D<sub>7</sub>-D<sub>0</sub> (pins 21-28) remain an 8-bit multiplexed address/data bus. A<sub>23</sub>-A<sub>8</sub>/D<sub>15</sub>-D<sub>0</sub> are three-state.

## A7-A4, A3-A0 [Address Bus]

A7-A4, A3-A0 function as the lower eight bits of the address bus. A7-A4 output memory addresses during the DMA cycle and become high impedance in the idle cycle. A3-A0 function as the lower four bits of the address bus. In the idle cycle, A3-A0 become address inputs to select internal registers for the CPU to read or write. In the DMA cycle, A3-A0 output memory addresses.

## V<sub>DD</sub> [Power Supply]

V<sub>DD</sub> connects to the +5-V power supply.

## ASTB [Address Strobe]

ASTB latches address A<sub>23</sub>-A<sub>8</sub> (16-bit mode)/A<sub>15</sub>-A<sub>8</sub> (8-bit mode) from the address/data bus into an external address latch at the falling edge of ASTB during a DMA cycle. Active high.

## AEN [Address Enable]

AEN enables the output of an external latch that holds DMA addresses. AEN becomes high during the DMA cycle.

## UB $\bar{E}$ [Upper Byte Enable]

UB $\bar{E}$  indicates the upper byte of the data bus is valid during 16-bit mode. In the idle cycle during data transfer, the μPD71071 acknowledges data on D<sub>15</sub>-D<sub>8</sub> when UB $\bar{E}$  is asserted. During a DMA cycle, UB $\bar{E}$  goes low to signify the presence of valid data on D<sub>15</sub>-D<sub>8</sub>. UB $\bar{E}$  has no meaning in 8-bit mode and becomes high impedance in the idle cycle and high level in the DMA cycle. Three-state, active low.

## I $\bar{O}R\bar{D}$ [I/O Read]

In the idle cycle, I $\bar{O}R\bar{D}$  inputs a read signal from the CPU. In the DMA cycle, I $\bar{O}R\bar{D}$  outputs a read signal to an I/O device. Three-state, active low.

## I $\bar{O}W\bar{R}$ [I/O Write]

In the idle cycle, I $\bar{O}W\bar{R}$  inputs a write signal from the CPU. In the DMA cycle, I $\bar{O}W\bar{R}$  outputs a write signal to an I/O device. Three-state, active low.

## MRD [Memory Read]

During the DMA cycle, MRD outputs a read signal to memory. MRD is high impedance during the idle cycle. Three-state, active low.

## MWR [Memory Write]

During the DMA cycle, MWR outputs a write signal to memory. MWR is high impedance during the idle cycle. Three-state, active low.

## $\bar{C}\bar{S}$ [Chip Select]

During the idle cycle,  $\bar{C}\bar{S}$  selects the μPD71071 as an I/O device. Active low.

## READY [Ready]

During a DMA operation, READY indicates that a data transfer for one cycle has been completed and may be terminated. To meet the requirements of low-speed I/O devices or memory, READY may be negated to insert wait states to extend the bus cycle until READY is again asserted.

## HLD $\bar{A}K$ [Hold Acknowledge]

When active, HLD $\bar{A}K$  indicates that the CPU has granted the μPD71071 the use of the system bus. Active high.

## HLD $\bar{R}Q$ [Hold Request]

HLD $\bar{R}Q$  outputs a bus hold request to the CPU. Active high.

## Block Diagram Description

The μPD71071 has the following functional units.

- Bus control unit
- DMA control unit
- Address registers
- Address incrementer/decrementer
- Count registers
- Count decremter
- Control registers

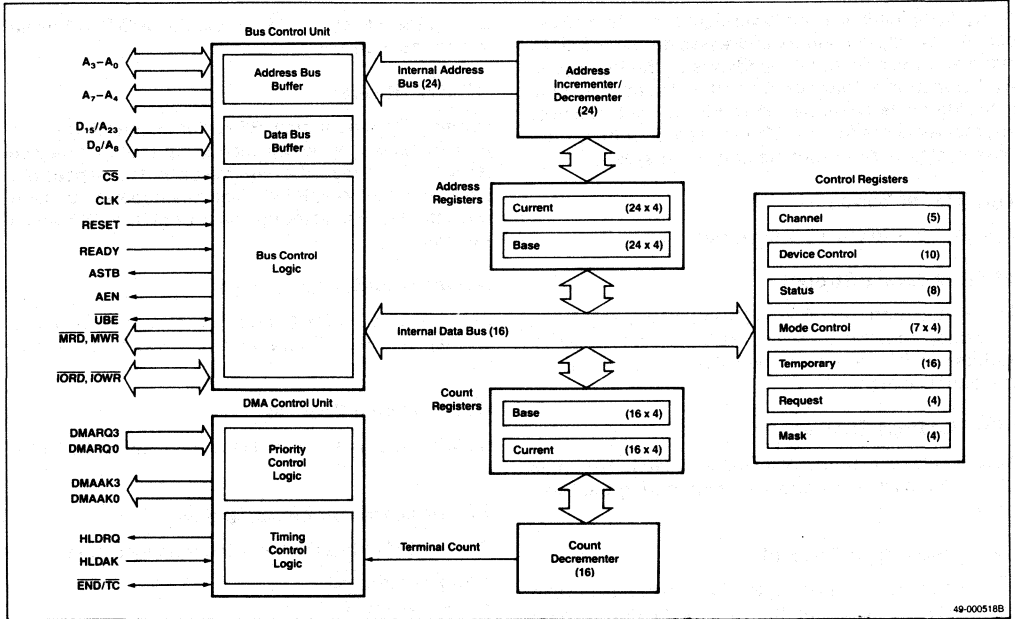
## Bus Control Unit

The bus control unit consists of the address and data buffers, and bus control logic. The bus control unit generates and receives signals that control addresses and data on the internal address and data buses.

## DMA Control Unit

The DMA control unit contains the priority and timing control logic. The priority control logic determines the priority level of DMA requests and arbitrates the use of the bus in accordance with this priority level. The DMA control unit also provides internal timing and controls DMA operations.

Block Diagram



49-000518B

Address Registers

Each of the four DMA channels has one 24-bit base address register and one 24-bit current address register. The base address register holds a value determined by the CPU and transfers this value to the current address register during autoinitialization (address and count are automatically initialized). The channel's current address register is incremented/decremented for each transfer and always contains the address of the data to be transferred next.

Address Incrementer/Decrementer

The address incrementer/decrementer updates the contents of the current address register whenever a DMA transfer completes.

Count Registers

Each of the four DMA channels has one 16-bit base count register and one 16-bit current count register. The base count register holds a value written by the CPU and transfers the value to the current count register during autoinitialization. A channel's current count register is decremented for each transfer and

generates a terminal count when the count register is decremented to FFFFH.

Note: The number of DMA transfer cycles is actually the value of the current count register + 1. Therefore, when programming the count register, specify the number of DMA transfers minus one.

Count Decrementer

The count decrementer decrements the contents of the current count register by one when each DMA transfer cycle ends.

Control Registers

The μPD71071 contains the following control registers.

- Channel
- Device
- Status
- Mode
- Temporary
- Request
- Mask

These registers control bus mode, pin active levels, DMA operation mode, mask bits, and other μPD71071 operating functions.

## Absolute Maximum Ratings

Power supply voltage, $V_{DD}$	-0.5 to +7.0 V
Input voltage, $V_I$	-0.5 to $V_{DD} + 0.3$ V
Output voltage, $V_O$	-0.5 to $V_{DD} + 0.3$ V
Operating temperature, $T_{OPT}$	-40 to +85°C
Storage temperature, $T_{STG}$ (D/G/L)	-65 to +150°C
Storage temperature, $T_{STG}$ (C)	-40 to +125°C

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## Capacitance

$T_A = 25^\circ\text{C}$

Parameter	Symbol	Limits		Unit	Test Conditions
		Typ	Max		
Output capacitance	$C_O$	4	8	pF	$f_c = 1.0$ MHz Unmeasured pins returned to 0 V
Input capacitance	$C_I$	8	15	pF	
I/O capacitance	$C_{I/O}$	10	18	pF	

## DC Characteristics

$T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD} = 5$  V  $\pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input high voltage	$V_{IH}$	3.3		$V_{DD} + 0.3$	V	CLK input pin
		2.2		$V_{DD} + 0.3$	V	Other inputs
Input low voltage	$V_{IL}$	-0.5		0.8	V	
Output high voltage	$V_{OH}$	$0.7 V_{DD}$			V	$I_{OH} = -400 \mu\text{A}$
Output low voltage	$V_{OL}$		0.4		V	$I_{OL} = 2.5$ mA; 4.5 mA (TC)
Input leakage current	$I_{LI}$			$\pm 10$	$\mu\text{A}$	$0 \text{ V} \leq V_I \leq V_{DD}$
Output leakage current	$I_{LO}$			$\pm 10$	$\mu\text{A}$	$0 \text{ V} \leq V_O \leq V_{DD}$
Supply current (dynamic)	$I_{DD1}$		15	30	mA	
Supply current (static)	$I_{DD2}$		10		$\mu\text{A}$	CLK stopped Inputs stable at 0.1 V or $V_{DD} - 0.1$ V outputs open

## AC Characteristics, μPD71071D (8 MHz)

$T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD} = 5$  V  $\pm 10\%$

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>DMA Mode</b>					
Clock cycle	$t_{CYK}$	125		ns	
Clock pulse width high	$t_{KHH}$	44		ns	
Clock pulse width low	$t_{KLL}$	55		ns	
Clock rise time	$t_{KR}$		10	ns	1.5 V $\rightarrow$ 3.0 V
Clock fall time	$t_{KF}$		10	ns	3.0 V $\rightarrow$ 1.5 V
Input rise time	$t_{IR}$		20	ns	
Input fall time	$t_{IF}$		12	ns	
Output rise time	$t_{OR}$		20	ns	
Output fall time	$t_{OF}$		12	ns	
DMARQ setup time to CLK high	$t_{SDQ}$	35		ns	SI, S0, S3, SW, S4w
HLDRQ high delay from CLK low	$t_{DHQH}$		100	ns	SI, S4w
HLDRQ low delay from CLK low	$t_{DHQL}$		100	ns	SI, S0, S4w
HLDRQ low level period	$t_{HOHQL}$	$2t_{CYK} - 50$		ns	S4w
HLDAK high setup time to CLK low	$t_{SHA}$	35		ns	S0, S4, S4w
AEN high delay from CLK low	$t_{DAEH}$		90	ns	S1, S2
AEN low delay time from CLK low	$t_{DAEL}$		90	ns	SI, S4w
ASTB high delay time from CLK low	$t_{DSTH}$		70	ns	S1
ASTB low delay time from CLK high	$t_{DSTL}$		70	ns	S1
ASTB high level period	$t_{STSTH}$	$t_{KLL} - 15$		ns	
ADR/UBE/RD/WR (1) active delay from CLK low	$t_{DA}$		100	ns	S1, S2
ADR/UBE/RD/WR float time from CLK low	$t_{FA}$		70	ns	SI, S4w
ADR setup time to ASTB low	$t_{SAST}$	$t_{KLL} - 50$		ns	
ADR hold time from ASTB low	$t_{HSTA}$	$t_{KHH} - 20$		ns	

AC Characteristics, μPD71071D (cont)

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>DMA Mode (cont)</b>					
ADR/UBE off delay time from CLK low	t <sub>DAF</sub>	0	70	ns	S1, S2
RD low delay time from ADR float	t <sub>DAR</sub>	-10		ns	
Input data delay time from MRD low	t <sub>DMRID</sub>		2t <sub>CYK</sub> - 100	ns	S12
Input data hold time from MRD high	t <sub>HMRID</sub>	0		ns	S14
Output data delay time from CLK low	t <sub>DOD</sub>	10	100	ns	S22
Output data hold time from CLK high	t <sub>HOD</sub>	10		ns	S24
Output data hold time from MWR high	t <sub>HMWOD</sub>	t <sub>KKL</sub> - 50		ns	
RD low delay time from CLK high	t <sub>DKHR</sub>		70	ns	S2 compressed timing
RD low level period	t <sub>RRL1</sub>	2t <sub>CYK</sub> - 50		ns	Normal timing
	t <sub>RRL2</sub>	t <sub>CYK</sub> + t <sub>KKH</sub> - 50		ns	Compressed timing
RD high delay time from CLK low	t <sub>DRH</sub>	15	100	ns	S4
ADR delay time from RD high	t <sub>DRA</sub>	t <sub>CYK</sub> - 40		ns	
WR low delay time from CLK low	t <sub>DWL1</sub>	10	70	ns	S3 normal write
WR low delay time from CLK low	t <sub>DWL2</sub>	10	70	ns	S2 extended write, normal timing
WR low delay time from CLK high	t <sub>DWL3</sub>	10	70	ns	S2 extended write, compressed timing
WR low level period	t <sub>WWL1</sub>	t <sub>CYK</sub> - 50		ns	Normal write
	t <sub>WWL2</sub>	2t <sub>CYK</sub> - 50		ns	Extended write, normal timing
	t <sub>WWL3</sub>	t <sub>CYK</sub> + t <sub>KKH</sub> - 50		ns	Extended write, compressed timing
WR high delay time from CLK low	t <sub>DWH</sub>	10	80	ns	S4

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
RD low delay time from CLK low	t <sub>DKLR</sub>		70	ns	S2 normal timing
RD, WR low delay time from DMAAK active	t <sub>DDARW</sub>	0		ns	S1, S2
RD high delay time from WR high	t <sub>DWHRH</sub>	5		ns	
DMAAK delay time from CLK high	t <sub>DKHDA</sub>	10	70	ns	S1 I/O memory timing
DMAAK delay time from CLK low	t <sub>DKLDA</sub>	10	115	ns	S1 cascade mode
DMAAK inactive delay time from CLK high	t <sub>DDAI1</sub>	10	70	ns	S4
DMAAK inactive delay time from HLDK low	t <sub>DDAI2</sub>	5	t <sub>KKL</sub> + 80	ns	S4 cascade mode, HLDK low in S4
	t <sub>DDAI3</sub>	t <sub>KKL</sub> + 80	4 t <sub>CYK</sub> + 80	ns	S4 cascade mode, HLDK low except in S4
DMAAK active level period	t <sub>DADA</sub>	2 t <sub>CYK</sub>		ns	Cascade mode
TC low delay time from CLK high	t <sub>DTCL</sub>		100	ns	S3
TC off delay time from CLK high	t <sub>DTCF</sub>		40	ns	S4
TC high delay time from CLK high	t <sub>DTCH</sub>		t <sub>KKH</sub> + t <sub>CYK</sub> - 10	ns	0 to 2.2 V (2)
TC low level period	t <sub>TCTCL</sub>	t <sub>CYK</sub> - 15		ns	
END low setup time to CLK high	t <sub>SED</sub>	35		ns	S2
END low level period	t <sub>EDFDL</sub>	100		ns	
READY setup time to CLK high	t <sub>SRY</sub>	35		ns	S3, SW
READY hold time from CLK high	t <sub>HRY</sub>	20		ns	S3, SW

**Note:**

- (1) RD/WR refers to IORD or MRD and IOWR or MWR, respectively.
- (2) For END/TC, output load capacitance = 75 pF maximum. To meet the t<sub>DTCH</sub> parameter use a 2.2-kΩ pull-up resistor with a load capacitance of 75 pF. For other than END/TC, output load capacitance = 100 pF maximum.

## AC Characteristics, μPD71071D (cont)

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Programming Mode and RESET</b>					
IOWR low level period	t <sub>WIWL</sub>	100		ns	
CS low setup time to IOWR high	t <sub>SCSIW</sub>	100		ns	
CS hold time from IOWR high	t <sub>HIWCS</sub>	0		ns	
ADR/ <u>BE</u> setup time to IOWR high	t <sub>SAIW</sub>	100		ns	
ADR/ <u>BE</u> hold time from IOWR high	t <sub>HIWA</sub>	0		ns	
Input data setup time to IOWR high	t <sub>SIDIW</sub>	100		ns	
Input data hold time from IOWR high	t <sub>HIWID</sub>	0		ns	
IORD low level period	t <sub>IIRIL</sub>	150		ns	
ADR/ <u>CS</u> setup time to IORD low	t <sub>SAIR</sub>	35		ns	
ADR/ <u>CS</u> hold time from IORD high	t <sub>HIRA</sub>	0		ns	
Output data delay time from IORD low	t <sub>DIROD</sub>		120	ns	
Output data float time from IORD high	t <sub>FIROD</sub>		100	ns	
RESET high level period	t <sub>RESET</sub>	2t <sub>CYK</sub>		ns	
V <sub>DD</sub> setup time to RESET low	t <sub>SVDD</sub>	500		ns	
IOWR/IORD wait time from RESET low	t <sub>SYIWR</sub>	2t <sub>CYK</sub>		ns	RESET low to first read/write
IOWR/IORD recovery time	t <sub>RVIWR</sub>	200		ns	

## AC Characteristics, μPD71071C/GC/L

T<sub>A</sub> = -40 to +85°C, V<sub>DD</sub> = 5 V ±10%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>DMA Mode</b>					
Clock cycle	t <sub>CYK</sub>	100		ns	
Clock pulse width high	t <sub>KKH</sub>	39		ns	
Clock pulse width low	t <sub>KKL</sub>	49		ns	
Clock rise time	t <sub>KR</sub>		10	ns	1.5 V → 3.0 V
Clock fall time	t <sub>KF</sub>		10	ns	3.0 V → 1.5 V
Input rise time	t <sub>IR</sub>		20	ns	
Input fall time	t <sub>IF</sub>		12	ns	
Output rise time	t <sub>OR</sub>		20	ns	
Output fall time	t <sub>OF</sub>		12	ns	
DMARQ setup time to CLK high	t <sub>SDQ</sub>	20		ns	SI, S0, S3, S4w, S4w
HLDRQ high delay from CLK low	t <sub>DHGH</sub>	5	70	ns	SI, S4w
HLDRQ low delay from CLK low	t <sub>DHGL</sub>	5	70	ns	SI, S0, S4w
HLDRQ low level period	t <sub>HGHGL</sub>	2t <sub>CYK</sub> - 50		ns	S4w
HLDK high setup time to CLK low	t <sub>SHA</sub>	20		ns	S0, S4, S4w
AEN high delay from CLK low	t <sub>DAEH</sub>	5	70	ns	S1, S2
AEN low delay time from CLK low	t <sub>DAEL</sub>	5	70	ns	SI, S4w
ASTB high delay time from CLK low	t <sub>DSTH</sub>	5	70	ns	S1
ASTB low delay time from CLK high	t <sub>DSTL</sub>	5	70	ns	S1
ASTB high level period	t <sub>STSTH</sub>	t <sub>KKL</sub> - 15		ns	
ADR/ <u>BE</u> / <u>RD</u> / <u>WR</u> (1) active delay from CLK low	t <sub>DA</sub>	5	80	ns	S1, S2
ADR/ <u>BE</u> / <u>RD</u> / <u>WR</u> float time from CLK low	t <sub>FA</sub>	0	70	ns	SI, S4w
ADR setup time to ASTB low	t <sub>SAST</sub>	t <sub>KKL</sub> - 40		ns	
ADR hold time from ASTB low	t <sub>HSTA</sub>	t <sub>KKH</sub> - 20		ns	

### AC Characteristics, μPD71071C/GC/L (cont)

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>DMA Mode (cont)</b>					
ADR/UBE off delay time from CLK low	t <sub>DAF</sub>	0	70	ns	S1, S2
RD low delay time from ADR float	t <sub>DAR</sub>	-10		ns	
Input data delay time from MRD low	t <sub>DMRID</sub>		2t <sub>CYK</sub> - 80	ns	S12
Input data hold time from MRD high	t <sub>HMRID</sub>	0		ns	S14
Output data delay time from CLK low	t <sub>DOD</sub>	10	80	ns	S22
Output data hold time from CLK high	t <sub>HOD</sub>	10		ns	S24
Output data hold time from MWR high	t <sub>HMWOD</sub>	t <sub>KKL</sub> - 35		ns	
RD low delay time from CLK high	t <sub>DKHR</sub>	5	50	ns	S2 compressed timing
RD low level period	t <sub>RRL1</sub>	2t <sub>CYK</sub> - 30		ns	Normal timing
	t <sub>RRL2</sub>	t <sub>CYK</sub> + t <sub>KKH</sub> - 30		ns	Compressed timing
RD high delay time from CLK low	t <sub>DRH</sub>	15	70	ns	S4
ADR delay time from RD high	t <sub>DRA</sub>	t <sub>CYK</sub> - 30		ns	
WR low delay time from CLK low	t <sub>DWL1</sub>	5	50	ns	S3 normal write
WR low delay time from CLK low	t <sub>DWL2</sub>	5	50	ns	S2 extended write, normal timing
WR low delay time from CLK high	t <sub>DWL3</sub>	5	50	ns	S2 extended write, compressed timing
WR low level period	t <sub>WWL1</sub>	t <sub>CYK</sub> - 30		ns	Normal write
	t <sub>WWL2</sub>	2t <sub>CYK</sub> - 30		ns	Extended write, normal timing
	t <sub>WWL3</sub>	t <sub>CYK</sub> + t <sub>KKH</sub> - 30		ns	Extended write, compressed timing
WR high delay from CLK low	t <sub>DWH</sub>	5	50	ns	S4

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
RD low delay time from CLK low	t <sub>DKLR</sub>	5	50	ns	S2 normal timing
RD, WR low delay time from DMAAK active	t <sub>DARW</sub>	0		ns	S1, S2
RD high delay time from WR high	t <sub>DWHRH</sub>	5		ns	
DMAAK delay time from CLK high	t <sub>DKHDA</sub>	5	70	ns	S1 I/O memory timing
DMAAK delay time from CLK low	t <sub>DKLDA</sub>	10	90	ns	S1 cascade mode
DMAAK inactive delay time from CLK high	t <sub>DDAI1</sub>	5	70	ns	S4
DMAAK inactive delay time from HLDAAK low	t <sub>DDAI2</sub>	5	t <sub>KKL</sub> + 70	ns	S4 cascade mode, HLDAAK low in S4
	t <sub>DDAI3</sub>	t <sub>KKL</sub> + 70	4t <sub>CYK</sub> + 70	ns	S4 cascade mode, HLDAAK low except in S4
DMAAK active level period	t <sub>DADA</sub>	2t <sub>CYK</sub>		ns	Cascade mode
TC low delay time from CLK high	t <sub>DTCL</sub>	5	70	ns	S3
TC off delay time from CLK high	t <sub>DTCF</sub>		30	ns	S4
TC high delay time from CLK high	t <sub>DTCH</sub>		t <sub>KKH</sub> + t <sub>CYK</sub> - 10	ns	0 to 2.2 V (2)
TC low level period	t <sub>TCTCL</sub>	t <sub>CYK</sub> - 15		ns	
DMAAK hold time from TC low	t <sub>HTCOA</sub>	t <sub>CYK</sub> - 15		ns	
END low setup time to CLK high	t <sub>SED</sub>	20		ns	S2
END low level period	t <sub>EDEL</sub>	50		ns	
READY setup time to CLK high	t <sub>SRY</sub>	20		ns	S3, SW
READY hold time from CLK high	t <sub>HRY</sub>	10		ns	S3, SW

**Note:**

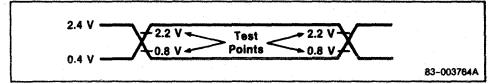
- (1) RD/WR refers to  $\overline{IORD}$  or  $\overline{MRD}$  and  $\overline{IOWR}$  or  $\overline{MWR}$ , respectively.
- (2) For  $\overline{END}/\overline{TC}$ , output load capacitance = 60 pF maximum. To meet the t<sub>DTCH</sub> parameter use a 2.2-kΩ pull-up resistor with a load capacitance of 75 pF. For other than  $\overline{END}/\overline{TC}$ , output load capacitance = 100 pF maximum.

## AC Characteristics, μPD71071C/GC/L (cont)

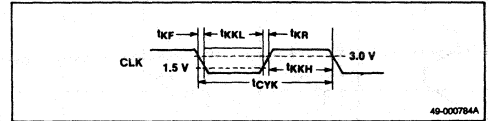
Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Programming Mode and RESET</b>					
I $\overline{O}$ WR low level period	t $\overline{I}$ WIWL	80		ns	
C $\overline{S}$ low setup time to I $\overline{O}$ WR high	t $\overline{S}$ CSIW	80		ns	
C $\overline{S}$ hold time from I $\overline{O}$ WR high	t $\overline{H}$ IWCS	0		ns	
ADR/ $\overline{U}$ BE setup time to I $\overline{O}$ WR high	t $\overline{S}$ AIW	80		ns	
ADR/ $\overline{U}$ BE hold time from I $\overline{O}$ WR high	t $\overline{H}$ IWA	0		ns	
Input data setup time to I $\overline{O}$ WR high	t $\overline{S}$ IDIW	80		ns	
Input data hold time from I $\overline{O}$ WR high	t $\overline{H}$ IWD	0		ns	
I $\overline{O}$ RD low level period	t $\overline{I}$ RIRL	120		ns	
ADR/ $\overline{C}$ S setup time to I $\overline{O}$ RD low	t $\overline{S}$ AIR	20		ns	
ADR/ $\overline{C}$ S hold time from I $\overline{O}$ RD high	t $\overline{H}$ IRA	0		ns	
Output data delay time from I $\overline{O}$ RD low	t $\overline{D}$ IROD	10	100	ns	
Output data float time from I $\overline{O}$ RD high	t $\overline{F}$ IROD		80	ns	
RESET high level period	t $\overline{R}$ RESET	2t $\overline{C}$ YK		ns	
V $\overline{D}$ D setup time to RESET low	t $\overline{S}$ VDD	500		ns	
I $\overline{O}$ WR/I $\overline{O}$ RD wait time from RESET low	t $\overline{S}$ YIWR	2t $\overline{C}$ YK		ns	RESET low to first read/write
I $\overline{O}$ WR/I $\overline{O}$ RD recovery time	t $\overline{R}$ VIWR	160		ns	

## Timing Waveforms

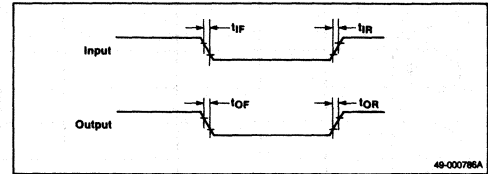
### Timing Measurement Points



### Clock Timing



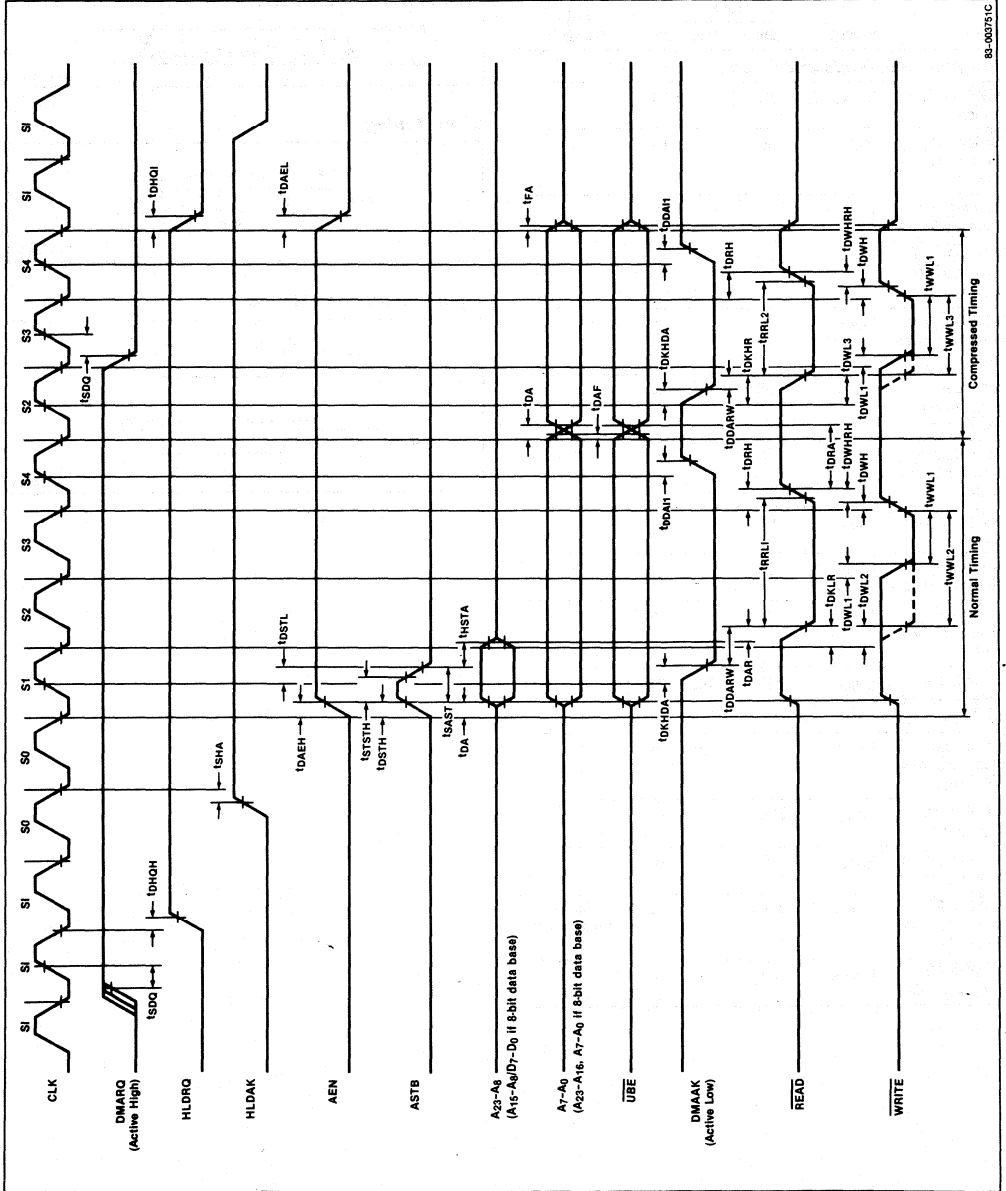
### Input/Output Edge Timing





Timing Waveforms (cont)

I/O-Memory Transfer Timing

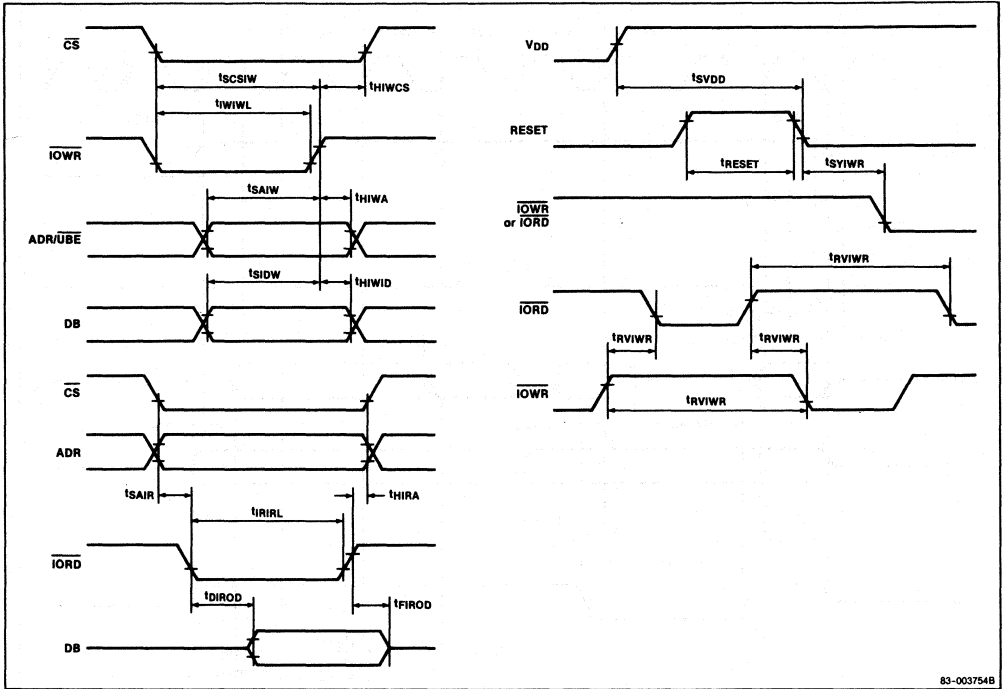


83-002751C



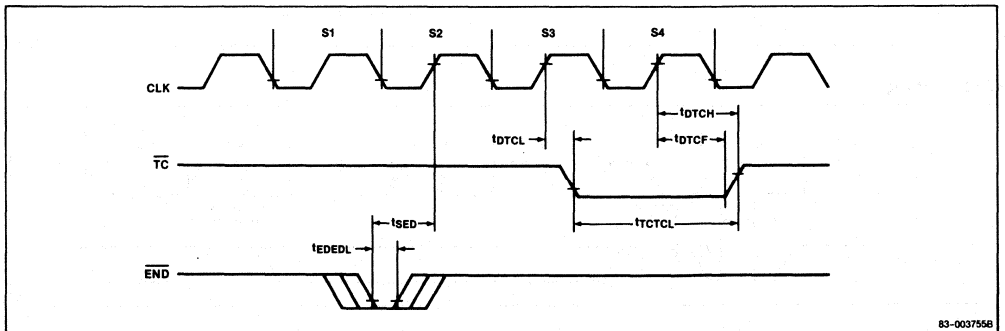
Timing Waveforms (cont)

Programming Mode and RESET Timing



83-003754B

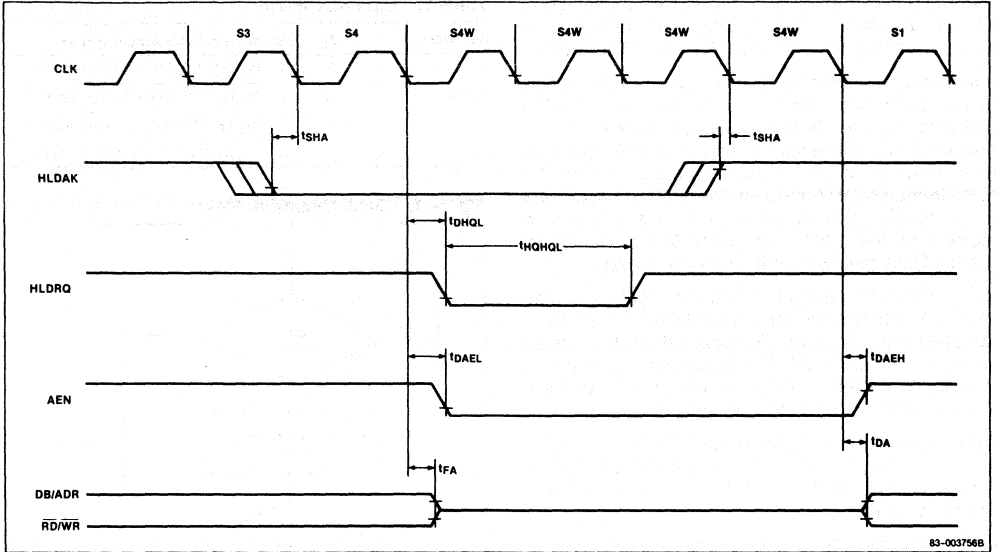
END/TC Timing



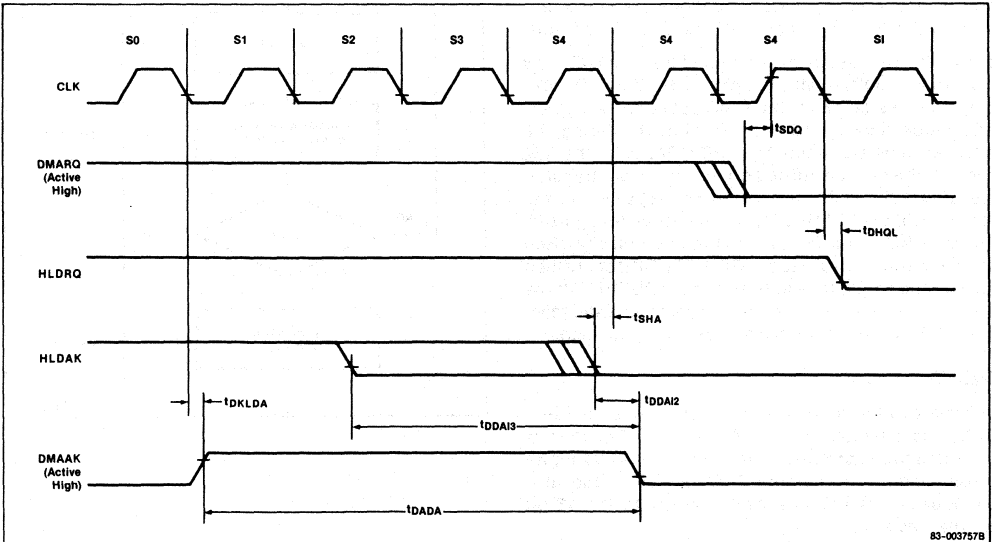
83-003755B

## Timing Waveforms (cont)

### Bus Wait Timing



### Cascade Timing



**Functional Description**

**DMA Operation**

The μPD71071 functions in three cycles: idle, DMA, and standby. In an idle or standby cycle, the CPU uses the bus, while in a DMA cycle, the μPD71071 uses it.

**Idle Cycle.** In an idle cycle, there are no DMA cycles active, but there may be one or more active DMA requests; however, the CPU has not released the bus. The μPD71071 will sample the four DMARQ input pins at every clock. If one or more inputs are active, the corresponding DMA request bits (RQ) are set in the status register and the μPD71071 sends a bus hold request to the CPU. The μPD71071 continues to sample DMA requests until it obtains the bus.

After the CPU returns a HLD $\overline{A}$ K signal and the μPD71071 obtains the bus, the μPD71071 stops DMA sampling and selects the DMA channel with the highest priority from the valid DMA request signals. Programming of the μPD71071 is done when the μPD71071 is in the idle cycle or the standby mode.

**DMA Cycle.** In a DMA cycle, the μPD71071 controls the bus and performs DMA transfer operations based on programmed information. Figure 1 outlines the sequential flow of a DMA operation.

**Standby Mode.** The μPD71071 can also be used in standby mode. It is in standby mode and consumes the static supply current (I<sub>DD2</sub>) when the clock is turned off and no I/O read or write operations are being performed. All internal registers will retain their contents.

The μPD71071 can be programmed (using  $\overline{IOWR}$ ) and read (using  $\overline{IORD}$ ) with the clock off. The μPD71071 only uses the clock for the DMA data transfer cycles. The clock may be turned off without altering the internal registers when the μPD71071 is in the idle cycle. If the clock is turned off during a DMA transfer, the μPD71071 will not operate correctly. When the clock is off, the DMARQ inputs will not be recognized. The DMARQ inputs could be externally logically ORed and cause an interrupt to the CPU. The CPU could then turn on the clock, thus activating the μPD71071. If the previously programmed mode of operation is still valid, the μPD71071 does not have to be reprogrammed.

**Data Bus Width**

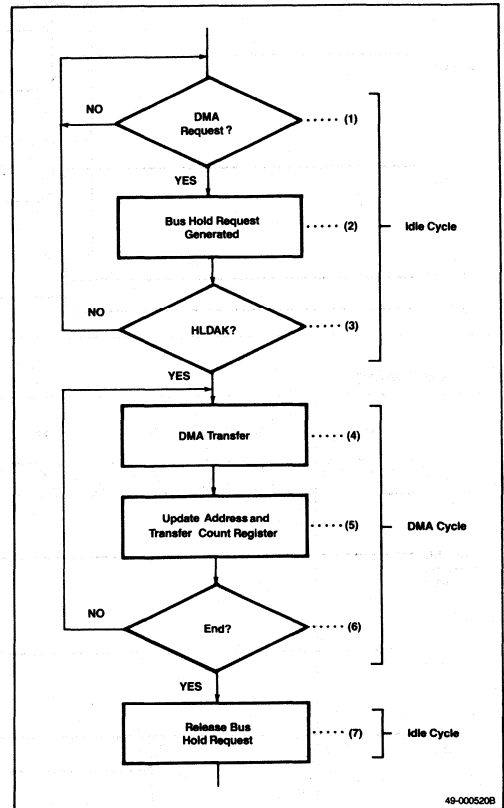
In order to allow an easy interface with an 8- or 16-bit CPU, the data bus width of the μPD71071 is user programmable for 8 or 16 bits. A 16-bit data bus allows 16-bit memory-to-memory DMA transfers and also provides a one-I/O bus cycle access to the 16-bit internal registers.

Table 1 shows the relationship of the data bus width, A<sub>0</sub>,  $\overline{UBE}$ , and the internal registers.

**Table 1. Data Bus Width**

Bus Width	A <sub>0</sub>	$\overline{UBE}$	Internal Read/Write Registers
8 bits	X	X	D <sub>7</sub> -D <sub>0</sub> ← 8-bit internal register
16 bits	0	1	D <sub>7</sub> -D <sub>0</sub> ← 8-bit internal register
	1	0	D <sub>15</sub> -D <sub>8</sub> ← 8-bit internal register
	0	0	D <sub>15</sub> -D <sub>0</sub> ← 16-bit internal register

**Figure 1. DMA Operation Flow**



49-0005208

## Terminal Count

The μPD71071 ends DMA service when it generates a terminal count ( $\overline{TC}$ ) or when the  $\overline{END}$  input becomes active. A terminal count is produced when a borrow is generated by the current count register and a low-level pulse is output to the  $\overline{TC}$  pin. Figure 2 shows that the current count register is tested after each DMA operation.

If autoinitialize is not set when DMA service ends, the mask register bit applicable to the channel where service ended is set, and the  $\overline{DMARQ}$  input of that channel is masked.

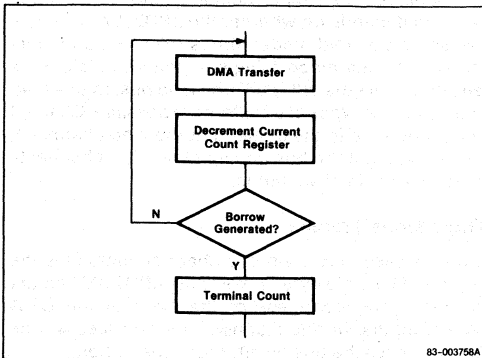
## DMA Transfer Type

The type of transfer the μPD71071 performs depends on the following conditions.

- Memory-to-memory transfer enable
- Direction of memory-to-I/O transfer (each channel)
- Transfer mode (each channel)
- Bus mode

**Memory-to-Memory Transfer Enable.** The μPD71071 can perform memory-to-I/O transfers (one transfer cycle in one bus cycle) and memory-to-memory transfers (one transfer in two bus cycles). To select memory-to-memory transfer, set bit 0 of the device control register to 1. The DMA channels used in memory-to-memory transfers are fixed, with channel 0 as the source channel and channel 1 as the destination channel. Channels 2 and 3 cannot be used in memory-to-memory transfers. The contents of the count registers and word/byte transfer modes of channels 0 and 1 should be the same when performing memory-to-memory transfer.

Figure 2. Generation of Terminal Count ( $\overline{TC}$ )



For memory-to-memory byte transfer in 16-bit data bus mode, a read data from upper data bus is to be written to upper data bus, while a read data from lower data bus is to be written to lower data bus. Therefore, start addresses for source and destination must be the even-even or odd-odd. For word transfer, only even-even addresses are to be set for source and destination. (See Byte/Word Transfer paragraphs below.) When  $\overline{DMARQ0}$  (channel 0) becomes active, the transfer is initiated.

During memory-to-memory bus cycles in the 16-bit mode, data read from the  $\overline{DMAC}$ 's upper (lower) data bus is written to the upper (lower) data bus of the destination device. Thus, for word transfers, only even source and destination addresses should be used.

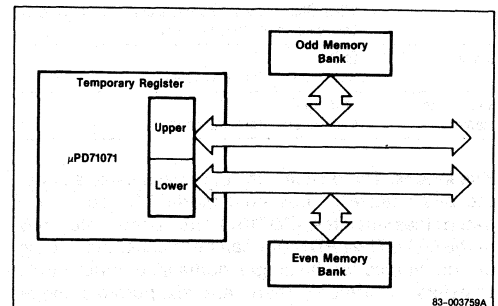
The DMA request input pin or a software DMA request to channel 0 may initiate memory-to-memory transfers. The μPD71071 performs the following operations until a channel 1 terminal count or  $\overline{END}$  input is present:

- During the first bus cycle, the memory data pointed to by the current address register of channel 0 is read into the temporary register of the μPD71071 and the address and count of channel 0 are updated.
- During the second bus cycle, the temporary register data is written to the memory location shown by the current address register of channel 1, and the address and count of channel 1 are updated.

**Note:** If  $\overline{DMARQ1}$  (channel 1) becomes active, the μPD71071 will perform memory-to-I/O transfer even though memory-to-memory transfer is selected. Since this may cause erroneous memory-to-memory transfers, mask out channel 1 ( $\overline{DMARQ1}$ ) by setting bit 1 of the mask register to 1 before starting memory-to-memory transfers.

During memory-to-memory transfers, the addresses on the source side (channel 0) can be fixed by setting bit 1 of the device control register to 1. In this manner, a

Figure 3. Memory-to-Memory Transfer in 16-Bit Data Bus Mode



range of memory can be initialized with the same value since the contents of the source address never change. During memory-to-memory transfer, the DMAAK signal and channel 0's terminal count (TC) pulse are not output. (See figure 3.)

**Direction of Memory-to-I/O Transfers.** All DMA transfers use memory as a reference point. Therefore, a DMA read reads a memory location and writes to an I/O port. A DMA write reads an I/O port and writes the data to a memory location. In memory-to-I/O transfer, use the mode control register to set one of the transfer directions in table 2 for each channel and activate the appropriate control signals.

**Table 2. Transfer Direction**

Transfer Direction	Activated Signals
Memory → I/O (DMA read)	IOWR, MRD
I/O → memory (DMA write)	IORD, MWR

Verify  
(Outputs addresses only. Does not perform a transfer.)

**Transfer Modes.** In memory-to-I/O transfer, the mode control register selects the single, demand, or block mode of DMA transfer for each channel. The conditions for the termination of each transfer characterize each transfer mode. Memory-to-memory transfers have no relationship to single, demand, or block mode. Memory-to-memory transfers are a separate and distinct type of transfer mode. Table 3 shows the various transfer modes and termination conditions.

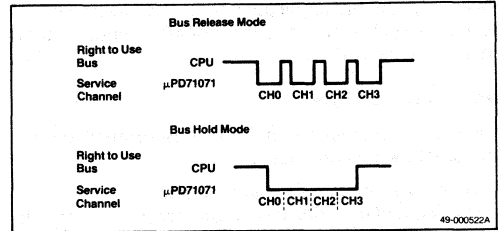
**Table 3. Transfer Termination**

Transfer Mode	End of Transfer Conditions
Single	After each byte/word
Demand	END input Generation of terminal count When DMA request of the channel in service becomes inactive When DMA request of a channel in higher priority becomes active (bus hold mode)
Block	END input Generation of terminal count
Memory-to-memory	END input Generation of terminal count

**Bus Modes.** The device control register selects either the bus release or bus hold mode. The bus mode determines when the μPD71071 returns the system bus to the CPU. The μPD71071 can be in either the release or hold modes for the single, demand, or block mode transfers. Therefore, there are six possible mode combinations.

Figure 4 shows that in bus release mode, only one channel can receive service after obtaining the bus. When DMA service ends (end of transfer conditions depend on the transfer mode), the channel returns the bus to the CPU (regardless of the state of other DMA requests) and the μPD71071 enters the idle cycle. When the μPD71071 regains use of the bus, a new DMA operation begins.

**Figure 4. Bus Modes**



In bus hold mode, several channels can receive service without releasing the bus after obtaining it. If there is another valid DMA request when a channel's DMA service is finished, the new DMA service can begin after the previous service without returning the bus to the CPU. End of transfer conditions depend on the transfer mode. A channel cannot terminate (end count) a transfer mode and immediately start on its next set of transfers. There must be another DMA channel service interleaved or the μPD71071 will put in an idle cycle. The following shows an example of the possible sequences for Channel 2.

CHAN2 → CHANn (n = 0,1,3) → CHAN2  
or,  
CHAN2 → idle → CHAN2

The operation of single, demand, and block mode transfers depends on whether the μPD71071 is in bus release or bus hold mode. In bus release mode, only one type of bus mode (single, demand, or block) is used each time the μPD71071 has the bus. In bus hold mode, multiple types of transfers are possible. Channel 0 might operate in the demand mode, and channel 1, which could get the bus immediately after channel 0, could operate in block mode.

**Single Mode Transfer**

In bus release mode, when a channel completes the transfer of a single byte or word, the μPD71071 enters the idle cycle regardless of the state of the DMA request inputs. In this manner, other devices will be able to access the bus on alternate bus cycles.

In bus hold mode, when a channel completes the transfer of a single byte or word, the μPD71071 terminates the channel's service even if it is still asserting a DMA request signal. The μPD71071 will then service the highest priority channel requesting the bus. If there are no requests from any other channel, the μPD71071 releases the bus and enters the idle cycle.

### Demand Mode Transfer

In bus release mode, the currently active channel continues its data transfer as long as the DMA request of that channel is active, even though other DMA channels are issuing higher priority requests. When the DMA request of the serviced channel becomes inactive, the μPD71071 releases the bus and enters the idle state, even if the DMA request lines of other channels are active.

In bus hold mode, when the active channel completes a single transfer, the μPD71071 checks DMA request lines (other request lines when  $\overline{\text{END}}$  or  $\overline{\text{TC}}$ , all request lines including the last serviced channel when there is no  $\overline{\text{END}}$  or  $\overline{\text{TC}}$ ). If there are active requests, the μPD71071 starts servicing the highest priority channel requesting service. If there is no request, the μPD71071 releases the bus and enters the idle state.

### Block Mode Transfer

In bus release mode, the current channel continues data transfer until a terminal count or the external  $\overline{\text{END}}$  signal becomes active. During this time, the μPD71071 ignores all other DMA requests. After completion of the block transfer, the μPD71071 releases the bus and enters the idle cycle even if DMA requests from other channels are active.

In bus hold mode, the current channel transfers data until a terminal count or the external  $\overline{\text{END}}$  signal becomes active. When the service is complete, the μPD71071 checks all DMA requests without releasing the bus. If there is an active request, the μPD71071 immediately begins servicing the request. The μPD71071 releases the bus after it honors all DMA requests or a higher priority bus master requests the bus.

Figure 5 shows the operation flow for the six possible transfer and bus mode operations in DMA transfer.

### Byte/Word Transfer

If the initialize command selects a 16-bit data bus width, the mode control register can specify DMA transfer in byte or word units for each channel. Table 4 shows the update of the address and count registers during byte/word transfer.

**Table 4. Address and Count Registers**

Register	Byte Transfer	Word Transfer
Address	± 1	± 2
Count	-1	-1

During word transfers, two bytes starting at an even address are handled as one word. If word transfer is selected and the initial value of the set address is odd, the μPD71071 will always decrement that address by 1, thus making the address even for the data transfer. For this reason, it is best to select even addresses when transferring words, to avoid destroying data.  $A_0$  and  $\overline{\text{UBE}}$  control byte and word transfers.

Table 5 shows the relationship between the data bus width,  $A_0$  and  $\overline{\text{UBE}}$  signals, and data bus status.

**Table 5. Data Bus Status**

Data Bus Width	$A_0$	$\overline{\text{UBE}}$	Data Bus Status
8 bits	X	1 (1)	$D_7$ - $D_0$ valid byte
16 bits	0	1	$D_7$ - $D_0$ valid byte
	1	0	$D_{15}$ - $D_8$ valid byte
	0	0	$D_{15}$ - $D_0$ valid word

Note:

(1) Always 1 for an 8-bit bus.

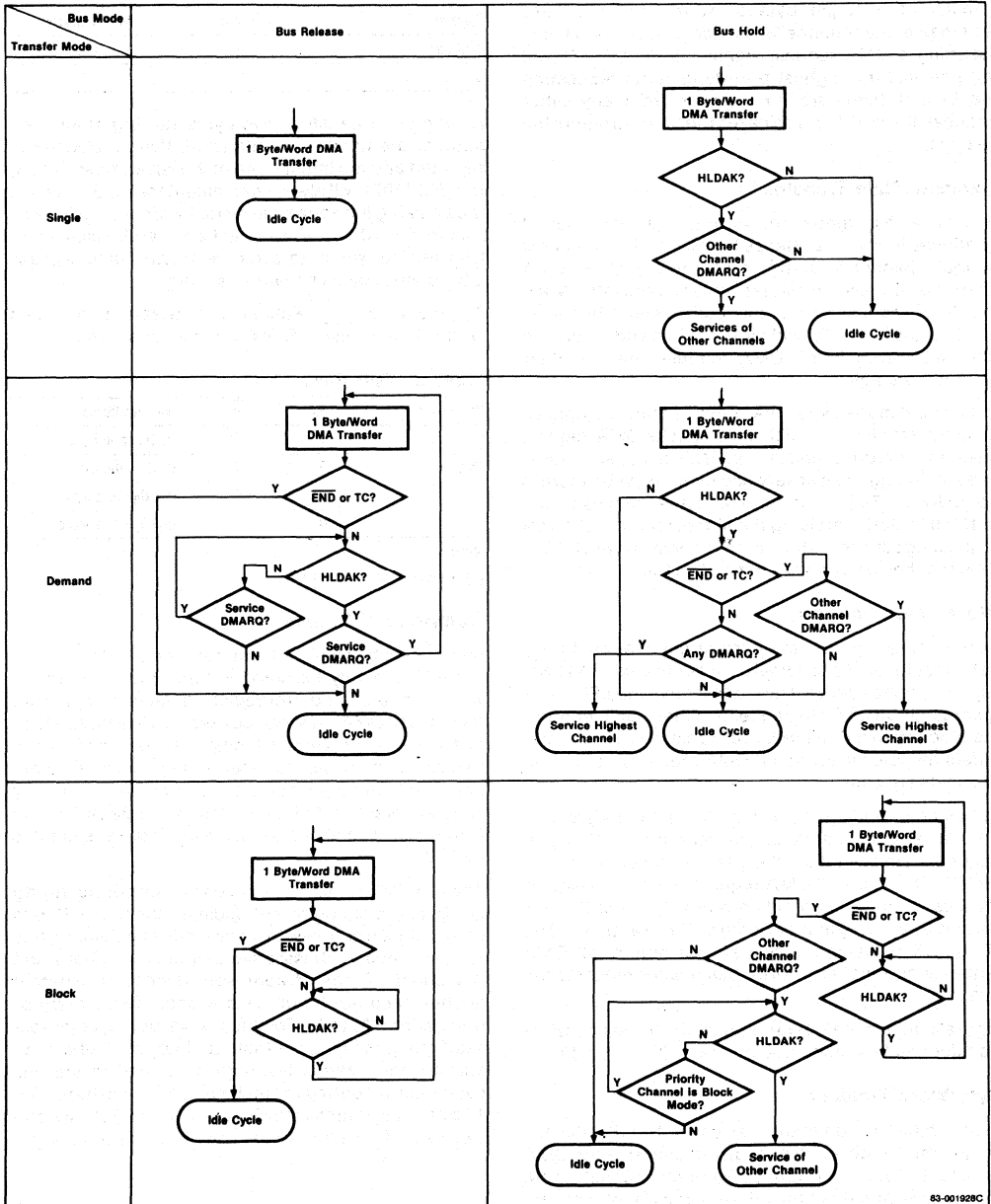
### Compressed Timing

In transfers between I/O and memory, a DMA transfer cycle is normally executed in four clocks. However, when the device control register selects compressed timing, one DMA cycle can be executed in a three-clock bus cycle. Compressed timing may be used in the release or hold modes when doing block transfers between I/O and memory. In the demand mode, only use compressed timing in the bus release mode. Compressed timing mode increases data transfer rates by 33%.

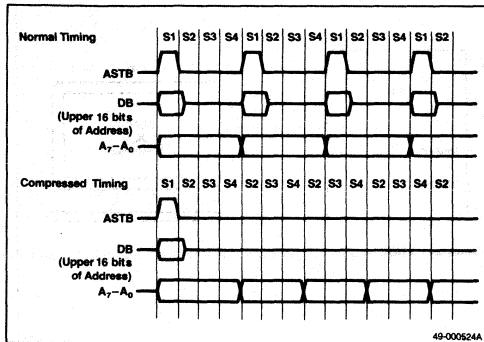
The μPD71071 is able to omit one clock period during compressed timing by not updating the upper 16 bits of the latched address. In block mode and demand bus release mode, addresses are output sequentially and the upper 16 bits of addresses latched in external latches need not be updated except after a carry or borrow from  $A_7$  to  $A_8$ . For this reason, during compressed timing, the S1 state (output of upper 16 bits of an address for external latching) is omitted in the bus cycles except during the first bus cycle when the upper 16 bits of an address are changed. Figure 6 shows one word waveforms for normal and compressed timing.



Figure 5. Transfer and Bus Modes Operations



**Figure 6. Normal and Compressed Timing Waveforms**



### Software DMA Requests

The μPD71071 can accept software DMA requests in addition to DMA requests from the four DMARQ pins. Setting the appropriate bit in the request register generates a software DMA request. The mask register does not mask software DMA requests. Software DMA requests operate differently depending on which bus or transfer mode is used.

**Bus Mode.** When bus release mode is set, the highest priority channel among software DMA requests and DMARQ pins is serviced, and all bits of the request register are cleared when the service is over. Therefore, there is a chance that other software DMA requests will be cancelled.

When bus hold mode is set, only the corresponding bit of the request register is cleared after a DMA service is over. Therefore, all software DMA requests will be serviced in the sequence of their priority level.

Software DMA requests for cascade channels (see Cascade Connection) must be performed in bus hold mode. When a cascade channel is serviced, the master μPD71071 operational mode is changed to bus release mode temporarily and all bits of the request register are cleared when the cascade channel service is over. To avoid this, it is necessary to mask any cascade channels before issuing a software DMA request. After confirming that all DMA software services are complete and all bits of the request register are cleared, the cascade channel masks can be cleared.

**Transfer Mode.** When single or demand mode is set, the applicable request bits are cleared and software DMA service ends with the transfer of one byte/word. When block mode or memory-to-memory modes are set, service continues until END is input or a terminal count is generated. Applicable request bits are cleared when service ends.

### Autoinitialize

When the mode control register is set to autoinitialize a channel, the μPD71071 automatically initializes the address and count registers when END is input or a terminal count is generated. The contents of the base address and base count registers are transferred to the current address and current count registers, respectively. The applicable bit of the mask register is unaffected. The applicable bit of the mask register is set for channels not programmed for autoinitialize.

The autoinitialize function is useful for the following types of transfers.

**Repetitive Input/Output of Memory Area.** Figure 7 shows an example of DMA transfer between a CRT controller and memory. After setting the value in the base and current registers, autoinitialize allows repetitive DMA transfer between the CRT controller and the video memory area without CPU involvement.

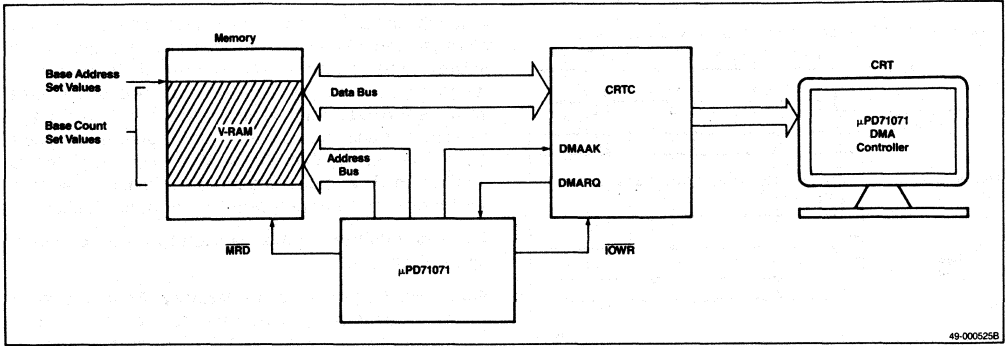
**Continuous Transfer of Several Memory Areas.** The CPU can indirectly write to the address or count registers by writing to the base registers. New values can be written to the base registers. In the autoinitialize mode, the value in the base register will be transferred to the address/count registers when termination is reached in the address/count registers. Because of this, the autoinitialize function can perform continuous transfer of several contiguous or noncontiguous memory areas during single or demand bus release modes in the following manner.

During the transfer of data in area 1 (the first area being transferred), the CPU can write address and count information about area 2 (the second area to be transferred). Generation of a terminal count for area 1 results in the transfer of information of area 2 to the address and count registers. This will cause area 2 to be transferred. Figure 8 illustrates this procedure.

### Channel Priority

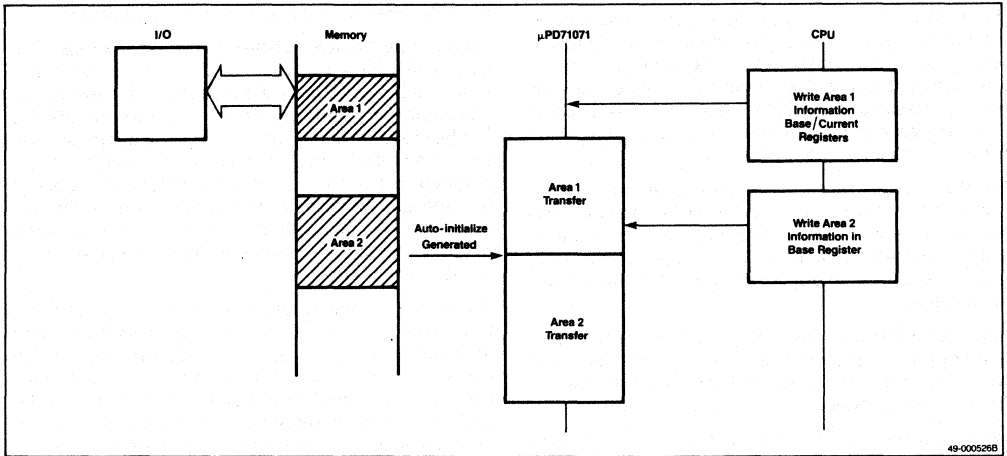
Each of the μPD71071's four channels has its own priority. When there are DMA requests from several channels simultaneously, the channel with the highest priority will be serviced. The device control register selects one of two channel priority methods: fixed and rotational priority. In fixed priority, the priority (starting with the highest) is channel 0, 1, 2, and 3, respectively. In rotational priority, priority order is rotated so that the channel that has just been given service receives the lowest priority and the next highest channel number is given the highest priority. This method prevents exclusive servicing of some channel(s). Figure 9 shows the two priority order methods.

Figure 7. Autoinitialize Application 1



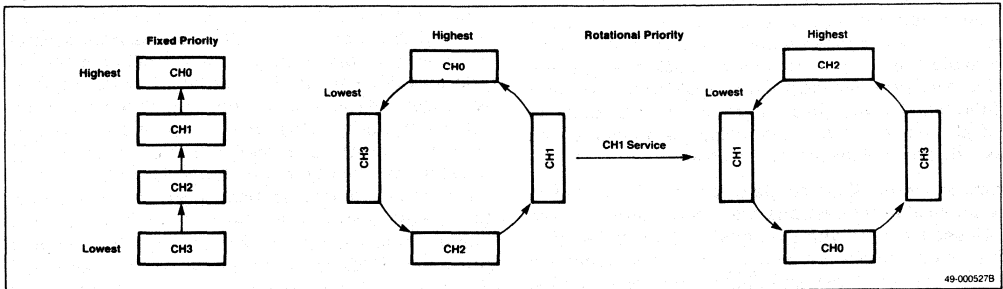
49-000526B

Figure 8. Autoinitialize Application 2



49-000526B

Figure 9. Priority Order



49-000527B

## Cascade Connection

The μPD71071 can be cascaded to expand the system DMA channel capacity. To connect a μPD71071 for cascading (figure 10), perform the following operations.

- (1) Connect pins HLDRQ and HLDAAK of the second-stage (slave) μPD71071 to pins DMARQ and DMAAK of any channel of the first-stage (master) μPD71071.
- (2) To select the cascade mode of a particular channel of a master μPD71071, set bits 7 and 6 of that channel's mode control register to 11.

When a channel is set to the cascade mode in a master μPD71071, DMARQ, DMAAK, HLDRQ, HLDAAK, and RESET are the only valid signals in the master μPD71071. The other signals are disabled. The master cascade channel only intermediates hold request/hold acknowledge between the slave and CPU.

The master μPD71071 always operates in the bus release mode when a cascade channel is in service (even when the bus hold mode is set). Other DMA requests are ignored while a cascade channel is in service. When the slave μPD71071 ends DMA service and moves into an idle cycle, the master also moves to an idle cycle and releases the bus. At this time, all bits of the master's request register are cleared. The master operates its non-cascaded channels normally.

## Bus Wait Operation

In systems using a μPD70208/70216 (V40/V50) as the CPU, the refresh control unit in the CPU changes the HLDAAK signal to inactive (even during a DMA cycle) and uses the bus. Here, the μPD71071 automatically performs a bus wait operation. This system has a bus master (V40/V50) whose priority level is higher than that of the μPD71071.

The μPD71071 executes the bus wait operation when the HLDAAK signal becomes inactive in an operating mode where transfer is executed continuously in block mode, during demand bus release mode, or during memory-to-memory transfer.

When HLDAAK becomes inactive during service in other operating modes, the operation returns to the idle cycle and transfers control of the bus to the higher bus master.

Figure 11 shows that when the HLDAAK signal becomes inactive during a continuous transfer, the μPD71071 is set up in an S4w state (bus wait). Operation moves to the idle cycle if DMARQ is inactive in the demand mode. The HLDRQ signal is made inactive for a period of about two clocks and the bus is released. The S4w state is repeated until the HLDAAK signal again becomes active and the interrupted service is immediately restarted.

Figure 10. Cascade Connection Example

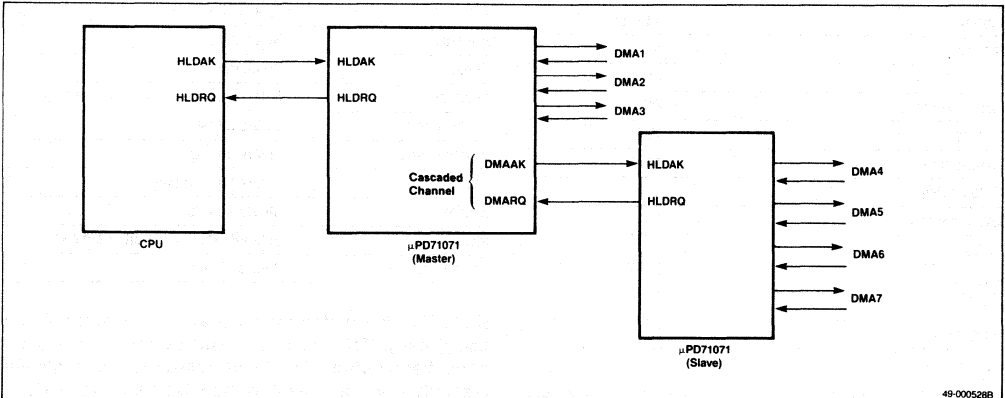
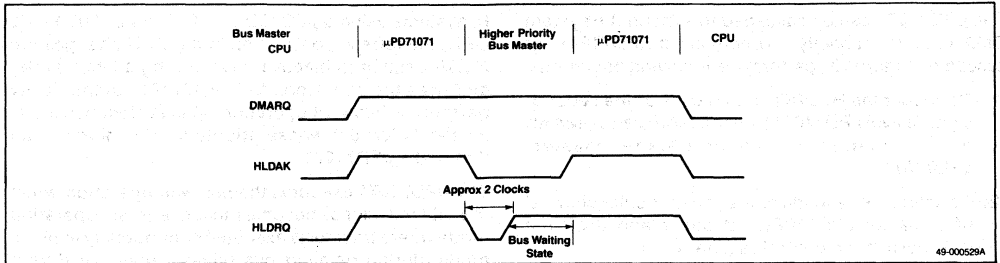


Figure 11. Bus Wait Operation



49-000529A

**Programming the μPD71071**

To prepare a channel for DMA transfer, you must select the following characteristics.

- Starting address for the transfer
- Number of byte/word transfers
- DMA operating modes
- Data bus widths
- Active levels of the DMARQ and DMAAK signals

When reading from or writing to a μPD71071 internal register, address lines A<sub>3</sub>-A<sub>0</sub> select the register, IORD or IOWR select the data transfer direction, and CS enables the transfer. Table 6 shows the register and command configurations.

Table 6. Register Configuration

Register	Bit size
Channel	5
Base address	24 (4)
Current address	24 (4)
Base count	16 (4)
Current count	16 (4)
Mode control	7 (4)
Device control	10
Status	8
Request	4
Mask	4
Temporary	16

**Note:**

When using a 16-bit CPU and selecting a 16-bit data bus, the word IN/OUT instruction can be used to read/write information two bytes at a time. However, commands in table 7 suffixed with B must be issued with the byte IN/OUT instruction.

**Initialize**

Use the initialize command as a software initialize to the μPD71071 or to set the width of the data bus. When using a 16-bit CPU, set the data bus width to 16 bits first. Figure 12 shows the initialize command format.

**Bit 0.** When the RES bit is set, the internal state of the μPD71071 is initialized and will be the same as when a hardware reset is used (except for data bus width selection). A software reset leaves bit 16B intact whereas a hardware reset selects the 8-bit data bus. After initialization, the registers are as in table 8 and the RES bit is cleared automatically.

Table 8. Register Initialization

Register	Initialization Operation
Initialize	Clears bit 0 only
Address	No change
Count	No change
Channel	Selects channel 0, current and base
Mode control	Clears all bits
Device control	Clears all bits
Status	Clears bits 3-0 only
Request	Clears all bits
Mask	Sets all bits (masks all channels)
Temporary	Clears all bits

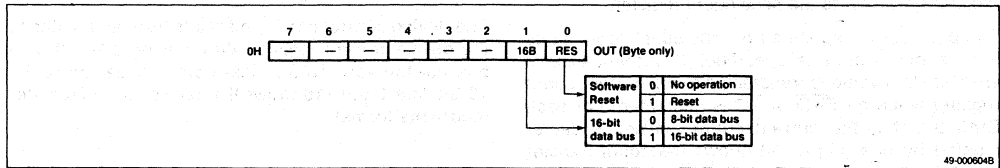
**Bit 1.** The 16B bit determines the data bus width. When using the μPD71071 in a 16-bit system, set this bit immediately after a hardware reset since a hardware reset always initializes it to the 8-bit data bus mode.

**Table 7. Command Configuration**

Address	R/W	Command Name	MSB	Format	LSB					
0H	W(B)	Initialize	—	—	16B RES					
1H	R(B)	Channel Register Read	—	—	BASE SEL3 SEL2 SEL1 SEL0					
	W(B)	Channel Register Write	—	—	BASE SELCH					
2H	R/W	Count Register Read/Write	C7	C6	C5	C4	C3	C2	C1	C0
3H	R/W		C15	C14	C13	C12	C11	C10	C9	C8
4H	R/W	Address Register Read/Write	A7	A6	A5	A4	A3	A2	A1	A0
5H	R/W		A15	A14	A13	A12	A11	A10	A9	A8
6H	R/W(B)		A23	A22	A21	A20	A19	A18	A17	A16
8H	R/W	Device Control Reg. Read/Write	AKL	RQL	EXW	ROT	CMP	DDMA	AHLD	MTM
9H	R/W		—	—	—	—	—	—	WEV	BHLD
0AH	R/W(B)	Mode Control Reg. Read/Write	TMODE	ADIR	AUT1	TDIR	—	—	W/B	
0BH	R(B)	Status Register Read	RQ3	RQ2	RQ1	RQ0	TC3	TC2	TC1	TC0
0CH	R	Temporary Reg. (lower) Read	T7	T6	T5	T4	T3	T2	T1	T0
0DH	R	Temporary Reg. (higher) Read	T15	T14	T13	T12	T11	T10	T9	T8
0EH	R/W(B)	Request Reg. Read/Write	—	—	—	—	SRQ3	SRQ2	SRQ1	SRQ0
0FH	R/W(B)	Mask Reg. Read/Write	—	—	—	—	M3	M2	M1	M0

49-00093B

**Figure 12. Initialize Command Format**



49-000904B

## Channel Register

This command reads and writes the channel register that selects one of four DMA channels for programming the address, count, and mode control registers. Figure 13 shows the channel register read/write format.

### Channel Register Read

**SEL3-SEL0.** These mutually exclusive bits show which of the four channels is currently selected for programming.

**BASE.** Base = 0. The current register may be read. During a write, the base and current registers will be written to simultaneously.

Base = 1. Only the base registers may be read or written to.

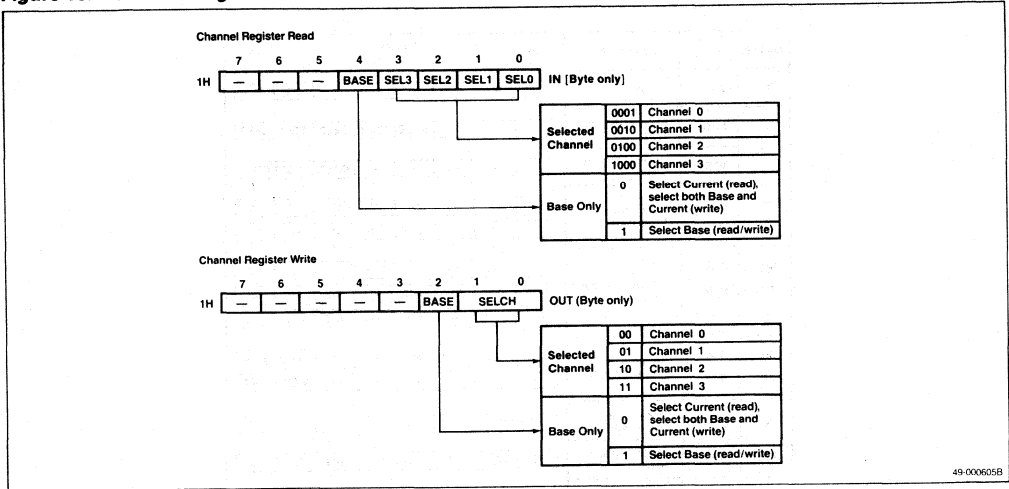
### Channel Register Write

**SELCH.** This bit selects the channel to be programmed.

**BASE.** Base = 0. The current register may be read. During a write, the base and current registers will be written to simultaneously.

Base = 1. Only the base registers may be read or written to.

Figure 13. Channel Register Format



**Count Register Read/Write**

When the 16-bit bus mode is selected, the IN/OUT instruction can directly transfer 16-bit data. The channel register selects one of the count registers. When bit 2 of the channel register write is cleared, a write to the count register updates both the base and current count registers with the new data. If bit 2 of the channel register write is set, a write to the count register only affects the base count register.

The base count registers hold the initial count value until a new count is specified. If autoinitialize is enabled, this value is transferred to the current count register when an END or TC is generated. For each DMA transfer, the current count register is decremented by one. Figure 14 shows the count register read/write format.

**Address Register Read/Write**

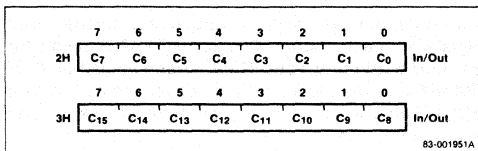
When a 16-bit data bus width is selected, the IN/OUT instruction can directly transfer the lower two bytes (4H and 5H) of the register. You must use the byte IN/OUT instruction with the upper byte (6H) of the register. The channel register selects one of the address registers. When bit 2 of the channel register is cleared, a write to the address register updates both the base and current address registers with the new data. If bit 2 of the channel register is set, a write to the address register only affects the base address register.

The base register holds the starting address value until a new setting is made and this value is transferred to the current address register during autoinitialization. For each DMA transfer, the current address register is updated ±2 during word transfer and ±1 during byte transfer. Figure 15 shows the address register read/write format.

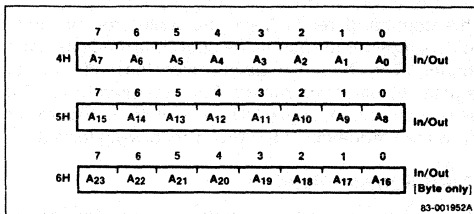
**Device Control Register Read/Write**

The device control command reads from and writes to the device control register. When using a 16-bit data bus, use the word IN/OUT instruction to read and write 16-bit data. Figure 16 shows the device control register read/write format.

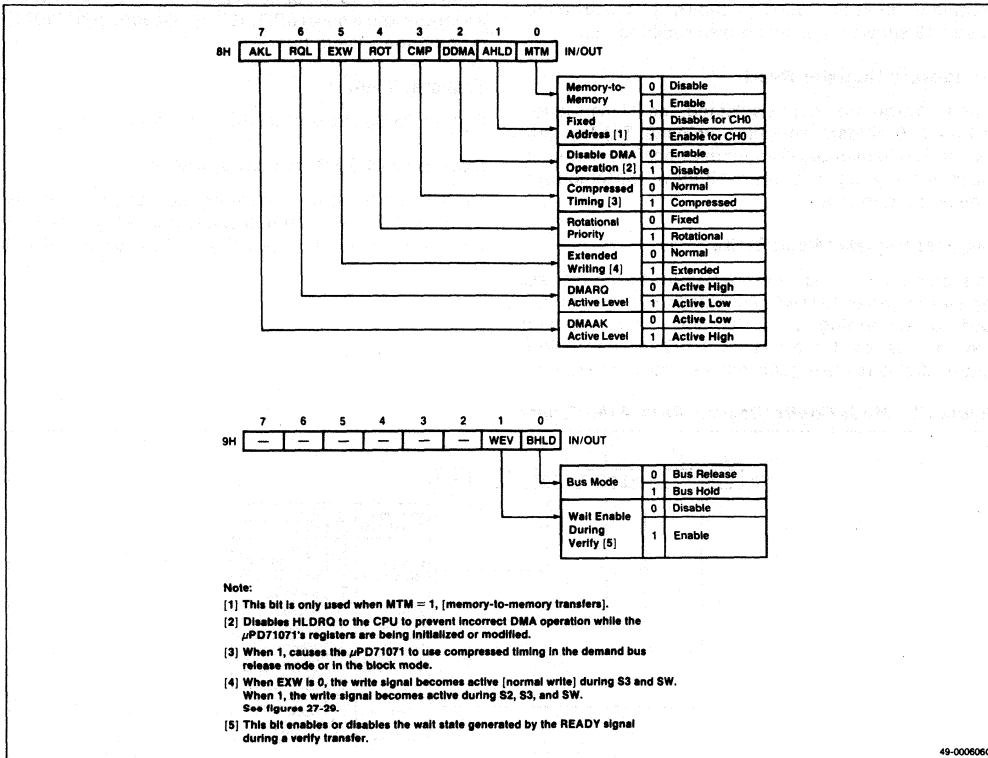
**Figure 14. Count Register Read/Write Format**



**Figure 15. Address Register Read/Write Format**



**Figure 16. Device Control Register Read/Write Format**





### Mode Control Register Read/Write

This command reads from and writes to the mode control register to specify the operating mode for each channel. The channel register selects the mode control register to be programmed. This command must be issued by the byte IN/OUT instruction. Figure 17 shows the mode control register read/write format.

### Status Register Read

This command reads the status register for the individual DMA channels. The register has DMA request states and terminal count or END information. This command must be issued by the byte IN instruction. Figure 18 shows the status register read format.

### Temporary Register Read

When a 16-bit data bus is selected, the IN instruction will read 16-bit data with this command. The last data transferred in memory-to-memory transfer is stored in the temporary register. Figure 19 shows the temporary register read format.

### Request Register Read/Write

This command reads from and writes to the request register to generate DMA requests by software for the four corresponding DMA channels. This command may be issued by the byte IN/OUT instruction. Figure 20 shows the request register read/write format.

### Mask Register Read/Write

This command reads from and writes to the mask register to mask or unmask external DMA requests for the corresponding four DMA channels (DMARQ3-DMARQ0). This command may be issued by the byte IN/OUT instruction. Figure 21 shows the mask register read/write format.

### DMA Transfer Modes

Figures 22-27 show state transition diagrams for the different modes of DMA transfer.

Figure 23 shows the state of a master μPD71071 when an input from a slave μPD71071 (cascaded μPD71071) is using the system bus.

### Transfer Timing

Figures 28-30 show μPD71071 timing waveforms.

### Examples of System Configuration

Figures 31-32 show system configuration examples using the 8-bit μPD70108 CPU and the 16-bit μPD70116 CPU. The μPD71082 externally latches addresses and data.

Figure 17. Mode Control Register Read/Write Format

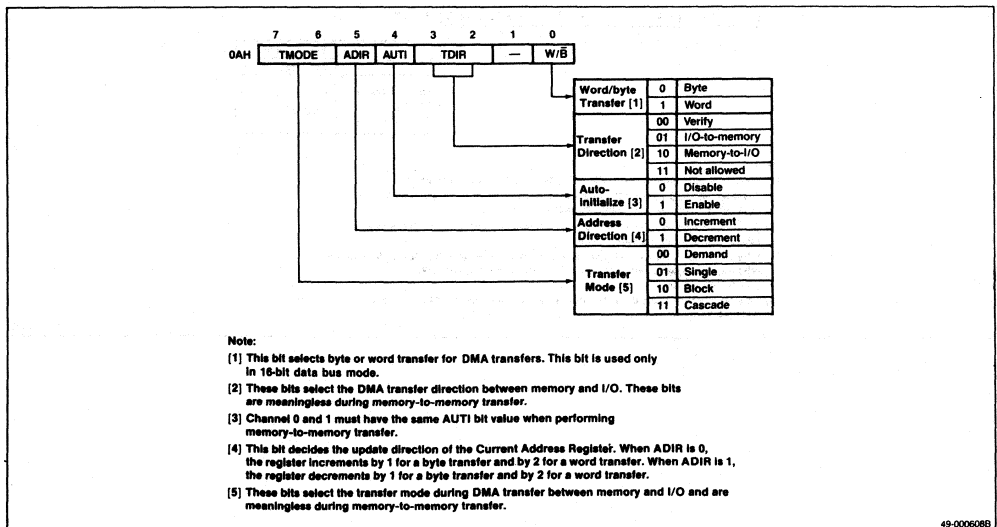


Figure 18. Status Register Read Format

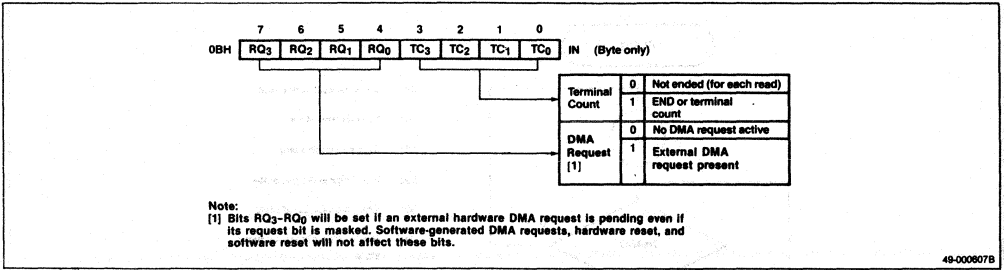


Figure 19. Temporary Register Read Format

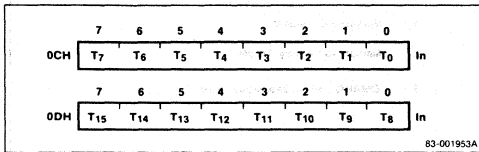


Figure 20. Request Register Read/Write Format

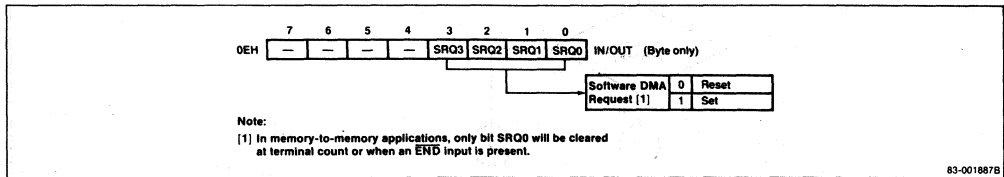


Figure 21. Mask Register Read/Write Format

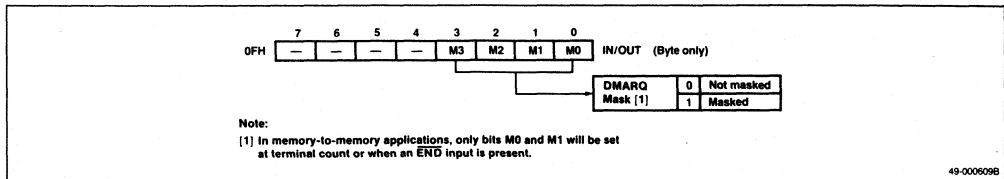


Figure 22. Idle Cycle

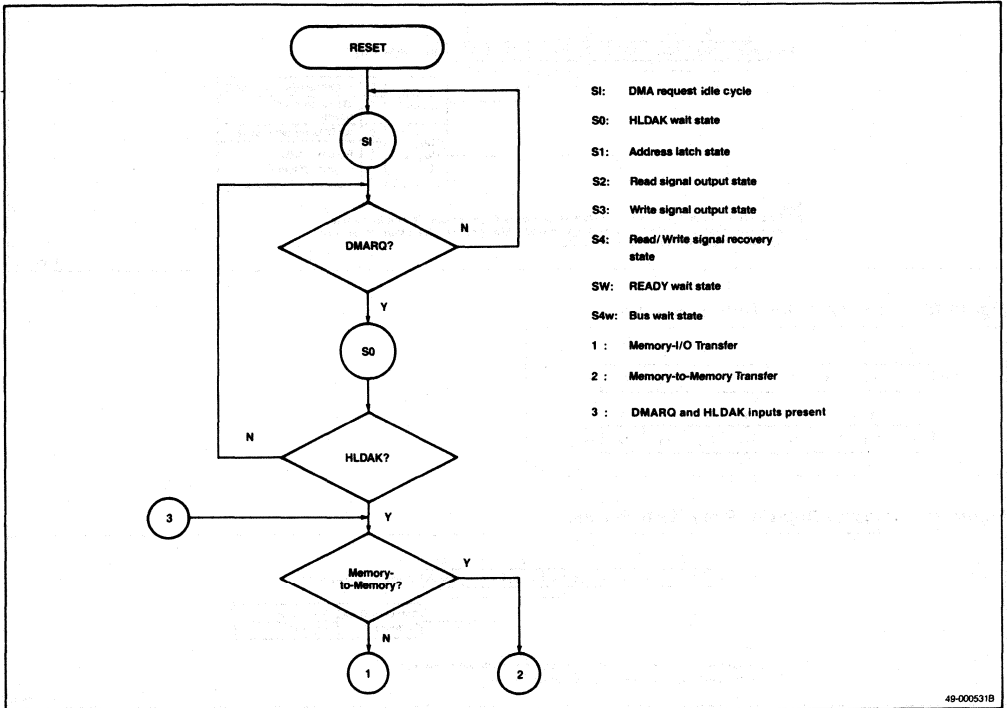


Figure 23. DMA Cycle, Cascade Mode

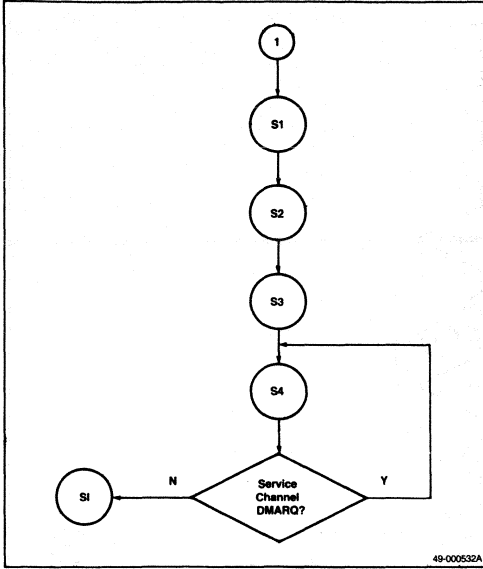


Figure 24. DMA Cycle, Single Mode

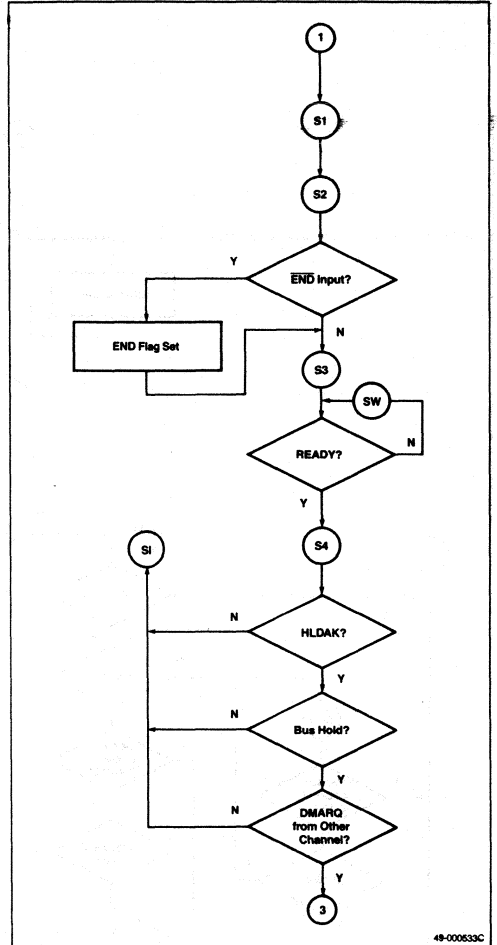
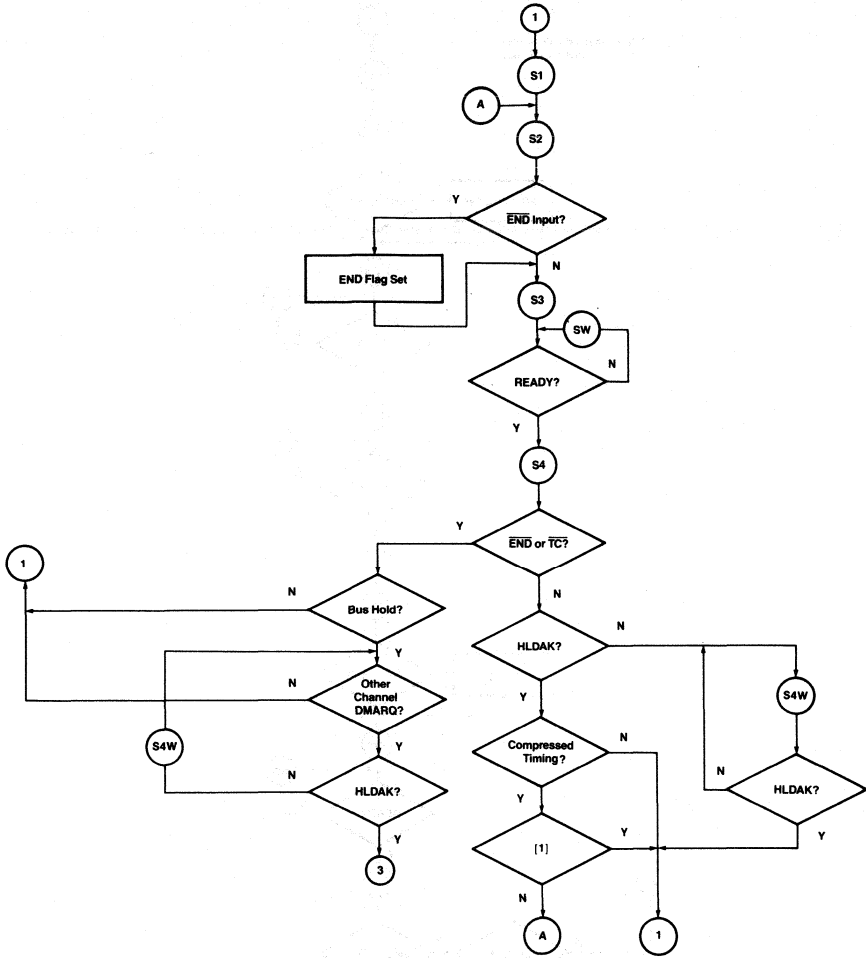




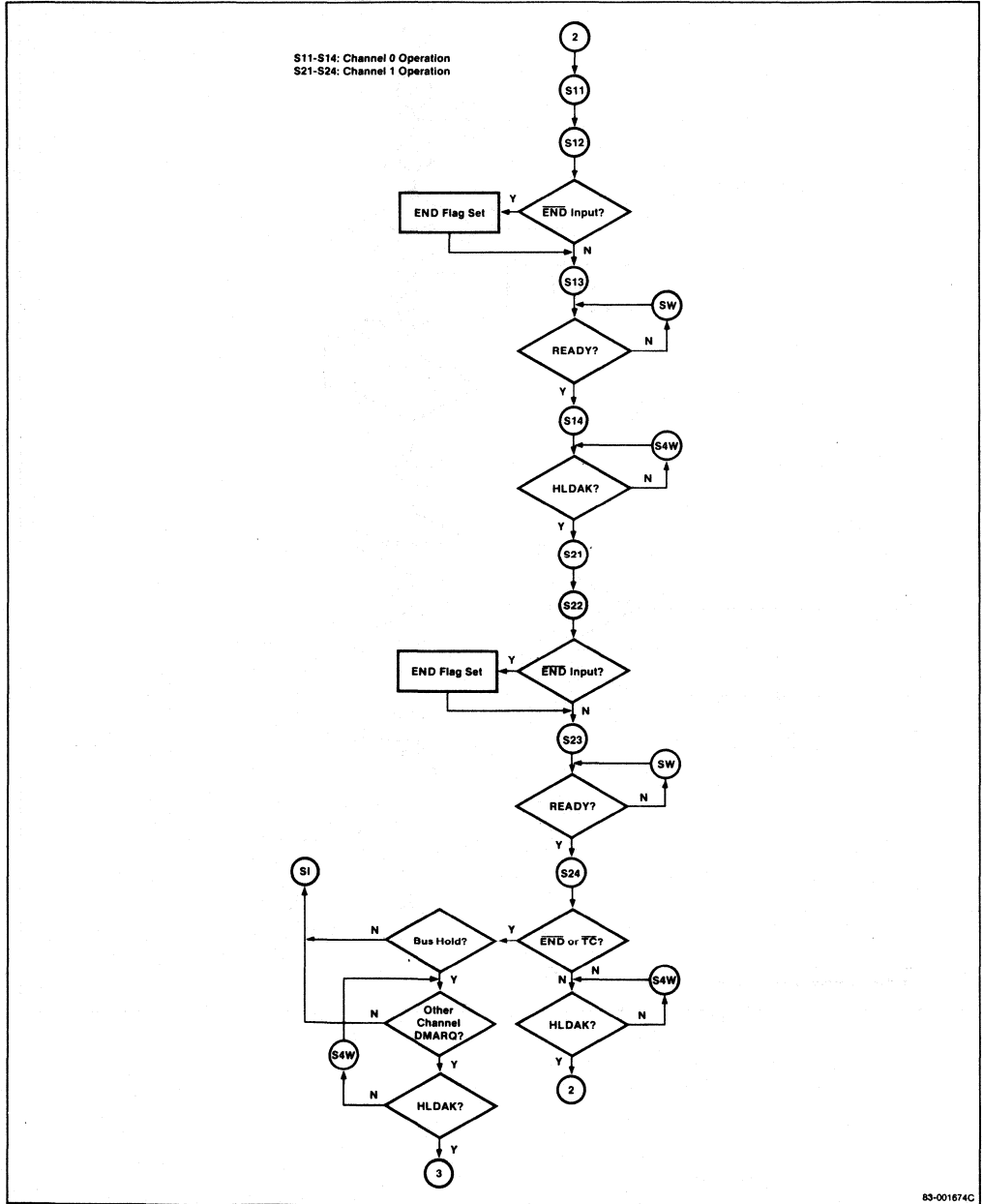
Figure 26. DMA Cycle, Block Mode



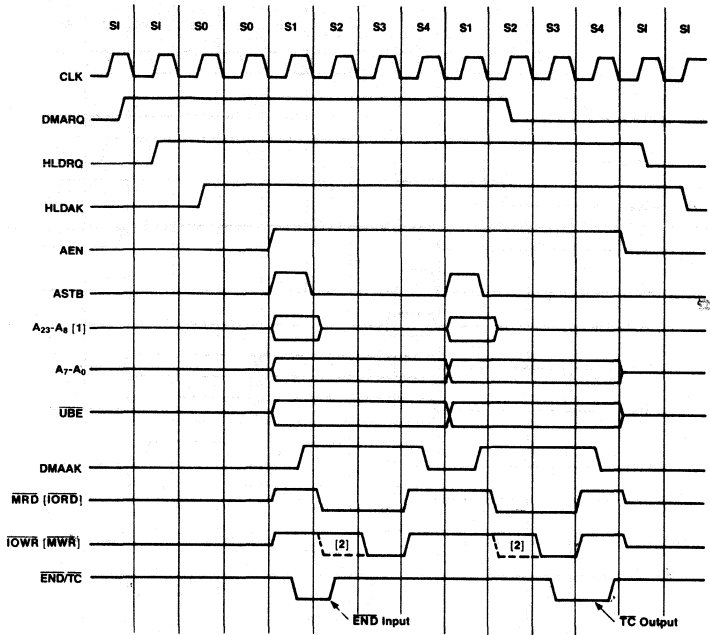
Note:

[1] Carry or borrow to upper two bytes of address?

Figure 27. DMA Cycle, Memory-to-Memory Transfer



**Figure 28. Memory-I/O Transfer, Normal Timing**

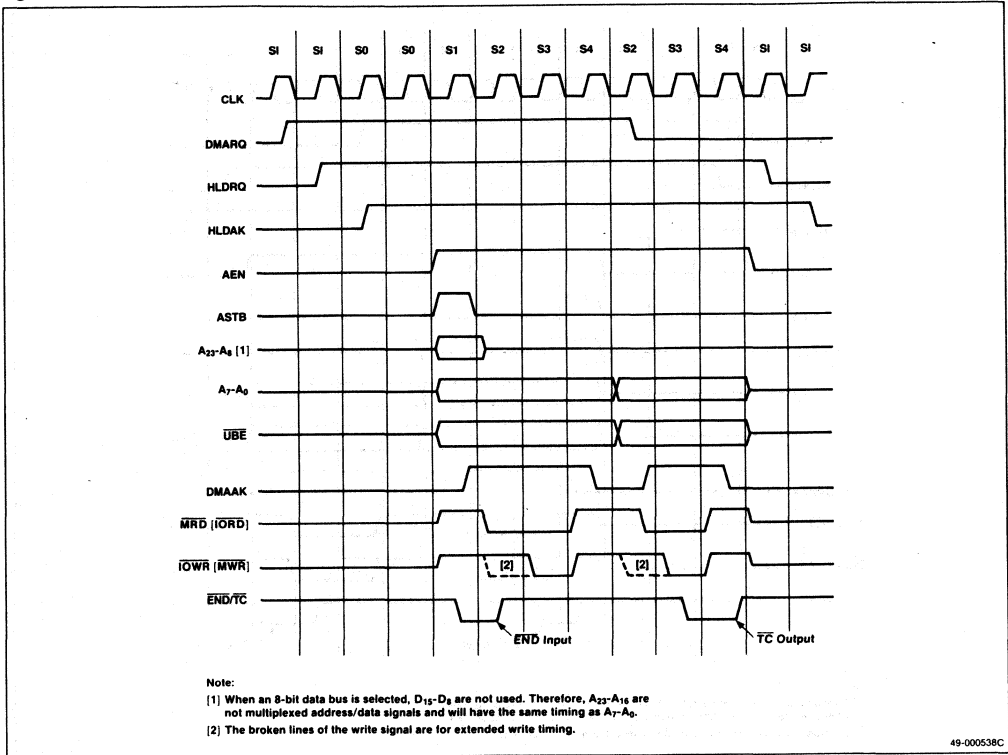


**Note:**  
 [1] When an 8-bit data bus is selected,  $D_{15}-D_8$  are not used. Therefore,  $A_{22}-A_{16}$  are not multiplexed address/data signals and will have the same timing as  $A_7-A_0$ .  
 [2] The broken lines of the write signal are for extended write timing.

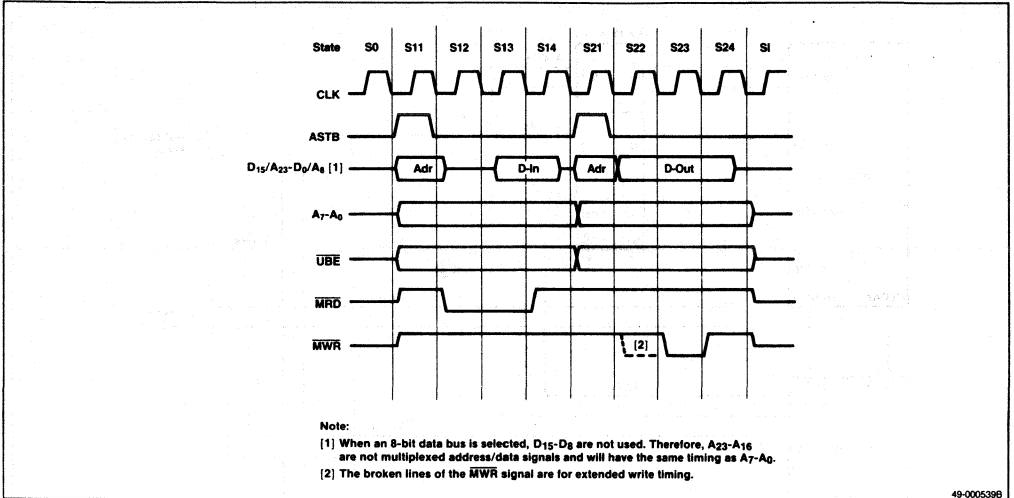
49-000537C



Figure 29. Memory-I/O Transfer, Compressed Timing



**Figure 30. Memory-to-Memory Transfer**



**Figure 31. END/TC Input/Output**

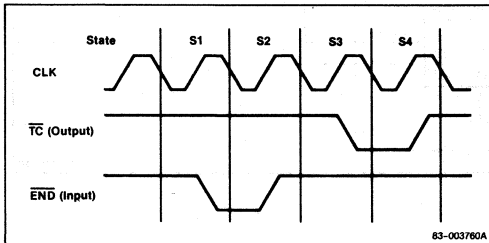
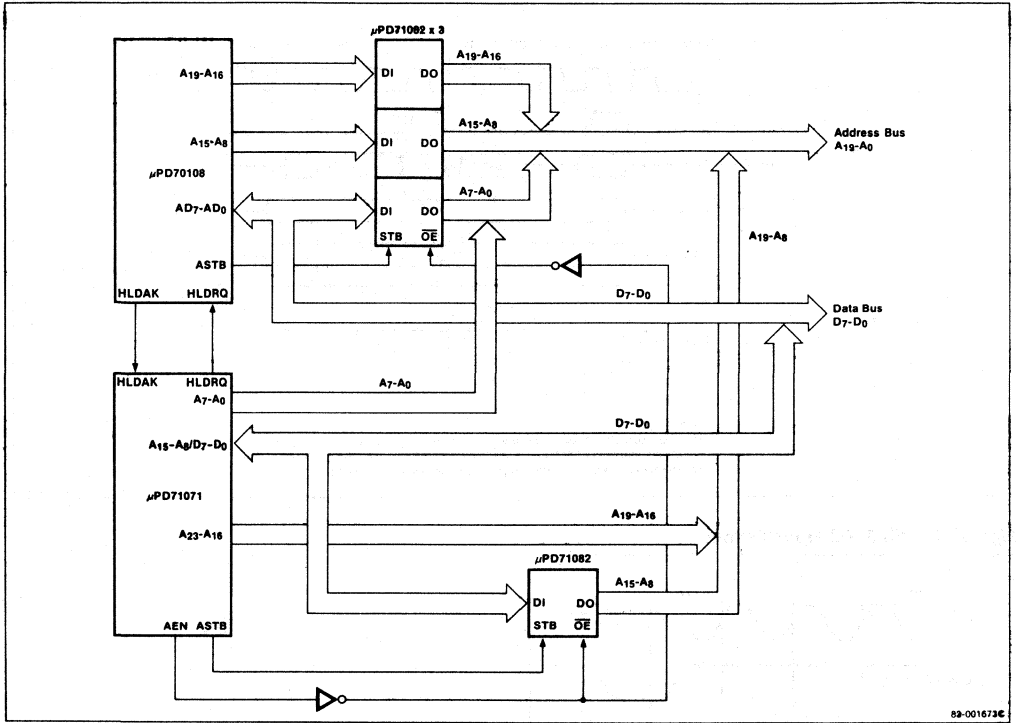
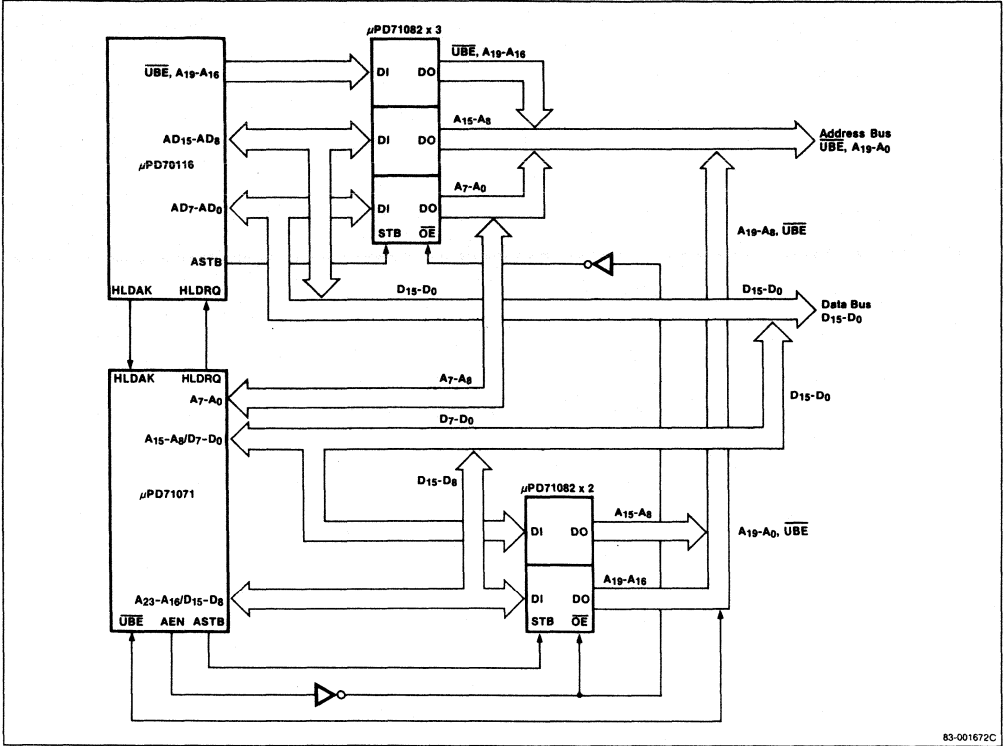


Figure 32. System Configuration with μPD70108



89-001673E

Figure 33. System Configuration with μPD70116



83-001672C



## Description

μPD71082 and μPD71083 are CMOS 8-bit transparent latches with three-state output buffers. They are used as bus buffers or bus multiplexers in microprocessor systems. Their high-drive capability makes them suitable for data latch, buffer, or I/O port applications.

## Features

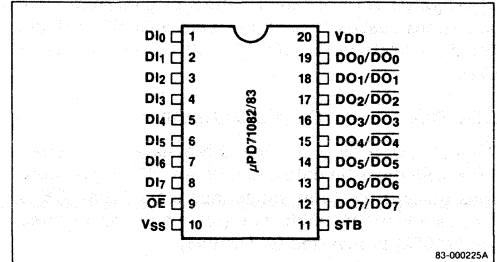
- CMOS technology
- 8-bit parallel data register
- Three-state output buffer
- High drive capability output buffer ( $I_{OL} = 12 \text{ mA}$ )
- μPD8085A, 8048, 8086, 8088, μPD70108/116 and μPD70208/216
- μPD71082 — non-inverted output;  
μPD71083 — inverted output
- Single +5 V ±10% power supply
- Transparent operation
- Industrial temperature range: -40 to +85°C

## Ordering Information

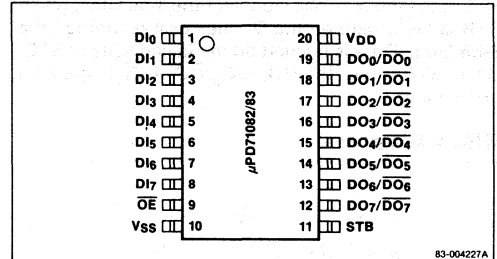
Part Number	Package Type	Output
μPD71082C	20-pin plastic DIP	Non-inverted
μPD71082G	20-pin plastic SO	
μPD71083C	20-pin plastic DIP	Inverted
μPD71083G	20-pin plastic SO	

## Pin Configurations

### 20-Pin Plastic DIP



### 20-Pin Plastic SO



## Pin Identification

Symbol	Function
D17-D10	Data input, bits 7-0
D07-D00/ $\overline{D07-D00}$	Data output, bits 7-0; non-inverted (μPD71082) or inverted (μPD71083)
STB	Strobe input
$\overline{OE}$	Output enable input
VDD	+5 V power supply
VSS	Ground

**Pin Functions**

**DI<sub>7</sub>-DI<sub>0</sub> [Data Input]**

DI<sub>7</sub>-DI<sub>0</sub> are data input lines to the 8-bit data latch. Data on DI lines passes through the latch while STB is high. The data is latched to DO/DO with the falling edge of STB.

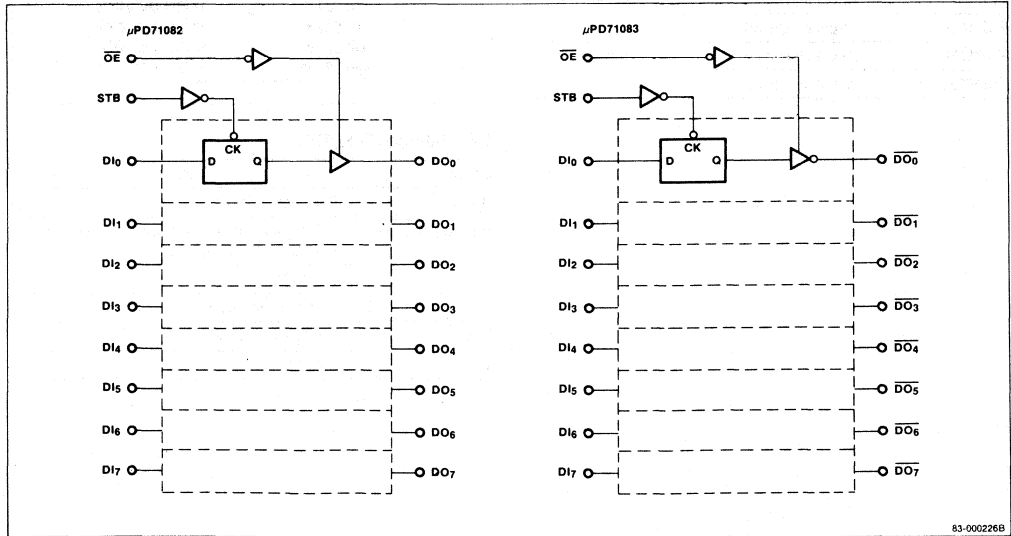
**DO<sub>7</sub>-DO<sub>0</sub>/ $\overline{DO_7-\overline{DO_0}}$  [Data Output]**

DO<sub>7</sub>-DO<sub>0</sub>/ $\overline{DO_7-\overline{DO_0}}$  are the three-state data output lines from the 8-bit data latch. When OE is high, these lines go into the high-impedance state. When OE is low, data from the latch is output, either non-inverted (μPD71082) or inverted (μPD71083).

**STB [Strobe]**

STB is the input strobe signal for the 8-bit latch. When STB is high, data on the DI lines passes through the 8-bit latch. Data is latched on the falling edge of STB. When STB is low, the DO<sub>7</sub>-DO<sub>0</sub>/ $\overline{DO_7-\overline{DO_0}}$  outputs do not change.

**Block Diagram**



**$\overline{OE}$  [Output Enable]**

$\overline{OE}$  input is the output enable signal for the three-state DO/ $\overline{DO}$  lines. When  $\overline{OE}$  is high, DO/ $\overline{DO}$  lines are high impedance. When  $\overline{OE}$  is low, data from the 8-bit latch is output to DO<sub>7</sub>-DO<sub>0</sub>/ $\overline{DO_7-\overline{DO_0}}$ . See table 1.

**Table 1. Latch Operation**

STB	$\overline{OE}$	DO <sub>7</sub> -DO <sub>0</sub> / $\overline{DO_7-\overline{DO_0}}$	8-Bit Data Latch
Low	Low	Latched data from 8-bit data latch is enabled	DI line data has been latched with falling edge of STB.
	High	High impedance	
High	Low	Data on DI <sub>7</sub> -DI <sub>0</sub>	DI passes through to DO/ $\overline{DO}$
	High	High impedance	

## Functional Description

The μPD71082 and μPD71083 are 8-bit data latches strobed by the STB signal. They have high-drive capability output buffers controlled by the  $\overline{OE}$  signal. Data on the DI lines is latched by the trailing edge of STB (high to low). When STB is high, data passes through the latch. When  $\overline{OE}$  is high, DO lines are high impedance. When  $\overline{OE}$  is low, the contents of the latches are output on DO<sub>7</sub>-DO<sub>0</sub>. The DO lines are isolated from  $\overline{OE}$  switching noise.

## Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$ ,  $V_{SS} = 0\text{ V}$

Power supply voltage, $V_{DD}$	-0.5 to +7.0 V
Input voltage, $V_I$	-1.0 to $V_{DD} + 1\text{ V}$
Output voltage, $V_O$	-0.5 to $V_{DD} + 0.5\text{ V}$
Power dissipation, $P_{D\text{MAX}}$ , DIP	500 mW
Power dissipation, $P_{D\text{MAX}}$ , SO	200 mW
Operating temperature, $T_{\text{opt}}$	-40°C to +85°C
Storage temperature, $T_{\text{stg}}$	-65°C to +150°C

**Comment:** Exposing the device to stresses above those listed in the absolute maximum ratings could cause permanent damage. Exposure to absolute maximum ratings for extended periods may affect device reliability.

## DC Characteristics

$T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD} = 5\text{ V} \pm 10\%$

Parameter	Symbol	Limits		Units	Test Conditions
		Min	Max		
Input voltage high	$V_{IH}$	2.2		V	$V_{OL} = 0.45\text{ V}$ $V_{OH} = V_{DD} - 0.8\text{ V}$
Input voltage low	$V_{IL}$	0.8		V	$V_{OL} = 0.45\text{ V}$ $V_{OH} = V_{DD} - 0.8\text{ V}$
Output voltage high	$V_{OH}$	$V_{DD} - 0.8$		V	$I_{OH} = -4\text{ mA}$
Output voltage low	$V_{OL}$	0.45		V	$I_{OL} = 12\text{ mA}$
Input current	$I_I$	-1.0	1.0	μA	$V_I = V_{DD}$ , $V_{SS}$
Leakage current, high impedance	$I_{\text{OFF}}$	-10	10	μA	$\overline{OE} = V_{DD}$
Power supply current (static)	$I_{DD}$		80	μA	$V_I = V_{DD}$ , $V_{SS}$
Power supply current (dynamic)	$I_{DD\text{dyn}}$		20	mA	$f_{\text{in}} = 10\text{ MHz}$

## Capacitance

$T_A = 25^\circ\text{C}$ ,  $V_{DD} = +5\text{ V}$

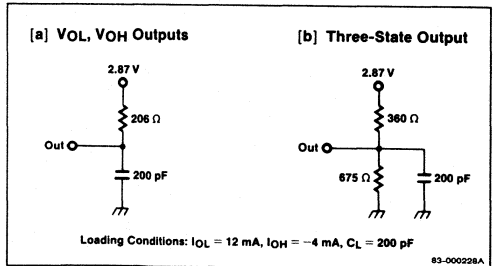
Parameter	Symbol	Limits		Units	Test Conditions
		Min	Max		
Input capacitance	$C_{\text{in}}$		12	pF	$f = 1\text{ MHz}$

## AC Characteristics

$T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD} = 5\text{ V} \pm 10\%$

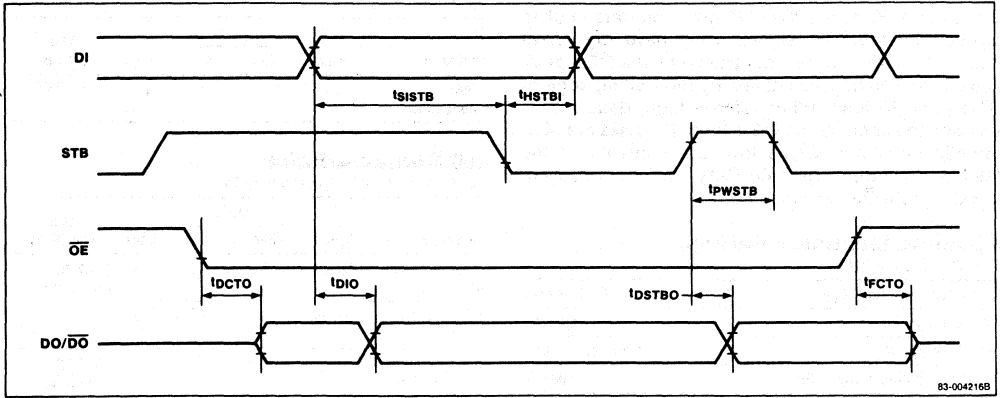
Parameter	Symbol	Limits		Units	Test Conditions
		Min	Max		
Input to output delay	$t_{DIO}$	5	40	ns	Loading Circuit (a)
STB to output delay	$t_{\text{OSTBO}}$	10	60	ns	
Data float time from $\overline{OE}$ high	$t_{\text{FCTO}}$	5	30	ns	Loading Circuit (b)
Data output delay from $\overline{OE}$ low	$t_{\text{OCTO}}$	10	40	ns	
Input to STB setup time	$t_{\text{SISTB}}$	0		ns	Loading Circuit (a)
Input to STB hold time	$t_{\text{HSTBI}}$	25		ns	
STB high pulse width	$t_{\text{PWSTB}}$	20		ns	
Signal rise time	$t_{\text{LH}}$		20	ns	0.8 V to 2.0 V
Signal fall time	$t_{\text{HL}}$		12	ns	2.0 V to 0.8 V

## Loading Circuits for AC Testing



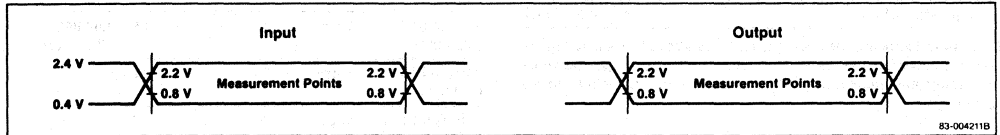


**Timing Waveforms**



83-004216B

**Timing Measurement Points**



83-004211B

## Description

The  $\mu$ PD71084 is a clock pulse generator/driver for the V20™/V30™ microprocessors and their peripherals using NEC's high-speed CMOS technology.

## Features

- CMOS technology
- Clock pulse generator/driver for  $\mu$ PD70108/70116 or other CMOS or NMOS CPUs and their peripherals
- Frequency source can be crystal or external clock input
- Reset signal with Schmitt-trigger circuit for CPU or peripherals
- Bus ready signal with two-bus system synchronization
- Clock synchronization with other  $\mu$ PD71084s
- Single +5 V  $\pm$ 10% power supply
- Industrial temperature range: -40 to +85°C

## Ordering Information

Part Number	Package Type	Max Clock Frequency
$\mu$ PD71084C	18-pin plastic DIP	8 MHz
$\mu$ PD71084G	20-pin plastic SO	8 MHz

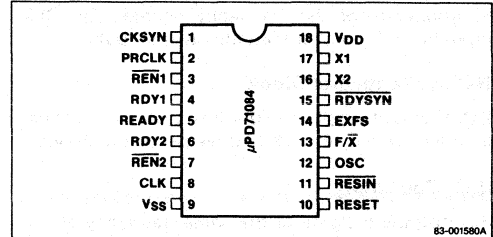
## Pin Identification

Symbol	Function
CKSYN	Clock synchronization input
PRCLK	Peripheral clock output
$\overline{\text{REN1}}$	Bus ready enable input 1
RDY1	Bus ready input 1
READY	Ready output
RDY2	Bus ready input 2
$\overline{\text{REN2}}$	Bus ready enable input 2
CLK	Processor clock output
V <sub>SS</sub>	Ground potential
RESET	Reset output
$\overline{\text{RESIN}}$	Reset input
OSC	Oscillator output
F/ $\overline{\text{X}}$	External frequency source/crystal select
EXFS	External frequency source input
RDYSYN	Ready synchronization select input
X2	Crystal input
X1	Crystal input
V <sub>DD</sub>	+5-volt power supply
NC	No connection

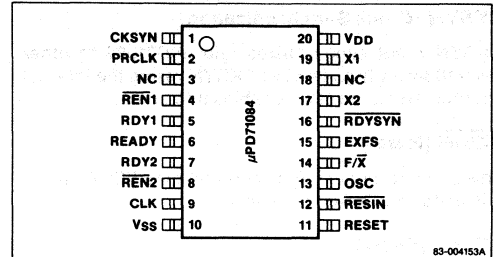
V20 and V30 are trademarks of NEC Corporation.

## Pin Configuration

### 18-Pin Plastic DIP



### 20-Pin Plastic SO (Small Outline)



## Pin Functions

### X1, X2 [Crystal]

When the F/ $\overline{\text{X}}$  input is low, a crystal connected to X1 and X2 will be the frequency source to generate clocks for a CPU and its peripherals. The crystal frequency should be three times the frequency of CLK.

### EXFS [External Frequency]

EXFS is the external frequency input in the external TTL frequency source mode (F/ $\overline{\text{X}}$  high). A TTL-level clock signal three times the frequency of CLK's output should be used for the source.

### F/ $\overline{\text{X}}$ [Frequency/Crystal Select]

F/ $\overline{\text{X}}$  input selects whether an external TTL-level input or an external crystal input is the frequency source of the CLK output. When F/ $\overline{\text{X}}$  is low, CLK is generated from the crystal connected to X1 and X2. When F/ $\overline{\text{X}}$  is high, CLK is generated from an external TTL-level frequency input on the EXFS pin. At the same time, the internal oscillator circuit will stop and the OSC output will be high.

**CLK [Processor Clock]**

CLK output supplies the CPU and its local bus peripherals. CLK is a 33% duty cycle clock of one-third the frequency of the frequency source. The CLK output is +0.4 V higher than the other outputs.

**PRCLK [Peripheral Clock]**

PRCLK output supplies a 50% duty cycle clock at one-half the frequency of CLK to drive peripheral devices.

**OSC [Oscillator]**

OSC outputs a signal at the same frequency as the crystal input. When EXFS is selected, the OSC output is powered down, and its output will be high.

**CKSYN [Clock Synchronization]**

CKSYN input synchronizes one μPD71084 to other μPD71084s. A high level at CKSYN resets the internal counter, and a low level enables it to count.

**RESIN [Reset]**

This Schmitt-trigger input generates the RESET output. It is used as a power-on reset.

**RESET [Reset]**

This output is a reset signal for the CPU. Reset timing is provided by the RESIN input to a Schmitt-trigger input gate and a flip-flop which will synchronize the reset timing to the falling edge of CLK. Power-on reset can be provided by a simple RC circuit on the RESIN input.

**RDY1, RDY2 [Bus Ready]**

A peripheral device drives the RDY1 or RDY2 inputs to signal that the data on the system bus has been received or is ready to be sent. REN1 and REN2 enable the RDY1 and RDY2 signals.

**REN1, REN2 [Address Enable]**

REN1 and REN2 inputs qualify their respective RDY inputs.

**RDYSYN [Ready Synchronization Select]**

RDYSYN input selects the mode of READY signal synchronization. A low-level signal makes the synchronization a two-step process. Two-step synchronization is used when RDY1 or RDY2 are not synchronized to the microprocessor clock and therefore cannot be

guaranteed to meet the READY setup time. A high-level signal makes synchronization a one-step process. One-step synchronization is used when RDY1 and RDY2 are synchronized to the processor clock. See Block Diagram.

**READY [Ready]**

The READY output signal to the processor is synchronized by the RDY inputs to the processor CLK. READY is cleared after RDY goes low and the guaranteed hold time of the processor has been met.

**Crystal**

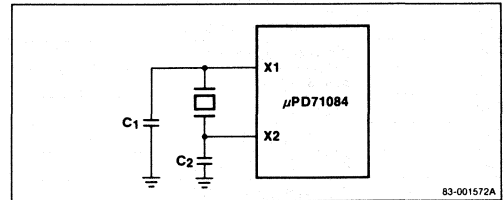
The oscillator circuit of the μPD71084 works with a parallel-resonant, fundamental mode, "AT-cut" crystal connected to pins X1 and X2.

Figure 1 shows the recommended circuit configuration. Capacitors C1 and C2 are required for frequency stability. The values of C1 and C2 (C1 = C2) can be calculated from the load capacitance (CL) specified by the crystal manufacturer.

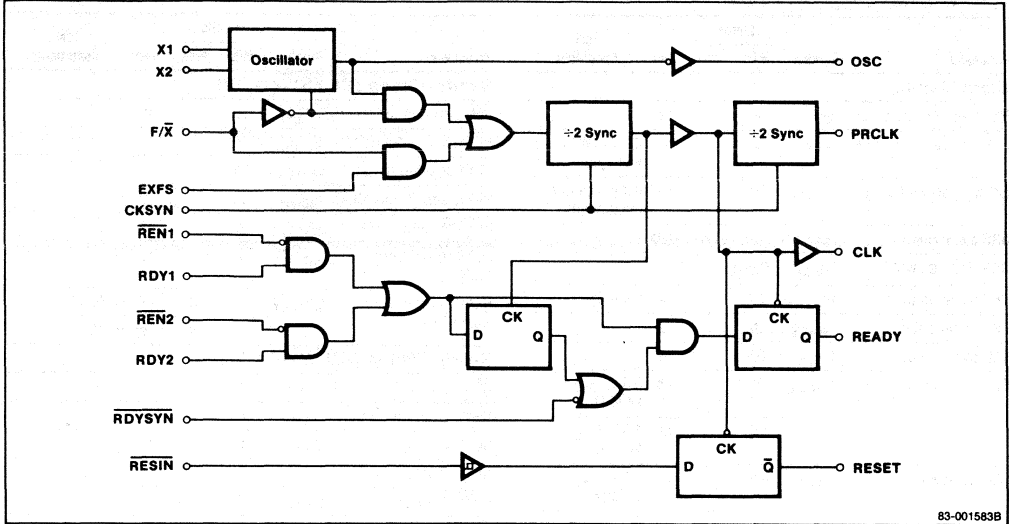
$$C_L = \frac{C_1 \times C_2}{C_1 + C_2} + C_S$$

CS is any stray capacitance in parallel with the crystal, such as the μPD71084 input capacitance CI.

Figure 1. Crystal Configuration Circuit



## Block Diagram



## Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$ ;  $V_{SS} = 0\text{ V}$

Power supply voltage, $V_{DD}$	-0.5 to +7.0 V
Input voltage, $V_I$	-1.0 V to $V_{DD} + 1.0\text{ V}$
Output voltage, $V_O$	-0.5 V to $V_{DD} + 0.5\text{ V}$
Operating temperature, $T_{OPT}$	-40 to +85°C
Storage temperature, $T_{STG}$	-65 to +150°C
Power dissipation, $P_D$ (DIP)	500 mW
Power dissipation, $P_D$ (SO package)	200 mW

**Comment:** Exposure to Absolute Maximum Ratings for extended periods may affect device reliability; exceeding the ratings could cause permanent damage. The device should be operated within the limits specified under DC and AC Characteristics.

## DC Characteristics

$T_A = -40$  to  $+85^\circ\text{C}$ ;  $V_{DD} = 5\text{ V} \pm 10\%$

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input voltage, high	$V_{IH}$	2.2		V	
			2.6	V	RESIN input
Input voltage, low	$V_{IL}$		0.8	V	
Output voltage, high	$V_{OH}$	$V_{DD}$		V	CLK output, $I_{OH} = -4\text{ mA}$
		-0.4		V	$I_{OH} = -4\text{ mA}$
Output voltage, low	$V_{OL}$		0.45	V	$I_{OL} = 4\text{ mA}$
Input leakage current	$I_{IN}$	-1.0	1.0	μA	
		-400	1.0	μA	RDYSYN input
RESIN hysteresis		0.25		V	
Power supply current (static)	$I_{DD}$		200	μA	
Power supply current (dynamic)	$I_{DDdyn}$		30	mA	$f_{in} = 24\text{ MHz}$

## Capacitance

$T_A = 25^\circ\text{C}$ ;  $V_{DD} = +5\text{ V}$

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input capacitance	$C_I$		12	pF	$f_C = 1\text{ MHz}$

**AC Characteristics**

T<sub>A</sub> = -40 to +85°C; V<sub>DD</sub> = +5 V ±10%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Clock Timing</b>					
EXFS input cycle time	t <sub>CYFS</sub>	50		ns	
EXFS pulse width, high	t <sub>PWFH</sub>	20		ns	2.2 V measurement point
EXFS pulse width, low	t <sub>PWFL</sub>	20		ns	0.8 V measurement point
OSC cycle time	t <sub>OSC</sub>	12	25	MHz	
CKSYN pulse width	t <sub>PWCT</sub>	2t <sub>CYFS</sub>			
CKSYN setup time	t <sub>HFSCTK</sub>	20		ns	from EXFS
CKSYN hold time	t <sub>SCTFS</sub>	20		ns	
CLK cycle time	t <sub>CYCK</sub>	125		ns	
CLK pulse width, high	t <sub>PWCKH</sub>	41		ns	3.0 V, f <sub>OSC</sub> = 24 MHz
		(1)		ns	1.5 V, f <sub>OSC</sub> = 24 MHz
CLK pulse width, low	t <sub>PWCKL</sub>	68		ns	1.5 V, f <sub>OSC</sub> = 10 MHz
		(2)		ns	1.5 V, f <sub>OSC</sub> = 16 MHz
CLK rise time	t <sub>LHCK</sub>	10		ns	1.5 V → 3.0 V, f <sub>OSC</sub> = 10 MHz
CLK fall time	t <sub>HLCK</sub>	10		ns	3.0 V → 1.5 V, f <sub>OSC</sub> = 10 MHz
CLK delay from OSC	t <sub>DCK</sub>	-5	22	ns	CLK↑
		2	35	ns	CLK↓
PRCLK cycle time	t <sub>CYPRK</sub>	250		ns	
PRCLK pulse width, high	t <sub>PWCKH</sub>	t <sub>CYCK</sub>		ns	-20
PRCLK pulse width, low	t <sub>PWCKL</sub>	t <sub>CYCK</sub>		ns	-20
PRCLK↑ delay from CLK↓	t <sub>DPRKH</sub>	22		ns	
PRCLK↓ delay from CLK↓	t <sub>DPRKL</sub>	22		ns	
<b>Reset Timing</b>					
RESIN setup to CLK↓	t <sub>SRIICK</sub>	65		ns	
RESIN hold from CLK↓	t <sub>HCKRI</sub>	20		ns	
RESET delay from CLK↓	t <sub>DCKRS</sub>	40		ns	

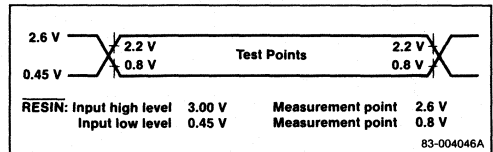
Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
<b>Ready Timing (RDYSYN = 'H')</b>					
REN1, 2 setup to RDY1, 2	t <sub>SRERY</sub>	15		ns	
REN1, 2 hold from CLK↓	t <sub>HCKRE</sub>	0		ns	
RDY1, 2 setup to CLK↓	t <sub>SRYCK</sub>	35		ns	
RDY1, 2 hold from CLK↓	t <sub>HCKRY</sub>	0		ns	
RDYSYN setup to CLK↓	t <sub>SRYSCK</sub>	50		ns	
RDYSYN hold from CLK↓	t <sub>HCKRYS</sub>	0		ns	
READY output delay from CLK↓	t <sub>DCKRDY</sub>	8		ns	READY↑
		8		ns	READY↓
<b>Ready Timing (RDYSYN = 'L')</b>					
REN1, 2 setup to RDY1, 2	t <sub>SRERY</sub>	15		ns	
REN1, 2 hold from CLK↓	t <sub>HCKRE</sub>	0		ns	
RDY1, 2 setup to CLK	t <sub>SRYCK</sub>	35		ns	
RDY1, 2 hold from CLK↓	t <sub>HCKRY</sub>	0		ns	
RDYSYN setup to CLK↓	t <sub>SRYSCK</sub>	50		ns	
RDYSYN hold from CLK↓	t <sub>HCKRYS</sub>	0		ns	
READY output delay from CLK↓	t <sub>DCKRDY</sub>	8		ns	READY↑
		8		ns	READY↓
<b>Output Pin Timing</b>					
Rise time	t <sub>LH</sub>	20		ns	0.8 V → 2.0 V
Fall time	t <sub>HL</sub>	12		ns	2.0 V → 0.8 V

**Note:**

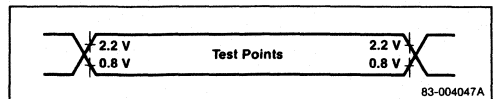
- (1) t<sub>PWCKH</sub> minimum = (1/3) t<sub>CYCK</sub> + 2
- (2) t<sub>PWCKL</sub> minimum = (2/3) t<sub>CYCK</sub> - 15

**Timing Waveforms**

**AC Test Input (except RESIN)**

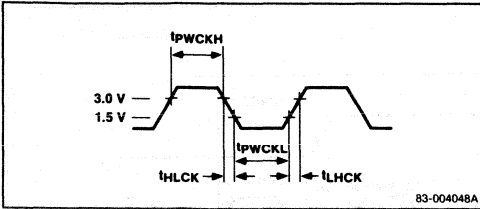


**AC Test Output (except CLK)**

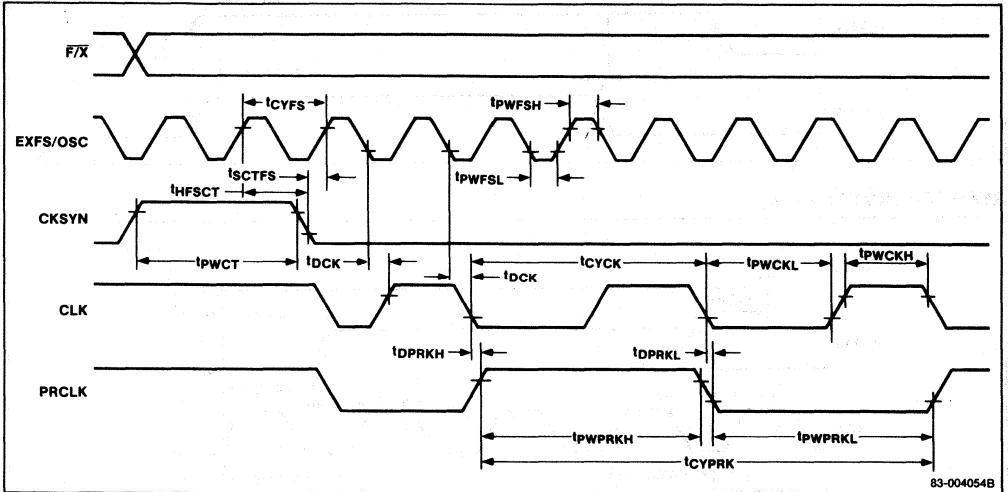


## Timing Waveforms (cont)

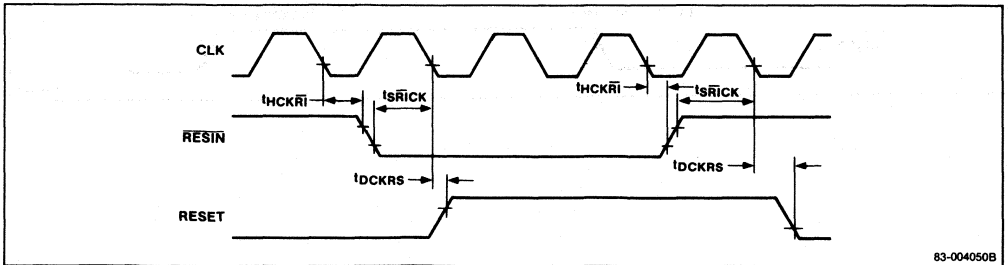
### AC Test Input (CLK)



### CLK Pin

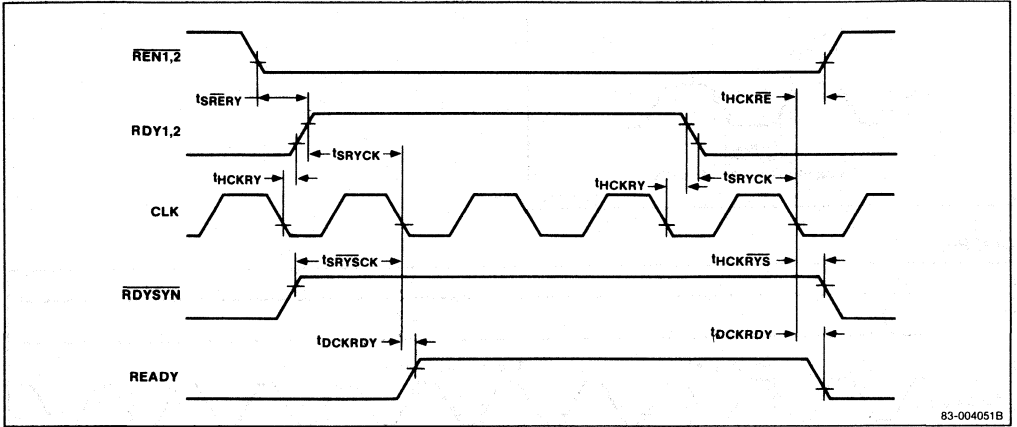


### RESET Pin

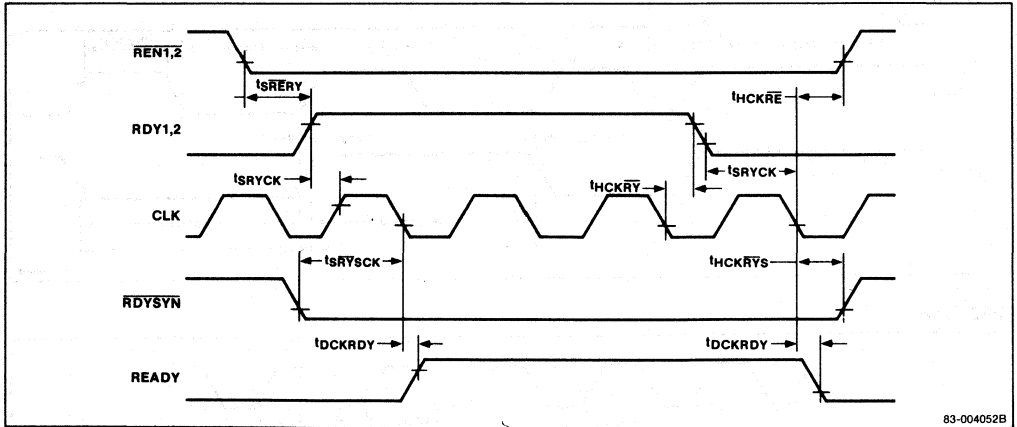


**Timing Waveforms (cont)**

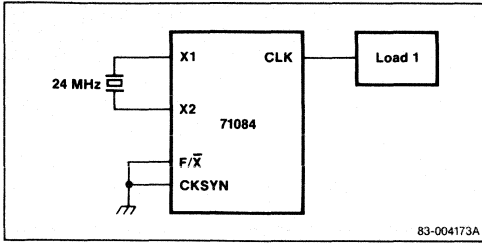
**READY Pin ( $\overline{RDYSYN} = 'H'$ )**



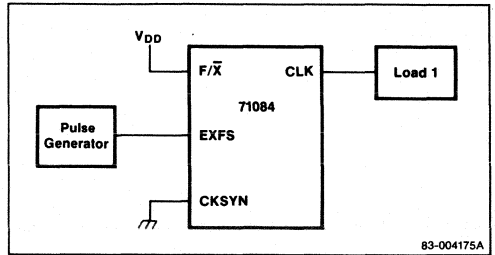
**READY Pin ( $\overline{RDYSYN} = 'L'$ )**



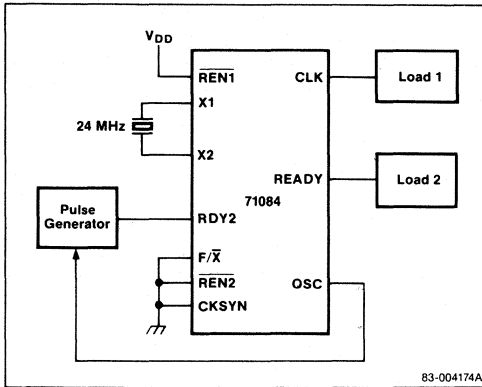
**Test Circuit for CLK High or Low Time  
(In Crystal Oscillation Mode)**



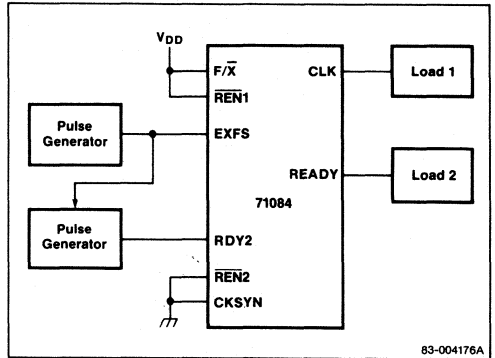
**Test Circuit for CLK High or Low Time  
(In EXFS Oscillation Mode)**



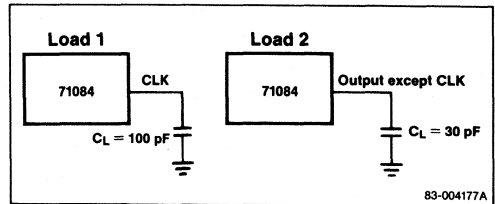
**Test Circuit for CLK to READY  
(In Crystal Oscillation Mode)**



**Test Circuit for CLK to READY  
(In EXFS Oscillation Mode)**



### Loading Circuits







### Description

μPD71086 and μPD71087 are 8-bit, bidirectional bus buffer/drivers with three-state outputs. The μPD71086 provides a non-inverted system bus. The μPD71087 provides an inverted system bus. These devices are used to expand CPU bus drive capability. The input/output lines are isolated from  $\overline{OE}$  and  $\overline{BUF\overline{R}/W}$  switching noise.

### Features

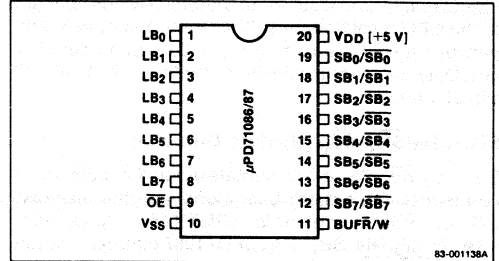
- CMOS technology
- Bidirectional 8-bit parallel bus buffer
- Three-state output
- High drive capability system bus output ( $I_{OL} = 12 \text{ mA}$ )
- Compatible with μPD70108/116, μPD70208/216 and other CMOS or NMOS designs
- μPD71086 — non-inverted system bus output
- μPD71087 — inverted system bus output
- Single +5 V ±10% power supply
- Industrial temperature range: -40 to +85°C

### Ordering Information

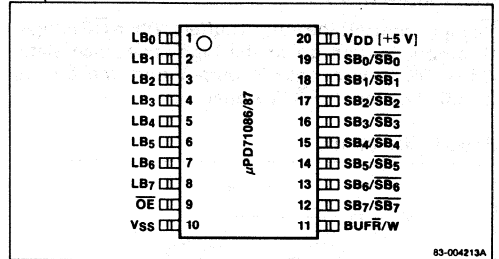
Part Number	Package Type	Output
μPD71086C	20-pin plastic DIP	Non-inverted
μPD71087C	20-pin plastic DIP	Inverted
μPD71086G	20-pin plastic SO	Non-inverted
μPD71087G	20-pin plastic SO	Inverted

### Pin Configurations

#### 20-Pin Plastic DIP



#### 20-Pin Plastic SO



### Pin Identification

Symbol	Function
LB7-LB0	CPU local I/O data bus, bits 7-0
SB7-SB0/ $\overline{SB7-SB0}$	System I/O data bus, bits 7-0; non-inverted (μPD71086) or inverted (μPD71087)
$\overline{OE}$	Output enable input
$\overline{BUF\overline{R}/W}$	Buffer read/write input
VDD	+5 V power supply
VSS	Ground

**Pin Functions**

**LB<sub>7</sub>-LB<sub>0</sub> [Local Data Bus]**

LB<sub>7</sub>-LB<sub>0</sub> are three state Inputs/Outputs which connect to the CPU local data bus. They input and output data between the CPU and memory, I/O, or other peripherals. Data read/write mode is controlled by the BUF $\bar{R}$ /W signal input.

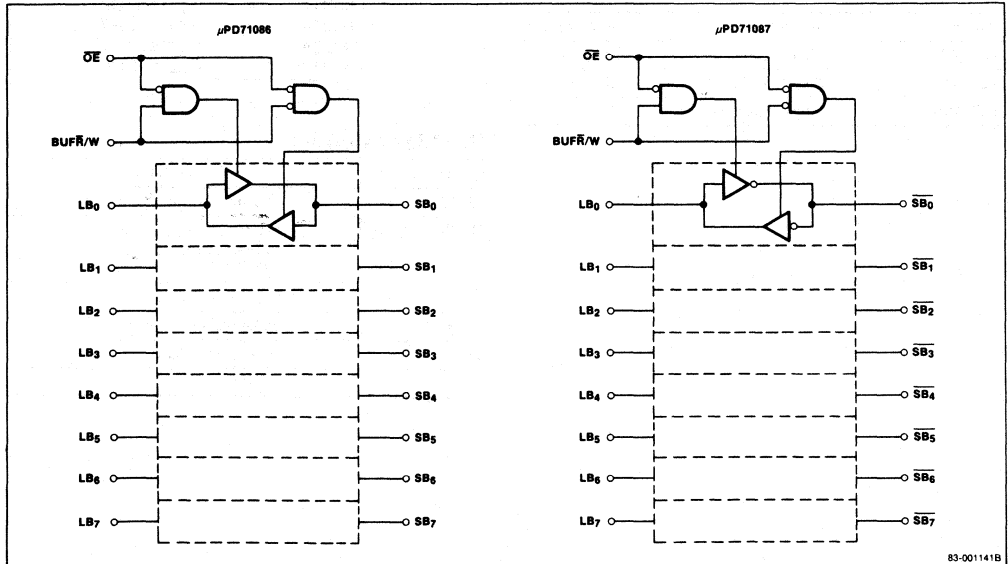
**SB<sub>7</sub>-SB<sub>0</sub>/ $\bar{S}B_7$ - $\bar{S}B_0$  [System Data Bus]**

SB<sub>7</sub>-SB<sub>0</sub>/ $\bar{S}B_7$ - $\bar{S}B_0$  are three state Inputs/Outputs which connect to the system bus, along with the memory, I/O, or other peripherals. μPD71086 outputs non-inverted signals, SB<sub>7</sub>-SB<sub>0</sub>. μPD71087 outputs inverted signals,  $\bar{S}B_7$ - $\bar{S}B_0$ .

**$\bar{O}E$  [Output Enable]**

$\bar{O}E$  input controls the output buffers. When  $\bar{O}E$  is high, all output buffers go to the high-impedance state. When  $\bar{O}E$  is low, data is output from the buffers specified by the BUF $\bar{R}$ /W signal.

**Block Diagram**



**BUF $\bar{R}$ /W [Buffer Read/Write]**

The data read/write mode is controlled by the BUF $\bar{R}$ /W signal input. When BUF $\bar{R}$ /W is high, LB lines are in input mode and SB lines are in output mode. When BUF $\bar{R}$ /W is low, SB lines are in input mode, and LB lines are output. See table 1.

**Table 1. Data Read/Write Mode**

$\bar{O}E$	BUF $\bar{R}$ /W	LB Pins	SB/ $\bar{S}B$ Pins	Mode
Low	Low	Output	Input	System bus to local bus
Low	High	Input	Output	Local bus to system bus
High	Low	—	—	High impedance
High	High	—	—	High impedance

**Note:** When  $\bar{O}E$  is high, all local and system bus pins go to high-impedance state.

## Absolute Maximum Ratings

( $T_A = 25^\circ\text{C}$ ,  $V_{SS} = 0\text{V}$ )

Power supply voltage, $V_{DD}$	-0.5 to +7.0 V
Input voltage, $V_i$	-1.0 to $V_{DD} + 1\text{V}$
Output voltage, $V_O$	-0.5 to $V_{DD} + 0.5\text{V}$
Power dissipation, $P_D$	500 mW
Operating temperature, $T_{opt}$	-40°C to +85°C
Storage temperature, $T_{stg}$	-65°C to +150 °C

**Comment:** Exposing the device to stresses above those listed in the absolute maximum ratings could cause permanent damage. Exposure to absolute maximum ratings for extended periods may affect device reliability.

## Capacitance

( $T_A = 25^\circ\text{C}$ ,  $V_{DD} = +5\text{V}$ )

Parameter	Symbol	Limits		Units	Test Conditions
		Min	Max		
Input Capacitance	$C_i$		24	pF	$f_c = 1\text{MHz}$

## DC Characteristics

( $T_A = -40^\circ\text{C}$  to +85°C,  $V_{DD} = 5\text{V} \pm 10\%$ )

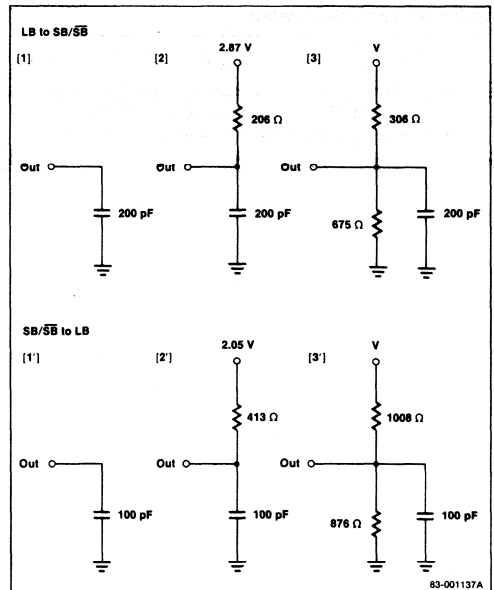
Parameter	Symbol	Limits		Units	Test Conditions
		Min	Max		
Input voltage high	$V_{IH}$	2.2		V	
Input voltage low	$V_{IL}$		0.8	V	
Output voltage high	$V_{OH}$	$V_{DD}$ -0.8		V	$I_{OH} = -4\text{mA}$
Output voltage low	$V_{OL}$	0.45		V	LB, $I_{OL} = 4\text{mA}$
Output voltage low	$V_{OL}$	0.45		V	SB, $I_{OL} = 12\text{mA}$
Input leakage current	$I_{iL}$	-1.0	1.0	μA	$V_i = V_{DD}, V_{SS}$
Leakage current, high impedance	$I_{OFF}$	-1.0	1.0	μA	$\overline{OE} = V_{DD}$
Power supply current (static)	$I_{DD}$	80		μA	$V_i = V_{DD}, V_{SS}$
Power supply current (dynamic)	$I_{DDdyn}$	40		mA	$f_{in} = 2\text{MHz}$

## AC Characteristics

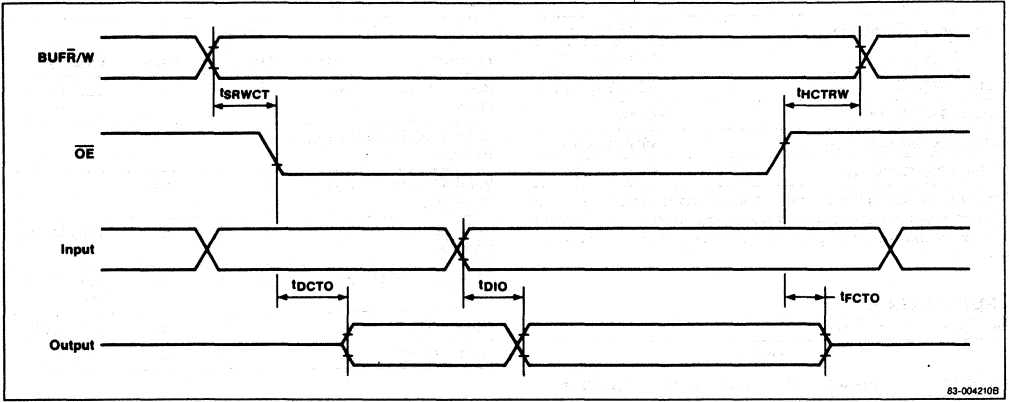
( $T_A = -40^\circ\text{C}$  to +85°C,  $V_{DD} = 5\text{V} \pm 10\%$ )

Parameter	Symbol	Limits		Units	Test Conditions
		Min	Max		
Input to output delay	$t_{DIO}$	5	40	ns	Load (1), (1') and (2), (2')
BUFR/W hold time from OE	$t_{HCTR W}$	5		ns	
BUFR/W setup time to OE	$t_{SRWCT}$	10		ns	
Data float time from OE	$t_{FCTO}$	5	30	ns	Load (3) and (3')
Data output delay from OE	$t_{DCTO}$	10	40	ns	Load (3) and (3')
Signal rise time	$t_R$		20	ns	0.8 to 2.0 V
Signal fall time	$t_F$		12	ns	2.0 to 0.8 V

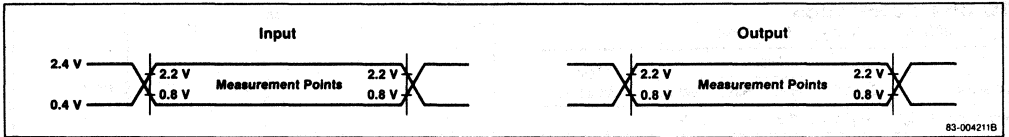
## Loading Circuit for AC Test



Timing Waveforms



Timing Measurement Points



## Description

The μPD71088 is a CMOS system bus controller for a μPD70108 (V20™) or μPD70116 (V30™) microprocessor system. It controls the memory or I/O system bus.

## Features

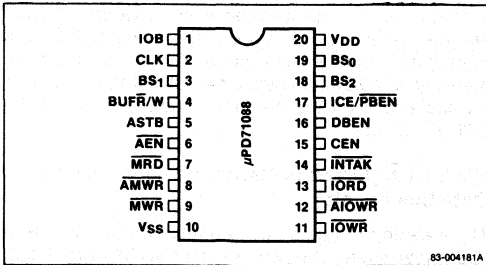
- CMOS technology
- Bus controller for microcomputer system expansion
- Command outputs for system bus control
- Control outputs for I/O peripheral bus control
- High drive capability for command and control outputs ( $I_{OL} = 12 \text{ mA}$ )
- Three-state outputs for command outputs
- Advanced I/O and memory write command outputs
- μPD70108, μPD70116 compatible
- +5-volt ±10% single power supply
- 20-pin plastic DIP (300 mil) or SO package
- Industrial temperature range: -40 to +85°C

## Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD71088C	20-pin plastic DIP	8 MHz
C-10	20-pin plastic DIP	10 MHz
G	20-pin plastic SO	8 MHz

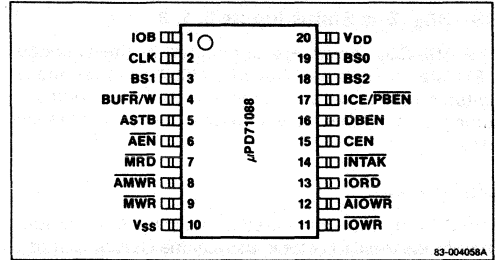
## Pin Configurations

### 20-Pin Plastic DIP



83-004181A

### 20-Pin Plastic SO



83-004058A

## Pin Identification

Symbol	Function
IOB	Input/output bus mode input
CLK	Clock input
BS <sub>1</sub>	Bus status input 1
BUF̄R/W	Buffer read/write output
ASTB	Address strobe output
AEN	Address enable input
MRD	Memory read output
AMWR	Advanced memory write output
MWR	Memory write command output
V <sub>SS</sub>	Ground
IOWR	I/O write command output
AIOWR	Advanced I/O write command output
IORD	I/O read command output
INTAK	Interrupt acknowledge output
CEN	Command enable input
DBEN	Data buffer enable output
ICE/PBEN	Interrupt cascade enable/Peripheral data bus enable output
BS <sub>2</sub>	Bus status input 2
BS <sub>0</sub>	Bus status input 0
V <sub>DD</sub>	Power supply

V20 and V30 are trademarks of NEC Corporation.

## Pin Functions

### **BS<sub>0</sub>-BS<sub>2</sub> [Bus Status Inputs 0, 1, 2]**

The BS<sub>0</sub>-BS<sub>2</sub> inputs are connected to the encoded CPU status outputs. The μPD71088 decodes these status outputs into command and control outputs for timing control. See table 1 for an explanation of these inputs.

### **CLK [Clock]**

The CLK input is connected to the same clock output that drives the CPU clock, usually the CLK output of a μPD71084 or a μPD71011. It is the internal system clock of the μPD71088.

### **ĀEN [Address Enable]**

The ĀEN input controls the command output buffers. When IOB is low, a low-level ĀEN causes the command buffers to output command output signals. A high-level ĀEN makes all command lines go to high impedance. When IOB is high, the μPD71088 is in I/O bus mode, and the command lines are not affected by ĀEN.

### **CEN [Command Enable]**

The CEN input controls DBEN,  $\overline{\text{PBEN}}$  and all command outputs. When CEN is high, all these outputs are active. When CEN is low, they are inactive.

### **IOB [I/O Bus Mode]**

When the IOB input is high, the bus control mode is I/O bus mode. When IOB is low, the bus control mode is system bus mode.

### **MRD [Memory Read Command]**

The MRD output is the signal to read data from a memory device. MRD is three-state, active low.

### **MWR [Memory Write Command]**

The MWR output is the signal to write data to a memory device. MWR is three-state, active low.

### **ĀMWR [Advanced Memory Write Command]**

This command output is the same as MWR, except that it is generated one state (clock cycle) earlier than MWR.

### **ĪORD [I/O Read Command]**

The ĪORD output is the signal to read data from an I/O device. ĪORD is three-state, active low.

### **ĪOWR [I/O Write Command]**

The ĪOWR output is the signal to write data to an I/O device. ĪOWR is three-state, active low.

### **ĀĪOWR [Advanced I/O Write Command]**

This command output is the same as ĪOWR, except that it is generated one state (clock cycle) earlier than ĪOWR.

### **ĪNTAK [Interrupt Acknowledge]**

The ĪNTAK output acknowledges interrupt requests. Requesting devices output an interrupt vector address in response to ĪNTAK. ĪNTAK is three-state, active low.

### **ASTB [Address Strobe]**

The ASTB output control signal latches the address outputs from the CPU into an external address latch, such as a μPD71082 or μPD71083. Address data should be strobed with the trailing edge (high to low) of ASTB.

### **DBEN [Data Buffer Enable]**

The DBEN output activates a data bus buffer/driver such as a μPD71086 or μPD71087 to input or output data between the CPU local bus and the memory or I/O system bus.

### **BUFR̄/W [Buffer Read/Write]**

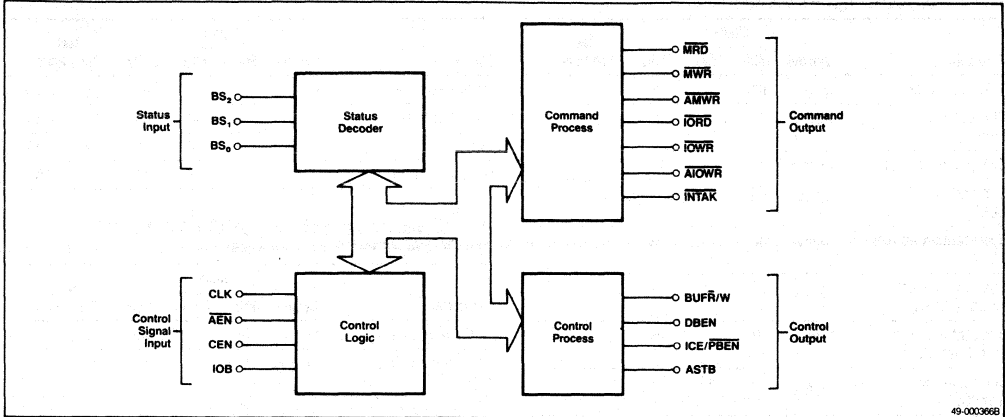
The BUFR̄/W output controls the direction in which data moves through a transceiver between the CPU and the memory or I/O peripherals. When BUFR̄/W is high, data is transferred from the CPU local bus to the memory or I/O system bus. When BUFR̄/W is low, data is transferred from the memory or I/O system bus to the CPU local bus.

### **ICE/ $\overline{\text{PBEN}}$ [Interrupt Cascade Enable/Peripheral Data Bus Enable]**

The meaning of this output signal depends on IOB. If IOB is low (system bus mode), it is the ICE output. ICE controls the cascade address transfer from a master priority interrupt controller to slave priority interrupt controllers. The slave reads the address from the master when ICE goes high.

When IOB is high, it becomes  $\overline{\text{PBEN}}$ .  $\overline{\text{PBEN}}$  controls the I/O bus the same way that DBEN controls the system bus. In this case, however, the output is active low.

## Block Diagram



## Absolute Maximum Ratings

T<sub>A</sub> = 25°C; V<sub>SS</sub> = 0 V

Power supply voltage, V <sub>DD</sub>	-0.5 to +7.0 V
Input voltage, V <sub>I</sub>	-1.0 to V <sub>DD</sub> + 1.0 V
Output voltage, V <sub>O</sub>	-0.5 to V <sub>DD</sub> ± 0.5 V
Operating temperature, T <sub>opt</sub>	-40 to +85°C
Storage temperature, T <sub>stg</sub>	-65 to +150°C
Power dissipation, P <sub>D</sub> (DIP)	500 mW
Power dissipation, P <sub>D</sub> (SO)	200 mW

**Comment:** Exposure to Absolute Maximum Ratings for extended periods may affect device reliability; exceeding the ratings could cause permanent damage. The device should be operated within the limits specified under DC and AC Characteristics.

## DC Characteristics

T<sub>A</sub> = -40°C to +85°C; V<sub>DD</sub> = 5 V ± 10%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input voltage, high	V <sub>IH</sub>	2.2		V	
Input voltage, low	V <sub>IL</sub>		0.8	V	
Output voltage, high	V <sub>OH</sub>	V <sub>DD</sub> - 0.8		V	Commands, I <sub>OH</sub> = -12 mA; Controls, I <sub>OH</sub> = -4 mA
Output voltage, low	V <sub>OL</sub>		0.45	V	Commands, I <sub>OL</sub> = 12 mA; Controls, I <sub>OL</sub> = 4 mA;
Input current leakage	I <sub>IL</sub>	-1.0	1.0	μA	V <sub>I</sub> = V <sub>DD</sub> , V <sub>SS</sub>
Leakage current at high impedance	I <sub>OFF</sub>	-10	10	μA	
Power supply current (static)	I <sub>DD</sub>		80	μA	V <sub>I</sub> = V <sub>DD</sub> , V <sub>SS</sub>
Power supply current (dynamic)	I <sub>DDdyn</sub>		20	mA	f <sub>in</sub> = 10 MHz



**AC Characteristics, μPD71088**

T<sub>A</sub> = -40 to +85°C; V<sub>DD</sub> = 5 V ±10%

Parameter	Symbol	Limits		Units	Test Conditions
		Min	Max		
CLK cycle time	t <sub>CYCK</sub>	125		ns	
CLK pulse width, high	t <sub>PWCKH</sub>	40		ns	
CLK pulse width, low	t <sub>PWCKL</sub>	60		ns	
Bus status setup to CLK↑	t <sub>SBSV</sub>	40		ns	
Bus status hold from CLK↓	t <sub>HBSV</sub>	10		ns	
Bus status inactive setup to CLK↓	t <sub>SBSIV</sub>	35		ns	
Bus status inactive hold from CLK↑	t <sub>HBSIV</sub>	10		ns	
Command active delay from CLK↓	t <sub>DCML</sub>	10	40	ns	
Command inactive delay from CLK↓	t <sub>DCMH</sub>	10	40	ns	
Command output on delay from AEN↓	t <sub>DAECM</sub>		40	ns	
Command active delay from AEN↓	t <sub>DAECML</sub>	100	295	ns	
Command active delay from CEN↑	t <sub>DCECM</sub>		t <sub>DCML</sub>	ns	
Command output float delay from AEN↑	t <sub>FAECM</sub>		50	ns	
ASTB active delay from CLK↓	t <sub>DCKSTH</sub>		30	ns	
ASTB active delay from BS2, 1, 0	t <sub>DBSST</sub>		25	ns	
ASTB inactive delay from CLK↑	t <sub>DCKSTL</sub>	7	25	ns	
DBEN, PBEN active delay from CLK↓	t <sub>DCTV</sub>	10	50	ns	
DBEN, PBEN inactive delay from CLK↑	t <sub>DCT</sub>	10	50	ns	
DBEN, PBEN active delay from AEN↓	t <sub>DAECTK</sub>		30	ns	
DBEN, PBEN active delay from CEN↑	t <sub>DCECT</sub>		30	ns	
BUFR/W↑ delay from CLK↑	t <sub>DCKWR</sub>		40	ns	
BUFR/W↓ delay from CLK↑	t <sub>DCKRD</sub>		60	ns	
ICE active delay from CLK↓	t <sub>DCKIC</sub>		30	ns	
ICE active delay from BS2, 1, 0	t <sub>DBSIC</sub>		25	ns	
ICE inactive delay from CLK↓	t <sub>DICL</sub>	10	50	ns	

Parameter	Symbol	Limits		Units	Test Conditions
		Min	Max		
Input rise time	t <sub>RI</sub>		20	ns	0.8 V → 2.0 V
Input fall time	t <sub>FI</sub>		12	ns	2.0 V → 0.8 V
Output rise time	t <sub>RO</sub>		20	ns	0.8 V → 2.0 V
Output fall time	t <sub>FO</sub>		12	ns	2.0 V → 0.8 V

**AC Characteristics, μPD71088-10**

T<sub>A</sub> = -10 to +70°C; V<sub>DD</sub> = 5 V ±10%

Parameter	Symbol	Limits		Units	Test Conditions	
		Min	Max			
CLK cycle time	t <sub>CYCK</sub>	100		ns		
CLK pulse width, high	t <sub>PWCKH</sub>	41		ns		
CLK pulse width, low	t <sub>PWCKL</sub>	49		ns		
Bus status setup to CLK↑	t <sub>SBSV</sub>		35	ns		
Bus status hold from CLK↓	t <sub>HBSV</sub>		10	ns		
Bus status inactive setup to CLK↓	t <sub>SBSIV</sub>		35	ns		
Bus status inactive hold from CLK↑	t <sub>HBSIV</sub>		10	ns		
Command active delay from CLK↓	t <sub>DCML</sub>		10	35	ns	
Command inactive delay from CLK↓	t <sub>DCMH</sub>		10	35	ns	
Command output on delay from AEN↓	t <sub>DAECM</sub>			40	ns	
Command active delay from AEN↓	t <sub>DAECML</sub>		115	200	ns	
Command active delay from CEN↑	t <sub>DCECM</sub>		t <sub>CLML</sub>		ns	
Command output float delay from AEN↑	t <sub>FAECM</sub>			40	ns	
ASTB active delay from CLK↓	t <sub>DCKSTH</sub>			20	ns	
ASTB active delay from BS2, 1, 0	t <sub>DBSST</sub>			20	ns	
ASTB inactive delay from CLK↑	t <sub>DCKSTL</sub>			7	25	ns
DBEN, PBEN active delay from CLK↓	t <sub>DCTV</sub>			10	35	ns
DBEN, PBEN inactive delay from CLK↑	t <sub>DCT</sub>			10	35	ns
DBEN, PBEN active delay from AEN↓	t <sub>DAECTK</sub>				30	ns
DBEN, PBEN active delay from CEN↑	t <sub>DCECT</sub>				30	ns

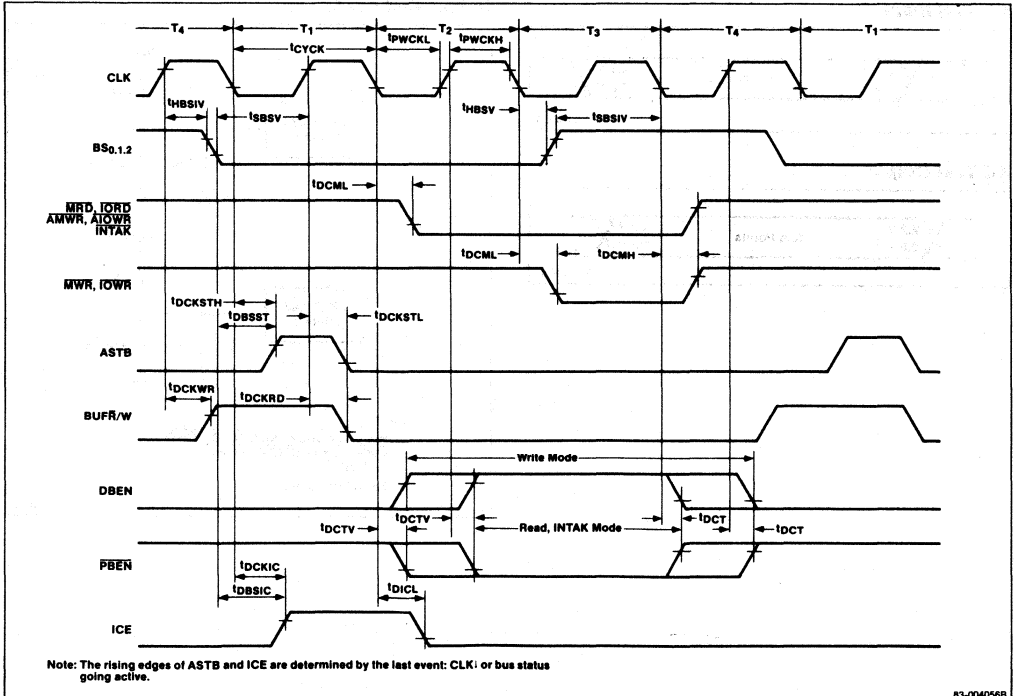
## AC Characteristics, μPD71088-10 (cont)

$T_A = -10$  to  $+70^\circ\text{C}$ ;  $V_{DD} = 5\text{ V} \pm 10\%$

Parameter	Symbol	Limits		Units	Test Conditions
		Min	Max		
BUFR/W↑ delay from CLK↑	$t_{\text{DCKWR}}$		40	ns	
BUFR/W↓ delay from CLK↑	$t_{\text{DCKRD}}$		40	ns	
ICE active delay from CLK↓	$t_{\text{DCKIC}}$		30	ns	
ICE active delay from BS2, 1, 0	$t_{\text{DBSIC}}$		20	ns	
ICE inactive delay from CLK↓	$t_{\text{DICL}}$	10	40	ns	
Input rise time	$t_{\text{RI}}$		20	ns	$0.8\text{ V} \rightarrow 2.0\text{ V}$
Input fall time	$t_{\text{FI}}$		12	ns	$2.0\text{ V} \rightarrow 0.8\text{ V}$
Output rise time	$t_{\text{RO}}$		20	ns	$0.8\text{ V} \rightarrow 2.0\text{ V}$
Output fall time	$t_{\text{FO}}$		12	ns	$2.0\text{ V} \rightarrow 0.8\text{ V}$

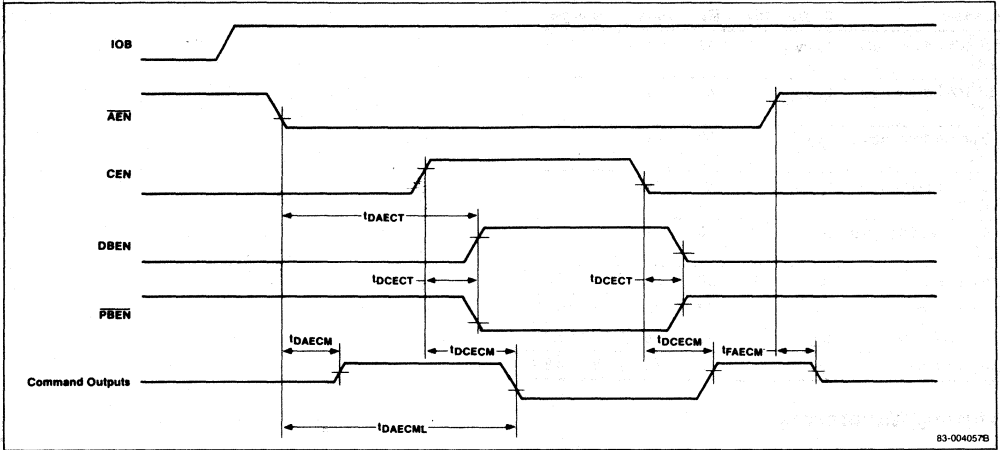
## Timing Waveforms

### General



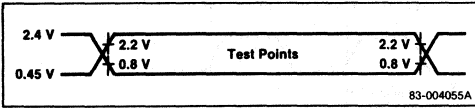
**Timing Waveforms (cont)**

**DBEN, PBEN, and Command Output**

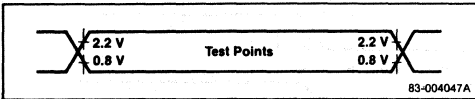


83-004057B

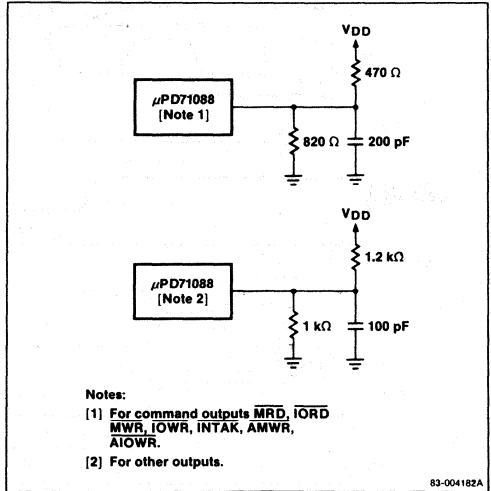
**AC Test Input**



**AC Test Output**



**Output Test Loads**



**Notes:**

[1] For command outputs  $\overline{MRD}$ ,  $\overline{IORD}$ ,  $\overline{MWR}$ ,  $\overline{IOWR}$ ,  $\overline{INTAK}$ ,  $\overline{AMWR}$ ,  $\overline{AIOWR}$ .

[2] For other outputs.

83-004182A

## Bus Controller Functional Description

### Command Logic

The μPD71088 decodes the CPU bus status outputs into command outputs. The bus status outputs (BS<sub>0</sub>-BS<sub>2</sub>) and their decoded commands are shown in table 1.

**Table 1. Command Logic**

BS <sub>2</sub>	BS <sub>1</sub>	BS <sub>0</sub>	CPU Status	μPD71088 Command Output
Low	Low	Low	Interrupt acknowledge	INTAK
Low	Low	High	I/O read mode	IORD
Low	High	Low	I/O write mode	IOWR, AIOWR
Low	High	High	Halt mode	None
High	Low	Low	Instruction fetch mode	MRD
High	Low	High	Memory read mode	MRD
High	High	Low	Memory write mode	MWR, AMWR
High	High	High	No bus cycle mode	None

### Bus Control Mode

The CEN, IOB, and  $\overline{\text{AEN}}$  signals control the bus controller mode as shown in table 2.

**Table 2. Bus Control Mode**

Control Input			Command Output		Control Output	
CEN	IOB	$\overline{\text{AEN}}$	Memory	I/O	$\overline{\text{ICE/PBEN}}$	ASTB, BUFR/W, DBEN
			MRD, MWR, AMWR	IOWR, AIOWR, IORD, INTAK		
H	H (I/O bus mode)	H	High impedance	Outputs enabled (NC)	$\overline{\text{PBEN}}$ (NC)	Outputs enabled (NC)
		L	Outputs enabled			
H	L (System bus mode)	H	High impedance	High impedance	ICE (NC)	Outputs enabled (NC)
		L	Outputs enabled	Outputs enabled		
L (Command disable mode)	x	x	H	H	$\overline{\text{PBEN}} = \text{H}$	Outputs enabled (DBEN = L: ASTB, BUFR/W are NC)

**Note:**

x = Don't care, NC = No change, H = High, L = Low



### Description

The μPD71611 is a high performance CMOS clock generator/driver for the μPD70616 (V60) microprocessor. The μPD71611 contains a crystal oscillator for use with a fundamental mode crystal. An external clock source can also be used if it is necessary to synchronize the μPD70616 to an external clock.

Reset logic with a Schmitt-trigger input is also included to synchronize a power-on reset signal to the system clock and be used as a system-wide reset signal. The μPD71611 also contains wait state logic for generation of 0 to 7 wait states into any bus cycle.

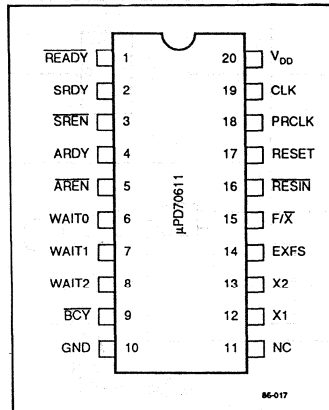
### Features

- ◆ μPD70616 system clock generator/driver
- ◆ 16 MHz system clock (32MHz input divided by two)
- ◆ Programmable wait state generator
- ◆ Schmitt-trigger reset logic
- ◆ Low power CMOS technology
- ◆ 20-pin plastic DIP

### Ordering Information

Part Number	Package	Maximum Frequency
μPD71611C	20-pin plastic DIP	16 MHz

### Pin Configuration





## Pin Identificaton

Symbol		I/O	Function
X1, X2	Crystal Inputs	I/O	I/O terminals for crystal, twice the μPD70616 frequency
EXFS	External Clock Source	In	External clock input pin, twice the μPD70616 frequency
F/X*	Frequency/Crystal Select	In	Clock source multiplexer control
CLK	Clock	Out	μPD70616 system clock, 1/2 the input clock frequency
PRCLK	Peripheral Clock	Out	Peripheral clock, 1/4 the input clock frequency
RESIN*	Reset Input	In	Schmitt trigger reset input
RESET	Reset Out	Out	μPD70616 and system reset output
ARDY	Asynchronous Ready	In	Asynchronous ready input
AREN*	Asynchronous Ready Enable	In	Enable input for ARDY
SRDY	Synchronous Ready	In	Synchronous ready input
SREN*	Synchronous Ready Enable	In	Enable input for SRDY
BCY*	Bus Cycle	In	μPD70616 bus cycle output
WAIT0 WAIT1 WAIT2	Wait State Controls	In	Wait state inputs to add 0 to 7 wait states to a bus cycle
READY*	Ready	Out	Ready output to the μPD70616

## Pin Functions

This section describes the operation of the μPD71611 I/O terminals. Inputs and outputs are considered at a logic "0" level when a low level signal is present. Likewise, a logic "1" is represented by a high level signal. Bus states are defined and measured from the rising edge of the clock to the rising edge of the next clock.

**X1, X2 [Crystal Inputs]** ..... inputs/outputs

The X1 and X2 inputs are used to connect a fundamental mode crystal to the internal oscillator circuitry. The crystal is selected to oscillate at a frequency twice the μPD70616 operating frequency.

**EXFS [External Frequency Source]** ..... input

The EXFS pin allows an external TTL-level clock source twice the desired operating frequency to be used as the input clock.

**F/X\* [Frequency/Crystal Select]** ..... input

The F/X\* input selects either the internal crystal oscillator or an external clock oscillator as the clock source for the μPD70616 and other devices.

**CLK [Clock]** ..... output

CLK is the μPD70616 processor clock output. The CLK output has additional drive capabilities matched to the requirements of the μPD70616 and associated microprocessors.

**PRCLK [Peripheral Clock]** ..... output

CLK is the μPD70616 system clock output available for use by peripheral devices.



**RESIN\* [Reset Input]** ..... input

RESIN\* is a Schmitt-trigger input used to synchronize the asynchronous reset input to the rising edge of the processor clock before use by the rest of the system.

**RESET [Reset Output]** ..... output

RESET is an active high synchronized reset signal for the μPD70616 and peripheral devices.

**ARDY [Asynchronous Ready Input]** ..... input

The ARDY input is used by ready signals which are not synchronized to the μPD70616 clock. This input is internally applied to a synchronizer before being used as a ready output to the processor.

**AREN\* [Asynchronous Ready Enable]** ..... input

AREN\* is an enable for the ARDY input. If ARDY is not used in a system, AREN\* can be pulled up to the positive power supply to disable the asynchronous ready logic.

**SRDY [Synchronous Ready Input]** ..... input

The SRDY input is used by ready signals which are synchronized to the μPD70616 clock.

**SREN\* [Synchronous Ready Enable]** ..... input

SREN\* is an enable for the SRDY input. If SRDY is not used in a system, SREN\* can be pulled up to the positive power supply to disable the synchronous ready logic.

**BCY\* [Bus Cycle]** ..... input

The μPD70616 indicates the start of a new bus cycle by asserting the BCY\* output. The μPD71611 uses BCY\* to control the operation of the wait state logic.

**WAIT2-0 [Wait State Mode Inputs]** ..... inputs

The WAIT2-0 inputs are decoded to determine the number of wait states the μPD71611 should automatically insert into the bus cycle. These inputs can be dynamically changed on a bus cycle basis to insert from 0 to 7 seven wait states into a bus cycle.

**READY\* [Ready Output]** ..... output

READY\* is an active low output used by slow memory and peripheral devices to insert wait states into a μPD70616 bus cycle.

**VDD [Power Supply]**

The VDD pin supplies +5 Volt power to the μPD71611.

**GND [Ground]**

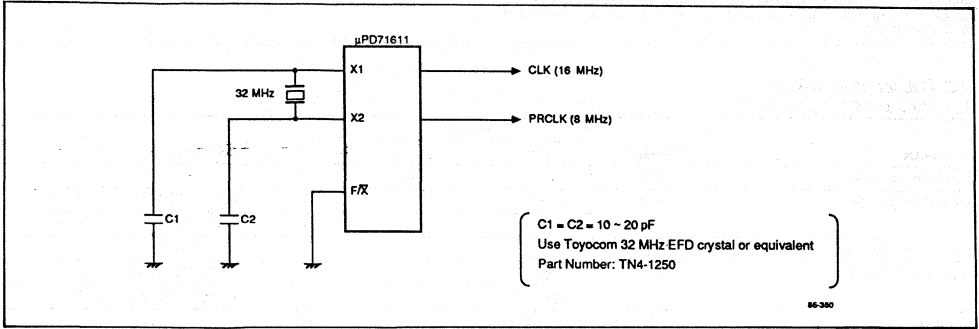
The GND pin is the power supply return.

### Operational Description

The μPD71611 serves as the system clock generator/driver for the μPD70616 (V60) microprocessor. The primary clock source can be selected from an internal crystal oscillator and fundamental mode crystal or from an external TTL-level clock. The source clock is then divided by two to generate the μPD70616 clock (CLK) and by four to generate the peripheral clock (PRCLK).

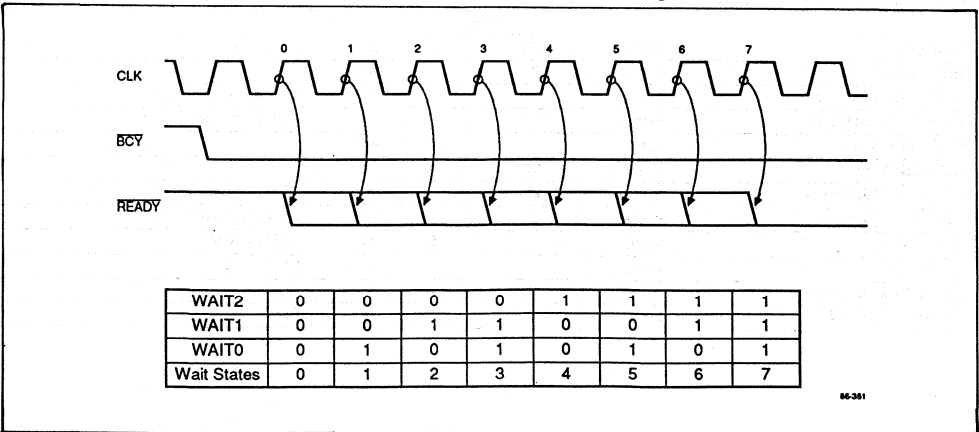
The μPD71611 contains additional logic for synchronization of a reset input and the generation of processor wait states. Figure 1 shows a typical μPD71611 design example using the on-chip clock oscillator logic.

Figure 1. Crystal Oscillator Example



Three different types of wait states are accommodated by the μPD71611. Both synchronous and asynchronous wait ready inputs are available, each with a corresponding enable input. Zero to seven wait states can also be automatically inserted into a bus cycle using the WAIT2-0 inputs.

Figure 2. WAIT2-0 and READY\* Timing



**Absolute Maximum Ratings**

T<sub>A</sub> = +25°C

Power Supply Voltage, V <sub>DD</sub>	-0.5 V to +7.0 V
Input Voltage, V <sub>I</sub>	-1.0 V to V <sub>DD</sub> + 1.0 V
Output Voltage, V <sub>O</sub>	-0.5 V to V <sub>DD</sub> + 0.5 V
Operating Temperature, T <sub>OPT</sub>	-40°C to +85°C
Storage Temperature, T <sub>STG</sub>	-65°C to +150°C
Power Dissipation, P <sub>D</sub>	500mW

Comment: Exposing the device to stresses above those listed in the Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Capacitance**

T<sub>A</sub> = +25°C, V<sub>DD</sub> = 0 V

Parameter	Symbol	Limits		Unit	Conditions
		Min	Max		
Input Capacitance	C <sub>I</sub>		10	pf	f = 1 MHz

**DC Characteristics**

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V ±5%

Parameter	Symbol	Limits		Unit	Conditions
		Min	Max		
Input Voltage Low	V <sub>IL</sub>		0.8	V	
Input Voltage High	V <sub>IH</sub>	2.2		V	except RESIN*
		2.6		V	RESIN* input
Output Voltage Low	V <sub>OL</sub>		0.45	V	I <sub>OL</sub> = 4mA
Output Voltage High	V <sub>OH</sub>	V <sub>DD</sub> - 0.4		V	CLK output, I <sub>OH</sub> = -4mA
		V <sub>DD</sub> - 0.8		V	except CLK, I <sub>OH</sub> = -4mA
Input Current	I <sub>I</sub>	-1.0	1.0	μA	
RESIN* Hysteresis		0.25		V	
Supply Current (Static)	I <sub>DD</sub>		200	μA	
Supply Current (Dynamic)	I <sub>DDdyn</sub>		50	mA	f <sub>in</sub> = 32 MHz, outputs open

**AC Characteristics**

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V ±5%

Parameter	Symbol	Limits		Unit	Conditions
		Min	Max		
EXFS Cycle Time	t <sub>CYFS</sub>	28		ns	
EXFS Pulse Width High	t <sub>PWFH</sub>	10		ns	2.2 V measurement point
EXFS Pulse Width Low	t <sub>PWFL</sub>	10		ns	0.8 V measurement point
OSC Cycle Time	f <sub>OSC</sub>		32	MHz	
CLK Cycle Time	t <sub>CYCK</sub>	60		ns	
CLK Pulse Width High	t <sub>PWCKH</sub>	25		ns	3.0 V measurement point
CLK Pulse Width Low	t <sub>PWCKL</sub>	25		ns	1.5 V measurement point
CLK Rise Time	t <sub>LHCK</sub>		5	ns	1.5 V → 3.0 V
CLK Fall Time	t <sub>HLCK</sub>		5	ns	3.0 V → 1.5 V
CLK Delay from EXFS↓	t <sub>DCKH</sub>	0	20	ns	CLK↑
CLK Delay from EXFS↓	t <sub>DCKL</sub>	0	20	ns	CLK↑

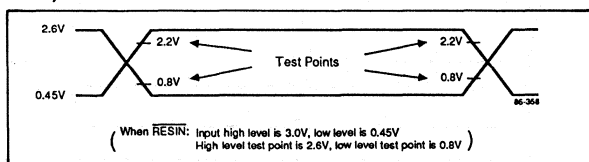
## AC Characteristics (cont)

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V ±5%

Parameter	Symbol	Limits		Unit	Conditions
		Min	Max		
PRCLK Cycle Time	t <sub>CYPRK</sub>	120		ns	
PRCLK Pulse Width High	t <sub>WPRKH</sub>	t <sub>CYCK</sub> - 15		ns	
PRCLK Pulse Width Low	t <sub>WPRKL</sub>	t <sub>CYCK</sub> - 15		ns	
PRCLK↑ Delay from CLK↓	t <sub>DPKHH</sub>		20	ns	
PRCLK↓ Delay from CLK↓	t <sub>DPKRL</sub>		20	ns	
RESIN* Setup to CLK↓	t <sub>SRICK</sub>	20		ns	
RESIN* Hold from CLK↓	t <sub>HCKRI</sub>	0		ns	
RESET* Delay from CLK↓	t <sub>DCKRS</sub>		10	ns	
SREN* Setup to CLK↑	t <sub>SSRECK</sub>	20		ns	
SREN* Hold from CLK↑	t <sub>HCKSRE</sub>	0		ns	
SRDY Setup to CLK↑	t <sub>SSRYCK</sub>	20		ns	
SRDY Hold from CLK↑	t <sub>HCKSRY</sub>	0		ns	
AREN* Setup to CLK↓	t <sub>SARECK</sub>	20		ns	
AREN* Hold from CLK↓	t <sub>HCKARE</sub>	0		ns	
ARDY Setup to CLK↓	t <sub>SARYCK</sub>	20		ns	
ARDY Hold from CLK↓	t <sub>HCKARY</sub>	0		ns	
READY* Output Delay from CLK↑	t <sub>DCKRDY</sub>		10	ns	READY*↑
				8	ns
BCY* Setup to CLK↑	t <sub>SBCKC</sub>	15		ns	
BCY* Hold from CLK↑	t <sub>HCKRE</sub>	5		ns	
WAIT0, 1, 2 Setup to CLK↑	t <sub>SRYCK</sub>	15		ns	
WAIT0, 1, 2 Hold from CLK↑	t <sub>HCKRY</sub>	5		ns	
Input Rise Time	t <sub>RI</sub>		15	ns	0.8 V → 2.2 V
Input Fall Time	t <sub>FI</sub>		8	ns	2.2 V → 0.8 V
Output Rise Time	t <sub>RO</sub>		10	ns	0.8 V → 2.2 V
Output Fall Time	t <sub>FO</sub>		6	ns	2.2 V → 0.8 V

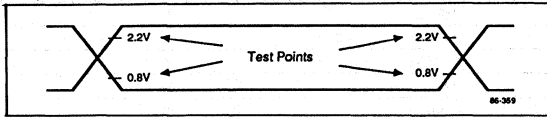
## Timing Waveforms

### AC Test Input (except RESIN\*)

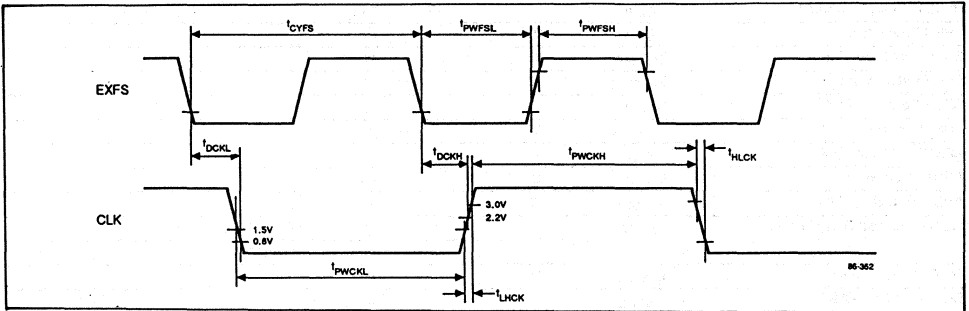


Timing Waveforms (cont)

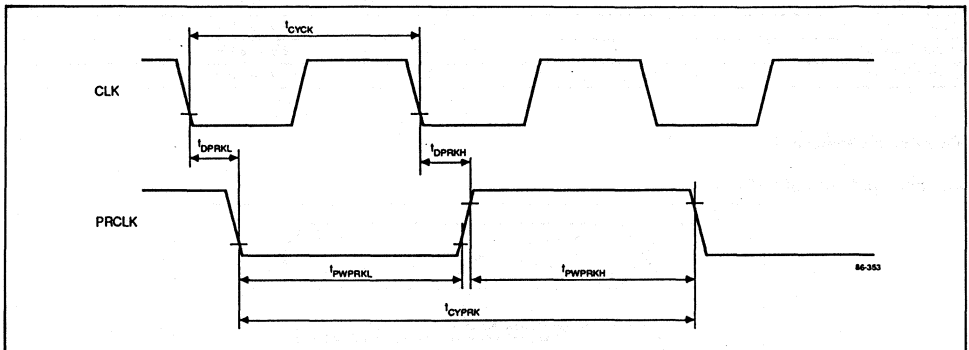
AC Test Output (except CLK)



Clock (CLK) Timing

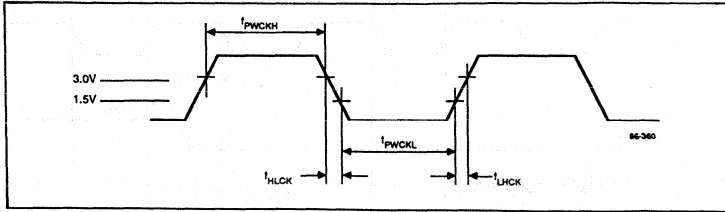


Peripheral Clock (PRCLK) Timing

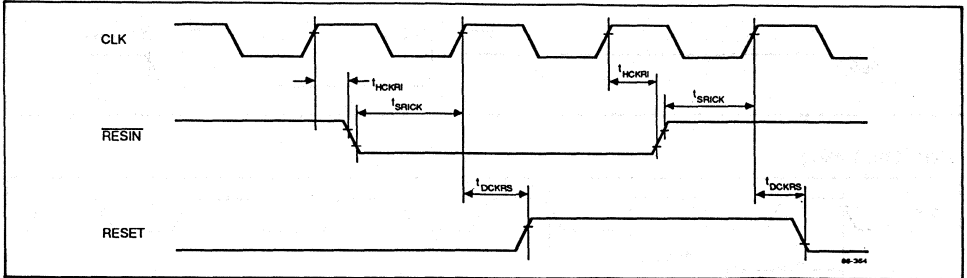


## Timing Waveforms (cont)

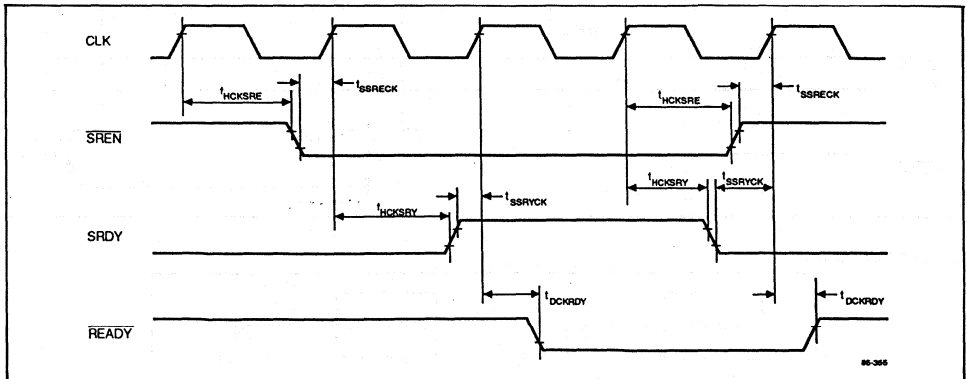
### Clock (CLK) Output Timing



### Reset Timing



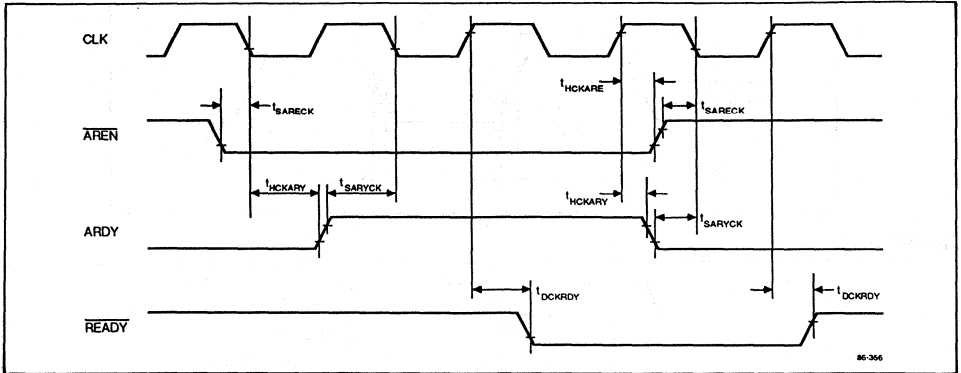
### Synchronous Ready Timing



## $\mu$ PD71611

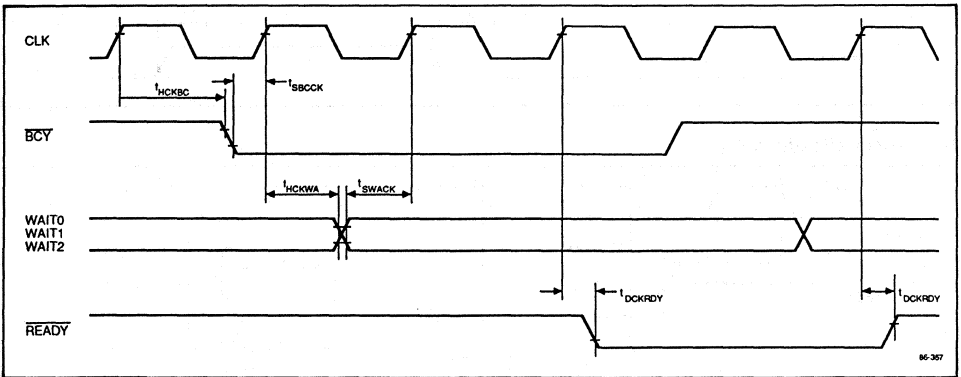
### Timing Waveforms (cont)

#### Asynchronous Ready Timing



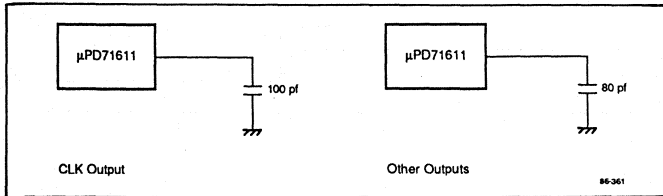
86-356

#### Wait State Timing



86-357

#### Loading Circuits



86-361

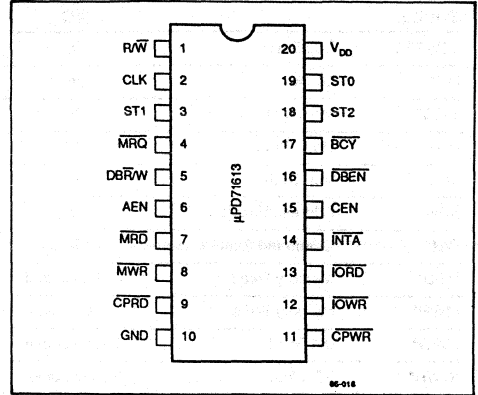
### Description

The μPD71613 is a high performance CMOS bus controller for the μPD70616 (V60) microprocessor. The μPD71613 decodes the μPD70616 status outputs and generates the memory, I/O and coprocessor read/write strobes. A separate interrupt acknowledge output is also provided for interfacing to interrupt controllers.

### Features

- ◆ High output drive (I<sub>OL</sub> = 16 mA)
- ◆ Decodes μPD70616 bus status for
  - memory read/write
  - I/O read/write
  - coprocessor read/write
  - interrupt acknowledge
- ◆ 3-state command outputs
- ◆ CMOS technology
- ◆ 20-pin plastic DIP

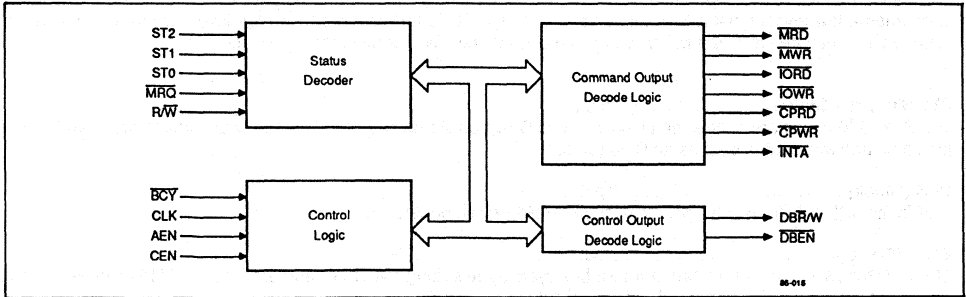
### Pin Configuration



### Ordering Information

Part Number	Package	Maximum Frequency
μPD71613C	20-pin plastic DIP	16 MHz

### Block Diagram





**Pin Identification**

Symbol		I/O	Function
ST2-ST0	Bus Status	In	μPD70616 bus status inputs
CLK	Clock	In	μPD70616 clock input
MRQ*	Memory Request	In	μPD70616 memory request strobe
R/W*	Read/Write	In	μPD70616 bus cycle direction input
BCY*	Bus Cycle	In	μPD70616 bus cycle active input
AEN	Address Enable	In	Command output buffer control pin
CEN	Command Enable	In	Command output control pin
MRD*	Memory Read	3-state output	Memory read strobe output
MWR*	Memory Write	3-state output	Memory write strobe output
IORD*	I/O Read	3-state output	I/O read strobe output
IOWR*	I/O Write	3-state output	I/O write strobe output
CPRD*	Coprocessor Read	3-state output	Coprocessor read strobe output
CPWR*	Coprocessor Write	3-state output	Coprocessor write strobe output
INTA*	Interrupt Acknowledge	Out	Interrupt acknowledge output
DBEN*	Data Bus Enable	Out	External data bus buffer enable output
DBR*/W	Data Buffer Read/Write	Out	Data bus buffer direction output

**Pin Functions**

This section describes the operation of the μPD71613 terminals. Inputs and outputs are considered at a logic "0" level when a low level signal is present. Likewise, a logic "1" is represented by a high level signal. Bus states are defined and measured from the rising edge of the clock to the rising edge of the next clock.

**ST2-ST0 [Bus Status]**..... inputs

The ST2-ST0 inputs are connected to the encoded CPU bus status outputs. The bus status inputs are decoded into the command and control outputs for timing control.

**CLK [Clock]**..... input

CLK is the μPD70616 system clock from the μPD71611 clock generator.

**BCY\* [Bus Cycle]**..... input

The μPD70616 indicates the start of a new bus cycle by asserting the BCY\* output. The μPD71613 uses BCY\* to generate the timing for all command and control outputs.

**MRQ\* [Memory Request]**..... input

MRQ\* is a μPD70616 output which indicates whether or not the bus cycle is in the memory address space or in some other address space.

**R/W\* [Read/Write]**..... input

R/W\* is the bus cycle direction status from the μPD70616.

**AEN [Address Enable]**..... input

The AEN input is the three-state control for the command output buffers. When AEN is high, all output buffers are enabled and the decoded command is determined by the input status. When AEN is low, all command output buffers are in the high impedance state.

**CEN [Command Enable]**..... input

CEN controls the DBEN\* and all command outputs. When CEN is high, all outputs are enabled and controlled by the status inputs. When CEN is low, DBEN\* and the command outputs are forced to the inactive state.

**MRD\* [Memory Read]**..... output

MRD\* is an active low three-state output strobe used to read data from memory.

**MWR\* [Memory Write]**..... output

MWR\* is an active low three-state output strobe used to write data to memory.

**IRD\* [I/O Read]**..... output

IRD\* is an active low three-state output strobe used to read data from a peripheral device.

**IOWR\* [I/O Write]**..... output

IOWR\* is an active low three-state output strobe used to write data to a peripheral device.

**CPRD\* [Coprocessor Read]**..... output

CPRD\* is an active low three-state output strobe used to read data from a coprocessor device.

**CPWR\* [Coprocessor Write]**..... output

CPWR\* is an active low three-state output strobe used to write data to a coprocessor.

**INTA\* [Interrupt Acknowledge]**..... output

The INTA\* output is used to indicate to interrupt controllers that an interrupt acknowledge bus cycle is taking place. The selected interrupt controller is responsible for supplying the interrupt vector.

**DBEN\* [Data Buffer Enable]**..... output

DBEN\* is used to enable the three-state output buffers of the data bus transceivers. This output is only asserted during valid μPD70616 bus cycles.

**DBR\*/W [Data Buffer Read/Write]**..... output

This output controls the direction of the external data bus transceivers. When DBR\*/W is high, data will be transferred from the μPD70616 local bus to an I/O or memory location. When DBR\*/W is low, data will be transferred from memory or I/O peripherals to the μPD70616 local bus.

**VDD [Power Supply]**

The VDD pin supplies +5 Volt power to the μPD71613.

**GND [Ground]**

The GND pin is the power supply return.

**Operational Description**

The μPD71613 serves as the system bus controller for the μPD70616 (V60) microprocessor. The μPD71613 monitors the μPD70616 status (ST2–ST0), memory request (MRQ\*) and read/write (R/W\*) outputs and during valid bus cycles (determined by the μPD70616 BCY\* output), drives the command and control outputs. Command outputs consist of the memory, I/O, coprocessor and interrupt acknowledge strobe signals. Control outputs are the data buffer enable and data buffer direction signals.

Figure 1 depicts a typical μPD70616 system design using both the μPD71613 and the μPD71611 clock generator. Table 1 contains a state table of the command and control outputs.

Figure 1. Basic System Configuration

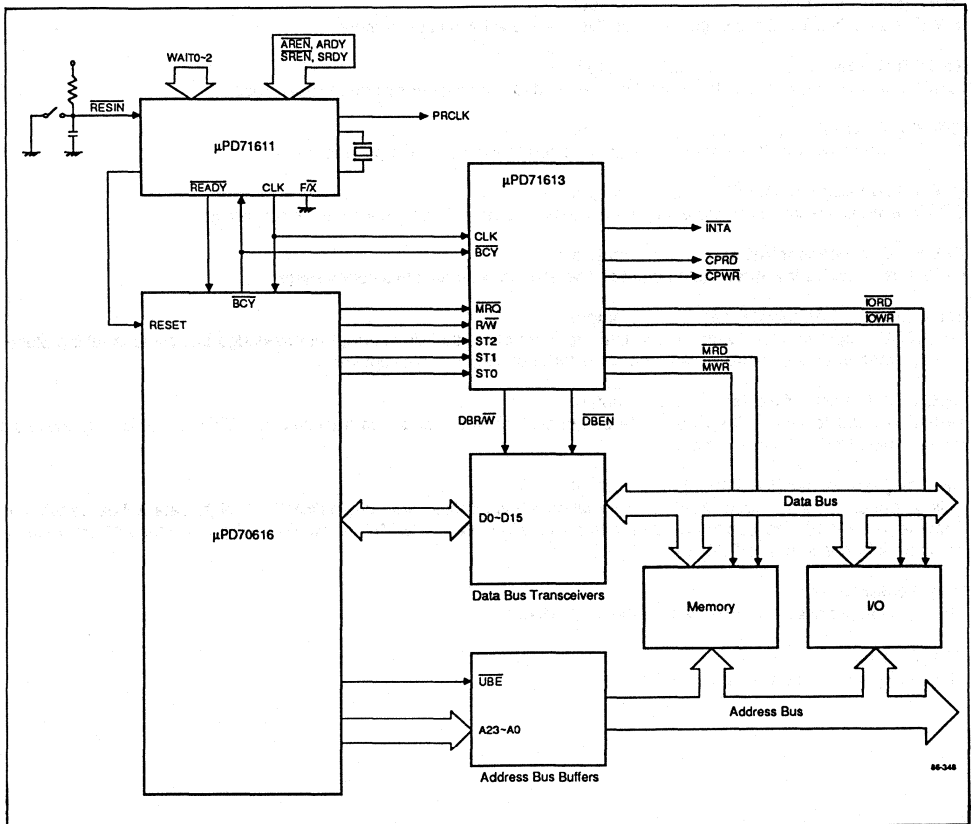


Table 1. Command and Control Output State Table

MRQ	R/W	Status			Bus Cycle	Command Outputs	Control Outputs	
		ST2	ST0	ST1			DBEN	DBR/W
0	0	0	0	0	Coprocessor Memory Write	MWR, CPRD	0	1
		0	0	1	String Mode Memory Write	MWR	0	1
		0	1	0	Memory Write with Short Pass	MWR	0	1
		0	1	1	Single Mode Memory Write	MWR	0	1
		1	0	0	Single Mode Memory Write	MWR	0	1
		1	0	1	Translation Table Memory Write	MWR	0	1
		1	1	0	Translation Table Memory Write	MWR	0	1
		1	1	1	Translation Table Memory Write	MWR	0	1
	1	0	0	0	Coprocessor Memory Read	MRD, CPWR	0	0
		0	0	1	String Mode Memory Read	MRD	0	0
		0	1	0	String Mode Memory Read	MRD	0	0
		0	1	1	Single Mode Memory Read	MRD	0	0
		1	0	0	System Base Table Access	MRD	0	0
		1	0	1	Translation Table Access	MRD	0	0
		1	1	0	Demand Mode Instruction Fetch	MRD	0	0
		1	1	1	Instruction Prefetch	MRD	0	0
1	0	0	0	0	Coprocessor Write	CPWR	0	1
		0	0	1	String Mode I/O Write	IOWR	0	1
		0	1	0	Reserved	-	0	1
		0	1	1	Single Mode I/O Write	IOWR	0	1
		1	0	0	Halt Acknowledge	-	1	1
		1	0	1	Halt Acknowledge	-	1	1
		1	1	0	None	-	0	1
		1	1	1	Reserved	-	0	1
	1	0	0	0	Coprocessor Read	CPRD	0	0
		0	0	1	String Mode I/O Read	IORD	0	0
		0	1	0	Reserved	-	0	0
		0	1	1	Single Mode I/O Read	IORD	0	0
		1	0	0	Halt Acknowledge	-	1	0
		1	0	1	Halt Acknowledge	-	1	0
		1	1	0	Interrupt Acknowledge	INTA	0	0
		1	1	1	Reserved	-	0	0

"0" is a logic low level, "1" is a logic high level

**Absolute Maximum Ratings**

T<sub>A</sub> = +25°C

Power Supply Voltage, V <sub>DD</sub>	-0.5 V to +7.0 V
Input Voltage, V <sub>I</sub>	-1.0 V to V <sub>DD</sub> + 1.0 V
Output Voltage, V <sub>O</sub>	-0.5 V to V <sub>DD</sub> + 0.5 V
Operating Temperature, T <sub>OPT</sub>	-40°C to +85°C
Storage Temperature, T <sub>STG</sub>	-65°C to +150°C
Power Dissipation, P <sub>D</sub>	500 mW

Comment: Exposing the device to stresses above those listed in the Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Capacitance**

T<sub>A</sub> = +25°C, V<sub>DD</sub> = 0 V

Parameter	Symbol	Limits		
		Min	Max	Unit
Input Capacitance	C <sub>I</sub>		12	pf

Conditions: f = 16 MHz

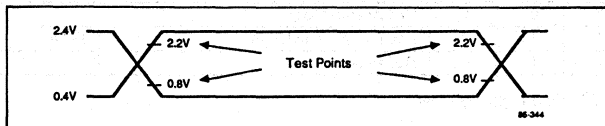
**DC Characteristics**

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V ±10%

Parameter	Symbol	Limits		Unit	Conditions
		Min	Max		
Input Voltage High	V <sub>IH</sub>	2.2		V	
Input Voltage Low	V <sub>IL</sub>		0.8	V	
Output Voltage High	V <sub>OH</sub>	V <sub>DD</sub> - 0.8		V	Command outputs, I <sub>OH</sub> = -4mA
		V <sub>DD</sub> - 0.8		V	Control outputs, I <sub>OH</sub> = -4mA
Output Voltage Low	V <sub>OL</sub>		0.45	V	Command outputs, I <sub>OL</sub> = 16mA
			0.45	V	Control outputs, I <sub>OL</sub> = 8mA
Input Current	I <sub>I</sub>	-1.0	1.0	μA	V <sub>I</sub> = V <sub>DD</sub> , V <sub>SS</sub>
3-state Output Leakage Current	I <sub>OFF</sub>	-10	10	μA	
Static Supply Current	I <sub>DD</sub>		80	μA	V <sub>I</sub> = V <sub>DD</sub> , V <sub>SS</sub>
Dynamic Supply Current	I <sub>DDdyn</sub>		30	mA	f <sub>in</sub> = 16 MHz

**Timing Waveforms**

**AC Input/Output Test**



## AC Characteristics

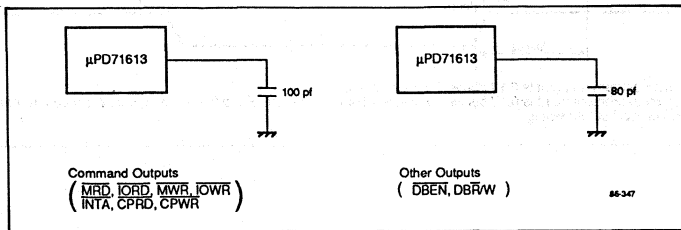
T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V ±10%

Output Pin Load Capacitance: 100 pf

Parameter	Symbol	Limits		Unit	Conditions
		Min	Max		
CLK Cycle Time	t <sub>CYCK</sub>	60		ns	
CLK Pulse Width High	t <sub>PWCKH</sub>	25		ns	
CLK Pulse Width Low	t <sub>PWCKL</sub>	25		ns	
ST2-ST0, MRQ*, R/W* Setup to CLK↑	t <sub>SSCK</sub>	15		ns	
ST2-ST0, MRQ*, R/W* Hold from CLK↑	t <sub>HCKST</sub>	5		ns	
BCY* Setup to CLK↑	t <sub>SBYCK</sub>	15		ns	
BCY* Hold from CLK↑	t <sub>HCKBY</sub>	5		ns	
Command Active Delay from CLK↑	t <sub>DCKCML</sub>	3	15	ns	
Command Inactive Delay from CLK↓	t <sub>DCKCMH</sub>	3	15	ns	
Command Inactive Delay from BCY*↑	t <sub>DBYCMH</sub>	3	15	ns	
Command Output On Delay from AEN↑	t <sub>DAECM</sub>	3	15	ns	
Command Active Delay from AEN↑	t <sub>DAECML</sub>	80	160	ns	
Command Output Float Delay from AEN↓	t <sub>FAECM</sub>		40	ns	
Command Active Delay from CEN↑	t <sub>DCECML</sub>	3	15	ns	
Command Inactive Delay from CEN↓	t <sub>DCECMH</sub>	3	15	ns	
DBEN* Inactive Delay from CLK↑	t <sub>DCKDE</sub>	3	15	ns	
DBEN* Active Delay from AEN↑	t <sub>DAEDE</sub>	3	15	ns	
DBEN* Active Delay from CEN↑	t <sub>DCEDEL</sub>	3	15	ns	
DBEN* Inactive Delay from CEN↓	t <sub>DCEDEH</sub>	3	15	ns	
DBR*W↑ Delay from CLK↑	t <sub>DCKDM</sub>	3	15	ns	
DBR*W↓ Delay from R/W*	t <sub>DRWDM</sub>	3	15	ns	
Input Rise Time	t <sub>RI</sub>		12	ns	0.8 V → 2.2 V
Input Fall Time	t <sub>FI</sub>		8	ns	2.2 V → 0.8 V
Output Rise Time	t <sub>RO</sub>		10	ns	0.8 V → 2.2 V
Output Fall Time	t <sub>FO</sub>		6	ns	2.2 V → 0.8 V

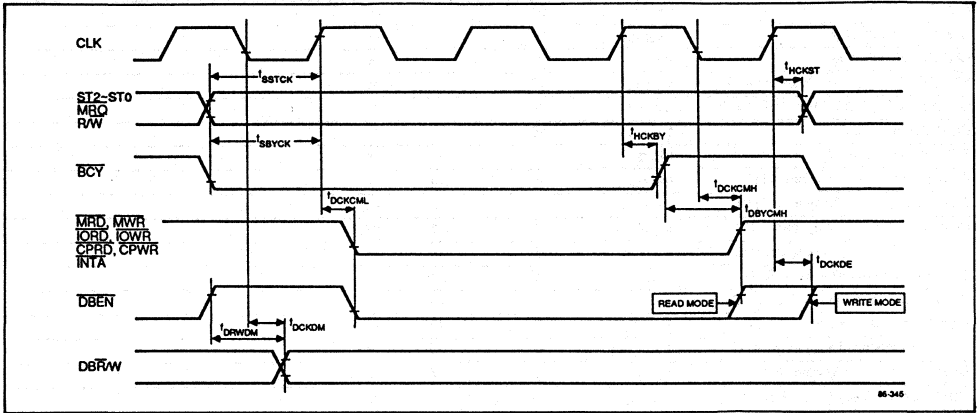
## Timing Waveforms (cont)

### Loading Circuits



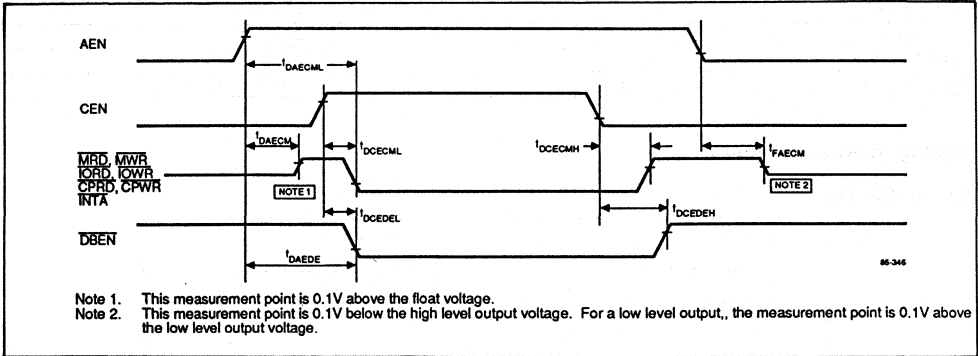
Timing waveforms (cont)

Command/Control Output Timing



88-345

Command Output Timing



88-346

Note 1. This measurement point is 0.1V above the float voltage.

Note 2. This measurement point is 0.1V below the high level output voltage. For a low level output, the measurement point is 0.1V above the low level output voltage.

### Description

The μPD82C43 input/output expander is directly compatible with the μPD8048/C48 family of single-chip microcomputers. Using CMOS technology, the μPD82C43 provides high drive capabilities while requiring only a single +5 V supply voltage.

The μPD82C43 interfaces to the μPD8048/C48 family through a 4-bit I/O port and offers four 4-bit bidirectional static I/O ports. The ease of expansion allows for multiple μPD82C43s to be added using the bus port.

The bidirectional I/O ports of the μPD82C43 act as an extension of the I/O capabilities of the μPD8048/C48 microcomputer family. They are accessible with their own ANL, MOV, and ORL instructions.

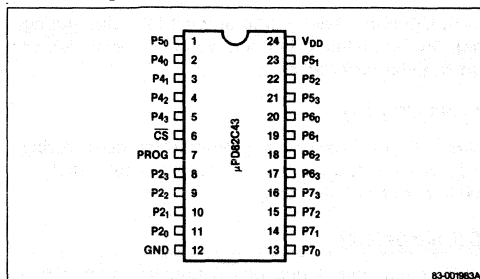
### Features

- Four 4-bit I/O ports
- High output drive
- Logical AND and OR directly to ports
- Compatible with industry standard 8243
- Direct extension of resident μPD8048/C48 I/O ports
- Fully compatible with μPD8048/C48 microcomputer family
- CMOS technology
- Single +5 V supply

### Ordering Information

Part Number	Package Type
μPD82C43C	24-pin plastic DIP
μPD82C43CX	24-pin plastic skinny DIP

### Pin Configuration



### Pin Identification

No.	Symbol	Function
1, 23-21	P50-P53	4-bit I/O port 5
2-5	P40-P43	4-bit I/O port 4
6	CS	Chip select input
7	PROG	Clock input
8-11	P23-P20	4-bit I/O CPU interface port 2
12	GND	Ground
13-16	P70-P73	4-bit I/O port 7
17-20	P63-P60	4-bit I/O port 6
24	VDD	+5 V power supply

### Pin Functions

#### P20-P23 (Port 2)

A 4-bit bidirectional port which contains the I/O port address and instruction code on a high to low transition of PROG. During a low to high transition of PROG, port 2 contains either the data for a selected output port if a write operation, or the data from a selected output port (before a low to high transition) if a read operation. Data on port 2 may be directly written, read, ANDed or ORed with previous data.

#### P40-P43 (Port 4)

A 4-bit I/O port. May be programmed for input (during read), low impedance latched output (after write), or high impedance (after read).

#### P50-P53 (Port 5)

A 4-bit I/O port. May be programmed for input (during read), low impedance latched output (after write), or high impedance (after read).



**P60-P63 (Port 6)**

A 4-bit I/O port. May be programmed for input (during read), low impedance latched output (after write), or high impedance (after read).

**P70-P73 (Port 7)**

A 4-bit I/O port. May be programmed for input (during read), low impedance latched output (after write), or high impedance (after read).

**$\overline{CS}$  (Chip Select)**

A chip select input. A high on  $\overline{CS}$  inhibits any change of output or internal status.

**PROG (Clock Input)**

A high to low transition on PROG indicates that the op-code and the addressed port information are available on port 2. A low to high transition indicates that data is available on port 2.

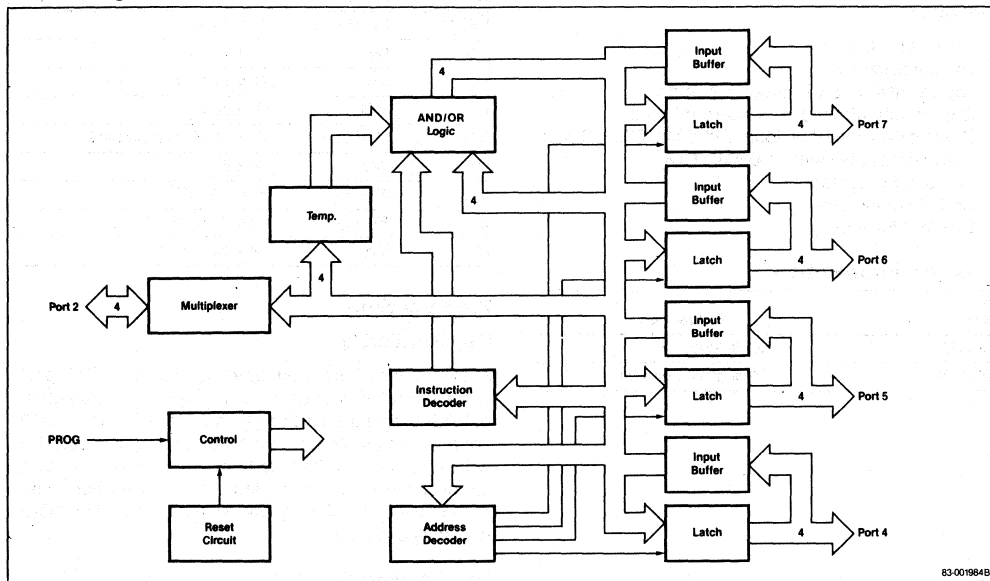
**Ground**

Ground.

**VDD (Power Supply)**

+5V power supply input.

**Block Diagram**



83-001984B

## Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$	
Power supply voltage, $V_{DD}$	-0.5 V to +7 V(1)
Input voltage, $V_I$	-0.3 V to $V_{DD} + 0.3$ V
Output voltage, $V_O$	-0.3 V to $V_{DD} + 0.3$ V
Operating temperature, $T_{OPT}$	-40°C to +85°C
Storage temperature, $T_{STG}$	-65°C to +150°C
Power dissipation, $P_D$	1.0 W

### Note:

(1) With respect to ground.

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## DC Characteristics

$T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ ;  $V_{DD} = +5\text{ V} \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage high	$V_{IH}$	$V_{DD} - 2.0$		$V_{DD}$	V	
Input voltage low	$V_{IL}$	-0.3		+0.8	V	
Output voltage high (port 4-7)	$V_{OH1}$	$V_{DD} - 0.5$			V	$I_{OH} = -240 \mu\text{A}$
Output voltage high (port 2)	$V_{OH2}$	$V_{DD} - 0.5$			V	$I_{OH} = -100 \mu\text{A}$
Output voltage low (port 4-7)	$V_{OL1}$		+0.45		V	$I_{OL} = 5 \text{ mA}$ , (Note 1)
Output voltage low (port 7)	$V_{OL2}$		+1		V	$I_{OL} = +20 \text{ mA}$
Output voltage low (port 2)	$V_{OL3}$		+0.45		V	$I_{OL} = 0.6 \text{ mA}$
Sum of all $I_{OL}$ from 16 outputs	$I_{OL}$		80	mA	5 mA	5 mA each pin
Input leakage current (port 4-7)	$I_{IL1}$		$\pm 1$	$\mu\text{A}$		$V_{IN} = V_{DD}$ to 0 V
Input leakage current (port 2, CS, PROG)	$I_{IL2}$		$\pm 1$	$\mu\text{A}$		$V_{IN} = V_{DD}$ to 0 V
$V_{DD}$ supply current	$I_{DD1}$	100	300	$\mu\text{A}$		Operation mode, (Note 1)
Power down supply current	$I_{DD2}$	1	10	$\mu\text{A}$		Standby mode

### Note:

(1) Refer to graph of additional sink current drive.

## DC Characteristics (cont)

$T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ ;  $V_{DD} = +2.5\text{ V}$  to  $+6\text{ V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage high	$V_{IH}$	$0.7 V_{DD}$		$V_{DD}$	V	
Input voltage low	$V_{IL}$	-0.3		+0.18 $V_{DD}$	V	
Output voltage high (port 4-7)	$V_{OH1}$	$0.75 V_{DD}$			V	$I_{OH} = -120 \mu\text{A}$
Output voltage high (port 2)	$V_{OH2}$	$0.75 V_{DD}$			V	$I_{OH} = -50 \mu\text{A}$
Output voltage low (port 4-7)	$V_{OL1}$		+0.45		V	$I_{OL} = +2.5 \text{ mA}$
Output voltage low (port 7)	$V_{OL2}$		+1		V	$I_{OL} = +7 \text{ mA}$
Output voltage low (port 2)	$V_{OL3}$		+0.45		V	$I_{OL} = +0.3 \text{ mA}$
Output current low (port 4-7)	$I_{OL}$		40	mA		+2.5 mA each pin
Input leakage current (port 4-7)	$I_{IL1}$		$\pm 1$	$\mu\text{A}$		$V_{IN} = V_{DD}$ to 0 V
Input leakage current (port 2, CS, PROG)	$I_{IL2}$		$\pm 1$	$\mu\text{A}$		$V_{IN} = V_{DD}$ to 0 V
$V_{DD}$ supply current	$I_{DD1}$	100	300	$\mu\text{A}$		Operation mode, (Note 1)
Power down supply current	$I_{DD2}$	1	10	$\mu\text{A}$		Standby mode

### Note:

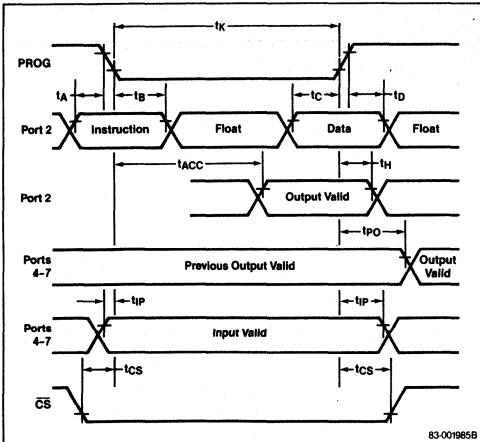
(1)  $I_{OH} = 0 \mu\text{A}$ , PROG pulse cycle = 5  $\mu\text{s}$  min.

**AC Characteristics**

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5V ± 10%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Code valid before PROG	t <sub>A</sub>	100			ns	80 pF load
Code valid after PROG	t <sub>B</sub>	0			ns	20 pF load
Data valid before PROG	t <sub>C</sub>	200			ns	80 pF load
Data valid after PROG	t <sub>D</sub>	20			ns	20 pF load
Port 2 floating after PROG	t <sub>H</sub>	0	150		ns	20 pF load
PROG negative pulse width	t <sub>K</sub>	700			ns	
Ports 4-7 valid after PROG	t <sub>P0</sub>		700		ns	100 pF load
Ports 4-7 valid before / after PROG	t <sub>IP</sub>	100			ns	
Port 2 valid after PROG	t <sub>ACC</sub>	90	650		ns	80 pF load
CS valid before / after PROG	t <sub>CS</sub>	50			ns	

**Timing Waveform**

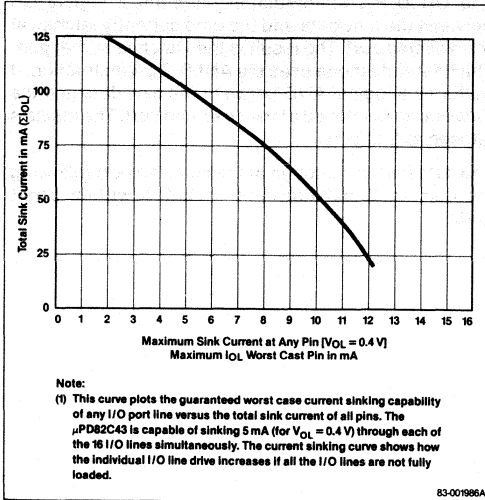


**AC Characteristics**

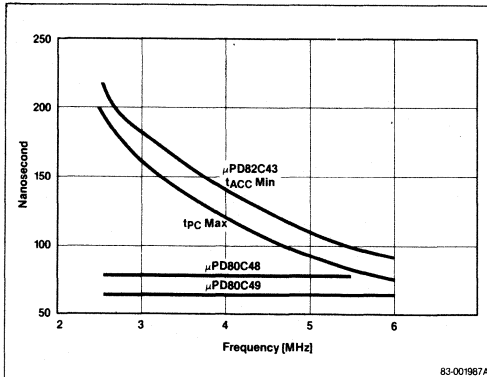
T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +2.5V to +6V

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Command input setup time to PROG ↓	t <sub>A</sub>	300			ns	Port 2 (control, port, address); 80 pF load
Command input setup time after PROG ↓	t <sub>B</sub>	0			ns	Port 2 (control, port, address); 20 pF load
Data input setup time to PROG ↑	t <sub>C</sub>	600			ns	Port 2 (write mode); 80 pF load
Data input hold time after PROG ↑	t <sub>D</sub>	80			ns	Port 2 (write mode); 20 pF load
Data float delay time from PROG ↑	t <sub>H</sub>	0	400		ns	Port 2 (read mode); 20 pF load
PROG pulse width	t <sub>K</sub>	2			μs	
CS input setup time to PROG ↓ CS input hold time after PROG ↑	t <sub>CS</sub>	200			ns	
Data output delay time from PROG ↑	t <sub>P0</sub>		2		ns	Port 4-7; 100 pF load
Data input setup time to PROG ↓ Data input hold time after PROG ↑	t <sub>IP</sub>	100			ns	Port 4-7
Data output delay time from PROG ↓	t <sub>ACC</sub>		3.5		μs	Port 2; 80 pF load

**Figure 1. Current Sinking Capability (Note 1)**



**Figure 2.  $t_{ACC}$  (Min)/μPD82C43 vs  $t_{PC}$  (Max)/μPD80C48, μPD80C49**



## Functional Description

The I/O capabilities of the μPD8048/C48 family can be enhanced in four I/O port increments of 4 bits each using one or more μPD82C43s. These additional I/O lines are addressed as ports 4–7. The following lists the operations which can be performed on ports 4–7.

- Logical AND accumulator to port
- Logical OR accumulator to port
- Transfer port to accumulator
- Transfer accumulator to port

Port 2 (P2<sub>0</sub>–P2<sub>3</sub>) forms the 4-bit bus through which the μPD82C43 communicates with the host processor. The PROG output from the μPD8048/C48 family provides the necessary timing to the μPD82C43. There are two 4-bit nibbles involved in each data transfer. The first nibble contains the opcode and port address followed by the second nibble containing the 4-bit data. Multiple μPD82C43s can be used for additional I/O. The output lines from the μPD8048/C48 family can be used to form the chip selects for additional μPD82C43s.

## Power On Initialization

Applying power to the μPD82C43 sets ports 4–7 to the high impedance mode and port 2 to the input mode. The state of the PROG pin at power on may be either high or low. The PROG pin must make a high to low transition in order to exit from the power on mode. The power on sequence is initiated any time  $V_{DD}$  drops below 1V. Table 1 following shows how the first 4-bit nibble of a data transfer instruction is decoded.

**Table 1. Port 2 Instruction Decoding**

P2 <sub>3</sub>	P2 <sub>2</sub>	Instruction Code	P2 <sub>1</sub>	P2 <sub>0</sub>	Address Code
0	0	Read	0	0	Port 4
0	1	Write	0	1	Port 5
1	0	ORLD	1	0	Port 6
1	1	ANLD	1	1	Port 7

For example, a 0010 appearing on P2<sub>3</sub>–P2<sub>0</sub>, respectively, would result in a read of port 6.

## Read Mode

There is one read mode in the μPD82C43. A falling edge on the PROG pin latches the op code and port address from input port 2. The port address and read operation are then decoded, causing the appropriate outputs to be high impedance and the input buffers switched on. The rising edge of PROG terminates the read operation. The port (4, 5, 6, or 7) that was selected by the port address (P2<sub>1</sub>–P2<sub>0</sub>) is returned to the high impedance mode, and port 2 is switched to the input mode.

Generally, in the read mode a port will be an input and in the write mode it will be an output. If during program operation the μPD82C43's modes are changed, the first read pulse immediately following a write should be ignored. The subsequent read signals are valid. Reading a port will then force that port to a high impedance state.

### **Write Modes**

There are three write modes in the μPD82C43. The **MOVD P<sub>p</sub>, A** instruction from the μPD8048/C48 family writes the new data directly to the specified port (4, 5, 6,

or 7). The old data previously latched at that port is lost. The **ORLD P<sub>p</sub>, A** instruction performs a logical OR between the new data and the data currently latched at the selected port. The result is then latched at that port. The final write mode uses the **ANLD P<sub>p</sub>, A** instruction. It performs a logical AND between the new data and the data currently latched at the specified port. The result is latched at that port.

The data remains latched at the selected port following the logical manipulation until new data is written to that port.

# NMOS + BIPOLAR SYSTEM SUPPORT PRODUCTS

## Section 7 – NMOS System Support Products

μPD8155/56	2048-Bit Static MOS RAM with I/O Ports and Timer .....	7.3
μPB8216/26	4-Bit Parallel Bidirectional Bus Drivers .....	7.11
μPD8237A	High-Performance Programmable DMA Controller .....	7.15
μPD8243/H	Input/Output Expander for μPD8048 Family .....	7.33
μPD8251A/AF	Programmable Communications Interface (USART) .....	7.39
μPD8253	Programmable Interval Timer .....	7.57
μPD8255A	Programmable Peripheral Interface .....	7.69
μPD8257	Programmable DMA Controller .....	7.79
μPD8259A	Programmable Interrupt Controller .....	7.91
μPD8279	Programmable Keyboard/Display Interface .....	7.109

## Section 7 – BIPOLAR System Support Products

μPB8282/83	8-Bit Latches .....	7.117
μPB8284A	Clock Generator and Driver for 8086/8088 Microprocessors .....	7.121
μPB8286/87	8-Bit Bus Transceivers .....	7.127
μPB8288	CPU System Bus Controller .....	7.131
μPB8289	Bus Arbiter .....	7.139

### Description

The μPD8155 and μPD8156 are μPD8085A family components having 256 × 8-bit static RAM, 3 programmable I/O ports, and a programmable timer. They directly interface to the multiplexed μPD8085A bus with no external logic. The μPD8155 has an active low chip enable while the μPD8156 is active high.

The μPD8155 and μPD8156 contain 2048 bits (256 × 8) of static RAM. The 256 words of memory may be selected anywhere within the system's 64K memory space by coding the upper 8 bits of address from the μPD8085A as a chip select.

The two general purpose 8-bit ports (PA and PB) may be programmed for input or output either in interrupt or status mode. The single 6-bit port (PC) may be used as a control for PA and PB or as a general purpose I/O port. The μPD8155 and μPD8156 are programmed for their system personalities by writing into their command/status (C/S) registers upon system initialization.

The timer is a single 14-bit down counter which is programmable for 4 modes of output operation; see table 3.

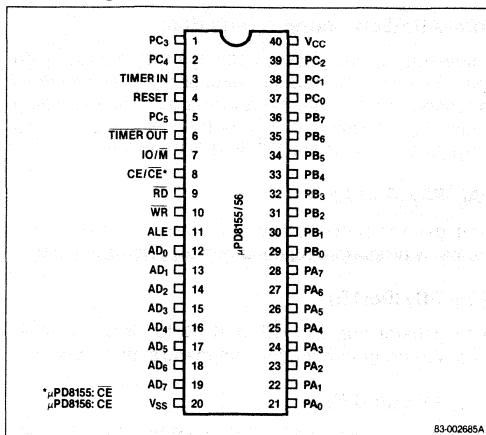
### Features

- 256 × 8-bit static RAM
- Two programmable 8-bit I/O ports
- One programmable 6-bit I/O port
- Single +5 V ± 10% power supply
- Directly interfaces to the μPD8085A and μPD8085A-2
- Programmable 14-bit binary counter/timer

### Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD8155C / 55HC	40-pin plastic DIP	3 MHz
μPD8155C-2 / 55HC-2	40-pin plastic DIP	5 MHz
μPD8156C / 56HC	40-pin plastic DIP	3 MHz
μPD8156C-2 / 56HC-2	40-pin plastic DIP	5 MHz

### Pin Configuration



### Pin Identification

No.	Symbol	Function
1, 2, 5, 37-39	PC <sub>0</sub> -PC <sub>5</sub>	6-bit I/O port or control lines
3	TIMER IN	Timer clock input
4	RESET	Reset input
6	TIMER OUT	Timer counter output
7	IO / $\bar{M}$	I/O or memory select input
8	CE / $\bar{C}E$	Chip enable input
9	RD	Read strobe input
10	WR	Write strobe input
11	ALE	Address low enable input
12-19	AD <sub>0</sub> -AD <sub>7</sub>	Low address / data bus I/O
20	V <sub>SS</sub>	Ground
21-28	PA <sub>0</sub> -PA <sub>7</sub>	8-bit I/O port A
29-36	PB <sub>0</sub> -PB <sub>7</sub>	8-bit I/O port B
40	V <sub>CC</sub>	+5 V power supply



**Pin Functions****AD<sub>0</sub>-AD<sub>7</sub> (Low Address/Data Bus)**

Three-state address/data (AD) lines that interface with the CPU lower 8-bit address/data bus. The 8-bit address is loaded into the internal address latch on the falling edge of ALE. The 8-bit data is then written to or read from the chip, based on  $\overline{WR}$  and  $\overline{RD}$  strobe inputs.

**PA<sub>0</sub>-PA<sub>7</sub> (Port A)**

8-bit general purpose I/O port. Data direction is selected by programming the command status register.

**PB<sub>0</sub>-PB<sub>7</sub> (Port B)**

8-bit general purpose I/O port. Data direction is selected by programming the command status register.

**PC<sub>0</sub>-PC<sub>5</sub> (Port C)**

6-bit general purpose I/O port or control signals for PA and PB. Port C function is selected by programming the command status register.

**ALE (Address Low Enable)**

This input control signal latches the address on the AD<sub>0</sub>-AD<sub>7</sub> lines and the states of  $\overline{CE}/\overline{CE}$  and  $\overline{IO}/\overline{M}$  into the chip on the falling edge of ALE.

 **$\overline{CE}/\overline{CE}$  (Chip Enable)**

The chip enable input is active low for μPD8155 and active high for μPD8156.

 **$\overline{IO}/\overline{M}$  (I/O or Memory Select)**

This input selects either internal RAM memory if low or I/O and command status registers if high.

**RESET (Reset)**

The reset input from μPD8085A initializes ports A, B, and C to the input mode.

**TIMER IN (Timer Clock In)**

Clock input to the 14-bit binary down counter.

 **$\overline{TIMER OUT}$  (Timer Counter Output)**

The timer output is programmable for 4 output waveform modes. The selected output waveform can be a single pulse or a continuous pulse train, or it can be a single square wave or a continuous square wave.

 **$\overline{RD}$  (Read Strobe)**

The  $\overline{RD}$  input will strobe the addressed RAM data onto the AD bus if the  $\overline{IO}/\overline{M}$  pin is low; otherwise the content of the selected I/O port or command status registers will be strobed onto the AD bus.

 **$\overline{WR}$  (Write Strobe)**

The  $\overline{WR}$  input will strobe the available data on the AD bus into addressed RAM location or I/O ports and command status registers depending on  $\overline{IO}/\overline{M}$ .

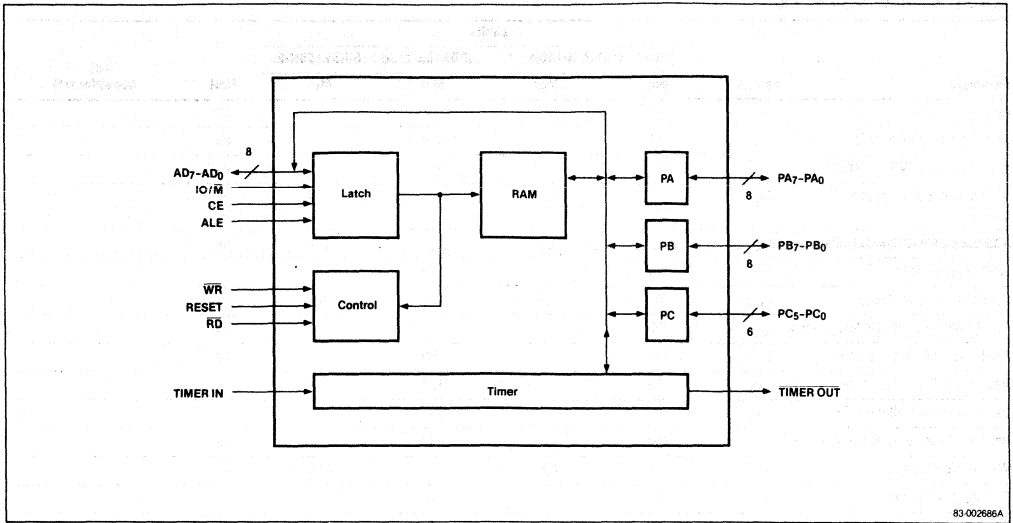
**V<sub>CC</sub> (Power Supply)**

+5 V power supply input.

**V<sub>SS</sub> (Ground)**

Ground.

## Block Diagram



83 002686A

## Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$

Power supply voltage, $V_{CC}$	-0.5 V to +7 V
Operating temperature, $T_{OP}$	$0^\circ\text{C}$ to $+70^\circ\text{C}$
Storage temperature, $T_{STG}$	$-65^\circ\text{C}$ to $+150^\circ\text{C}$
Power dissipation, $P_D$	1.5 W

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## DC Characteristics

$T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = +5\text{ V} \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage low	$V_{IL}$	-0.5		0.8	V	
Input voltage high	$V_{IH}$	2.0		$V_{CC} + 0.5$	V	
Output voltage low	$V_{OL}$			0.45	V	$I_{OL} = 2.0\text{ mA}$
Output voltage high	$V_{OH}$	2.4			V	$I_{OH} = 400\text{ }\mu\text{A}$
Input leakage current	$I_{LI}$			$\pm 10$	$\mu\text{A}$	$V_I = V_{CC}$ to 0 V
Output leakage current	$I_{LO}$			$\pm 10$	$\mu\text{A}$	$0.45\text{ V} \leq V_{OUT} \leq V_{CC}$
Power supply current ( $V_{CC}$ )	$I_{CC}$			180	mA	8155/56, 8155-2/56-2
				125	mA	8155H/56H, 8155H-2/56H-2
Chip enable leakage	$\mu\text{PD8155 } I_{IL}(\text{CE})$			+100	$\mu\text{A}$	$V_I = V_{CC}$ to 0 V
	$\mu\text{PD8156}$			-100	$\mu\text{A}$	$V_I = V_{CC}$ to 0 V

**AC Characteristics**

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5 V ± 10%

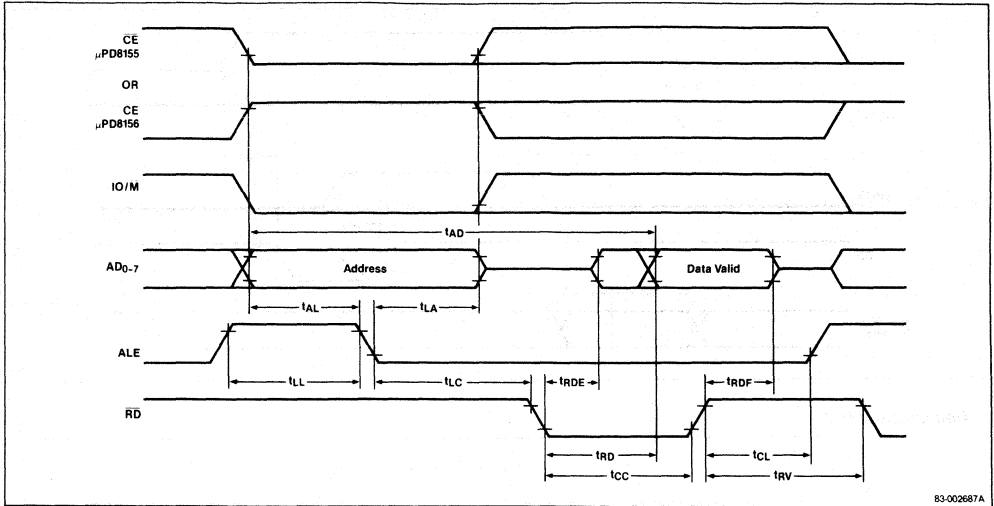
Parameter	Symbol	Limits				Unit	Test Conditions(1)
		μPD8155/56/55H/56H		μPD8155-2/56-2/55H-2/56H-2			
		Min	Max	Min	Max		
Address to latch setup time	t <sub>AL</sub>	50		30		ns	
Address hold time after latch	t <sub>LA</sub>	80		30		ns	
Latch to READ / WRITE control	t <sub>LC</sub>	100		40		ns	
Valid data out delay from READ control	t <sub>RD</sub>		170		140	ns	
Address stable to data out valid	t <sub>AD</sub>		400		330	ns	
Latch enable width	t <sub>LL</sub>	100		70		ns	
Data bus float after READ	t <sub>RDF</sub>	0	100	0	80	ns	
READ / WRITE control to latch enable	t <sub>CL</sub>	20		10		ns	
READ / WRITE control width	t <sub>CC</sub>	250		200		ns	
Data in to WRITE setup time	t <sub>DW</sub>	150		100		ns	
Data in hold time after WRITE	t <sub>WD</sub>	0		0		ns	
Recovery time between controls	t <sub>RV</sub>	300		200		ns	
WRITE to port output	t <sub>WP</sub>		400		300	ns	
Port input setup time	t <sub>PR</sub>	70		50		ns	
Port input hold time	t <sub>RP</sub>	50		10		ns	
Strobe to buffer full	t <sub>SBF</sub>		400		300	ns	
Strobe width	t <sub>SS</sub>	200		150		ns	
READ to buffer empty	t <sub>RBE</sub>		400		300	ns	
Strobe to INTR on	t <sub>SI</sub>		400		300	ns	
READ to INTR off	t <sub>RDI</sub>		400		300	ns	
Port setup time to strobe	t <sub>PSS</sub>	50		0		ns	
Port hold time after strobe	t <sub>PHS</sub>	120		100		ns	
Strobe to buffer empty	t <sub>SBE</sub>		400		300	ns	
WRITE to buffer full	t <sub>WBE</sub>		400		300	ns	
WRITE to INTR off	t <sub>WI</sub>		400		300	ns	
TIMER IN to TIMER OUT low	t <sub>TL</sub>		400		300	ns	
TIMER IN to TIMER OUT high	t <sub>TH</sub>		400		300	ns	
Data bus enable from READ control	t <sub>RDE</sub>	10		10		ns	
Clock TIMER IN	t <sub>CYC</sub>	320		200		ns	
CLK rise and fall time	t <sub>r</sub> , t <sub>f</sub>		30		30	ns	
CLK pulse width	t <sub>1</sub>	80		40		ns	
	t <sub>2</sub>	120		70		ns	

**Note:**

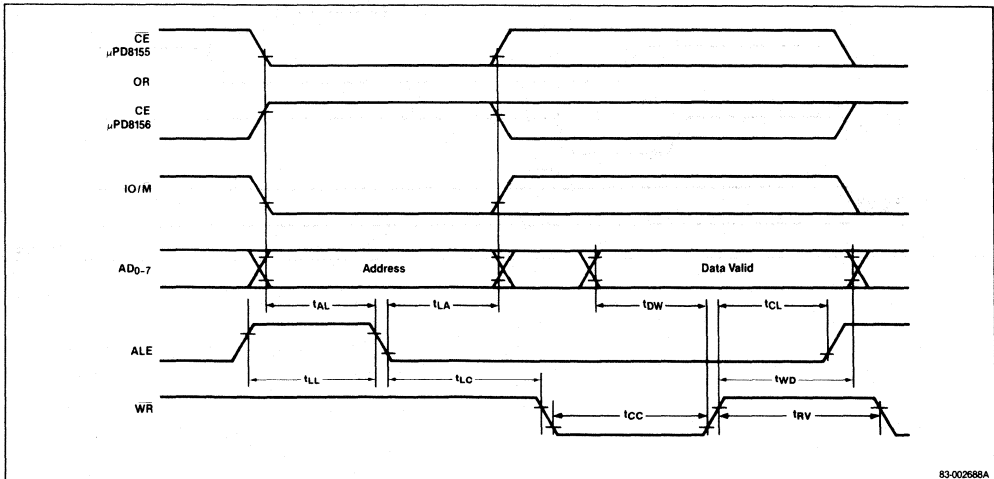
(1) 150 pF load

## Timing Waveforms

### Read Cycle

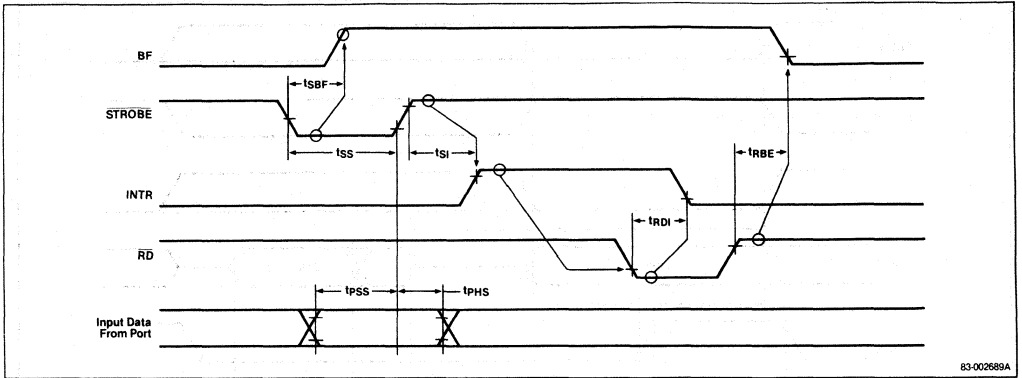


### Write Cycle

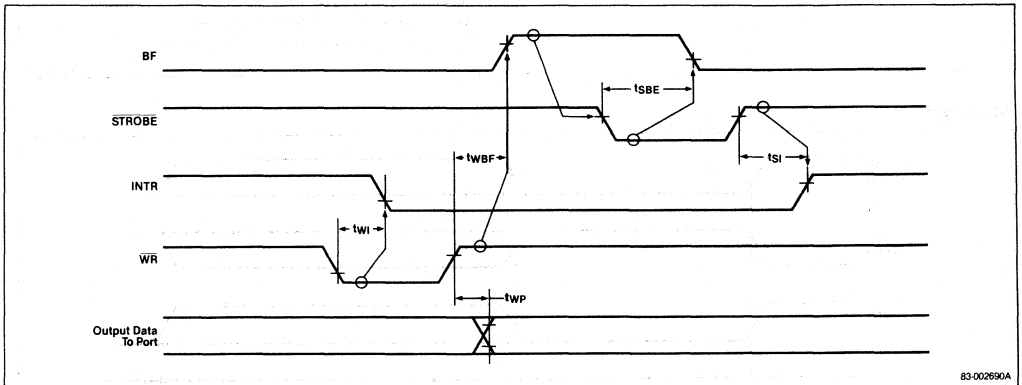


Timing Waveforms (cont)

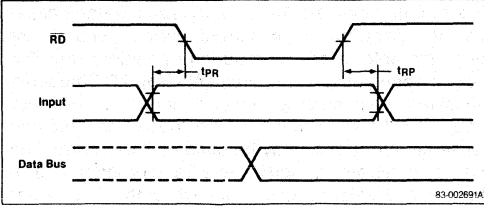
Strobed Input Mode



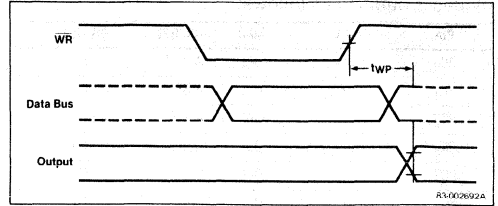
Strobed Output Mode



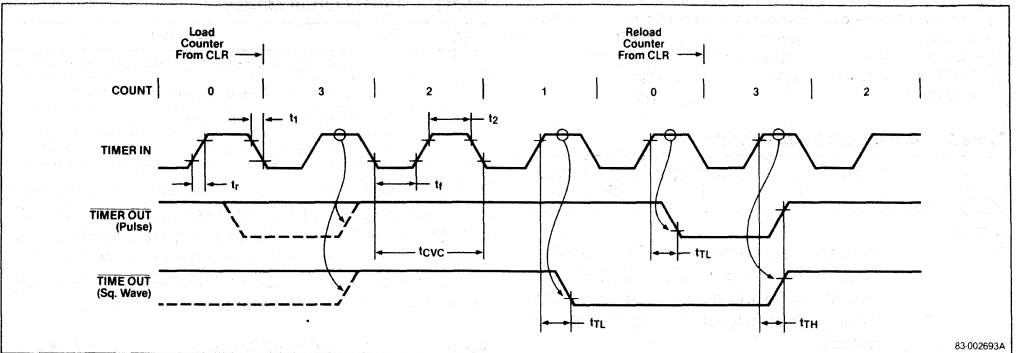
### Basic Input Mode



### Basic Output Mode



### Time Output



## Functional Description

### Command Status Register

The command status register is an 8-bit register which must be programmed before the  $\mu$ PD8155/56 can perform any useful functions. Its purpose is to define the mode of operation of the three ports and the timer. Programming of the device may be accomplished by writing to I/O address XXXXX000 (X = don't care) with a specific bit pattern. Reading of the command status register can be accomplished by performing an I/O read operation at address XXXXX000. The pattern returned will be a 7-bit status report of PA, PB and the timer. The bit patterns for the command status register read and write are shown in tables 1 and 2.

Table 1. Command Status Write

TM2	TM1	IEB	IEA	PC <sub>2</sub>	PC <sub>1</sub>	PB	PA
-----	-----	-----	-----	-----------------	-----------------	----	----

where:

- TM2-TM1 = Define timer mode
- IEB = Enable port B interrupt
- IEA = Enable port A interrupt
- PC<sub>2</sub>-PC<sub>1</sub> = Define port C mode
- PB/PA = Define port B/A as in or out(1)

The timer mode of operation is programmed as follows during command status write:

TM2	TM1	Timer Mode
0	0	Don't affect timer operation
0	1	Stop timer counting
1	0	Stop counting after TC
1	1	Start timer operation

Interrupt enable status is programmed as follows:

IEB/IEA	Interrupt Enable Port B/A
0	No
1	Yes

Port C may be placed in four possible (Alt) modes of operation as outlined below. The modes are selected during command status write as follows:

PC <sub>2</sub>	PC <sub>1</sub>	Port C Mode
0	0	Alt 1
0	1	Alt 3
1	0	Alt 4
1	1	Alt 2

The function of each pin of port C in the four possible modes is outlined as follows:

Pin	Alt 1	Alt 2	Alt 3(2)	Alt 4(2)
PC <sub>0</sub>	In	Out	A INTR	A INTR
PC <sub>1</sub>	In	Out	A BF	A BF
PC <sub>2</sub>	In	Out	A STB	A STB
PC <sub>3</sub>	In	Out	Out	B INTR
PC <sub>4</sub>	In	Out	Out	B BF
PC <sub>5</sub>	In	Out	Out	B STB

**Note:**

- (1) PB/PA sets port B/A mode: 0 = input; 1 = output
- (2) In Alt 3 and Alt 4 modes, the control signals are initialized as follows:

Control	Input	Output
STB (Input strobe)	Input control	Output control
INTR (Interrupt request)	Low	High
BF (Buffer full)	Low	Low

**Table 2. Command Status Read**

X	TI	INTE B	B BF	INTR B	INTE A	A BF	INTR A
---	----	-----------	---------	-----------	-----------	---------	-----------

where:

TI = Indicates a timer interrupt. This bit is set when terminal count is reached. It is reset when starting a new count, or a hardware reset occurs, or after reading the CS register.

INTE B/A = Port B/A interrupt. High = active.

B/A BF = Indicates whether port B/A is full if in input mode or empty if in output mode. High = active.

INTR B/A = Port B/A interrupt request. High = active.

The programming address summary for the status, ports, and timer are as follows:

I/O Address	Number of Bits	Function
XXXXX000	8	Command status
XXXXX001	8	PA
XXXXX010	8	PB
XXXXX011	6	PC
XXXXX100	8	Timer low
XXXXX101	8	Timer high

**Timer Operation**

The internal timer is a 14-bit binary down counter capable of operating in 4 output modes which are programmable at any time during operation. Any TTL clock meeting timer in requirements (see AC Characteristics) may be used as a time base and fed to the timer input. The timer output may be looped around and cause an interrupt or may be used as I/O control. The output modes are defined in table 3 and programmed as the two MSBs of the higher order byte of the timer count register.

**Table 3. Timer Output Modes**

M <sub>2</sub>	M <sub>1</sub>	Operation
0	0	Single square wave cycle from start to terminal count
0	1	Continuous square wave (period = count length)
1	0	Single pulse at terminal count
1	1	Continuous single pulse occurring at terminal count

Programming the timer requires two words to be written to the μPD8155/56 at I/O address XXXXX100 and XXXXX101 for the low and high order bytes, respectively. Valid count length must be between 0002H and 3FFFH. The bit assignments for the high and low programming words of the timer count register are as follows:

Word	Timer Count Register								I/O Address
High byte	M <sub>2</sub>	M <sub>1</sub>	T <sub>13</sub>	T <sub>12</sub>	T <sub>11</sub>	T <sub>10</sub>	T <sub>9</sub>	T <sub>8</sub>	XXXXX101
Low byte	T <sub>7</sub>	T <sub>6</sub>	T <sub>5</sub>	T <sub>4</sub>	T <sub>3</sub>	T <sub>2</sub>	T <sub>1</sub>	T <sub>0</sub>	XXXXX100

The control of the timer is performed by TM2 and TM1 of the command status word.

Note that counting will be stopped by a hardware reset. A start command must be issued via the command status register to begin counting. A new mode and/or count length can be loaded while the counter is counting, but will not be used until a start command is issued.

When an external nonsynchronous event is used as the timer input, the signal must first be synchronized to the system clock. A D-type flip-flop can be used for this purpose.

### Description

The μPB8216 and μPB8226 are 4-bit parallel bidirectional bus drivers specifically designed to buffer microcomputer system components. All inputs are low power TTL compatible. For driving MOS, the DO outputs provide a high 3.65 volts ( $V_{OH}$ ); for high-capacitance terminated bus structures, the DB outputs provide a high 55 mA ( $I_{OL}$ ) capability. The noninverting μPB8216 and the inverting μPB8226 bus drivers are available to meet a wide variety of applications for buffering in microcomputer systems.

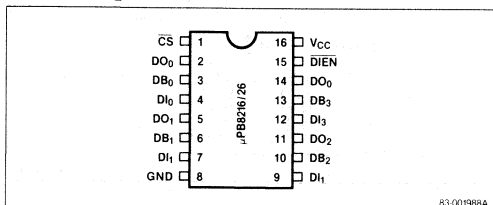
### Features

- Low input load current; 0.25 mA maximum
- High output drive capability for driving system data bus
- 3.65 V output high voltage for direct interface to CPU
- Three-state outputs
- Reduces system package count

### Ordering Information

Part Number	Package Type
μPB8216C	16-pin plastic DIP
μPB8226C	16-pin plastic DIP

### Pin Configuration



83 001988A

### Pin Identification

No.	Symbol	Function
1	CS	Chip select input
2	DO <sub>0</sub>	Data output, bit 0
3	DB <sub>0</sub>	Data bus, bit 0
4	DI <sub>0</sub>	Data input, bit 0
5	DO <sub>1</sub>	Data output, bit 1
6	DB <sub>1</sub>	Data bus, bit 1
7	DI <sub>1</sub>	Data input, bit 1
8	GND	Ground
9	DI <sub>2</sub>	Data input, bit 2
10	DB <sub>2</sub>	Data bus, bit 2
11	DO <sub>2</sub>	Data output, bit 2
12	DI <sub>3</sub>	Data input, bit 3
13	DB <sub>3</sub>	Data bus, bit 3
14	DO <sub>3</sub>	Data output, bit 3
15	DIEN	Data in enable
16	V <sub>CC</sub>	+5 V power supply

### Pin Functions

#### DB<sub>0</sub>-DB<sub>3</sub> (Bidirectional Data Bus)

Three-state data lines that interface with the system data bus. Data direction and high impedance output are functions of the CS and  $\overline{\text{DIEN}}$  control signals.

#### DI<sub>0</sub>-DI<sub>3</sub> (Data Input)

The four data input lines receive data from the CPU and make it available to the system data bus when both CS and  $\overline{\text{DIEN}}$  are active low.

#### DO<sub>0</sub>-DO<sub>3</sub> (Data Output)

The four data output lines make data available to the CPU from the system data bus when CS is active low and  $\overline{\text{DIEN}}$  is active high.

#### CS (Chip Select)

Chip select enables the chip's I/O capability when active low. When CS is high, the output drivers go to a high impedance state.



**DIEN (Data In Enable)**

$\overline{\text{DIEN}}$  is the data flow direction control signal. When low, data on the chip's input lines (DI<sub>0</sub>-DI<sub>3</sub>) from the CPU is made available to the system data bus (DB<sub>0</sub>-DB<sub>3</sub>). When high, data on the chip's data bus lines (DB<sub>0</sub>-DB<sub>3</sub>) is output to the CPU (providing  $\overline{\text{CS}}$  is active low enabled).

**VCC (Power Supply)**

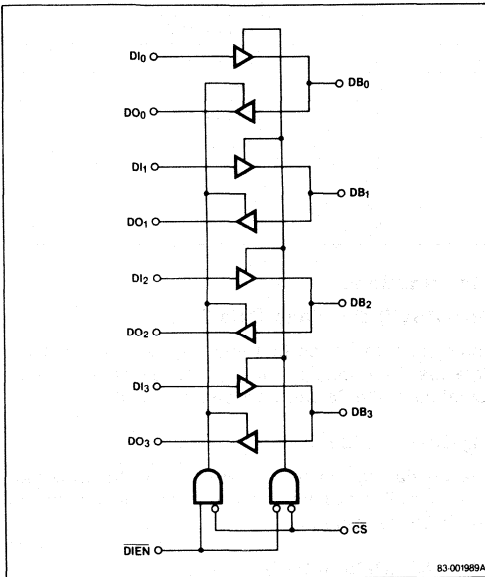
+5 V power supply input.

**GND (Ground)**

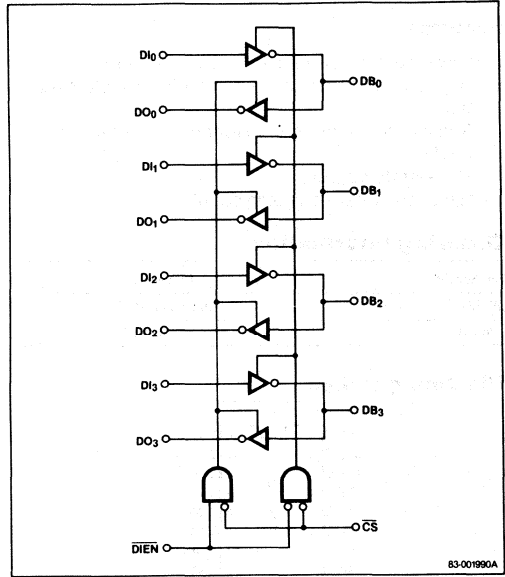
Ground.

**Block Diagrams**

$\mu$ PB8216



$\mu$ PB8226



## Functional Description

Microprocessors like the μPD8080A are MOS devices and are generally capable of driving a single TTL load. This also applies to MOS memory devices. This type of drive is sufficient for small systems with a few components, but often it is necessary to buffer the microprocessor and memories when adding components or expanding to a multiboard system.

## Bidirectional Driver

Each buffered line of the μPB8216/26 4-bit driver consists of two separate buffers. They are three-state in nature to achieve direct bus interface and bidirectional capability. On one side of the driver the output of one buffer and the input of another are tied together (DB). This is used to interface to the system side components such as memories, I/O, etc. Its interface is directly TTL-compatible and it has a high drive (55 mA). For maximum flexibility on the other side of the driver, the inputs and outputs are separate. They can be tied together so that the driver can be used to buffer a true bidirectional bus such as the 8080A data bus. The DO outputs on this side of the driver have a special high voltage output drive capability (3.65 V) so that direct interface to the 8080A processor is achieved with a maximum noise level of 650 mV.

## Control Gating $\overline{CS}$ , $\overline{DIEN}$

The  $\overline{CS}$  input is used for device selection. When  $\overline{CS}$  is high, the output drivers are all forced to their high impedance state. When it is low, the device is selected (enabled) and the data flow direction is determined by the  $\overline{DIEN}$  input.

The  $\overline{DIEN}$  input controls the data flow direction (see block diagrams for complete truth table). This directional control is accomplished by forcing one of the pair of buffers to its high impedance state. This allows the other to transmit its data. This is accomplished by a simple two-gate circuit.

The μPB8216/26 is a device that will reduce component count in microcomputer systems and at the same time enhance noise immunity to assure reliable, high performance operation.

## Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$

Power supply voltage, $V_{CC}$	-0.5 V to +7.0 V
Input voltage, $V_I$	-1.0 V to +5.5 V
Output voltage, $V_O$	-1.0 V to +5.5 V
Operating temperature, $T_{OP1}$	0°C to +70°C
Storage temperature, $T_{STG}$	-65°C to +150°C
Output current, $I_O$	125 mA

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## Capacitance (Note 1)

$T_A = 25^\circ\text{C}$ ,  $V_{CC} = 5\text{ V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	$C_I$			8	pF	$f = 1.0\text{ MHz}$
Output capacitance	$C_{O1}$			10(2)	pF	$V_{BIAS} = 2.5\text{ V}$
Output capacitance	$C_{O2}$			18(3)	pF	

### Note:

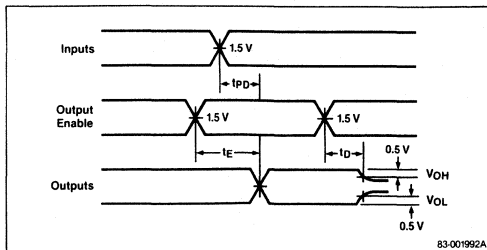
- (1) This parameter is not 100% tested.
- (2) DO output.
- (3) DB output.

**DC Characteristics**

T<sub>A</sub> = 0°C to +70°C; V<sub>CC</sub> = +5V ± 5%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage low	V <sub>IL</sub>			0.95	V	
Input voltage high	V <sub>IH</sub>	2.0			V	
Output voltage low	V <sub>OL1</sub>			0.48	V	DO outputs; I <sub>OL</sub> = 15 mA
				0.48	V	DB outputs I <sub>OL</sub> = 25 mA
	V <sub>OL2</sub>			0.7	V	8216; DB outputs; I <sub>OL</sub> = 55 mA
				0.7	V	8226; DB outputs I <sub>OL</sub> = 50 mA
Output voltage high	V <sub>OH1</sub>	3.65			V	DO outputs; I <sub>OH</sub> = -1 mA
	V <sub>OH2</sub>	2.4			V	DB outputs; I <sub>OH</sub> = -10 mA
Input forward voltage clamp	V <sub>C</sub>			-1.0	V	I <sub>C</sub> = -5 mA
Input load current	I <sub>F1</sub>			-0.5	mA	(DIEN, CS); V <sub>F</sub> = 0.45 V
	I <sub>F2</sub>			-0.25	mA	(All other inputs); V <sub>F</sub> = 0.45 V
Input leakage current	I <sub>R1</sub>			20	μA	(DIEN, CS); V <sub>R</sub> = 5.25 V
	I <sub>R2</sub>			10	μA	(DI inputs); V <sub>R</sub> = 5.25 V
Output leakage current (3-state)	I <sub>O</sub>			20	μA	DO outputs; V <sub>O</sub> = 0.45 / 5.25 V
				100	μA	DB outputs
Output short circuit current	I <sub>OS</sub>	-15		-65	mA	DO outputs; V <sub>O</sub> = 0 V
		-30		-120	mA	DB outputs V <sub>CC</sub> = 5.0 V
Power supply current	I <sub>CC</sub>			130	mA	8216
				120	mA	8226

**Timing Waveform**



**AC Characteristics**

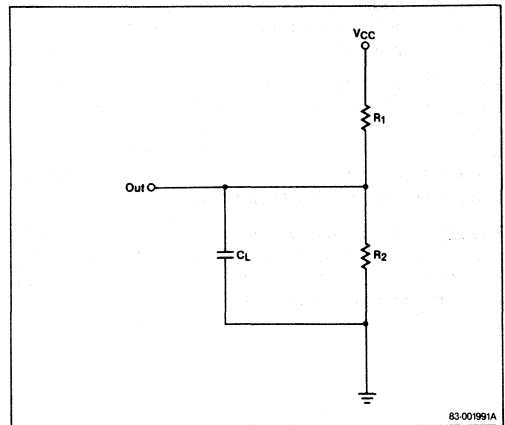
T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = +5V ± 5% (Note 1)

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input to output delay DO outputs	t <sub>PD1</sub>			25	ns	C <sub>L</sub> = 30 pF, R <sub>1</sub> = 300 Ω, R <sub>2</sub> = 600 Ω, (Note 4)
Input to output delay DB outputs	t <sub>PD2</sub>			30	ns	8216; C <sub>L</sub> = 300 pF, R <sub>1</sub> = 90 Ω, R <sub>2</sub> = 180 Ω, (Note 4)
				25	ns	8226; C <sub>L</sub> = 300 pF, R <sub>1</sub> = 90 Ω, R <sub>2</sub> = 180 Ω, (Note 4)
Output enable time	t <sub>E</sub>			65	ns	8216; (Notes 2 & 4)
				54	ns	8226; (Notes 2 & 4)
Output disable time	t <sub>D</sub>			35	ns	(Notes 3 & 4)

**Note:**

- (1) Typical values are for T<sub>A</sub> = 25°C, V<sub>CC</sub> = +5.0 V.
- (2) DO outputs, C<sub>L</sub> = 30 pF, R<sub>1</sub> = 300/10 kΩ, R<sub>2</sub> = 600/1 kΩ  
DB outputs, C<sub>L</sub> = 300 pF, R<sub>1</sub> = 90/10 kΩ, R<sub>2</sub> = 180/1 kΩ.
- (3) DO outputs, C<sub>L</sub> = 5 pF, R<sub>1</sub> = 300/10 kΩ, R<sub>2</sub> = 600/1 kΩ  
DB outputs, C<sub>L</sub> = 5 pF, R<sub>1</sub> = 90/10 kΩ, R<sub>2</sub> = 180/1 kΩ.
- (4) Input pulse amplitude: 2.5 V  
Input rise and fall times of 5 ns between 1 and 2 V.  
Output loading is 5 mA and 10 pF.  
Speed measurements are made at 1.5 V levels.

**Test Load Circuit**



## Description

The μPD8237A high performance 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 μPD8237A offers a wide variety of programmable control features to enhance data throughput and allow dynamic reconfiguration under program control.

The μPD8237A is designed to be used with an external 8-bit address register such as the 8282. 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 the user to program the types of DMA service. Each channel can be individually programmed to autoinitialize to its original condition following an end of process (EOP).

Each channel has a full 64K-byte address and word count capability.

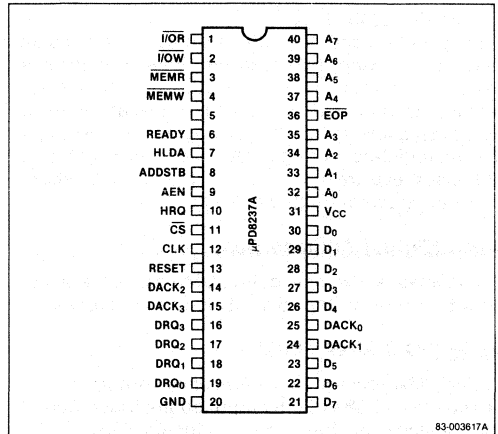
## Features

- Memory-to-memory transfers
- Memory block initialization
- Address increment or decrement
- Four independent DMA channels
- Multiple transfer modes: block, demand, single word, cascade
- Independent autoinitialization of all channels
- Enable/disable control of individual DMA requests
- Independent polarity control of DREQ and DACK signals
- End of process input for terminating transfers
- Software DMA requests
- High performance: transfers up to 1.6 Mbs
- Directly expandable to any number of channels

## Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD8237AC-5	40-pin plastic DIP	5 MHz

## Pin Configuration



83-003617A

## Pin Identification

No.	Symbol	Function
1	I/OR	I/O read control signal
2	I/OW	I/O write control signal
3	MEMR	Memory read output
4	MEMW	Memory write output
5	—	Fixed, high level input
6	READY	Ready input
7	HLDA	Hold acknowledge input
8	ADDSTB	Address strobe output
9	AEN	Address enable output
10	HRQ	Hold request output
11	CS	Chip select input
12	CLK	Clock input
13	RESET	Reset input
14, 15	DACK <sub>2</sub> , DACK <sub>3</sub>	DMA acknowledge output
24, 25	DACK <sub>1</sub> , DACK <sub>0</sub>	
16-19	DRQ <sub>3</sub> -DRQ <sub>0</sub>	DMA request input
20	GND	Ground
21-23,	D <sub>7</sub> -D <sub>5</sub>	I/O data bus
26-30	D <sub>4</sub> -D <sub>0</sub>	
31	V <sub>CC</sub>	Power supply
32-35	A <sub>0</sub> -A <sub>3</sub>	I/O address bus
36	EOP	I/O end of process
37-40	A <sub>4</sub> -A <sub>7</sub>	Output address bus

## Pin Functions

### D<sub>0</sub>-D<sub>7</sub> (I/O Data Bus)

During an I/O read, the CPU enables these lines as outputs, allowing it to read an address register, a word count register, or the status or temporary register. During an I/O write, these lines are enabled as inputs, allowing the CPU to program the μPD8237A control registers. During DMA cycles, the eight MSBs of the address are output to the data bus to be strobed to an external latch via ADDSTB.

### A<sub>4</sub>-A<sub>7</sub> (Output Address Bus)

These lines, active only during DMA service, are outputs that provide the four MSBs of the address.

### A<sub>0</sub>-A<sub>3</sub> (I/O Address Bus)

During DMA active states, these lines are outputs that provide the 4 LSBs of the output address bus. During DMA idle states, these lines are inputs, allowing the CPU to load or examine control registers.

### DRQ<sub>0</sub>-DRQ<sub>3</sub> (DMA Request Input)

These are asynchronous channel request inputs used by peripherals to request DMA service. In a fixed priority scheme, DRQ<sub>3</sub> has the lowest. The polarity of these lines is programmable; however, reset initializes them to active high.

### HLDA (Hold Acknowledge)

Indicates that the CPU has relinquished control of the system buses.

### HRQ (Hold Request)

Requests control of the system bus. The μPD8237A issues this signal in response to software requests or DRQ inputs from peripherals.

### DACK<sub>0</sub>-DACK<sub>3</sub> (DMA Acknowledge Output)

These lines indicate an active channel. They are sometimes used to select a peripheral. Only one DACK may be active at any time. All DACK lines are inactive unless DMA has control of the bus. The polarity of these lines is programmable; however, reset initializes them to active low.

### $\overline{\text{EOP}}$ (End of Process)

$\overline{\text{EOP}}$  signals that DMA service has been completed. When the word count of a channel becomes zero, the μPD8237A pulses  $\overline{\text{EOP}}$  low to notify the peripheral that DMA service is complete. The peripheral may pull  $\overline{\text{EOP}}$  low to prematurely end DMA service. Internal or external receipt of  $\overline{\text{EOP}}$  causes the currently active channel to end service, set its TC bit in the status register, and reset its request bit. If the channel is programmed for autoinitialization, the current registers are updated from the base registers. Otherwise, the channel's mask bit is set and the contents of the register are unaltered.

$\overline{\text{EOP}}$  is output when TC for channel 1 occurs during memory-to-memory transfers.  $\overline{\text{EOP}}$  applies to the channel with an active DACK. When DACK<sub>0</sub>-DACK<sub>3</sub> are inactive, external  $\overline{\text{EOP}}$ s are ignored.

Use of an external pull-up resistor of 3.3 kΩ or 4.7 kΩ is recommended. This pin ( $\overline{\text{EOP}}$ ) cannot sink the current passed by a 1 kΩ or 4.7 kΩ pull-up.

### RESET

Clears the command, status, request, and temporary registers, the first/last flip flop, and sets the mask register. The μPD8237A is in idle state after a reset.

### $\overline{\text{CS}}$ (Chip Select)

The CPU uses  $\overline{\text{CS}}$  to select the μPD8237A as an I/O device during an I/O read or write by the CPU. This provides CPU communication on the data bus.  $\overline{\text{CS}}$  may be held low during multiple transfers to or from the μPD8237A as long as  $\overline{\text{I/O}}$  or  $\overline{\text{I/O}}$  is toggled following each transfer.

### READY

This signal can extend memory read and write pulses for slow memories or I/O peripherals.

### CLK (Clock)

Controls internal operations and data transfer rate.

### AEN (Address Enable)

This signal allows the external latch to output the upper address byte by disabling the system bus during DMA cycles. Use HLDA and AEN to deselect I/O peripherals that may be erroneously accessed during DMA transfers. The μPD8237A deselects itself during DMA transfers.

### ADDSTB (Address Strobe)

This signal strobes the upper address byte from D<sub>0</sub>-D<sub>7</sub> into an external latch.

### MEMR (Memory Read)

This signal accesses data from a specified memory location during memory-to-peripheral or memory-to-memory transfers.

### MEMW (Memory Write)

This signal writes data to a specified memory location during peripheral-to-memory or memory-to-memory transfers.

### I/O $\overline{R}$ (I/O Read)

In the idle state, this signal is an input control line used by the CPU to read control registers. In the active state, the μPD8237A uses I/O $\overline{R}$  as an output control signal to access data from a peripheral during a DMA write.

### I/O $\overline{W}$ (I/O Write)

In the idle state, the CPU uses I/O $\overline{W}$ , as an input control signal to load information to the μPD8237A. In the active state, the μPD8237A uses I/O $\overline{W}$  as an output control signal to load data to a peripheral during a DMA read.

The rising edge of  $\overline{WR}$  must follow each data byte transfer in order for the CPU to write to the μPD8237A. Holding I/O $\overline{W}$  low while toggling  $\overline{CS}$  does not produce the same effect.

### Pin 5

Pin 5 is always tied high.

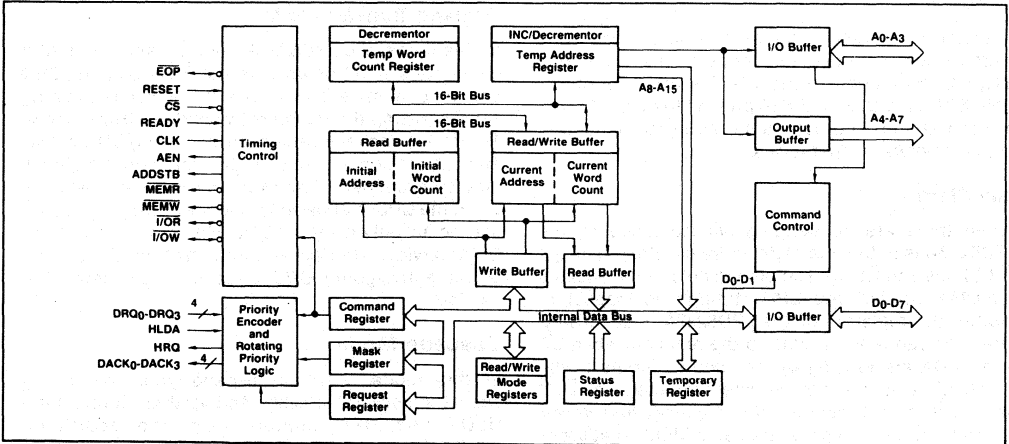
### V<sub>CC</sub>

Power supply.

### GND

Ground.

### Block Diagram



7

## Functional Description

The μPD8237A has three basic control logic blocks, as shown in the block diagram. The command control block decodes commands issued by the CPU to the μPD8237A before DMA requests are serviced. It also decodes the mode control word of each channel. The timing control block generates the external control signals and the internal timing. The priority encoder block settles priority contentions among channels simultaneously requesting service.

## DMA Operation

The μPD8237A operates in two states: idle and active. Each of these is made up of several smaller states equal to one clock cycle. The inactive state, S1, is entered when there are no pending DMA requests. The controller is inactive in S1, but the CPU may program it. S0 is the initial state for DMA service; the μPD8237A requests a hold, but the CPU has not acknowledged. Transfers may begin upon acknowledgement from the CPU. The normal working states of DMA service are S1, S2, S3, and S4. If more time is needed for a transfer, a wait state, SW, can be inserted using the READY line.

A memory-to-memory transfer requires read-from-memory and write-to-memory operations. The states S11, S12, S13, and S14 provide the read-from operation. S21, S22, S23, and S24 provide the write-to part of the transfer. The byte is stored in the temporary register between operations.

## Idle State

When there are no pending service requests, the μPD8237A is in the idle state; more specifically, in S1, DRQ lines and  $\overline{CS}$  are sampled to determine requests for DMA service and CPU attempts to inspect or modify the registers of the μPD8237A, respectively. The CPU can read or write to the registers when  $\overline{CS}$  and HLDA are low. A<sub>0</sub>-A<sub>3</sub> are used as inputs to the μPD8237A and select the registers affected. The  $\overline{I/OR}$  and  $\overline{I/OW}$  lines select and time the reads and writes. An internal flip-flop generates an additional address bit which determines the upper or lower byte of the address and word count registers. This flip-flop can be reset by master clear, reset, or a software command.

When  $\overline{CS}$  and HLDA are low (program phase), the μPD8237A can execute special software commands. When  $\overline{CS}$  and  $\overline{I/OW}$  are active, the commands are decoded as addresses and do not use the data bus.

## Active State

When a channel requests service while the μPD8237A is in idle state, the μPD8237A outputs an HRQ to the CPU and enters the active state. DMA service takes place in the active state, in one of the four modes described below.

DRQ is held active only until the corresponding DACK goes active when a single transfer is performed. If DRQ is held active for a longer period, HRQ will become inactive after each transfer, become active again, and a one-byte transfer will be made after each rising edge of HLDA. This assures a full machine cycle between DMA transfers in 8080A/8085A systems. Timing between the μPD8237A and other bus control protocols depends on the CPU being used.

## Block Transfer Mode

In this mode, the μPD8237A makes transfers until it encounters a TC or an external EOP. Hold DRQ active only until DACK goes active. The channel will autoinitialize at the end of the DMA service if it has been programmed to do so.

## Demand Transfer Mode

In this mode, the μPD8237A makes transfers until it encounters a TC or an external  $\overline{EOP}$ , or until DRQ becomes inactive. This allows the device requesting service to stop the transfers by sending DRQ inactive. The device can resume service by making DRQ active. The current address and current word count registers may be examined during the time between services when the CPU is allowed to operate. Autoinitialization can occur only after a TC or  $\overline{EOP}$  at the end of the DMA service. After an autoinitialization, there must be an active-going DRQ edge to begin new DMA service.

## Cascade Mode

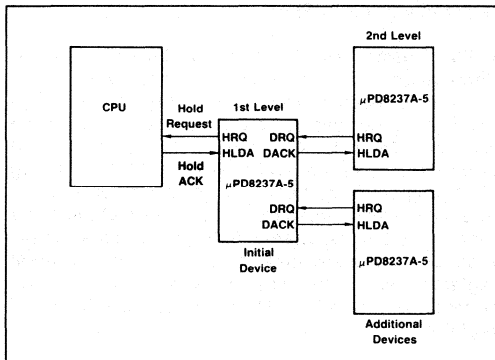
In this mode, you can expand your system by cascading several μPD8237As together. Connect the HLDA and HRQ signals from the additional μPD8237As to the DRQ and DACK signals of a channel of the initial μPD8237A. This scheme allows the additional devices to send the DMA requests through the priority resolution circuitry of the preceding device, preserving the priority chain and forcing the device to wait its turn to acknowledge requests. The cascade channel in the initial device does not output any address or control signals because its only function is that of assigning priorities. The μPD8237A responds to DRQ with DACK, but all outputs except HRQ are disabled.

Figure 1 shows two μPD8237As cascaded into two channels of another one, forming a two-level DMA system. You could add more devices at the second level by using the leftover channels of the first level; likewise, you could add more devices to form a third level by cascading into the channels of the second level.

## Transfers

There are three types of transfers that can be performed by the three active transfer modes: read, write, and verify. Read transfers activate MEMR and I/O to move memory data to an I/O device. Write transfers activate I/OR and MEMW to move data from an I/O device to memory. Verify transfers are not really transfers; the μPD8237A goes through the motions of a transfer but the memory and I/O lines are not active.

**Figure 1. Two-Level DMA System**



## Memory-to-Memory Transfers

Use block transfer mode for memory-to-memory transfers. Mask out channels 0 and 1, and initialize the channel 0 word count to the same value as channel 1. Setting bit C0 of the command register to 1 makes channels 0 and 1 operate as memory-to-memory transfer channels. Channel 0 is the source address, channel 1 is the destination address, and the channel 1 word count is used. Initiate the memory-to-memory transfer by setting a DMA request for channel 0. You can write a single source word to a block of memory when channel 0 is programmed for a fixed source address. The μPD8237A responds to external EOP signals during these transfers, but no DACK outputs are active. The EOP input may be used by data comparators doing block searches to end service when a match is found.

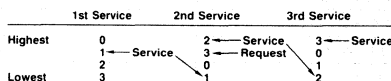
## Autoinitialization

A channel may be set for autoinitialize by programming a bit in the mode register. Autoinitialize restores the original values of the current address and current word count registers from the initial address and initial word count registers of that channel. The CPU loads the current and initial registers simultaneously and they are unchanged through DMA service. EOP does not set the mask bit when the channel is in autoinitialize. The channel can repeat its service following autoinitialize without CPU intervention.

## Priority Resolution

Two software-selectable priority resolution schemes are available on the μPD8237A: fixed priority and rotating priority. In the fixed priority scheme, priority is assigned by the value of the channel number. Channel 3 is the lowest priority and channel 0 is the highest priority.

In the rotating priority scheme, the channel that was just serviced assumes the lowest priority and the other channels move up accordingly. This guarantees that a device requesting service can be acknowledged after no more than three other devices have been serviced, preventing any channel from monopolizing the system.



The highest priority channel is selected on each active-going HLDA edge. Once service to a channel begins, it cannot be interrupted by a request from a higher priority channel. A higher priority channel gets control only when the lower priority channel releases HRQ. The CPU gets bus control when control passes from channel to channel, ensuring that a rising HLDA edge can be generated to select the new highest priority request.

## Transfer Timing

If, you can cut transfer timing, by compressing the transfer time to two clock periods. Since state 3 (S3) extends the access time for the read pulse, you can eliminate S3, making the width of the read pulse equal to the write pulse. A transfer is then made up of S2 to change the address and S4 to perform the read or write. When the address lines A<sub>8</sub>-A<sub>15</sub> need to be updated, S1 states occur.



## Generating Addresses

The eight MSBs of the address are multiplexed on the data lines. These bits are output to an external latch during S1, after which they can be placed on the address bus. The falling edge of ADDSTB loads the bits from the data lines to the latch. AEN places the bits on the address bus. The eight LSBs of the address are directly output on lines A<sub>0</sub>-A<sub>7</sub> to the address bus.

Sequential addresses are generated during block and demand transfer mode operations because they include several transfers. Often, data in the external address latch does not change; it changes only when a carry or borrow from A<sub>7</sub> to A<sub>8</sub> occurs in the sequence of addresses. S1 states are executed only when A<sub>8</sub>-A<sub>15</sub> need to be updated. In the course of lengthy transfers, S1 states may be executed only once every 256 transfers.

## Registers

Table 1 summarizes the registers of the  $\mu$ PD8237A.

**Table 1. Register Summary**

Register	No.	Bits
Current address registers	4	16
Current word count registers	4	16
Initial address registers	4	16
Initial word count registers	4	16
Command register	1	8
Mode registers	4	6
Request register	1	4
Mask register	1	4
Status register	1	8
Temporary register	1	8
Temporary address register	1	16
Temporary word count register	1	16

**Current Address Register.** There is a current address register for each channel. This register holds the address used for DMA transfers; the address is incremented or decremented after each transfer and the intermediate values are stored here during the transfer. The CPU writes or reads this register in 8-bit bytes. An autoinitialize restores this register to its initial value.

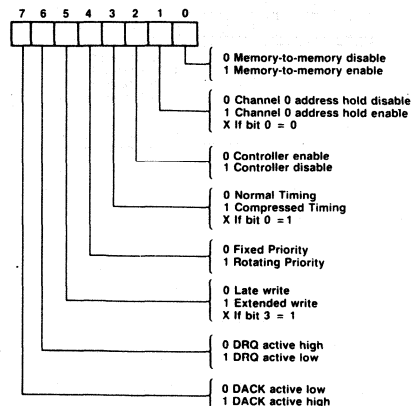
**Current Word Count Register.** There is a current word count register for each channel. Program this register with the value of the number of words to be transferred, minus one. The word count is decremented after each transfer and intermediate values are stored in this register during the transfer. A TC is generated when the word count is zero. The CPU writes or reads this register in 8-bit bytes during program phase. An autoinitialize restores this register to its initial value. After an internally generated  $\overline{EOP}$ , the contents of this register will be FFFFH.

**Initial Address and Initial Word Count Registers.** There is an initial register and an initial word count register for each channel. The initial values of the associated current registers are stored in these registers. The values in these registers are used to restore the current registers at autoinitialize. During DMA programming, the CPU writes the initial registers and the corresponding current registers at the same time, in 8-bit bytes. Intermediate values in the current registers are overwritten if you write to the initial registers while the current registers contain intermediate values. The CPU cannot read the initial registers.

**Table 2. Word Count and Address Register Command Codes**

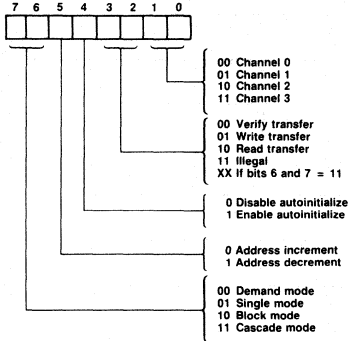
Channel	Operation	Signals						Internal Flip-Flop	D <sub>0</sub> -D <sub>7</sub>	
		CS	I/OR	I/OW	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>			A <sub>0</sub>
0	Initial & current address write	0	1	0	0	0	0	0	0	A <sub>0</sub> -A <sub>7</sub>
		0	1	0	0	0	0	0	0	A <sub>8</sub> -A <sub>15</sub>
	Current address read	0	0	1	0	0	0	0	0	A <sub>0</sub> -A <sub>7</sub>
		0	0	1	0	0	0	0	0	A <sub>8</sub> -A <sub>15</sub>
	Initial & current word count write	0	1	0	0	0	0	1	0	W <sub>0</sub> -W <sub>7</sub>
		0	1	0	0	0	0	1	1	W <sub>8</sub> -W <sub>15</sub>
Current word count read	0	0	1	0	0	0	1	0	W <sub>0</sub> -W <sub>7</sub>	
	0	0	1	0	0	0	1	1	W <sub>8</sub> -W <sub>15</sub>	
1	Initial & current address write	0	1	0	0	0	1	0	0	A <sub>0</sub> -A <sub>7</sub>
		0	1	0	0	0	1	0	1	A <sub>8</sub> -A <sub>15</sub>
	Current address read	0	0	1	0	0	1	0	0	A <sub>0</sub> -A <sub>7</sub>
		0	0	1	0	0	1	0	1	A <sub>8</sub> -A <sub>15</sub>
	Initial & current word count write	0	1	0	0	0	1	1	0	W <sub>0</sub> -W <sub>7</sub>
		0	1	0	0	0	1	1	1	W <sub>8</sub> -W <sub>15</sub>
Current word count read	0	0	1	0	0	1	1	0	W <sub>0</sub> -W <sub>7</sub>	
	0	0	1	0	0	1	1	1	W <sub>8</sub> -W <sub>15</sub>	
2	Initial & current address write	0	1	0	0	1	0	0	0	A <sub>0</sub> -A <sub>7</sub>
		0	1	0	0	1	0	0	1	A <sub>8</sub> -A <sub>15</sub>
	Current address read	0	0	1	0	1	0	0	0	A <sub>0</sub> -A <sub>7</sub>
		0	0	1	0	1	0	0	1	A <sub>8</sub> -A <sub>15</sub>
	Initial & current word count write	0	1	0	0	1	0	1	0	W <sub>0</sub> -W <sub>7</sub>
		0	1	0	0	1	0	1	1	W <sub>8</sub> -W <sub>15</sub>
Current word count read	0	0	1	0	1	0	1	0	W <sub>0</sub> -W <sub>7</sub>	
	0	0	1	0	1	0	1	1	W <sub>8</sub> -W <sub>15</sub>	
3	Initial & current address write	0	1	0	0	1	1	0	0	A <sub>0</sub> -A <sub>7</sub>
		0	1	0	0	1	1	0	1	A <sub>8</sub> -A <sub>15</sub>
	Current address read	0	0	1	0	1	1	0	0	A <sub>0</sub> -A <sub>7</sub>
		0	0	1	0	1	1	0	1	A <sub>8</sub> -A <sub>15</sub>
	Initial & current word count write	0	1	0	0	1	1	1	0	W <sub>0</sub> -W <sub>7</sub>
		0	1	0	0	1	1	1	1	W <sub>8</sub> -W <sub>15</sub>
Current word count read	0	0	1	0	1	1	1	0	W <sub>0</sub> -W <sub>7</sub>	
	0	0	1	0	1	1	1	1	W <sub>8</sub> -W <sub>15</sub>	

**Command Register.** The CPU programs this register during program phase. The register can be cleared with reset.



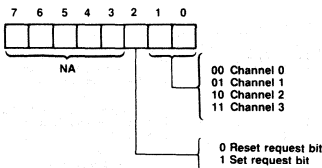
## μPD8237A

**Mode Register.** There is a mode register associated with each channel. When the CPU writes to this register during the program phase, bits 0 and 1 determine on which channel mode register the operation is performed.

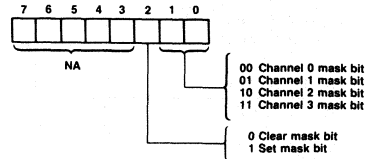


**Request Register.** This register allows the μPD8237A to respond to DMA requests from software as well as hardware. There is a bit pattern for each channel in the request register. These bits can be prioritized by the priority resolving circuitry and are not maskable. Each bit is set or reset under software control or cleared when TC or an external EOP is generated. A reset clears the entire register. The correct data word is loaded by software to set or reset a bit.

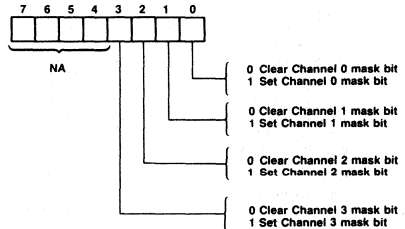
Software requests receive service only when the channel is in block mode. The software request for channel 0 should be set at the beginning of a memory-to-memory transfer.



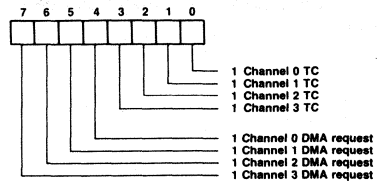
**Mask Register.** There is a mask bit for each channel which can disable an incoming DRQ. If the channel is not set for autoinitialize, each mask bit is set when its channel produces an EOP. Each bit can be set or cleared under software control. Reset clears the register. This disallows DMA requests until they are permitted by a clear mask register instruction.



You may also write all four bits of the mask register with a single command.



**Status Register.** The status register indicates which channels have made DMA requests and which channels have reached TC. Each time a channel reaches TC, including after autoinitialization, bits 0-3 are set. Status read and reset clear these bits. Bits 4-7 are set when a channel is requesting service. The CPU can read the status register.



**Temporary Register.** The temporary register holds data during memory-to-memory transfers. The CPU can read the last word moved when the transfer is complete. This register always contains the last byte transferred in a memory-to-memory transfer unless cleared by a reset.

### Software Commands

There are two software commands that can be executed in the program phase. These commands are independent of data on the data bus.

**Clear First/Last Flip-Flop.** You may issue this command before reading or writing any word count or address information. It allows the CPU to access registers, addressing upper and lower bytes correctly by initializing the flip-flop to an identifiable state.

**Master Clear.** This command produces the same effect as reset. It clears the command, status, request, temporary, and internal first/last flip-flop registers, sets the mask register, and causes the μPD8237A to enter idle state.

Table 3 illustrates address codes for the software commands.

**Table 3. Software Command Codes**

A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	I/OR	I/OW	(1)	Operation
1	0	0	0	0	1		Read status register
1	0	0	0	1	0		Write to command register
1	0	0	1	1	0		Write to request register
1	0	1	0	1	0		Write a mask register bit
1	0	1	1	1	0		Write to mode register
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	1	1	0		Write all mask register bits
1	1	1	0	0	1		Clear Mask register

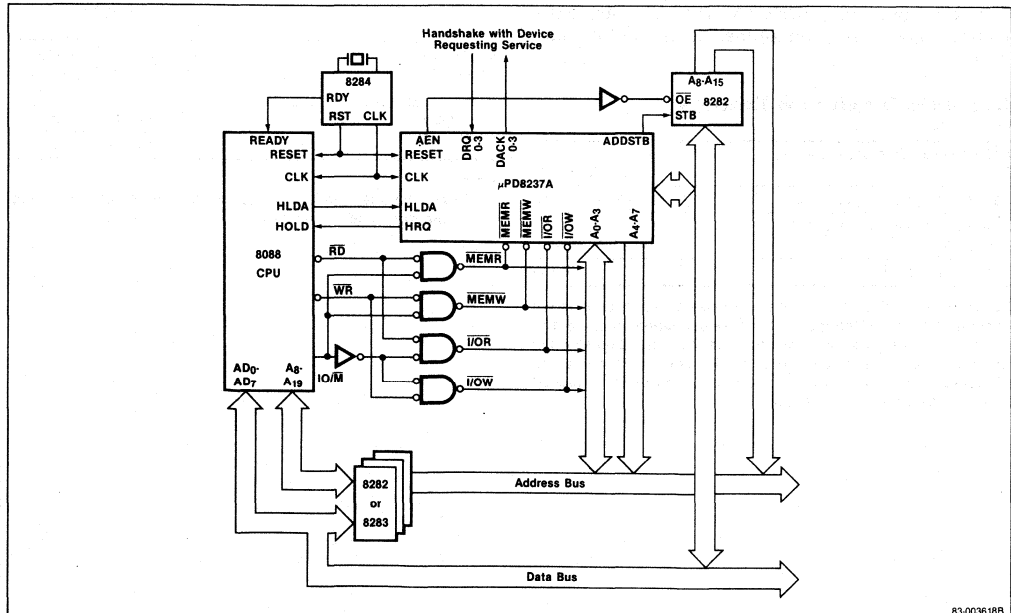
**Note:**

(1) All other bit combinations are illegal.

### Application Example

Figure 2 shows an application using the μPD8237A with an 8088. The μPD8237A sends a hold request to the CPU whenever there is a valid DMA request from a peripheral device. The μPD8237A takes control of the address, data, and control buses when the CPU

**Figure 2. μPD8237A DMA Controller Application with 8088 CPU**



## μPD8237A

replies with an HLDA signal. The address for the first transfer appears in two bytes: the eight LSBs are output on A<sub>0</sub>-A<sub>7</sub> and the eight MSBs are output on the data bus pins. The contents of the data bus pins are latched to the 8282 to make up the 16 bits of the address bus. Once the address is latched, the data bus transfers data to or from a memory location or I/O device, using the control bus signals generated by the μPD8237A.

### AC Characteristics Supplementary Information

All AC timing measurement points are 2.0 V for high and 0.8 V for low, for both inputs and outputs. The loading on the outputs is one TTL gate plus 100 pF of capacitance for the data bus pins, and one TTL gate plus 50 pF for all other outputs.

Recovery time between successive read and write inputs must be at least 400 ns. I/O or memory write pulse widths will be T<sub>CY</sub>-100 ns for normal DMA transfers and 2 T<sub>CY</sub>-100 ns for extended cycles. I/O or memory reads will be 2 T<sub>CY</sub>-50 ns for normal reads and T<sub>CY</sub>-50 ns for compressed cycles. T<sub>DQ1</sub> and T<sub>DQ2</sub> are measured on two different levels: T<sub>DQ1</sub> at 2.0 V, T<sub>DQ2</sub> at 3.3 V with a 3.3 kΩ pull-up resistor. DREQ and DACK are both active high and low. DREQ must be held in the active state (user defined) until DACK is returned from the μPD8237A. The AC waveforms assume these are programmed to the active high state.

### Absolute Maximum Ratings

T<sub>A</sub> = 25°C

Ambient temperature under bias, T <sub>OPT</sub>	0°C to +70°C
Storage temperature, T <sub>STG</sub>	-65°C to +150°C
Voltage on any pin with respect to Ground, V <sub>CC</sub>	-0.5V to +7V
Power dissipation, P <sub>D</sub>	1.5 W

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### Capacitance

T<sub>A</sub> = 25°C

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ <sup>(1)</sup>	Max		
Output capacitance	C <sub>O</sub>		4	8	pF	f <sub>c</sub> = 1.0 MHz, Inputs = 0 V
Input capacitance	C <sub>I</sub>		8	15	pF	
I/O capacitance	C <sub>I/O</sub>		10	18	pF	

**Note:**

(1) Typical values measured at T<sub>A</sub> = 25°C, nominal processing parameters, and nominal V<sub>CC</sub>.

### DC Characteristics

T<sub>A</sub> = 0°C to +70°C; V<sub>CC</sub> = +5 V ± 5%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ <sup>(1)</sup>	Max		
Output high voltage	V <sub>OH</sub>	2.4			V	I <sub>OH</sub> = -200 μA
		3.3			V	I <sub>OH</sub> = -100 μA (HRQ only)
Output low voltage	V <sub>OL</sub>			0.45		I <sub>OL</sub> = 2.0 mA (Data bus)
					V	I <sub>OL</sub> = 3.2 mA (Other outputs)
Input high voltage	V <sub>IH</sub>	2.0		V <sub>CC</sub> + 0.5	V	
Input low voltage	V <sub>IL</sub>	-0.5		0.8	V	
Input load current	I <sub>LI</sub>			±10	μA	0V ≤ V <sub>IN</sub> ≤ V <sub>CC</sub>
Output leakage current	I <sub>LO</sub>			±10	μA	0.45 ≤ V <sub>OUT</sub> ≤ V <sub>CC</sub>
V <sub>CC</sub> supply current	I <sub>CC</sub>		65	130	mA	T <sub>A</sub> = +25°C
				75	150	mA

**Note:**

(1) Typical values measured at T<sub>A</sub> = 25°C, nominal processing parameters, and nominal V<sub>CC</sub>.

## AC Characteristics

### DMA (Master) Mode

$T_A = 0^\circ\text{C to } +70^\circ\text{C}; V_{CC} = 5\text{V} \pm 5\%; V_{SS} = 0\text{V}$

Parameter	Symbol	Limits			Unit
		Min	Typ	Max	
AEN high from CLK low (S1) delay time	$t_{AEL}$		200		ns
AEN low from CLK high (S1) delay time	$t_{AET}$		130		ns
ADR active to float delay from CLK high	$t_{AFAB}$		90		ns
READ or WRITE float from CLK high	$t_{AFC}$		120		ns
DB active to float delay from CLK high	$t_{AFDB}$		170		ns
ADR from READ high hold time	$t_{AHR}$	$t_{CY}-100$			ns
DB from ADDSTB low hold time	$t_{AHS}$	30			ns
ADR from WRITE high hold time	$t_{AHW}$	$t_{CY}-50$			ns
DACK valid from CLK low delay time	$t_{AK}$		170		ns
EOP high from CLK high delay time	$t_{AK}$		170		ns
EOP low to CLK high delay time	$t_{AK}$		100		ns
ADR stable from CLK high	$t_{ASM}$		170		ns
Data bus to ADDSTB low setup time	$t_{ASS}$	100			ns
Clock high time (transitions $\leq 10$ ns)	$t_{CH}$	80			ns
Clock low time (transitions $\leq 10$ ns)	$t_{CL}$	68			ns
CLK cycle time	$t_{CY}$	200			ns
CLK high to READ or WRITE low delay <sup>(1)</sup>	$t_{DCL}$		190		ns
READ high from CLK high (S-4) delay time <sup>(1)</sup>	$t_{DCTR}$		190		ns
WRITE high from CLK high (S-4) delay time <sup>(1)</sup>	$t_{DCTW}$		130		ns
HRQ valid from CLK high delay time <sup>(2)</sup>	$t_{DQ1}$		120		ns
	$t_{DQ2}$		120		ns
EOP low from CLK low setup time	$t_{EPS}$	40			ns
EOP pulse width	$t_{EPW}$	220			ns
ADR float to active delay from CLK high	$t_{FAAB}$		170		ns
READ or WRITE active from CLK high	$t_{FAC}$		150		ns

Parameter	Symbol	Limits			Unit
		Min	Typ	Max	
Data bus float to active delay from CLK high	$t_{FADB}$			200	ns
HLDA valid to CLK high setup time	$t_{HS}$	75			ns
Input data from MEMR high hold time	$t_{IDH}$	0			ns
Input data to MEMR high setup time	$t_{IDS}$	170			ns
Output data from MEMW high hold time	$t_{ODH}$	10			ns
Output data valid to MEMW high	$t_{ODV}$	125			ns
DRQ to CLK low (S1, S4) setup time	$t_{QS}$	0			ns
CLK to READY low hold time	$t_{RH}$	20			ns
READY to CLK low setup time	$t_{RS}$	60			ns
ADDSTB high from CLK high delay time	$t_{STL}$			130	ns
ADDSTB low from CLK high delay time	$t_{STT}$			90	ns

#### Note:

(1) Net  $\overline{IOW}$  or  $\overline{MEMW}$  pulse width for normal write is  $t_{CY}-100$  ns and  $t_{CY}-100$  ns for extended write. Net  $\overline{IOR}$  or  $\overline{MEMR}$  pulse width for normal read to  $2t_{CY}-50$  ns and  $t_{CY}-50$  ns for compressed read.

(2)  $t_{DQ1}$  is measured at 2.0 V.  $t_{DQ2}$  is measured at 3.3 V. An external pullup resistor of  $3.3\Omega$  connected from HRQ to  $V_{CC}$  is assumed for  $t_{DQ2}$ .

**AC Characteristics (cont)**

**Peripheral Mode**

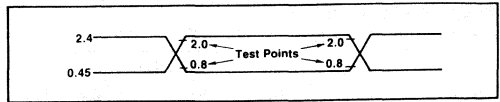
T<sub>A</sub> = 0°C to +70°C; V<sub>CC</sub> = 5 V ± 5%; V<sub>SS</sub> = 0 V

Parameter	Symbol	Limits			Unit
		Min	Typ	Max	
ADR valid or $\overline{\text{CS}}$ low to $\overline{\text{READ}}$ low	t <sub>AR</sub>	50			ns
ADR valid to $\overline{\text{WRITE}}$ high setup time	t <sub>AW</sub>	150			ns
CS low to $\overline{\text{WRITE}}$ high setup time	t <sub>CW</sub>	150			ns
Data valid to $\overline{\text{WRITE}}$ high setup time	t <sub>DW</sub>	150			ns
ADR or CS hold from $\overline{\text{READ}}$ high	t <sub>RA</sub>	0			ns
Data access from $\overline{\text{READ}}$ low <sup>(1)</sup>	t <sub>RDE</sub>			140	ns
Data bus float delay from $\overline{\text{READ}}$ high	t <sub>RDF</sub>	0		70	ns
Power supply high to RESET low setup time	t <sub>RSTD</sub>	500			ns
RESET to first I/OR or I/OW	t <sub>RSTS</sub>	2t <sub>CY</sub>			ns
RESET pulse width	t <sub>RSTW</sub>	300			ns
$\overline{\text{READ}}$ width	t <sub>RW</sub>	200			ns
ADR from $\overline{\text{WRITE}}$ high hold time	t <sub>WA</sub>	20			ns
CS high from $\overline{\text{WRITE}}$ high hold time	t <sub>WC</sub>	20			ns
Data from $\overline{\text{WRITE}}$ high hold time	t <sub>WD</sub>	30			ns
Write width	t <sub>WWS</sub>	160			ns

**Note:**

(1) Data bus output loading is 1 TTL gate plus 100 pF capacitance.

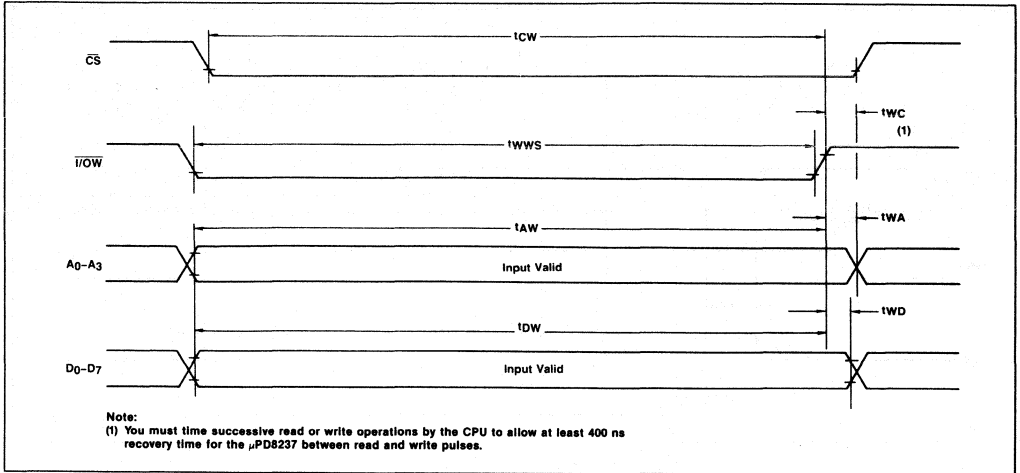
**AC Testing Input/Output Waveform**



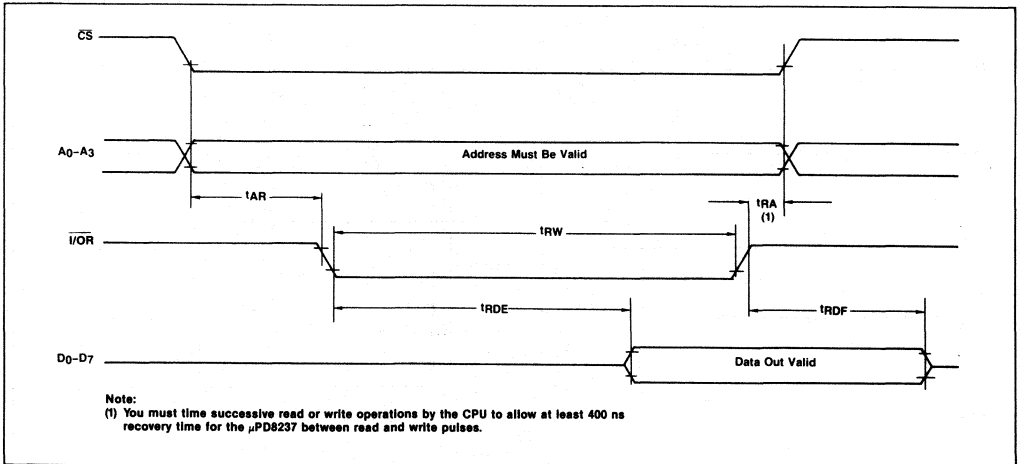
Inputs are driven at 2.4V for logic 1 and 0.45V for logic 0. These timing measurements are made at 2.0V for logic 1 and 0.8V for logic 0. A transition time of 20 ns or less is assumed for input timing parameters. Unless noted, output loading is 1 TTL gate plus 50 pF capacitance.

## Timing Waveforms

### Slave Mode Write



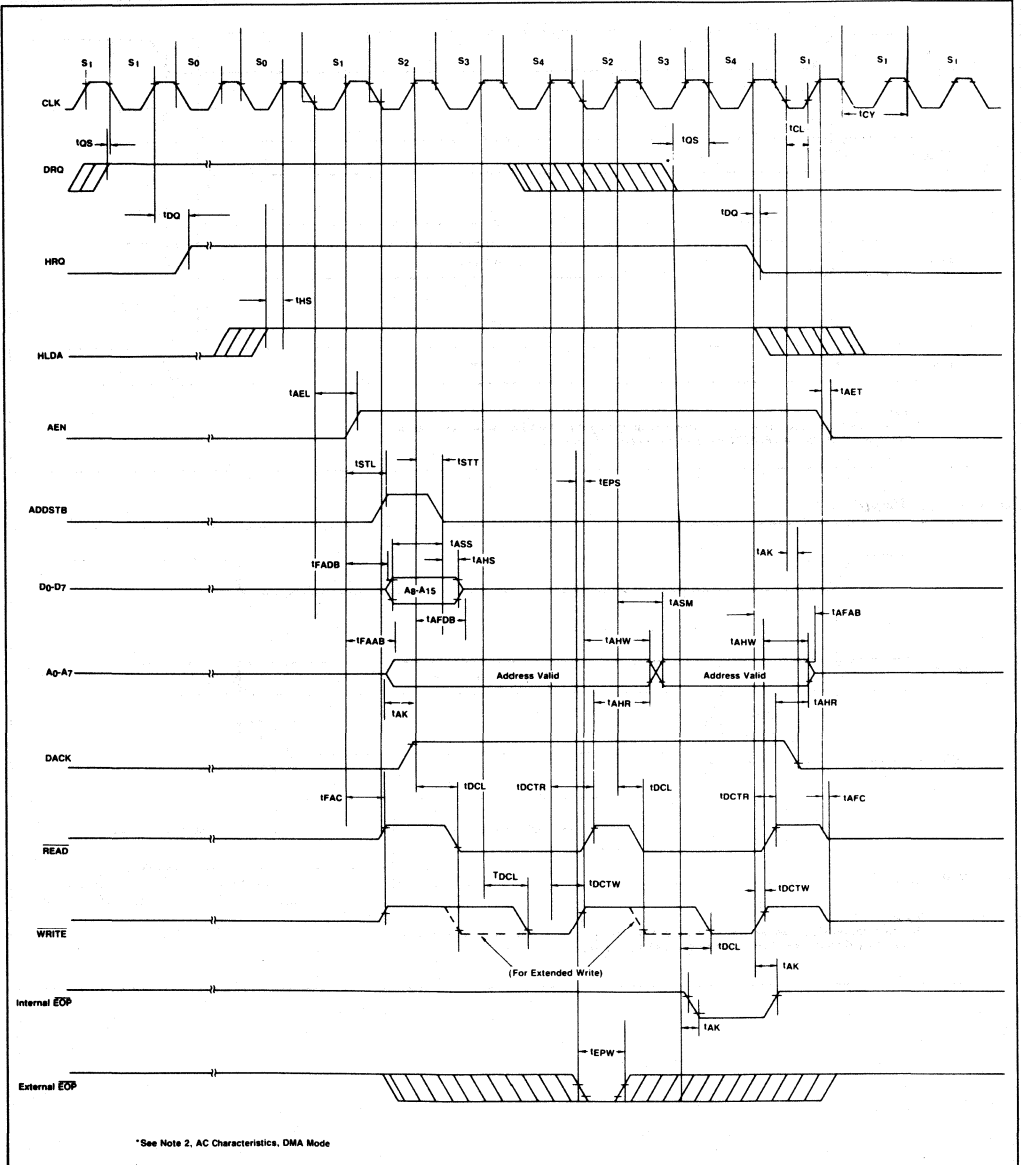
### Slave Mode Read





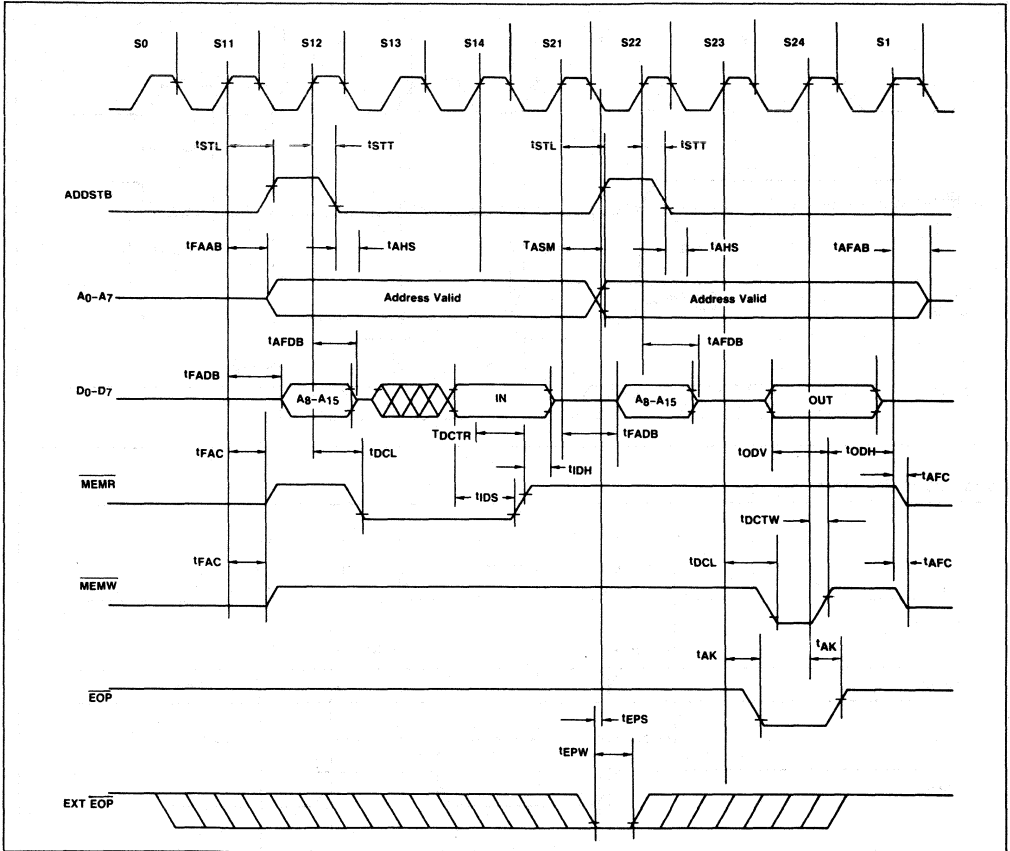
Timing Waveforms (cont)

DMA Transfer



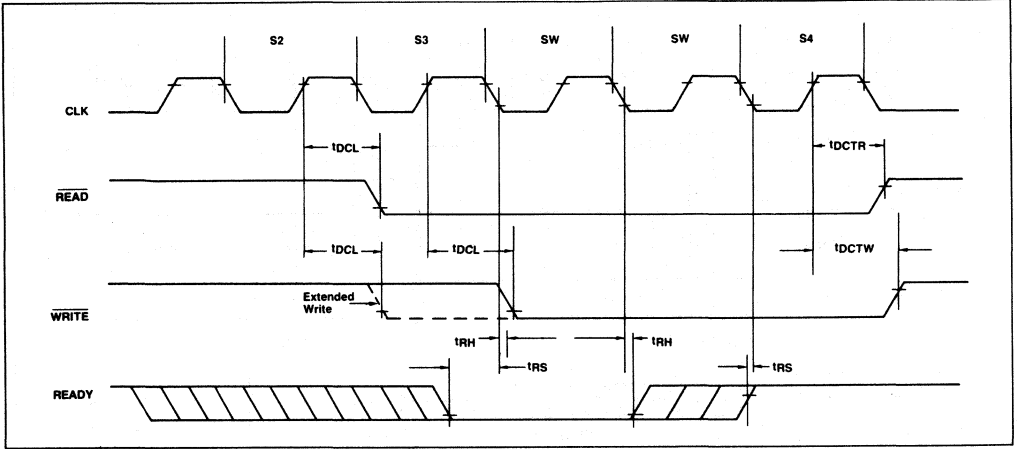
## Timing Waveforms (cont)

### Memory-to-Memory Transfer

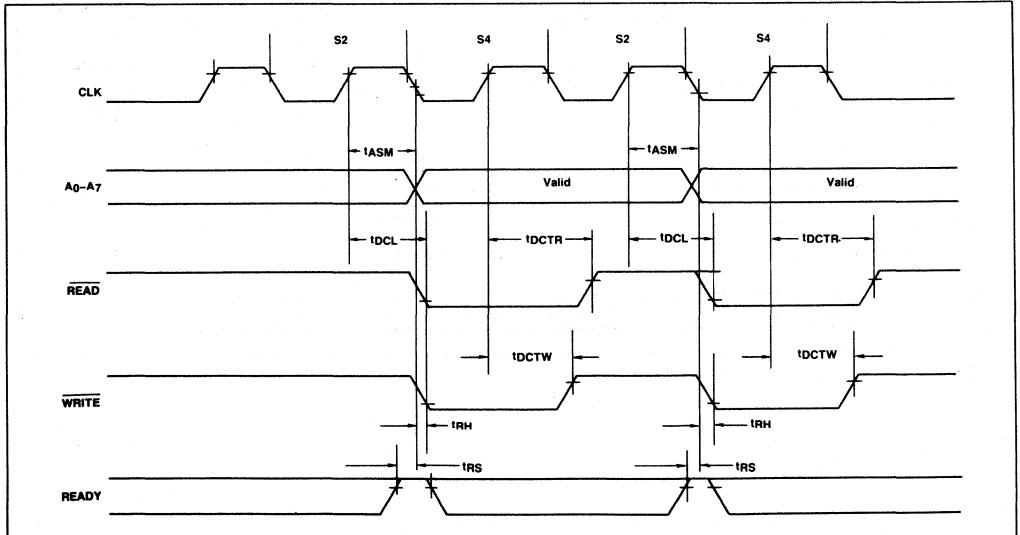


Timing Waveforms (cont)

Ready

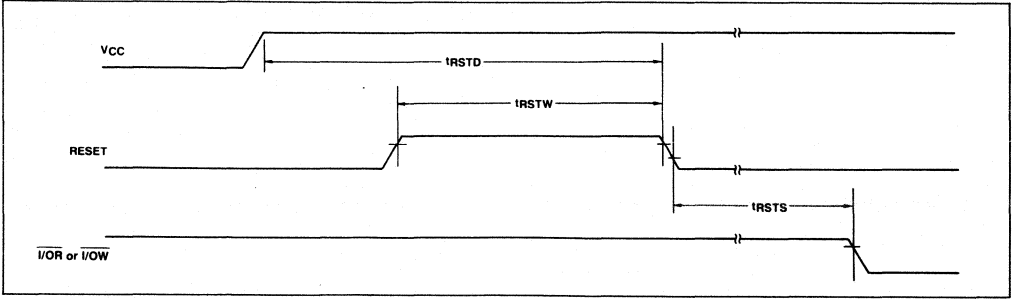


Compressed Transfer



## Timing Waveforms (cont)

### Reset





### Description

The μPD8243 and μPD8243H input/output expander is directly compatible with the μPD8048 family of single-chip microcomputers. Using NMOS technology the μPD8243 provides high drive capabilities while requiring only a single +5 V supply voltage.

The μPD8243 interfaces to the μPD8048 family through a 4-bit I/O port and offers four 4-bit bidirectional static I/O ports. The ease of expansion allows for multiple μPD8243s to be added using the bus port.

The bidirectional I/O ports of the μPD8243 act as an extension of the I/O capabilities of the μPD8048 microcomputer family. They are accessible with their own ANL, MOV, and ORL instructions.

Another version, μPD8243H, has less total output current sinking capability than μPD8243 but is otherwise identical.

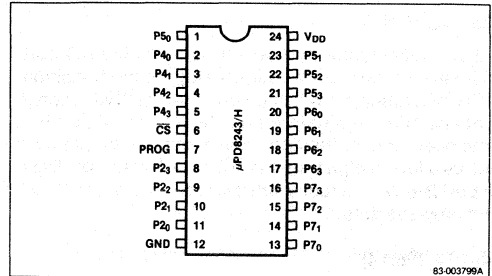
### Features

- Four 4-bit I/O ports
- High output drive
- Logical AND and OR directly to ports
- Compatible with industry standard 8243
- Direct extension of resident μPD8048 I/O ports
- Fully compatible with μPD8048 microcomputer family
- NMOS technology
- Single +5 V supply

### Ordering Information

Part Number	Package Type
μPD8243C	24-pin plastic DIP
μPD8243HC	24-pin plastic DIP

### Pin Configuration



### Pin Identification

No.	Symbol	Function
1, 21-23	P50-P53	4-bit I/O port 5
2-5	P40-P43	4-bit I/O port 4
6	CS	Chip select input
7	PROG	Clock input
8-11	P20-P23	4-bit I/O CPU interface port 2
12	GND	Ground
13-16	P70-P73	4-bit I/O port 7
17-20	P60-P63	4-bit I/O port 6
24	VDD	+5 V power supply

### Absolute Maximum Ratings

T<sub>A</sub> = 25°C

Power supply voltage, V <sub>DD</sub> (to ground)	-0.5 to +7 V
Operating temperature, T <sub>OP</sub>	0 to +70°C
Storage temperature, T <sub>STG</sub>	-65 to +150°C
Power dissipation, P <sub>D</sub>	1.0 W

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Pin Functions**

**P2<sub>0</sub>-P2<sub>3</sub> (Port 2)**

A 4-bit bidirectional port which contains the I/O port address and instruction code on a high to low transition of PROG. During a low to high transition of PROG, port 2 contains either the data for a selected output port if a write operation, or the data from a selected output port (before a low to high transition) if a read operation. Data on port 2 may be directly written, read, ANDed, or ORed with previous data.

**P4<sub>0</sub>-P4<sub>3</sub> (Port 4)**

A 4-bit I/O port. May be programmed for input (during read), low impedance latched output (after write), or high impedance (after read).

**P5<sub>0</sub>-P5<sub>3</sub> (Port 5)**

A 4-bit I/O port. May be programmed for input (during read), low impedance latched output (after write), or high impedance (after read).

**P6<sub>0</sub>-P6<sub>3</sub> (Port 6)**

A 4-bit I/O port. May be programmed for input (during read), low impedance latched output (after write), or high impedance (after read).

**P7<sub>0</sub>-P7<sub>3</sub> (Port 7)**

A 4-bit I/O port. May be programmed for input (during read), low impedance latched output (after write), or high impedance (after read).

**CS (Chip Select)**

Chip select input. A high on CS inhibits any change of output or internal status.

**PROG (Clock Input)**

A high to low transition on PROG indicates that the op-code and the addressed port information are available on port 2. A low to high transition indicates that data is available on port 2.

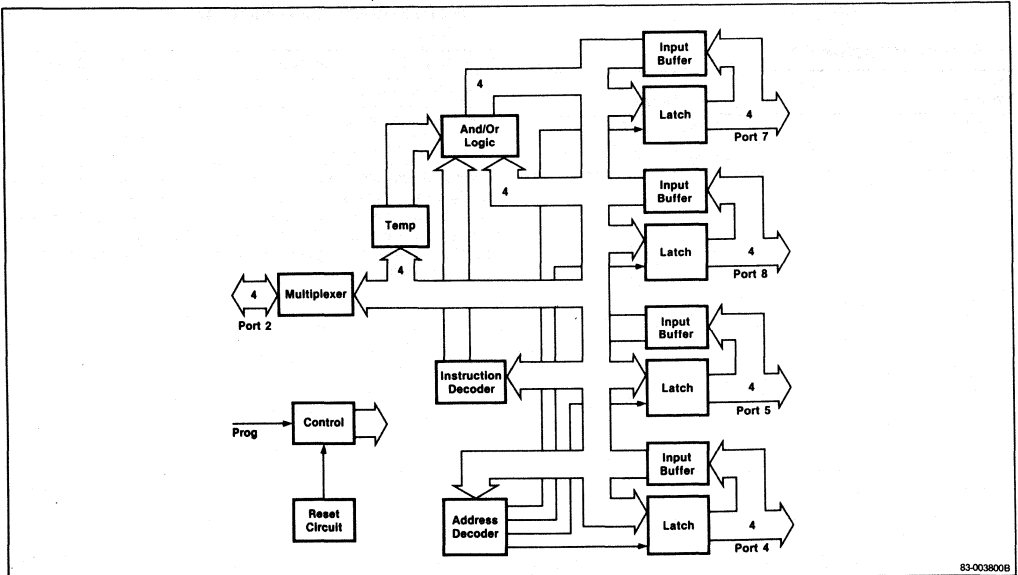
**Ground**

Ground.

**V<sub>DD</sub> (Power Supply)**

+5 V power supply input.

**Block Diagram**



83-0038008

## DC Characteristics

T<sub>A</sub> = 0°C to +70°C; V<sub>DD</sub> = +5 V ± 10%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage high	V <sub>IH</sub>	2		V <sub>DD</sub> +0.5	V	
Input voltage low	V <sub>IL</sub>	-0.5		+0.8	V	
Output voltage high (port 4-7)	V <sub>OH1</sub>	2.4			V	I <sub>OH</sub> = -240 μA
Output voltage high (port 2)	V <sub>OH2</sub>	2.4			V	I <sub>OH</sub> = -100 μA
Output voltage low (port 4-7)	V <sub>OL1</sub>		0.45		V	I <sub>OL</sub> = 5 mA, (Note 1)
Output voltage low (port 7)	V <sub>OL2</sub>		1		V	I <sub>OL</sub> = 20 mA
Output voltage low (port 2)	V <sub>OL3</sub>		0.45		V	I <sub>OL</sub> = 0.6 mA
Sum of all I <sub>OL</sub> from 16 outputs (Note 1)	I <sub>OL</sub>		100		mA	(8243) 5 mA each pin
			80		mA	(8243H) 5 mA each pin
Input leakage current (port 4-7)	I <sub>IL1</sub>	-10		20	μA	V <sub>IN</sub> = V <sub>DD</sub> to 0 V
Input leakage current (port 2, CS, PROG)	I <sub>IL2</sub>	-10		10	μA	V <sub>IN</sub> = V <sub>DD</sub> to 0 V
V <sub>DD</sub> supply current	I <sub>DD</sub>		10	16	mA	

### Note:

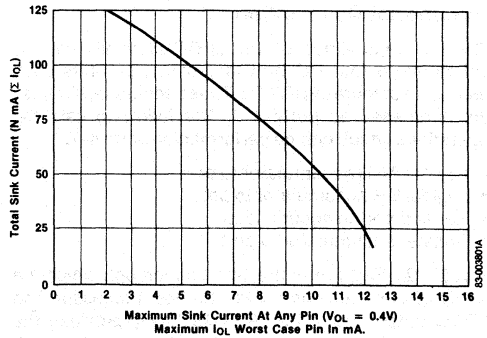
(1) Refer to graph of current sinking capability.

## AC Characteristics

T<sub>A</sub> = 0°C to +70°C; V<sub>DD</sub> = +5 V ± 10%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Code valid before PROG	t <sub>A</sub>	100			ns	80 pF load
Code valid after PROG	t <sub>B</sub>	60			ns	20 pF load
Data valid before PROG	t <sub>C</sub>	200			ns	80 pF load
Data valid after PROG	t <sub>D</sub>	20			ns	20 pF load
Port 2 floating after PROG	t <sub>H</sub>	0		150	ns	20 pF load
PROG negative pulse width	t <sub>K</sub>	700			ns	
Ports 4-7 valid after PROG	t <sub>P0</sub>			700	ns	100 pF load
Ports 4-7 valid before / after PROG	t <sub>IP</sub>	100			ns	
Port 2 valid after PROG	t <sub>ACC</sub>			650	ns	80 pF load
CS valid before / after PROG	t <sub>CS</sub>	50			ns	

## Current Sinking Capability

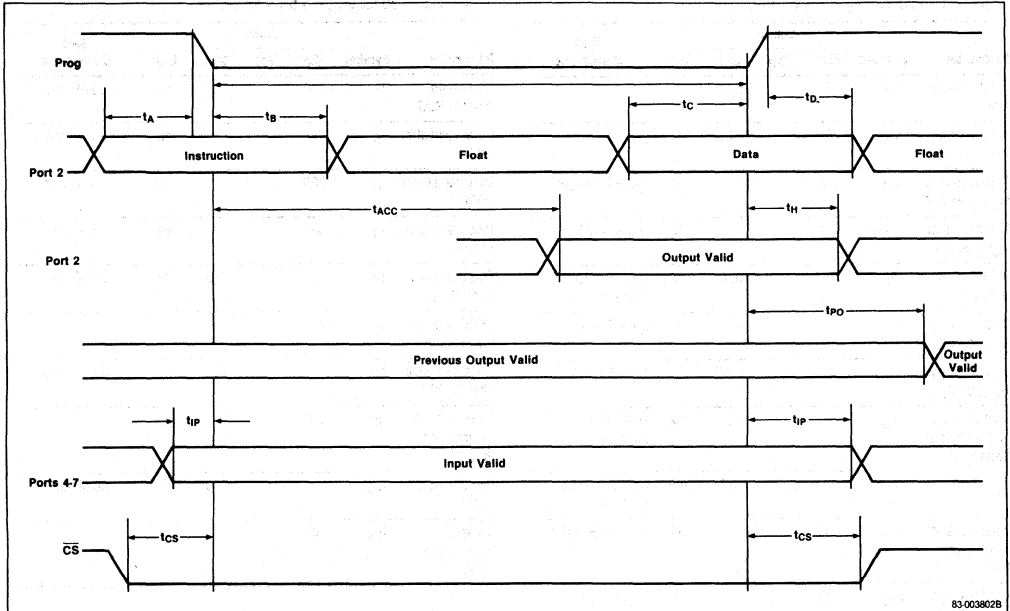


### Note:

This curve plots the guaranteed worst case current sinking capability of any I/O port line versus the total sink current of all pins. The μPD8243 is capable of sinking 5 mA (for V<sub>OL</sub> = 0.4 V) through each of the 16 I/O lines simultaneously. The current sinking curve shows how the individual I/O line drive increases if all the I/O lines are not fully loaded.



Timing Waveform



83-003802B

Functional Description

The I/O capabilities of the μPD8048 family can be enhanced in four I/O port increments of 4-bits each using one or more μPD8243's. These additional I/O lines are addressed as ports 4-7. The following lists the operations which can be performed on ports 4-7.

- Logical AND accumulator to port
- Logical OR accumulator to port
- Transfer port to accumulator
- Transfer accumulator to port

Port 2 (P2<sub>0</sub>-P2<sub>3</sub>) forms the 4-bit bus through which the μPD8243 communicates with the host processor. The PROG output from the μPD8048 family provides the necessary timing to the μPD8243. There are two 4-bit nibbles involved in each data transfer. The first nibble contains the opcode and port address followed by the second nibble containing the 4-bit data. Multiple μPD8243's can be used for additional I/O. The output lines from the μPD8048 family can be used to form the chip selects for additional μPD8243's.

Power On Initialization

Applying power to the μPD8243 sets ports 4-7 to the high impedance mode and port 2 to the input mode. The state of the PROG pin at power on may be either high or low. The PROG pin must make a high to low transition in order to exit from the power on mode. The power on sequence is initiated any time V<sub>DD</sub> drops below 1V. Table 1 following shows how the first 4-bit nibble of a data transfer instruction is decoded.

Table 1. Port 2 Instruction Decoding

P2 <sub>3</sub>	P2 <sub>2</sub>	Instruction Code	P2 <sub>1</sub>	P2 <sub>0</sub>	Address Code
0	0	Read	0	0	Port 4
0	1	Write	0	1	Port 5
1	0	ORLD	1	0	Port 6
1	1	ANLD	1	1	Port 7

For example, an 0010 appearing on P2<sub>3</sub>-P2<sub>0</sub>, respectively would result in a read of port 6.

### Read Mode

There is one read mode in the  $\mu$ PD8243. A falling edge on the PROG pin latches the opcode and port address from input port 2. The port address and read operation are then decoded causing the appropriate outputs to be high impedance and the input buffers switched on. The rising edge of PROG terminates the read operation. The port (4, 5, 6, or 7) that was selected by the port address (P2<sub>1</sub>-P2<sub>0</sub>) is returned to the high impedance mode, and port 2 is switched to the input mode.

Generally, in the read mode a port will be an input and in the write mode it will be an output. If during program operation, the  $\mu$ PD8243's modes are changed, the first read pulse immediately following a write should be ignored. The subsequent read signals are valid. Reading a port will then force that port to a high impedance state.

### Write Modes

There are three write modes in the  $\mu$ PD8243. The MOVD P<sub>p</sub>, A instruction from the  $\mu$ PD8048 family writes the new data directly to the specified port (4, 5, 6, or 7). The old data previously latched at that port is lost. The ORLD P<sub>p</sub>, A instruction performs a logical OR between the new data and the data currently latched at the selected port. The result is then latched at that port. The final write mode uses the ANLD P<sub>p</sub>, A instruction. It performs a logical AND between the new data and the data currently latched at the specified port. The result is latched at that port.

The data remains latched at the selected port following the logical manipulation until new data is written to that port.



### Description

The μPD8251A and μPD8251AF Universal Synchronous/Asynchronous Receiver/Transmitter (USART) are designed for microcomputer systems data communications. The USART is used as a peripheral and is programmed by the 8085A or other processor to communicate in commonly used serial data transmission techniques including IBM Bi-Sync. The USART receives serial data streams and converts them into parallel data characters for the processor. While receiving serial data, the USART will also accept data characters from the processor in parallel format, convert them to serial format, and transmit. The USART will signal the processor when it has completely received or transmitted a character and requires service. Complete USART status including data format errors and control signals such as TxE and SYNDET, is available to the processor at any time.

### Features

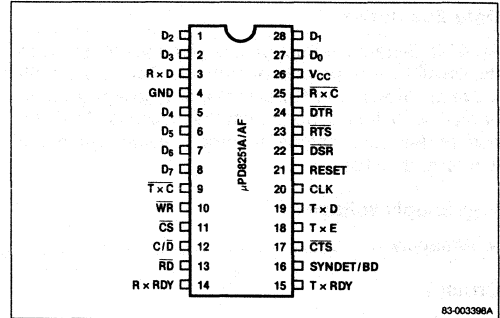
- Asynchronous or synchronous operation
  - Asynchronous:
    - Five 8-bit characters
    - Clock rate — 1, 16, or 64 x baud rate
    - Break character generation
    - Select 1, 1½, or 2 stop bits
    - False start bit detector
    - Automatic break detect and handling
  - Synchronous:
    - Five 8-bit characters
    - Internal or external character synchronization
    - Automatic sync insertion
    - Single or double sync characters
- Baud rate (1x mode) — DC to 64K baud
- Full-duplex, double buffered transmitter and receiver
- Parity, overrun and framing flags
- Fully compatible with 8085A/μPD780 (Z80®), etc.
- All inputs and outputs are TTL-compatible
- Single +5V supply, ±10%
- Separate device receive and transmit TTL clocks
- NMOS technology

### Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD8251AC	28-Pin plastic DIP	3/5 MHz
μPD8251AFC	28-Pin plastic DIP	3/5 MHz

Z80 is a registered trademark of Zilog, Inc.

### Pin Configuration



### Pin Identification

No.	Symbol	Function
1, 2, 27, 28, 5-8	D <sub>7</sub> -D <sub>0</sub>	Data bus buffer
26	V <sub>DD</sub>	V <sub>DD</sub> supply voltage
4	GND	Ground
21	RESET	Reset
20	CLK	Clock pulse
10	WR	Write data
13	RD	Read data
12	C / D	Control / data
11	CS	Chip select
22	DSR	Data set ready
24	DTR	Data terminal ready
23	RTS	Request to send
17	CTS	Clear to send
15	TxRDY	Transmitter ready
18	TxE	Transmitter empty
9	TxC	Transmitter clock
19	TxD	Transmitter data
14	RxRDY	Receiver ready
25	RxC	Receiver clock
3	RxD	Receiver data
16	SYNDET / BD	Sync detect / break detect

## Pin Functions

### Data Bus Buffer

An 8-bit, 3-state bi-directional buffer used to interface the USART to the processor data bus. Data is transmitted or received by the buffer in response to input/output or read/write instructions from the processor. The data bus buffer also transfers control words, command words, and status.

### V<sub>DD</sub> Supply Voltage

+5V supply

### Ground

Ground

### Read/Write Control Logic

This logic block accepts inputs from the processor control bus and generates control signals for overall USART operation. The mode instruction and command instruction registers that store the control formats for device functional definition are located in the read/write control logic.

### Reset

A "one" on this input forces the USART into the "Idle" mode where it will remain until reinitialized with a new set of control words. Minimum RESET pulse width is 6t<sub>cy</sub>.

### Clock Pulse

The CLK input provides for internal device timing and is usually connected to the phase 2 (TTL) output of the μPB8224 clock generator. External inputs and outputs are not referenced to CLK, but the CLK frequency must be at least 30 times the receiver or transmitter clocks in the synchronous mode and 4.5 times for the asynchronous mode.

### Write Data

A "zero" on this input instructs the USART to accept the data or control word which the processor is writing out on the data bus.

## Read Data

A "zero" on this input instructs the USART to place the data or status information onto the data bus for the processor to read.

## Control/Data

The control/data input, in conjunction with the  $\overline{WR}$  and  $\overline{RD}$  inputs, informs the USART to accept or provide either a data character, control word, or status information via the data bus. 0 = Data; 1 = Control.

## Chip Select

A "zero" on this input enables the USART to read from or write to the processor.

## Modem Control

The μPD8251A/51AF have a set of control inputs and outputs which may be used to simplify the interface to a modem.

## Data Set Ready

The data set ready input can be tested by the processor via status information. The DSR input is normally used to test the modem data set ready condition.

## Data Terminal Ready

The data terminal ready output can be controlled via the command word. The DTR output is normally used to drive modem data terminal ready or rate select lines.

## Request to Send

The request to send output can be controlled via the command word. The RTS output is normally used to drive the modem request to send line.

## Clear to Send

A "zero" on the clear to send input enables the USART to transmit serial data if the TxEN bit in the command instruction register is enabled (one).

## Transmit Control Logic

The transmit control logic accepts and outputs all external and internal signals necessary for serial data transmission.

### Transmitter Ready

Transmitter ready signals the processor that the transmitter is ready to accept a data character. TxRDY can be used as an interrupt or may be tested through the status information for polled operation. Loading a character from the processor automatically resets TxRDY, on the leading edge.

### Transmitter Empty

The transmitter empty output signals the processor that the USART has no further characters to transmit. TxE is automatically reset upon receiving a data character from the processor. In half-duplex, TxE can be used to signal the end of a transmission and request the processor to "turn the line around." The TxEn bit in the command instruction does not effect TxE.

In the synchronous mode, a "one" on this output indicates that a sync character or characters are about to be automatically transmitted as "fillers" because the next data character has not been loaded.

### Transmitter Clock

The transmitter clock controls the serial character transmission rate. In the asynchronous mode, the  $\overline{\text{TxC}}$  frequency is a multiple of the actual baud rate. Two bits of the mode instruction select the multiple to be 1x, 16x, or 64x the baud rate. In the synchronous mode, the  $\overline{\text{TxC}}$  frequency is automatically selected to equal the actual baud rate.

Note that for both synchronous and asynchronous modes, serial data is shifted out of the USART by the falling edge of  $\overline{\text{TxC}}$ .

### Transmitter Data

The transmit control logic outputs the composite serial data stream on this pin.

### Receiver Control Logic

This block manages all activities related to incoming data.

## Receiver Ready

The receiver ready output indicates that the receiver buffer is ready with an "assembled" character for input to the processor. For polled operation, the processor can check RxRDY using a status read or RxRDY can be connected to the processor interrupt structure. Note that reading the character to the processor automatically resets RxRDY.

## Receiver Clock

The receiver clock determines the rate at which the incoming character is received. In the asynchronous mode, the  $\overline{\text{RxC}}$  frequency may be 1, 16 or 64 times the actual baud rate, but in the synchronous mode the  $\overline{\text{RxC}}$  frequency must equal the baud rate. Two bits in the mode instruction select asynchronous at 1x, 16x, or 64x or synchronous operation at 1x the baud rate.

Unlike  $\overline{\text{TxC}}$ , data is sampled by the μPD8251A/51AF on the rising edge of  $\overline{\text{RxC}}$ . (Note 1)

## Receiver Data

A composite serial data stream is received by the receiver control logic on this pin.

## Sync Detect/Break Detect

The μPD8251A/51AF may be programmed through the mode instruction to operate in either the internal or external sync mode; the SYNDET/BD pin then functions as an output or input respectively. In the internal sync mode, the SYNDET output will go to "one" when the μPD8251A/51AF has located the SYNC character in the receive mode. If double SYNC character (bi-sync) operation has been programmed, SYNDET will go to "one" in the middle of the last bit of the second SYNC character. SYNDET is automatically reset to "zero" upon a status read or RESET. In the external SYNC mode, a "zero" to "one" transition on the SYNDET input will cause the μPD8251A/51AF to start assembling data character on the next falling edge of  $\overline{\text{RxC}}$ . The length of the SYNDET input should be at least one  $\overline{\text{RxC}}$  period, but may be removed once the μPD8251A/51AF is in SYNC.

In the asynchronous mode, the SYNDET/BD pin functions as a break detect. The BD output will go high when a word of all zeros is received. This word consists of: start bit, data bits, parity bit, and one stop bit. Reset only occurs when Rx data returns to a logic one state or upon chip reset. The state of break detect can be read as a status bit.

**Note:**

(1) Since the μPD8251A/51AF will frequently be handling both the reception and transmission for a given link, the receive and transmit baud rates will be the same. Rx̄C and Tx̄C then require the same frequency and may be tied together and connected to a single clock source or baud rate generator.

**Examples:**

If the baud rate equals 110 (Async):

Rx̄C or Tx̄C equals 110 Hz (1x)

Rx̄C or Tx̄C equals 1.76 kHz (16x)

Rx̄C or Tx̄C equals 7.04 kHz (64x)

If the baud rate equals 300:

Rx̄C or Tx̄C equals 300 Hz (1x) A or S

Rx̄C or Tx̄C equals 4800 Hz (16x) A only

Rx̄C or Tx̄C equals 19.2 kHz (64x) A only

**Absolute Maximum Ratings**

T<sub>A</sub> = 25°C

Power supply voltage, V <sub>DD</sub>	-0.5 V to +7 V
Input voltage, V <sub>I</sub>	-0.5 V to +7 V
Output voltage, V <sub>O</sub>	-0.5 V to +7 V
Operating temperature, T <sub>OP</sub>	-0°C to +70°C
Storage temperature, T <sub>STG</sub>	-65°C to +150°C

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Capacitance**

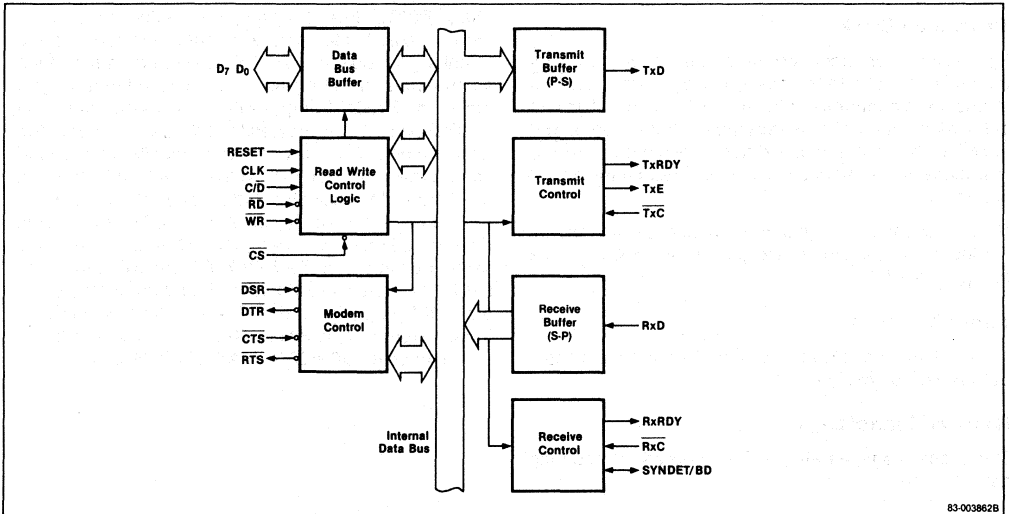
T<sub>A</sub> = 25°C, V<sub>CC</sub> = GND = 0 V, f<sub>C</sub> = 1.0 MHz

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	C <sub>IN</sub>			10	pF	(Note 1)
I/O capacitance	C <sub>IO</sub>			20	pF	(Note 1)

**Note:**

(1) All unmeasured pins returned to GND.

**Block Diagram**



83-003862B

## DC Characteristics

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = +5V ± 10%, GND = 0V

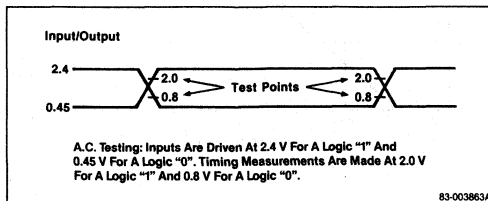
Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage low	V <sub>IL</sub>	-0.5		+0.8	V	
Input voltage high	V <sub>IH</sub>	2		V <sub>CC</sub>	V	
Output voltage low	V <sub>OL</sub>			+0.45	V	μPD8251: I <sub>OL</sub> = 1.7 mA μPD8251A: I <sub>OL</sub> = 2.2 mA
Output voltage high	V <sub>OH</sub>	2.4			V	μPD8251: I <sub>OH</sub> = -100 μA μPD8251A: I <sub>OH</sub> = -400 mA
Output float leakage current	I <sub>OFFL</sub>			±10	μA	V <sub>OUT</sub> = 0.45 V
				10	μA	0.45 V ≤ V <sub>OUT</sub> ≤ V <sub>CC</sub>
Input load current	I <sub>IL</sub>			10	μA	0.45 V ≤ V <sub>IN</sub> ≤ V <sub>CC</sub>
Power supply current	I <sub>CC</sub>			100	mA	All outputs = logic 1

## AC Characteristics

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5V ± 10%, GND = 0V

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8251A		μPD8251AF			
		Min	Max	Min	Max		
<b>Read</b>							
Address stable before RD, (CS, CD)	t <sub>AR</sub>	50		0		ns	(Note 7)
Address hold time for RD, (CS, CD)	t <sub>RA</sub>	50		0		ns	(Note 7)
RD pulse width	t <sub>RR</sub>	250		200		ns	
Data delay from RD	t <sub>RD</sub>		250		140	ns	μPD8251A; C <sub>L</sub> = 150 pF, (Note 8)
RD to data floating	t <sub>DF</sub>	10	100	10	80	ns	

## Testing Input, Output Waveform



83-00863A

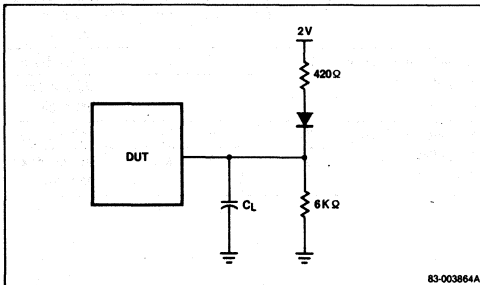
Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8251A		μPD8251AF			
		Min	Max	Min	Max		
<b>Writes</b>							
Address stable before WR	t <sub>AW</sub>	50		0		ns	
Address hold time for WR	t <sub>WA</sub>	50		0		ns	
WR pulse width	t <sub>WW</sub>	250		200		ns	
Data set-up time for WR	t <sub>DW</sub>	150		100		ns	
Data hold time for WR	t <sub>WD</sub>	30		0		ns	
Recovery time between WR's	t <sub>RV</sub>	6		6		t <sub>CY</sub> (Note 2)	
<b>Other Timing</b>							
Clock period	t <sub>CY</sub>	0.32	1.35	0.20	1.35	μs	(Note 3)
Clock pulse width high	t <sub>φW</sub>	140	t <sub>CY</sub> - 90	70	t <sub>CY</sub> - 40	ns	
Clock pulse width low	t <sub>φL</sub>	90		40		ns	
Clock rise and fall time	t <sub>R</sub> , t <sub>F</sub>	5	20	5	20	ns	
TxD delay from falling edge of TxC	t <sub>DTx</sub>		1		1	μs	
Rx data set-up time to sampling pulse	t <sub>SRx</sub>	2				μs	
Rx data hold time to sampling pulse	t <sub>HRx</sub>	2				μs	
Transmitter input clock frequency	f <sub>Tx</sub>		64	DC	64	kHz	1x baud rate
			310	DC	310	kHz	16x baud rate
			615	DC	615	kHz	64x baud rate
Transmitter input clock pulse width	t <sub>TPW</sub>	12		12	t <sub>CY</sub>		1x baud rate
		1		1	t <sub>CY</sub>		16x and 64x baud rate
Transmitter input clock pulse delay	t <sub>TPD</sub>	15		15	t <sub>CY</sub>		1x baud rate
		3		3	t <sub>CY</sub>		16x and 64x baud rate



**AC Characteristics (cont)**

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8251A		μPD8251AF			
		Min	Max	Min	Max		
<b>Other Timing (cont)</b>							
Receiver input clock frequency	t <sub>RX</sub>	64	DC	64	kHz	1x baud rate	
		310	DC	310	kHz	16x baud rate	
		615	DC	615	kHz	64x baud rate	
Receiver input clock pulse width	t <sub>RPW</sub>	12		12	t <sub>CY</sub>	1x baud rate	
		1		1	t <sub>CY</sub>	16x and 64x baud rate	
Receiver input clock pulse delay	t <sub>RPD</sub>	15		15	t <sub>CY</sub>	1x baud rate	
		3		3	t <sub>CY</sub>	16x and 64x baud rate	
TxRDY delay from center of data bit	t <sub>TX</sub>		8		t <sub>CY</sub>	(Note 9)	
TxRDY ↓ from leading edge of WR	t <sub>TXRDY CLEAR</sub>			300	ns	(Note 9)	
RxRDY delay from center of data bit	t <sub>RX</sub>		24		20	t <sub>CY</sub>	
Internal SYNDET delay from center of data bit	t <sub>IS</sub>		24			t <sub>CY</sub> (Note 9)	
RxRDY ↓ from leading edge of RD	t <sub>RXRDY CLEAR</sub>			300	ns	(Note 9)	
External SYNDET set-up time before falling edge of RxC	t <sub>ES</sub>	16		18		t <sub>CY</sub> (Note 9)	

**Test Load Circuit**

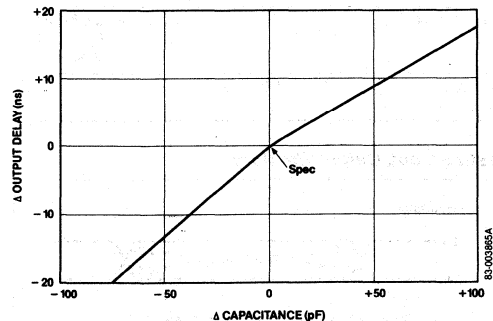


Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8251A		μPD8251AF			
		Min	Max	Min	Max		
<b>Other Timing (cont)</b>							
TxEMPTY delay from center of data bit	t <sub>TXE</sub>		20		20	t <sub>CY</sub>	(Note 9)
Control delay from rising edge of WR (TxE, DTR, RTS)	t <sub>WC</sub>		8		8	t <sub>CY</sub>	(Note 9)
Control to RD set-up time (DSR, CTS)	t <sub>CR</sub>	20		20		t <sub>CY</sub>	(Note 9)

**Note:**

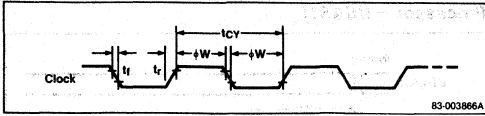
- AC timing measured at V<sub>OH</sub> = 2.0 V, V<sub>OL</sub> = 0.8 V, and with test load circuit below.
- This recovery time is for initialization only, when MODE, SYNC1, SYNC2, COMMAND and first DATA BYTES are written into the USART. Subsequent writing of both COMMAND and DATA are only allowed when TxRDY = 1.
- The TxC and RxC frequencies have the following limitations with respect to CLK:  
For 1x baud rate, f<sub>TX</sub> or f<sub>RX</sub> ≤ 1(30t<sub>CY</sub>)  
For 16x and 64x baud rate, f<sub>TX</sub> or f<sub>RX</sub> ≤ 1(4.5t<sub>CY</sub>)
- Reset pulse width = 6 t<sub>CY</sub> minimum.
- t<sub>TXRDYCCR</sub> - 2t<sub>CY</sub> + t<sub>φ</sub> + t<sub>R</sub> + 200 ns
- t<sub>RXRDYCCR</sub> - 2t<sub>CY</sub> + t<sub>φ</sub> + t<sub>R</sub> + 170 ns
- Chip Select (CS) and Command/Data (C/D) are considered as addresses.
- Assumes that address is valid before R<sub>O</sub> ↓.
- Status update can have a maximum delay of 28 clock periods from the event affecting the status.

**Typical Δ Output Delay Versus Δ Capacitance**

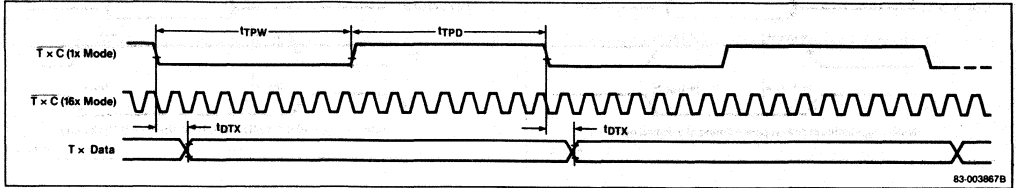


## Timing Waveforms

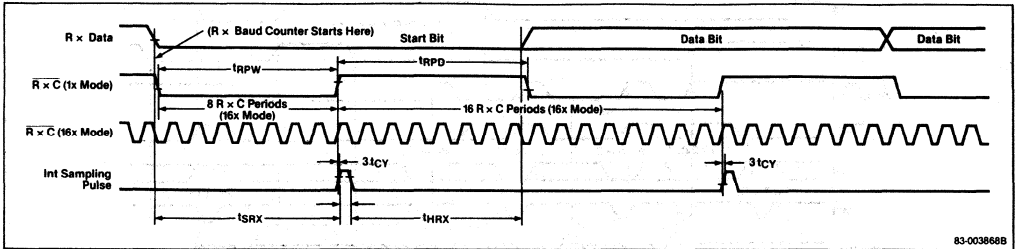
### System Clock Input



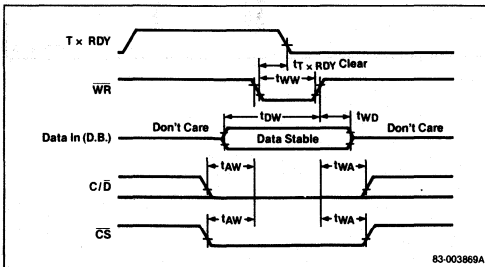
### Transmitter Clock and Data



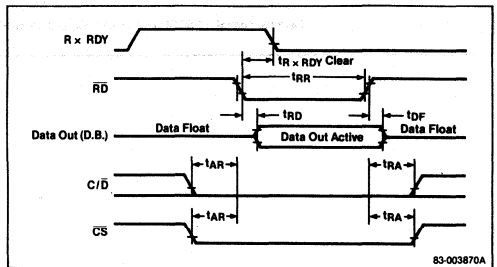
### Receiver Clock and Data



### Write Data Cycle (Processor → USART)

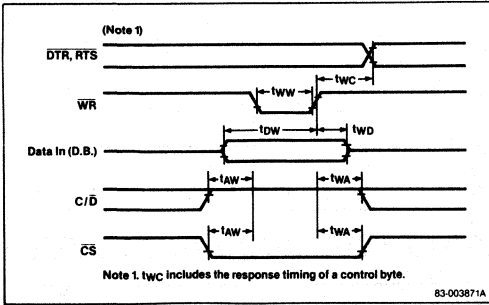


### Read Data Cycle (Processor ← USART)

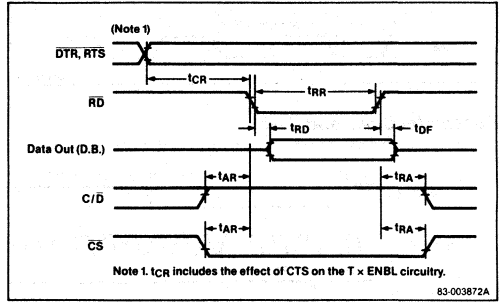


**Timing Waveforms (cont)**

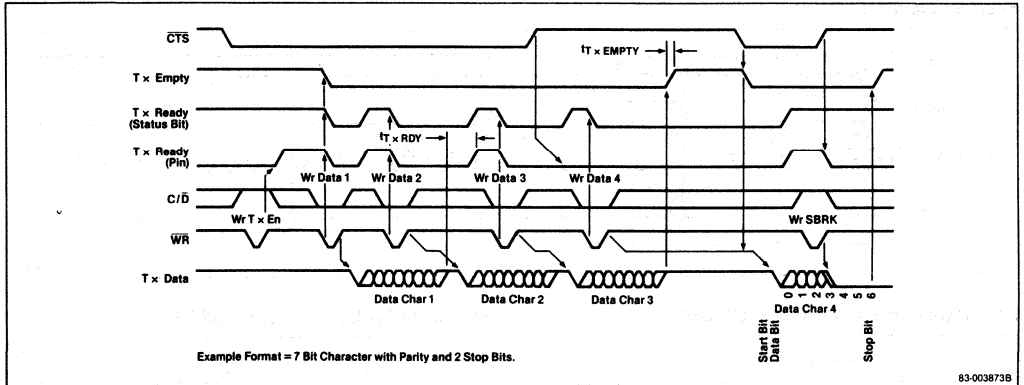
**Write Control or Output Port Cycle  
(Processor → USART)**



**Read Control or Input Port Cycle  
(Processor ← USART)**

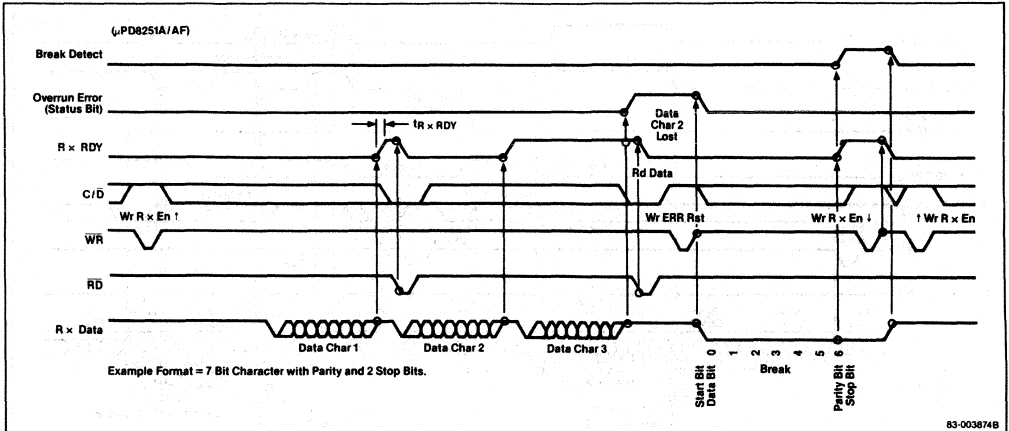


**Transmitter Control and Flag Timing (Async Mode)**

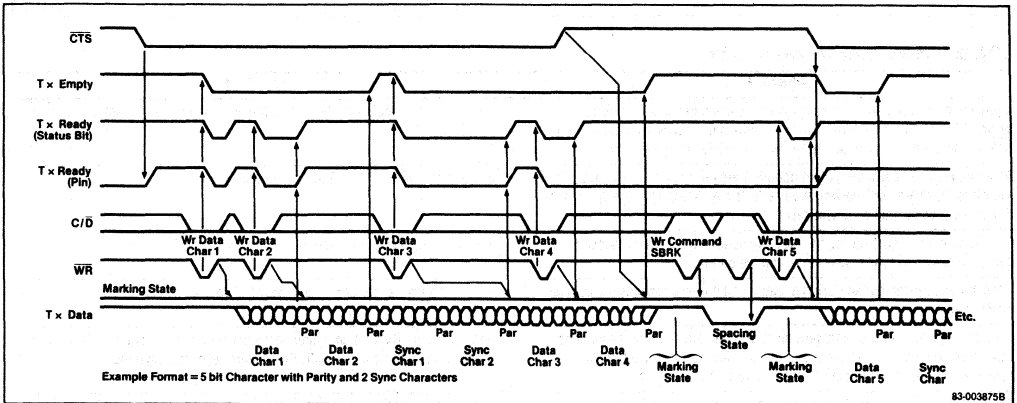


## Timing Waveforms (cont)

### Receiver Control and Flag Timing (Async Mode)



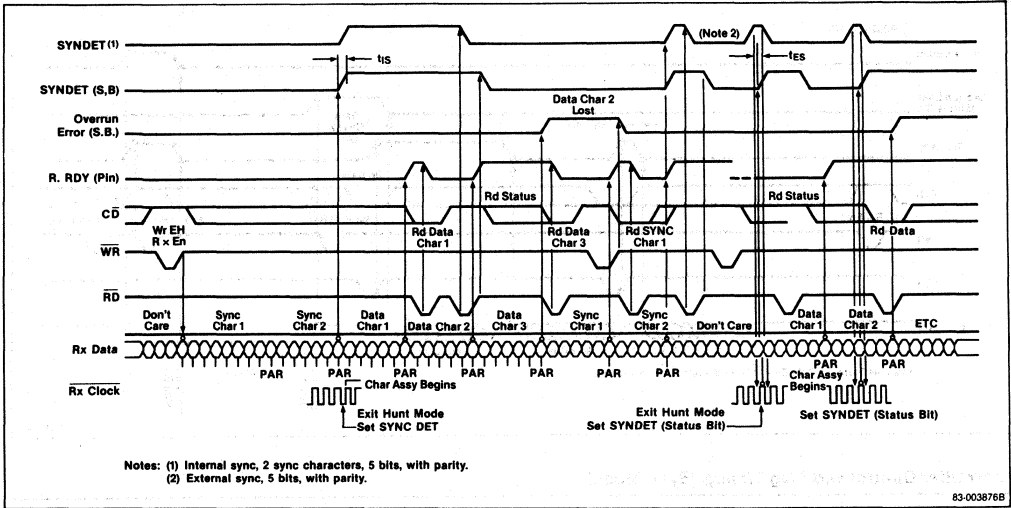
### Transmitter Control and Flag Timing (Sync Mode)



7

Timing Waveforms (cont)

Receiver Control and Flag Timing (Sync Mode)



83-003876B

μPD8251AF Enhancements

μPD8251A	μPD8251AF
A previously loaded data character will be retransmitted if Tx was disabled before TxEMPTY by TxEnable ↓ or CTS ↑, and is re-enabled by TxEnable ↑ or CTS ↓ before a new data character is sent to μPD8251A by the CPU.	A previously loaded character will be flushed out and not transmitted on CTS ↓ or TxEnable ↑.
Break detect does not always reset upon RxData returning to a '1' during the last bit of the character following the break. Break detect will latch up, and the device must be cleared by device reset.	Break detect will reset on RxData going to '1'.
On TxEnable ↓ or CTS ↑ during the first character of a double-character sync output, the second sync character will not be output.	Will output both sync characters on TxEnable ↓ or CTS ↑.
If the status register is read during a status update, an erroneous status read may result.	Some valid status (either new or old) will always be available.
In Rx mode, a hardware or software reset does not force asynchronous mode, clear hunt condition, or require a proper line initialization (1 to 0 transition) before receiving. This may cause reception of garbage characters.	Reset will clear Rx hunt condition, force asynchronous operation (64x clock), and require a proper line initialization before receiving anything.
Break detect will occur on the first complete (start bit to stop bit) break. This situation could be confused with a null frame (all zeros) that also has a framing error.	Will give a framing error at the end of the first complete or partial break and will give a break detect at the stop bit position of the second contiguous break character.
Sync detect does not reset on status read.	Sync detect will reset on status read.
RxRDY clears within 2 t <sub>cy</sub> 's of RD leading edge.	RxRDY will clear on RD leading edge.
TxEMPTY oscillates with internal clock when TxEnable ↓ or CTS ↑.	TxEMPTY will not oscillate this way.
TxRDY and TxEMPTY clear on WR trailing edge (data).	TxRDY, TxEMPTY will clear on WR leading edge.
Enter hunt command affects asynchronous Rx by loss of data characters.	Enter hunt will not affect asynchronous operation.
Writing a command will sometimes clear TxRDY or TxEMPTY if C/D set up or hold is marginal. Reading status will sometimes clear RxRDY if C/D set up or hold is marginal.	C/D set up and hold margin will be improved.

### μPD8251AF Enhancements (cont)

μPD8251A	μPD8251AF
Rx data overrun error will not occur and garbage data may result if $\overline{RD}$ and $\overline{CS}$ are active during an internal data update.	Will indicate an overrun error properly.
In asynchronous mode, after a reset, the first TxD bit may be shifted out on either the first or second $\overline{TxC}$ $\downarrow$ edge.	The first TxD bit will be shifted out on the first $\overline{TxC}$ $\downarrow$ edge.
RxRDY can glitch when CLK does not have a fixed phase relationship to $\overline{RxC}$ .	RxRDY will not glitch.
The receiver occasionally gives an extra character following the end of break condition.	No extra characters will occur.

### Functional Description

The μPD8251A/51AF Universal Synchronous/Asynchronous Receiver/Transmitters are designed specifically for 8085A microcomputer systems but work with most 8-bit processors. Operations of the μPD8251A/51AF, like other I/O devices in the 8085A family, are programmed by system software for maximum flexibility.

In the receive mode, the μPD8251A/51AF converts incoming serial format data into parallel data and makes certain format checks. In the transmit mode, it formats parallel data into serial form. The device also supplies or removes characters or bits that are unique to the communication format in use. By performing conversion and formatting services automatically, the USART appears to the processor as a simple or "transparent" input or output of byte-oriented parallel data.

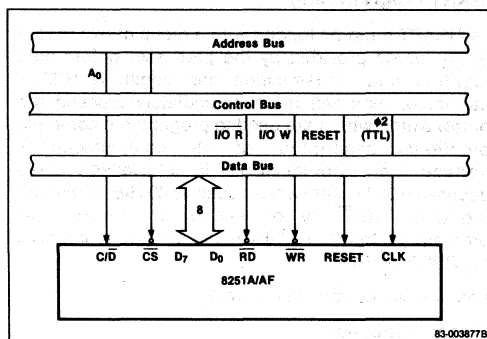
#### Truth Table

$\overline{C/D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	MODE
0	0	1	0	μPD8251A / 51AF → Data bus
0	1	0	0	Data bus → μPD8251A / 51AF
1	0	1	0	Status → Data bus
1	1	0	0	Data bus → Control
X	X	X	1	Data bus → 3-state
X	1	1	0	Data bus → 3-state

#### Transmit Buffer

The transmit buffer receives parallel data from the data bus buffer via the internal data bus, converts parallel to serial data, inserts the necessary characters or bits needed for the programmed communication format and outputs composite serial data on the TxD pin.

### μPD8251A/51AF Interface to 8085A Standard System Bus



#### Receive Buffer

The receive buffer accepts serial data input at the RxD pin and converts the data from serial to parallel format. Bits or characters required for the specific communication technique in use are checked and then an eight-bit "assembled" character is readied for the processor. For communication techniques which require fewer than eight bits, the μPD8251A/51AF set the extra bits to "zero".

#### Operation

A set of control words must be sent to the μPD8251A/51AF to define the desired mode and communications format. The control words will specify the baud rate factor (1x, 16x, 64x), character length (5 to 8), number of STOP bits (1, 1½, 2) asynchronous or synchronous mode, SYNDET (IN or OUT), parity, etc.

After receiving the control words, the μPD8251A/51AF are ready to communicate. TxRDY is raised to signal the processor that the USART is ready to receive a character for transmission. When the processor writes a character to the USART, TxRDY is automatically reset.

Concurrently, the μPD8251A/51AF may receive serial data; and after receiving an entire character, the RxRDY

output is raised to indicate a completed character is ready for the processor. The processor fetch will automatically reset RxRDY.

**Note:**

The μPD8251A/51AF may provide faulty RxRDY for the first read after power-on or for the first read after receive is re-enabled by a command instruction (RxE). A dummy read is recommended to clear faulty RxRDY. But this is not the case for the first read after hardware or software reset after the device operation has once been established.

The μPD8251A/51AF cannot transmit until the TxEN (transmitter enable) bit has been set by a command instruction and until the CTS (clear to send) input is a "zero". TxD is held in the "marking" state after reset awaiting new control words.

**USART Programming**

The USART must be loaded with a group of two to four control words provided by the processor before data reception and transmission can begin. A RESET (internal or external) must immediately proceed the control words which are used to program the complete operational description of the communications interface. If an external RESET is not available, three successive 00 Hex or two successive 80 Hex command instructions (C/D = 1) followed by a software reset command instruction (40 Hex) can be used to initialize the μPD8251A/51AF.

There are two control word formats:

1. Mode Instruction
2. Command Instruction

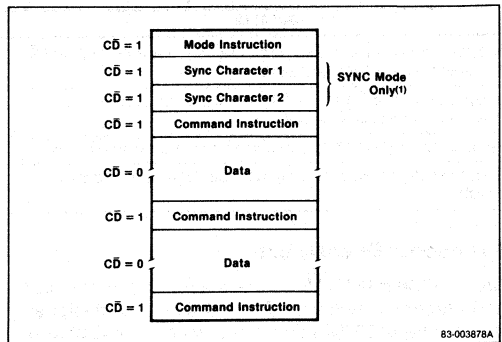
**Mode Instruction**

This control word specifies the general characteristics of the interface regarding the synchronous or asynchronous mode, baud rate factor, character length, parity, and number of stop bits. Once the mode instruction has been received, SYNC characters or command instructions may be inserted depending on the mode instruction content.

**Command Instruction**

This control word will be interpreted as a SYNC character definition if immediately preceded by a mode instruction which specified a synchronous format. After the SYNC character(s) are specified or after an asynchronous mode instruction, all subsequent control words will be interpreted as an update to the command instruction. Command instruction updates may occur at any time during the data block. To modify the mode instruction, a bit may be set in the command instruction which causes an internal reset which allows a new mode instruction to be accepted.

**Typical Data Block**



**Mode Instruction Definition**

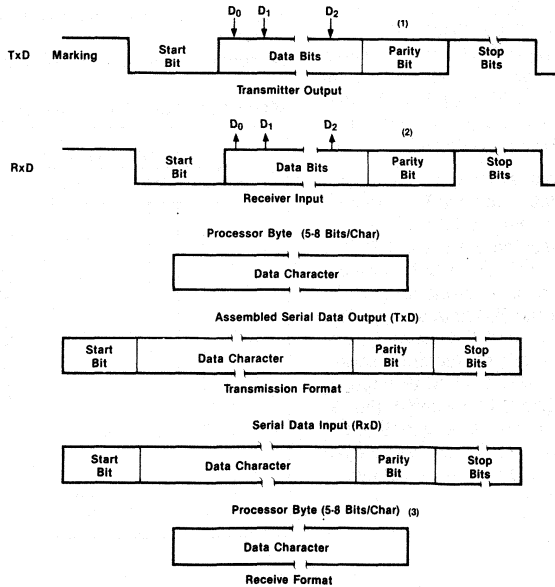
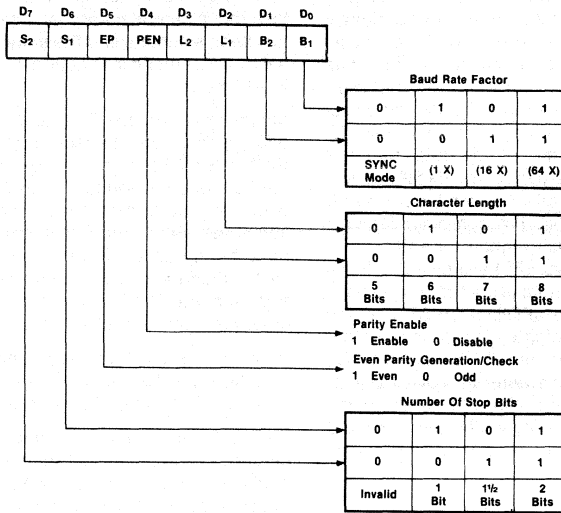
The μPD8251A/51AF can operate in either asynchronous or synchronous communication modes. Understanding how the mode instruction controls the functional operation of the USART is easiest when the device is considered to be two separate components (one asynchronous and the other synchronous) which share the same support circuits and package. Although the format definition can be changed at will or "on the fly," the two modes will be explained separately for clarity.

**Asynchronous Transmission**

When a data character is written into μPD8251A/51AF, the USART automatically adds a start bit (low level or "space") and the number of stop bits (high level or "mark") specified by the mode instruction. If parity has been enabled, an odd or even parity bit is inserted just before the stop bit(s), as specified by the mode instruction. Then, depending on CTS and TxEN, the character may be transmitted as a serial data stream at the TxD output. Data is shifted out by the falling edge of Tx̄C at Tx̄C, Tx̄C/16 or Tx̄C/64, as defined by the mode instruction.

If no data characters have been loaded into the μPD8251A/51AF, or if all available characters have been transmitted, the TxD output remains "high" (marking) in preparation for sending the start bit of the next character provided by the processor. TxD may be forced to send a break (continuously low) by setting the correct bit in the command instruction.

## Mode Instruction Format for Asynchronous Mode



**Notes:**

- (1) Generated by μPD8251A/AF
- (2) Does not appear on the Data Bus.
- (3) If character length is defined as 5, 6, or 7 bits, the unused bits are set to "zero."



Asynchronous Receive

The Rx̄D input line is normally held "high" (marking) by the transmitting device. A falling edge at Rx̄D signals the possible beginning of a start bit and a new character. The start bit is checked by testing for a "low" at its nominal center as specified by the baud rate. If a "low" is detected again, it is considered valid, and the bit assembling counter starts counting. The bit counter locates the approximate center of the data, parity (if specified), and stop bits. The parity error flag (PE) is set, if a parity error occurs. Input bits are sampled at the Rx̄D pin with the rising edge of Rx̄C. If a high is not detected for the stop bit, which normally signals the end of an input character, a framing error (FE) will be set. After a valid stop bit, the input character is loaded into the parallel data bus buffer of the μPD8251A/51AF and the Rx̄RDY signal is raised to indicate to the processor that a character is ready to be fetched. If the processor has failed to fetch the previous character, the new character replaces the old and the overrun flag (OE) is set. All the error flags can be reset by setting a bit in the command instruction. Error flag conditions will not stop subsequent USART operation.

Synchronous Transmission

As in asynchronous transmission, the Tx̄D output remains "high" (marking) until the μPD8251A/51AF receive the first character (usually a SYNC character) from the processor. After a command instruction has set Tx̄EN and after clear to send (CTS) goes low, the first character is serially transmitted. Data is shifted out on the falling edge of Tx̄C and the same rate as Tx̄C.

Once transmission has started, synchronous mode format requires that the serial data stream at Tx̄D continue at the Tx̄C rate or SYNC will be lost. If a data character is not provided by the processor before the μPD8251A/51AF transmit buffer becomes empty, the SYNC character(s) loaded directly following the mode instruction will be automatically inserted in the Tx̄D data stream. The SYNC character(s) are inserted to fill the line and maintain synchronization until new data characters are available for transmission. If the μPD8251A/51AF become empty, and must send the SYNC character(s), the Tx̄EMPTY output is raised to signal the processor that the transmitter buffer is empty and SYNC characters are begun transmitted. Tx̄EMPTY is automatically reset by the next character from the processor.

Synchronous Receive

In synchronous receive, character synchronization can be either external or internal. If the internal SYNC mode has been selected, and the enter hunt (EH) bit has been set by a command instruction, the receiver goes into the HUNT mode.

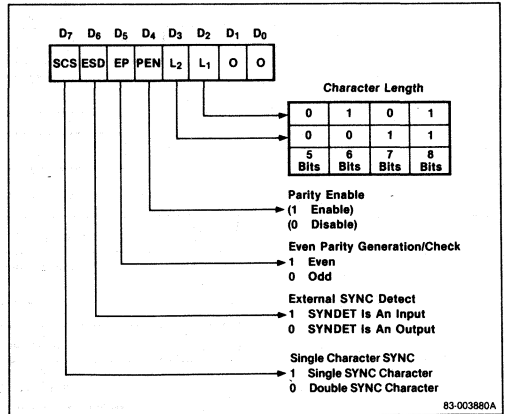
Incoming data on the Rx̄D input is sampled on the rising edge of Rx̄C, and the receive buffer is compared with the first SYNC character after each bit has been loaded until a match is found. If two SYNC characters have been programmed, the next received character is also compared. When the SYNC character(s) programmed have been detected, the μPD8251A/51AF leave the HUNT mode and are in character synchronization. At this time, the SYNDET (output) is set high. SYNDET is automatically reset by a status read.

If external SYNC has been specified in the mode instruction, a "one" applied to the SYNDET (input) for at least one Rx̄C cycle will synchronize the USART.

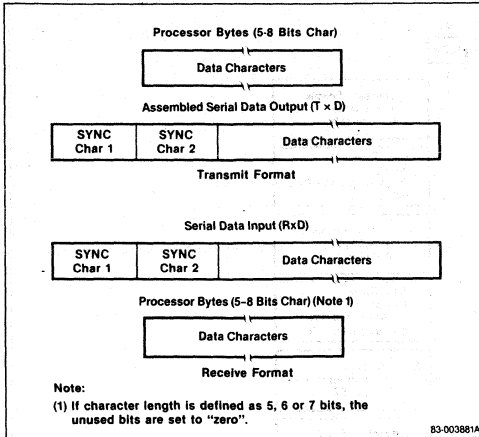
Parity and overrun errors are treated the same in the synchronous as in the asynchronous mode. If not in HUNT, parity will continue to be checked even if the receiver is not enabled. Framing errors do not apply in the synchronous format.

The processor may command the receiver to enter the HUNT mode with a command instruction which sets enter hunt (EH) if synchronization is lost.

Mode Instruction Format for Synchronous Mode



## Transmit/Receive Format Synchronous Mode



## Command Instruction Format

After the functional definition of the μPD8251A/51AF has been specified by the mode instruction and the SYNC character(s) have been entered (if in SYNC mode), the USART is ready to receive command instructions and begin communication. A command instruction is used to control the specific operation of the format selected by the mode instruction. Enable transmit, enable receive, error reset and modem controls are controlled by the command instruction.

After the mode instruction and the SYNC character(s) (as needed) are loaded, all subsequent "control writes" ( $C/\bar{D} = 1$ ) will load or overwrite the command instruction register. A reset operation (internal via CMD IR or external via the RESET input) will cause the μPD8251A/51AF to interpret the next "control write", which must immediately follow the reset, as a mode instruction.

## Status Read Format

It is frequently necessary for the processor to examine the status of an active interface device to determine if errors have occurred or if there are other conditions which require a response from the processor. The μPD8251A/51AF have features which allow the processor to read the device status at any time. A data fetch is issued by the processor while holding the C/D input "high" to obtain device status information. Many of the bits in the status register are copies of external pins. This dual status arrangement allows the μPD8251A/51AF to be used in both polled and interrupt driven environments. Status update can have a maximum delay of 28 clock periods in the μPD8251A/51AF.

## Parity Error

When a parity error is detected, the PE flag is set. It is cleared by setting the ER bit in a subsequent command instruction. PE being set does not inhibit USART operation.

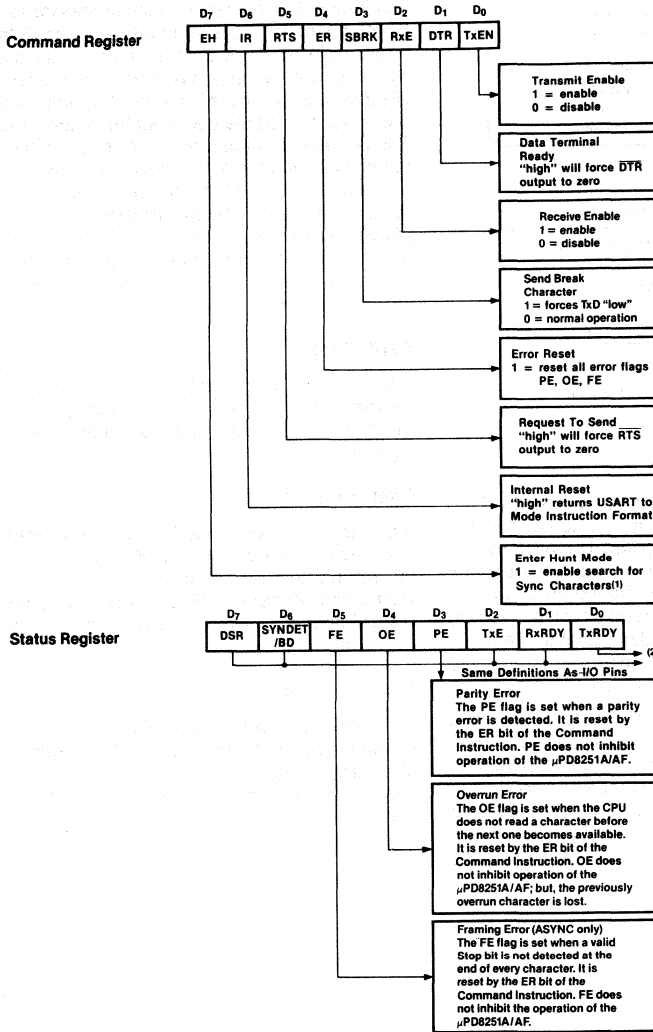
## Overrun Error

If the processor fails to read a data character before the one following is available, the OE flag is set. It is cleared by setting the ER bit in a subsequent command instruction. Although OE being set does not inhibit USART operation, the previously received character is overwritten and lost.

## Framing Error

If a valid STOP bit is not detected at the end of a character, the FE flag is set (ASYNC mode only). It is cleared by setting the ER bit in a subsequent command instruction. FE being set does not inhibit USART operation.

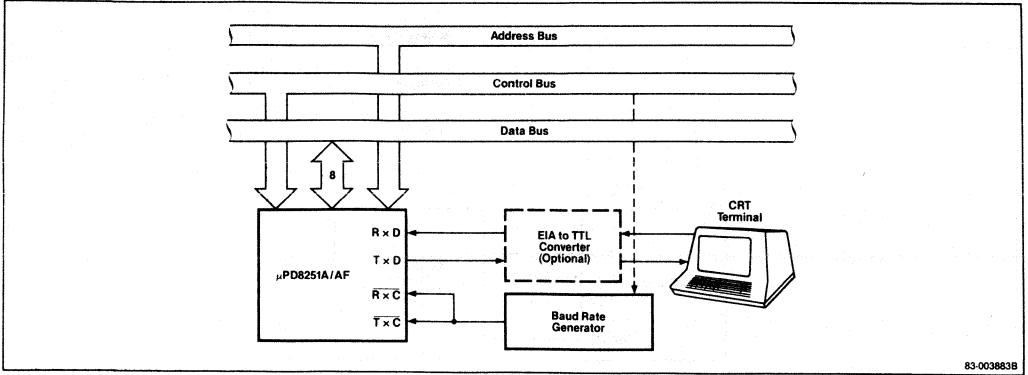
Command and Status Register Formats



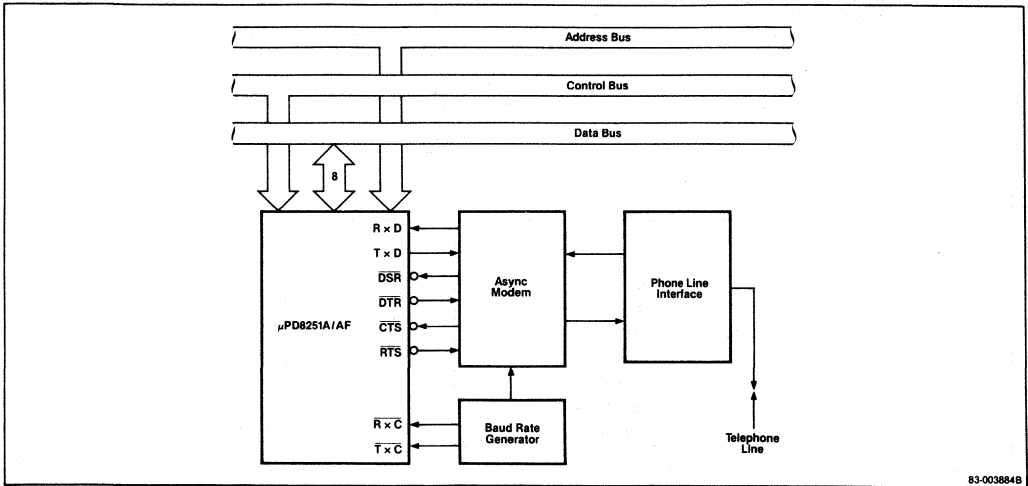
Notes:

- (1) NO effect in ASYNC mode.
- (2) TxRDY status bit is not totally equivalent to the TxRDY output pin, the relationship is as follows:  
TxRDY status bit = DB Buffer Empty  
TxRDY (pin 15) = DB Buffer Empty · CTS · TxEn

## Asynchronous Serial Interface to CRT Terminal, DC to 9600 Baud

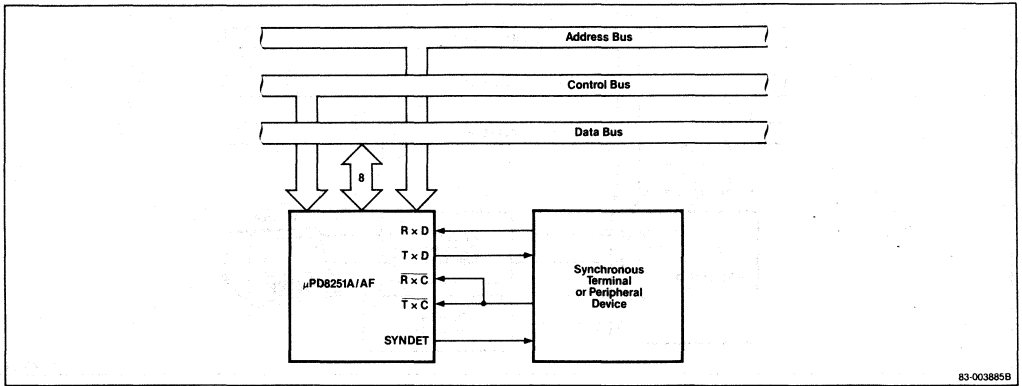


## Asynchronous Interface to Telephone Lines

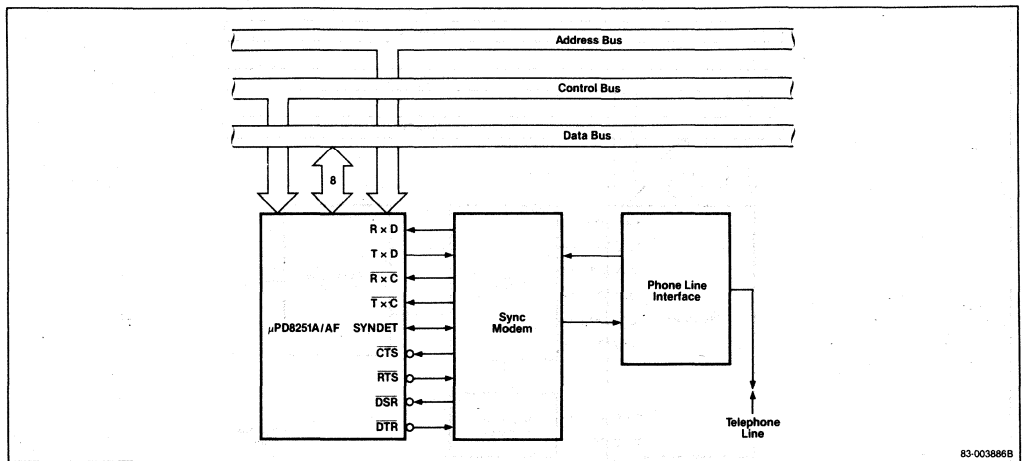


7

**Synchronous Interface to Terminal or Peripheral Device**



**Synchronous Interface to Telephone Lines**



## Description

The NEC  $\mu$ PD8253 contains three independent, programmable, multi-model 16-bit counter/timers. It is designed as a general purpose device, fully compatible with the 8080 family. The  $\mu$ PD8253 interfaces directly to the buses of the processor as an array of I/O ports.

The  $\mu$ PD8253 can generate accurate time delays under the control of system software. The three independent 16-bit counters can be clocked at rates from DC to 5 MHz. The system software controls the loading and starting of the counters to provide accurate multiple time delays. The counter output flags the processor at the completion of the time-out cycles.

System overhead is greatly improved by relieving the software from the maintenance of timing loops. Some other common uses for the  $\mu$ PD8253 in microprocessor based systems are:

- Programmable baud rate generator
- Event counter
- Binary rate multiplier
- Real time clock
- Digital one-shot
- Complex motor controller

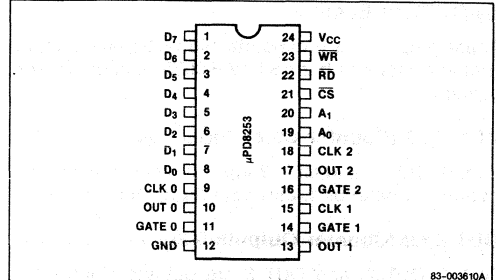
## Features

- Three independent 16-bit counters
- Clock rate: DC to 5 MHz
- Binary count or BCD
- Single +5 V power supply,  $\pm 10\%$

## Ordering Information

Part Number	Package Type	Max Frequency of Operation
$\mu$ PD8253C-2	24-pin plastic DIP	5 MHz
$\mu$ PD8253C-5	24-pin plastic DIP	4 MHz

## Pin Configuration



83-003610A

## Pin Identification

No.	Symbol	Function
1-8	D <sub>7</sub> -D <sub>0</sub>	Three-state data bus
9,15,18	CLK 0,1,2	Counter clock inputs 0-2
10,13,17	OUT 0,1,2	Counter outputs 0-2
11,14,16	GATE 0,1,2	Counter gate inputs 0-2
12	GND	Ground
19,20	A <sub>0</sub> ,A <sub>1</sub>	Counter select
21	CS	Chip select
22	$\overline{RD}$	Read counter
23	WR	Write command or data
24	V <sub>CC</sub>	+5 V power supply

**Pin Functions**

**D<sub>7</sub>-D<sub>0</sub> (Data Bus)**

These pins form a three-state, bidirectional data bus that interfaces with the 8080AF/8085 microprocessor system.

**CLK 0,1,2 (Counter Clock Inputs 0-2)**

CLK 0, CLK 1, and CLK 2 input the clock signal for counter 0, counter 1, and counter 2, respectively.

**OUT 0,1,2 (Counter Outputs 0-2)**

OUT 1, OUT 2, and OUT 3 are outputs signals for counter 0, counter 1, and counter 2, respectively.

**GATE 0,1,2 (Counter Gate Inputs 0-2)**

The GATE 0, GATE 1, and GATE 2 inputs gate counter 0, counter 1, and counter 2, respectively.

**GND (Ground)**

Connection to ground.

**A<sub>0</sub>, A<sub>1</sub> (Counter Select)**

These inputs are normally connected to the processor's address bus. Their function is to select which of the three counters will be operated on, and to address the control word register for mode selection.

**$\overline{CS}$  (Chip Select)**

A low level input to this pin enables the μPD8253. Reading and writing will not occur unless the device is selected. This input has no effect on the actual operation of the counters.

**$\overline{RD}$  (Read Counter)**

A low level input to this pin instructs the μPD8253 to send the selected counter value to the processor.

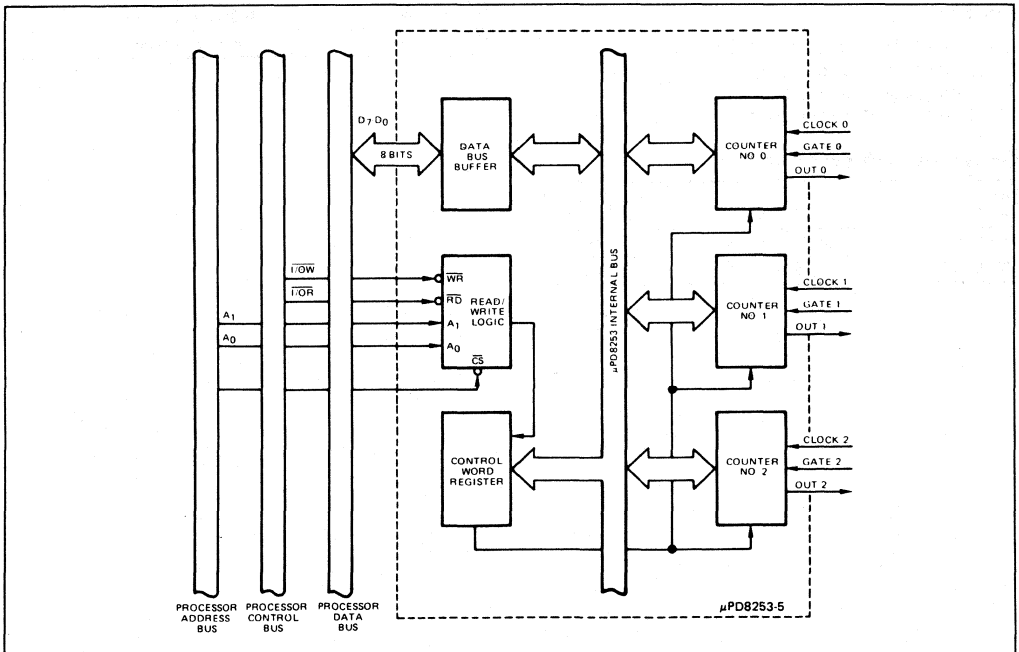
**$\overline{WR}$  (Write Command or Data)**

A low level input to this pin instructs the μPD8253 to receive mode information or counter input data from the processor.

**V<sub>cc</sub>**

+5 V power supply.

**Block Diagram**



## Functional Description

The three-state, bidirectional data bus buffer interfaces the μPD8253 to the 8080AF/8085A micro-processor system. Data transfer is according to the input or output instructions executed by the processor. The data bus buffer has three basic functions:

- Programming the μPD8253 modes
- Loading the count registers
- Reading the count values

The read/write logic controls the overall operation of the μPD8253 and is governed by inputs received from the processor system bus.

When  $A_0$  and  $A_1$  are high level, data from the data bus buffer is stored in the control word register. This data controls the operational mode of the counters, the selection of BCD or binary counting, and the loading of the count registers.

Counters 0, 1, and 2 are identical 16-bit down counters that are functionally independent, allowing for separate mode configurations and counting operations. Each counter can operate in either binary or BCD. Gate, input, and output line configurations are determined by the operational mode data stored in the control word register. System software overhead can be reduced by allowing the control word to govern the loading of the count data.

It is possible to read the contents of a counter when it is operating, without disturbing its operation. The following table shows how the counters are manipulated by input signals to the read/write logic.

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	$A_1$	$A_0$	Function
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

## Absolute Maximum Ratings

Operating temperature	0°C to +70°C
Storage temperature	-65°C to +150°C
Voltage on any pin	-0.5 to +7 volts (1)

### Note:

(1) With respect to ground.

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## Capacitance

$T_A = 25^\circ\text{C}$ ;  $V_{CC} = \text{GND} = 0\text{ V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	$C_{IN}$		10		pF	$f_c = 1\text{ MHz}$
Input/Output capacitance	$C_{I/O}$		20		pF	Unmeasured pins returned to $V_{SS}$

## DC Characteristics

$T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ;  $V_{CC} = +5\text{ V} \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input low voltage	$V_{IL}$	-0.5		0.8	V	
Input high voltage	$V_{IH}$ (1)	2.0		$V_{CC} + 0.5$	V	
Output low voltage	$V_{OL}$		0.45		V	$I_{OL} = 2.2\text{ mA}$
Output high voltage	$V_{OH}$	2.4			V	$I_{OH} = -400\text{ }\mu\text{A}$
Input load current	$I_{IL}$		$\pm 10$		$\mu\text{A}$	$0 \leq V_{IN} \leq V_{CC}$
Output float leakage current	$I_{OFL}$		$\pm 10$		$\mu\text{A}$	$0.45 \leq V_{OUT} \leq V_{CC}$
$V_{CC}$ supply current	$I_{CC}$			140	mA	

### Note:

(1)  $V_{IH}$  2.2 min for μPD8253-2.



**AC Characteristics**

$T_A = 0^\circ\text{C to } +70^\circ\text{C}; V_{CC} = +5\text{ V } \pm 10\%; \text{GND} = 0\text{ V}$

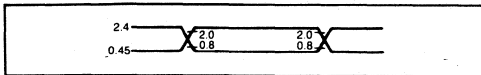
Parameter	Symbol	Limits μPD8253-2		Limits μPD8253-5		Unit	Test Conditions
		Min	Max	Min	Max		
<b>Read</b>							
Address stable before READ	$t_{AR}$	30		0		ns	
Address hold time for READ	$t_{RA}$	0		0		ns	
READ pulse width	$t_{RR}$	200		250		ns	
Data delay from READ	$t_{RD}$		140		170	ns	$C_L = 150\text{ pF}$
READ to data floating	$t_{DF}$	10	85	25	100	ns	$C_L = 150\text{ pF}$
Recovery time between READS	$t_{RV}$	200		1000		ns	
<b>Write</b>							
Address stable before WRITE	$t_{AW}$	0		0		ns	
Address hold time for WRITE	$t_{WA}$	0		0		ns	
WRITE pulse width	$t_{WW}$	160		250		ns	
Data set up time for WRITE	$t_{DW}$	130		150		ns	
Data hold time for WRITE	$t_{WD}$	0		0		ns	
Recovery time between WRITES	$t_{RW}$	200		1000		ns	
<b>Clock and Gate Timing</b>							
Clock period	$t_{CLK}$	200		250	DC	ns	
High pulse width	$t_{PWH}$	80		160		ns	
Low pulse width	$t_{PWL}$	80		90		ns	
Gate pulse width high	$t_{GW}$	120		150		ns	
Gate set up time to clock ↑	$t_{GS}$	70		100		ns	
Gate hold time after clock ↑	$t_{GH}$	50		50		ns	
Low gate width	$t_{GL}$	120		100		ns	
Output delay from Clock ↓	$t_{OD}$		250		300	ns	$C_L = 150\text{ pF}$
Output delay from gate	$t_{ODG}$		250		300	ns	$C_L = 150\text{ pF}$

**Note:**

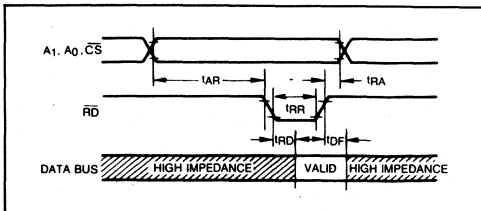
(1) AC timing measured at  $V_{OH} = 2.0\text{ V}; V_{OL} = 0.8\text{ V}$ .

**Timing Waveforms**

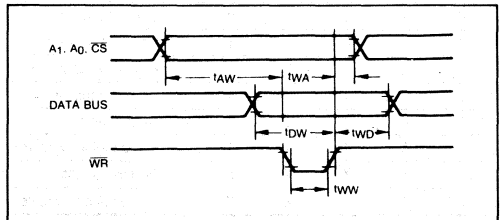
**AC Test Conditions**



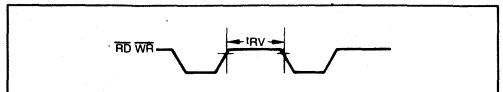
**Read Timing**



**Write Timing**

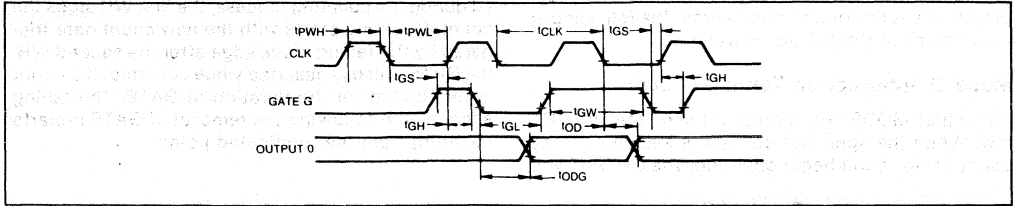


**Read and Write Timing**



## Timing Waveforms (cont)

### Clock and Gate Timing



### Programming the μPD8253

The programmer can select any of the six operational MODES for the counters using system software. Individual counter programming is accomplished by loading the CONTROL WORD REGISTER with the appropriate control word data ( $A_0, A_1 = 11$ ).

#### Control Word Format

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
SC <sub>1</sub>	SC <sub>0</sub>	RL <sub>1</sub>	RL <sub>0</sub>	M <sub>2</sub>	M <sub>1</sub>	M <sub>0</sub>	BCD

#### RL — Read/Load

RL <sub>1</sub>	RL <sub>0</sub>	
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

#### SC — Select Counter

SC <sub>1</sub>	SC <sub>0</sub>	
0	0	Select counter 0
0	1	Select counter 1
1	0	Select counter 2
1	1	Invalid

#### BCD

0	Binary counter, 16-bits
1	BCD counter, 4-decades

#### M-Mode

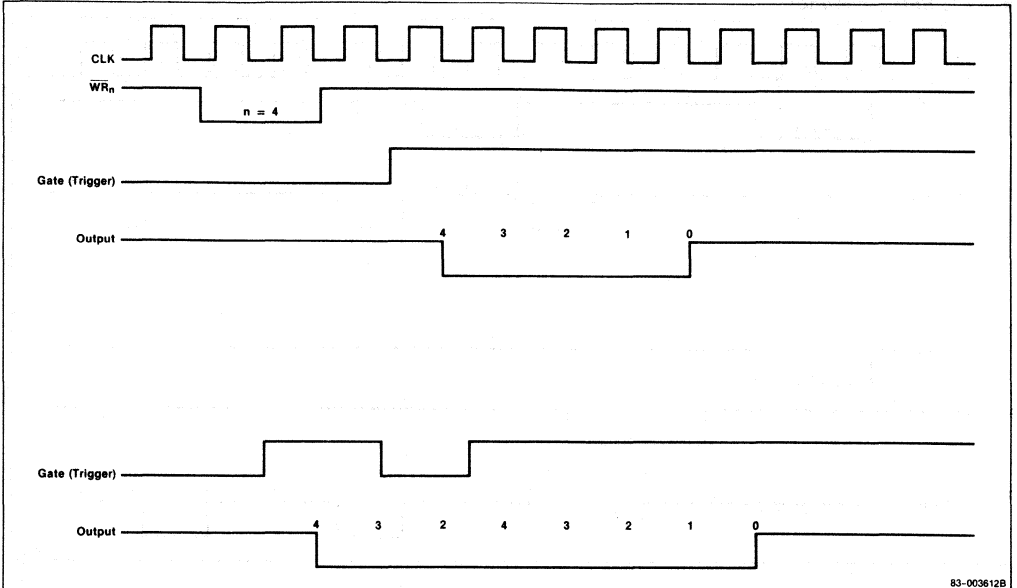
M <sub>2</sub>	M <sub>1</sub>	M <sub>0</sub>	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5



## Mode 1: Programmable One-Shot

The OUTPUT is set low by the falling edge of CLOCK following the trailing edge of GATE. The OUTPUT is set high again at the terminal count. The output pulse is not affected if new count data is loaded while the OUTPUT is low. The new data will be loaded on the rising edge of the next trigger pulse. The assertion of a trigger pulse while OUTPUT is low, resets and retriggers the one-shot. The OUTPUT will remain low for the full count value after the rising edge of TRIGGER.

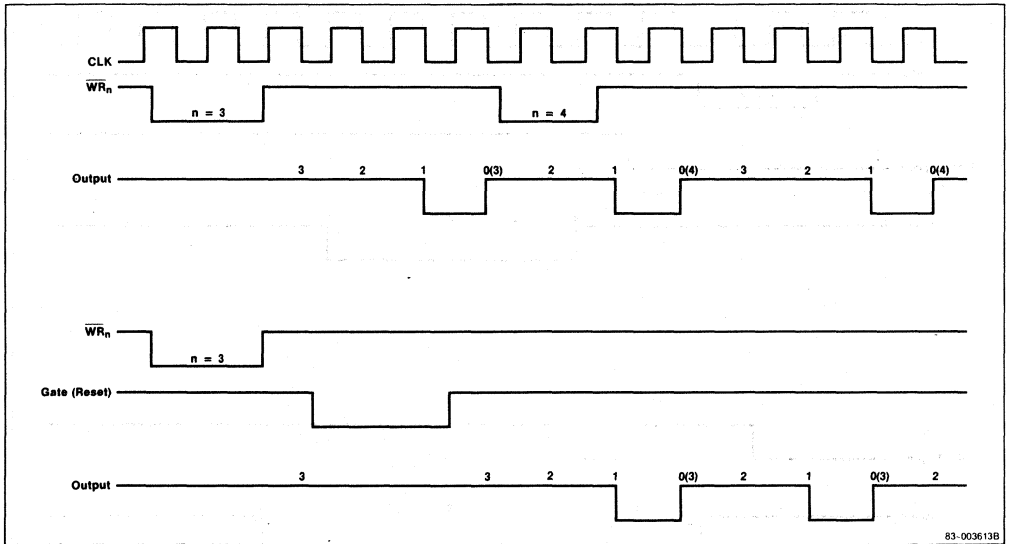
### Mode 1: Programmable One-Shot



**Mode 2: Rate Generator**

The RATE GENERATOR is a variable modulus counter. The OUTPUT goes low for one full CLOCK period as shown in the following timing diagram. The count data sets the time between OUTPUT pulses. If the count register is reloaded between output pulses the present period will not be affected. The subsequent period will reflect the new value. The OUTPUT will remain high for the duration of the asserted GATE input. Normal operation resumes on the falling CLOCK edge following the rising edge of GATE.

**Mode 2: Rate Generator**

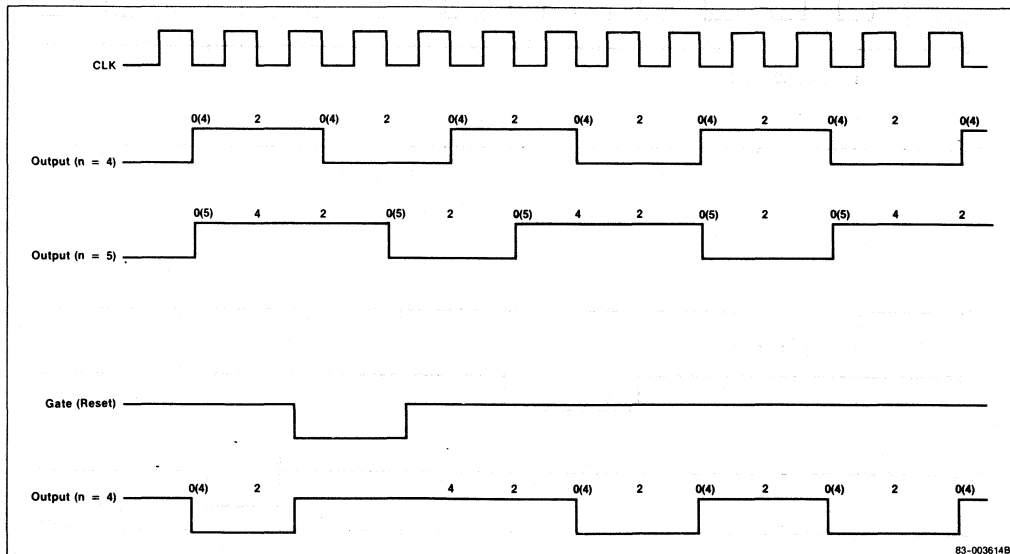


### Mode 3: Square Wave Generator

MODE 3 resembles MODE 2 except the OUTPUT will be high for half of the count and low for the other half (for even values of data). For odd values of count data the OUTPUT will be high one clock cycle longer than when it is low (High Period  $\rightarrow \frac{N+1}{2}$  clock cycles; Low Period  $\rightarrow \frac{N-1}{2}$  clock periods, where N is the decimal value of count data). If the count register is reloaded with a new value during counting, the new value will be reflected immediately after the output transition of the current count.

The OUTPUT will be held in the high state while GATE is asserted. Counting will start from the full count data after the GATE has been removed.

### Mode 3: Square Wave Generator

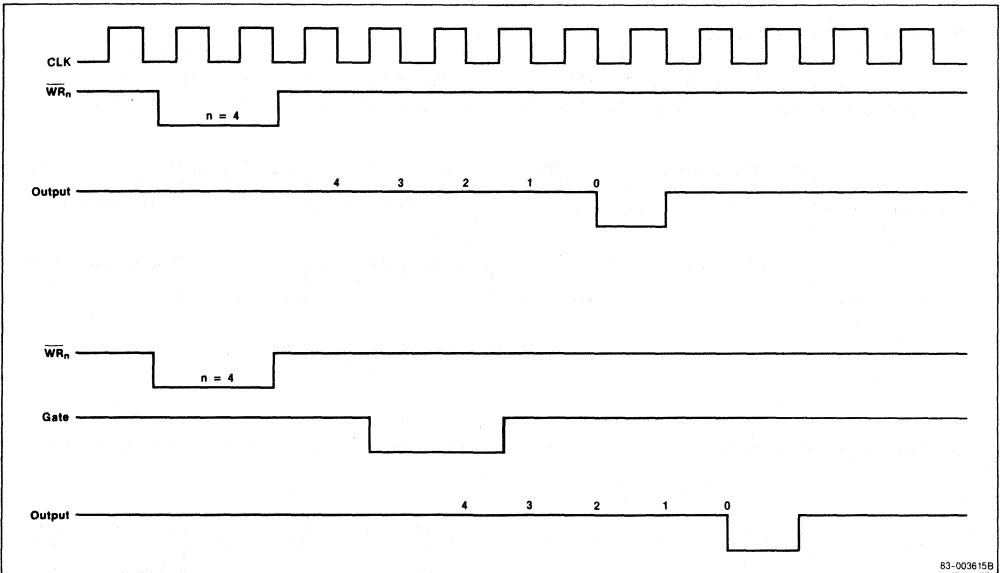


**Mode 4: Software Triggered Strobe**

The OUTPUT goes high when MODE 4 is set, and counting begins after the second byte of data has been loaded. When the terminal count is reached, the OUTPUT will pulse low for one clock period. Changes in count data are reflected in the OUTPUT as soon as the new data has been loaded into the count registers. During the loading of new data, the OUTPUT is held high and counting is inhibited.

The OUTPUT is held high for the duration of GATE. The counters are reset and counting begins from the full data value after GATE is removed.

**Mode 4: Software Triggered Strobe**

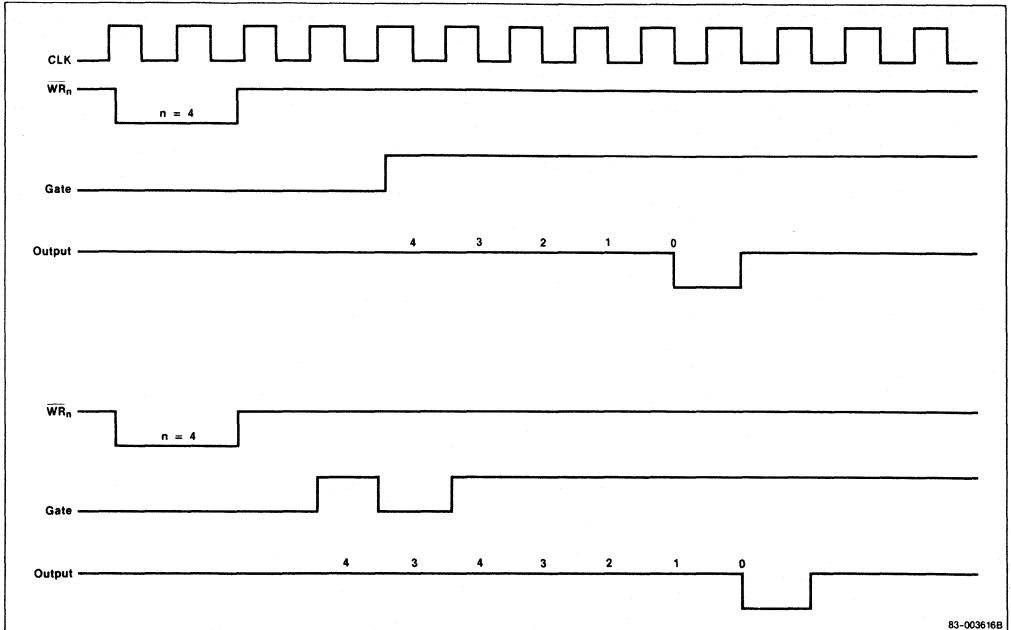


83-003615B

## Mode 5: Hardware Triggered Strobe

Loading MODE 5 sets OUTPUT high. Counting begins when count data is loaded and GATE goes high. After terminal count is reached, the OUTPUT will pulse low for one clock period. Subsequent trigger pulses will restart the counting sequence with the OUTPUT pulsing low on terminal count following the last rising edge of the trigger input. (Reference the bottom half of the timing diagram.)

### Mode 5: Hardware Triggered Strobe







## Description

The μPD8255A-2 and μPD8255A-5 are general purpose programmable input/output devices designed for use with the 8080A/8085A microprocessors. Twenty-four I/O lines may be programmed in two groups of twelve (group I and group II) and used in three modes of operation. In the basic mode, (Mode 0), each group of twelve I/O pins may be programmed in sets of 4 to input or output. In the strobed mode, (MODE 1), each group may be programmed to have 8 lines of input or output. Three of the remaining four pins in each group are used for handshaking strobes and interrupt control signals. The bidirectional bus mode, (MODE 2), uses the 8 lines of port A for a bi-directional bus, and five lines from port C for bus control signals.

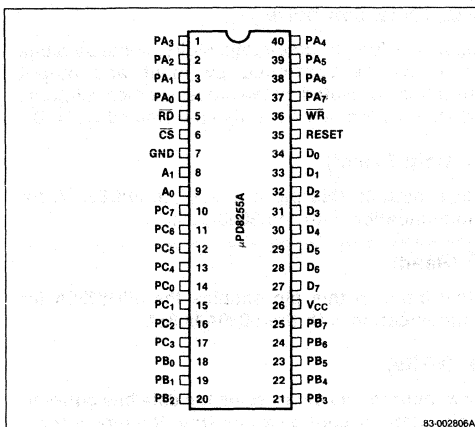
## Features

- Fully compatible with the 8080A/8085 microprocessor families
- All inputs and outputs TTL compatible
- 24 programmable I/O pins
- Direct bit set/reset eases control application interfaces
- Eight Darlington drive outputs for printers and displays
- LSI drastically reduces system package count

## Ordering Information

Part Number	Package Type	Max System Clock Frequency
μPD8255AC-2	40-pin plastic DIP	5 MHz
μPD8255AC-5	40-pin plastic DIP	.4 MHz

## Pin Configuration



## Pin Identification

No.	Symbol	Function
1-4, 37-40	PA <sub>7</sub> -PA <sub>0</sub>	Port A (I/O)
5	RD	Read input
6	CS	Chip select input
7	GND	Ground
8,9	A <sub>1</sub> ,A <sub>0</sub>	Port address inputs
10-17	PC <sub>7</sub> -PC <sub>0</sub>	Port C (I/O)
18-25	PB <sub>7</sub> -PB <sub>0</sub>	Port B (I/O)
26	V <sub>CC</sub>	+5 V power supply
27-34	D <sub>7</sub> -D <sub>0</sub>	Bidirectional data bus
35	RESET	Reset input
36	WR	Write input

**Pin Functions**

**D<sub>7</sub>-D<sub>0</sub> (Data Bus Buffer)**

These pins form a three-state, bidirectional data bus buffer that is controlled by input and output instructions executed by the processor. Control words and status information are also transmitted via D<sub>7</sub>-D<sub>0</sub>.

**$\overline{CS}$  (Chip Select)**

A low input to this pin enables the μPD8255A for communication with the 8080A/8085A.

**$\overline{RD}$  (Read)**

A low input to this pin enables the μPD8255A for communication with the 8080A/8085A.

**$\overline{WR}$  (Write)**

A low input to this pin enables the data bus buffer to receive data or control words from the processor.

**A<sub>1</sub>, A<sub>0</sub> (Port Address)**

These inputs are used in conjunction with  $\overline{CS}$ ,  $\overline{RD}$ , and  $\overline{WR}$  to control the selection of one of the three ports on the control word register. A<sub>0</sub> and A<sub>1</sub> are usually connected to A<sub>0</sub> and A<sub>1</sub> of the processor address bus.

**RESET (Reset)**

A high level input to this pin clears the control register and places ports A, B, and C in input mode. The input latches in ports A, B, and C are not cleared.

**PA<sub>7</sub>-PA<sub>0</sub>, PB<sub>7</sub>-PB<sub>0</sub>, PC<sub>7</sub>-PC<sub>0</sub> (Ports A, B, and C)**

These three 8-bit I/O ports can be configured to meet a variety of functional requirements through system software. The effectiveness and flexibility of the μPD8255A are further enhanced by special features unique to each of the ports, as follows:

- Port A has an 8-bit data output latch/buffer, data input latch/buffer, and data input latch.
- Port B has an 8-bit data I/O latch/buffer and an 8-bit data input buffer.
- Port C has an 8-bit output latch/buffer and a data input buffer (input not latched).

Port C may be divided into two independent 4-bit control and status ports for use with ports A and B.

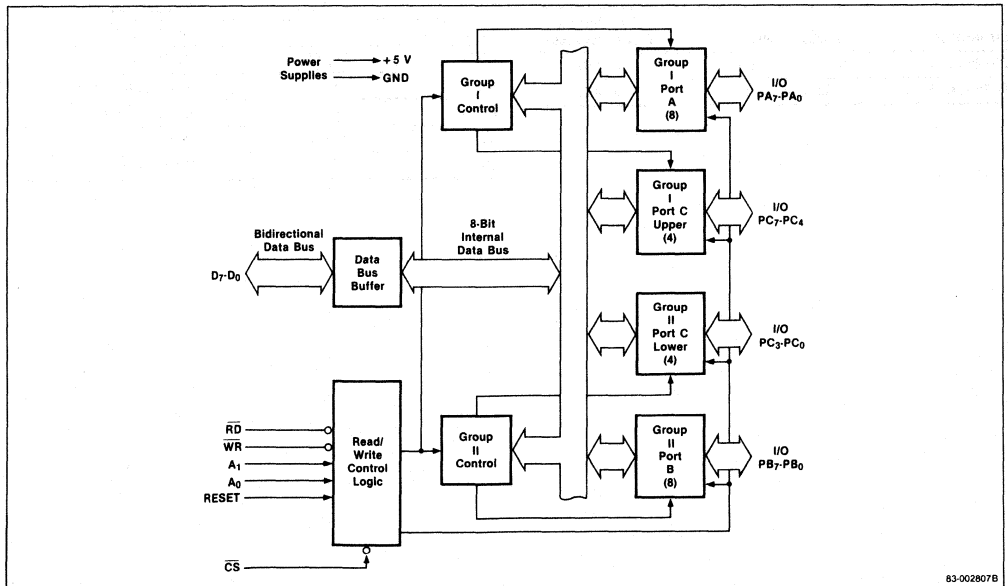
**V<sub>CC</sub>**

+5 V power supply.

**GND (Ground)**

Connection to ground.

**Block Diagram**



83-002807B

## Functional Description

The read/write and control logic manages all internal and external transfers of data, control, and status. It is through this block that the processor address and control buses control the peripheral interfaces.

Through an OUT instruction in system software from the processor, a control word is transmitted to the μPD8255A. Information such as the mode, bit set, and bit reset is used to initialize the functional configuration of each I/O port.

Both group I and group II accept commands from the read/write control logic and control words from the internal data bus and in turn controls its associated I/O ports, as follows:

- Group I: port A and upper port C (PC<sub>7</sub>-PC<sub>4</sub>)
- Group II: port B and lower port C (PC<sub>3</sub>-PC<sub>0</sub>)

While the control word register can be written to, the contents cannot be read back to the processor.

## Absolute Maximum Ratings

T<sub>A</sub> = 25°C

Operating temperature, T<sub>OP</sub> 0°C to +70°C

Storage temperature, T<sub>STG</sub> -65°C to +150°C

Voltage on any pin with respect to V<sub>SS</sub> -0.5 to +7 V

**Comment:** Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## DC Characteristics

T<sub>A</sub> = 0 to +70°C; V<sub>CC</sub> = +5 V ±10%; V<sub>SS</sub> = 0 V

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input low voltage	V <sub>IL</sub>	-0.5	0.8	V	
Input high voltage	V <sub>IH</sub>	2	V <sub>CC</sub>	V	
Output low voltage	V <sub>OL</sub>		0.45	V	(2)
Output high voltage	V <sub>OH</sub>	2.4		V	(3)
Darlington drive current	I <sub>OH</sub> (1)	-1	-4	mA	V <sub>EXT</sub> = 1.5 V R <sub>EXT</sub> = 750Ω
Power supply current	I <sub>CC</sub>		120	mA	V <sub>CC</sub> = +5 V, output open
Input leakage current	I <sub>LIH</sub>		10	μA	V <sub>IN</sub> = V = V <sub>CC</sub>
Input leakage current	I <sub>LIL</sub>		-10	μA	V <sub>IN</sub> = 0.4 V
Output leakage current	I <sub>LOH</sub>		±10	μA	V <sub>OUT</sub> = V <sub>CC</sub> ; CS = 2.0 V
Output leakage current	I <sub>LOL</sub>		-10	μA	V <sub>OUT</sub> = 0.4 V; CS = 2.0 V

### Note:

- (1) Any set of eight outputs from either port A, B, C can source 4 mA into 1.5 V.
- (2) I<sub>OL</sub> = 2.5 mA for DB port; 1.7 mA for peripheral ports.
- (3) I<sub>OH</sub> = -400μA for DB port; -200 μA for peripheral ports.

## Capacitance

T<sub>A</sub> = 25°C; V<sub>CC</sub> = 0V

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input capacitance	C <sub>I</sub>		10	pF	f <sub>c</sub> = 1 MHz
I/O capacitance	C <sub>I0</sub>		20	pF	Unmeasured pins returned to V <sub>SS</sub>

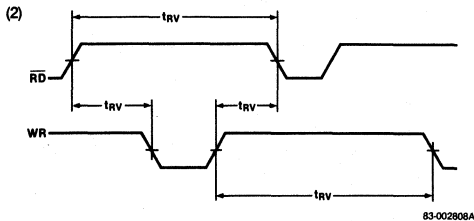
**AC Characteristics**

T<sub>A</sub> = 0°C to +70°C; V<sub>CC</sub> = +5 V ±5%; V<sub>SS</sub> = 0 V

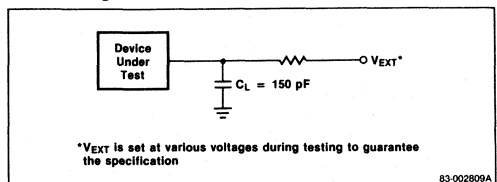
Parameter	Symbol	8255A-2 Limits		8255A-5 Limits		Unit	Test Conditions
		Min	Max	Min	Max		
Address stable before $\overline{\text{READ}}$	t <sub>AR</sub>	0		0		ns	
Address stable after $\overline{\text{READ}}$	t <sub>RA</sub>	0		0		ns	
READ pulse width	t <sub>RR</sub>	200		250		ns	
Data valid from $\overline{\text{READ}}$	t <sub>RD</sub>		140		170	ns	C <sub>L</sub> = 150 pF
Data float after $\overline{\text{READ}}$	t <sub>DF</sub>	10	100	10	100	ns	C <sub>L</sub> = 100 pF C <sub>L</sub> = 15 pF
Time between READS and /WRITES	t <sub>RV</sub>	200		850		ns	(Note 2)
<b>Write</b>							
Address stable before $\overline{\text{WRITE}}$	t <sub>AW</sub>	0		0		ns	
Address stable after $\overline{\text{WRITE}}$	t <sub>WA</sub>	20		20		ns	
WRITE pulse width	t <sub>WW</sub>	200		250		ns	
Data valid to $\overline{\text{WRITE}}$ (T.E.)	t <sub>DW</sub>	100		100		ns	
Data valid after $\overline{\text{WRITE}}$	t <sub>WD</sub>	0		0		ns	
<b>Other Timing</b>							
WR = 0 to output	t <sub>WB</sub>		350		350	ns	C <sub>L</sub> = 150 pF
Peripheral data before $\overline{\text{RD}}$	t <sub>IR</sub>	0		0		ns	
Peripheral data after $\overline{\text{RD}}$	t <sub>HR</sub>	0		0		ns	
ACK pulse width	t <sub>AK</sub>	300		300		ns	
STB pulse width	t <sub>ST</sub>	350		350		ns	
Per. data before T.E. of STB	t <sub>PS</sub>	0		0		ns	
Per. data after T.E. of STB	t <sub>PH</sub>	150		150		ns	
ACK = 0 to output	t <sub>AD</sub>		300		300	ns	C <sub>L</sub> = 150 pF
ACK = 0 to output float	t <sub>KD</sub>	20	250	20	250	ns	C <sub>L</sub> = 50 pF C <sub>L</sub> = 15 pF
WR = 1 to OBF = 0	t <sub>WOB</sub>		300		650	ns	
ACK = 0 to OBF = 1	t <sub>AOB</sub>		350		350	ns	
STB = 0 to IBF = 1	t <sub>SIB</sub>		300		300	ns	
RD = 1 to IBF = 0	t <sub>RIB</sub>		300		300	ns	
RD = 0 to INTR = 0	t <sub>RIT</sub>		400		400	ns	
STB = 1 to INTR = 1	t <sub>SIT</sub>		300		300	ns	C <sub>L</sub> = 150 pF
ACK = 1 to INTR = 1	t <sub>AIT</sub>		350		350	ns	
WR = 0 to INTR = 0	t <sub>WIT</sub>		450		850	ns	C <sub>L</sub> = 150 pF (Note 3)

**Note:**

- (1) Period of reset pulse must be at least 50 μs during or after power on. Subsequent reset pulse can be 500 ns min.
- (3) INTRt may occur as early as  $\overline{\text{WR}}t$ .

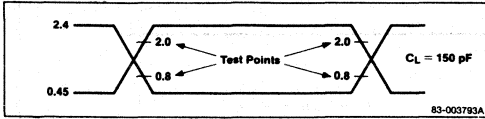


**AC Testing Load Circuit**

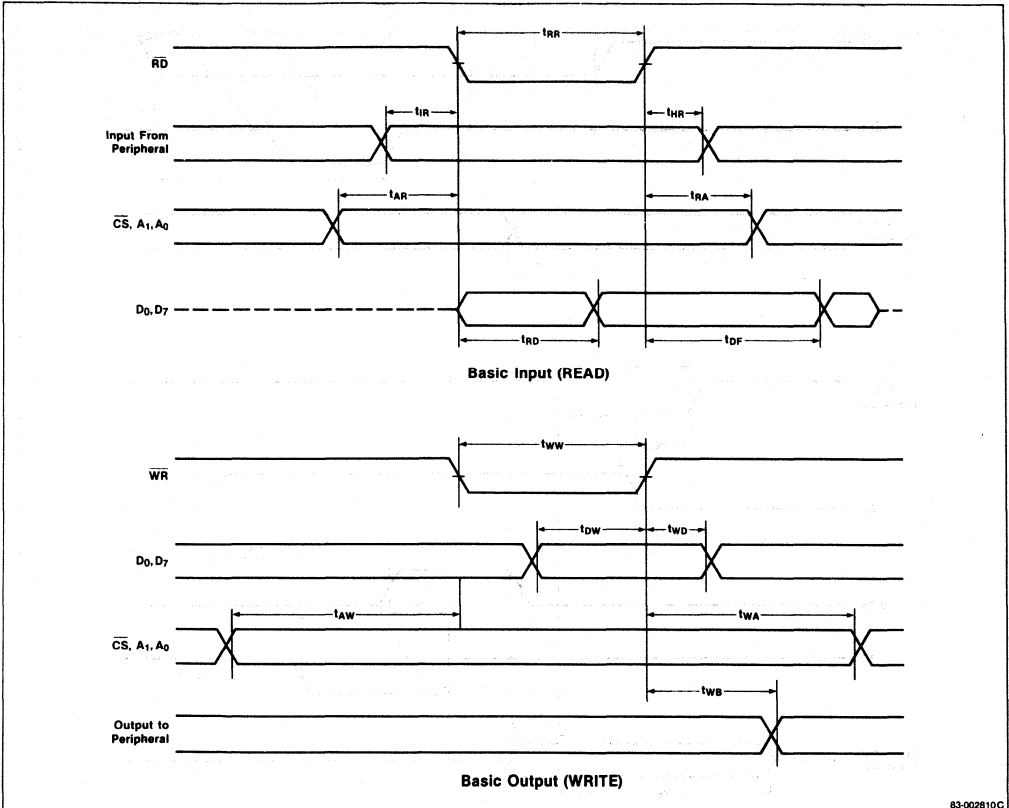


## Timing Waveforms

### AC Testing Input, Output Waveform

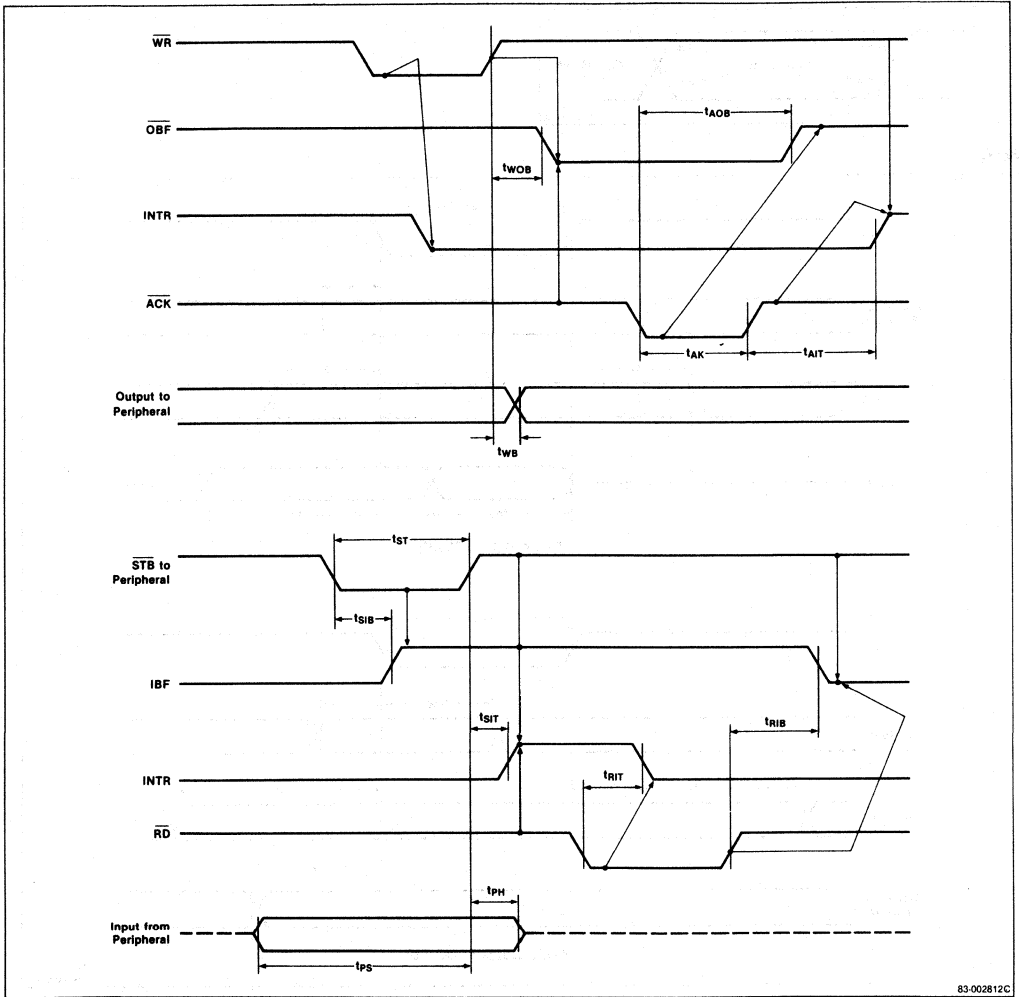


### Mode 0



Timing Waveforms (cont)

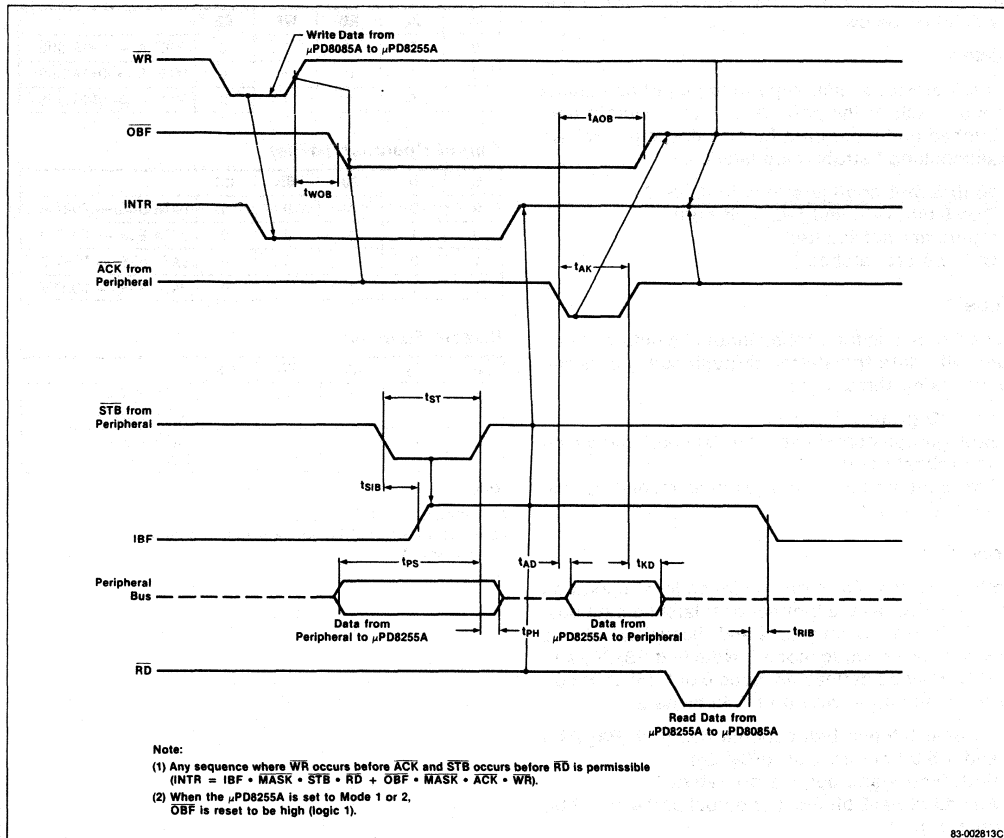
Mode 1



83-002812C

## Timing Waveforms (cont)

### Mode 2



83-002813C



**Modes**

The μPD8255A can be operated in modes 0, 1 or 2 which are selected by appropriate control words and are detailed below.

**Mode 0**

Mode 0 provides basic input and output operations through each of the ports A, B, and C. Output data is latched and input data follows the peripheral. No "handshaking" strobes are needed.

- 16 different configurations in mode 0
- Two 8-bit ports and two 4-bit ports
- Inputs are not latched
- Outputs are latched

**Mode 1**

Mode 1 provides for strobed input and output operations with data transferred through port A or B and handshaking through port C.

- Two I/O groups (I and II)
- Both groups contain an 8-bit data port and a 4-bit control/data port
- Both 8-bit data ports can be either latched input or latched output

**Mode 2**

Mode 2 provides for strobed bidirectional operation using PA<sub>0</sub>PA<sub>7</sub> as the bidirectional latched data bus. PC<sub>3</sub>PC<sub>7</sub> is used for interrupts and "handshaking" bus flow control similar to mode 1. Note that PB<sub>0</sub>PB<sub>7</sub> and PC<sub>0</sub>PC<sub>2</sub> may be defined as mode 0 or 1, input or output in conjunction with port A in mode 2.

- An 8-bit latched bidirectional bus port (PA<sub>0</sub>-PA<sub>7</sub>) and a 5-bit control port (PC<sub>3</sub>PC<sub>7</sub>)
- Both inputs and outputs are latched
- An additional 8-bit input or output port with a 3-bit control port.

**Basic Operation****Input Operation (Read)**

A <sub>1</sub>	A <sub>0</sub>	RD	WR	CS	
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)**

A <sub>1</sub>	A <sub>0</sub>	RD	WR	CS	
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**

A <sub>1</sub>	A <sub>0</sub>	RD	WR	CS	
X	X	X	X	1	DATA BUS → HIGH Z STATE
X	X	1	1	0	DATA BUS → HIGH Z STATE

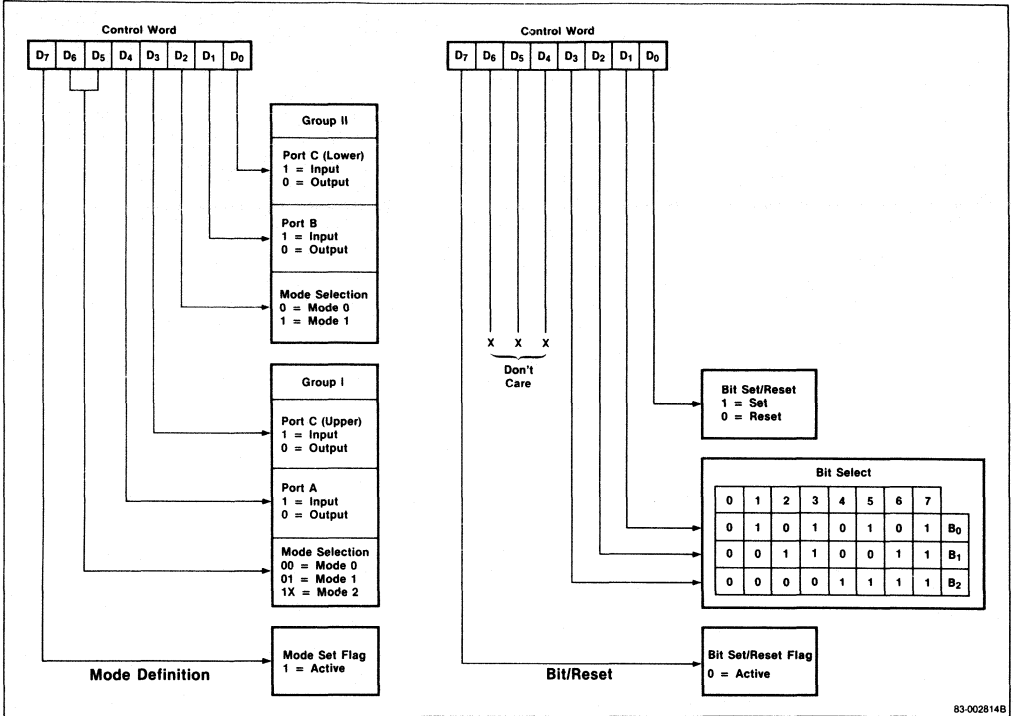
**Note:**

(1) X means "DO NOT CARE"

(2) All conditions not listed are illegal and should be avoided.

## Formats

### Mode Definition, Bit/Rest Format





## Description

The μPD8257 is a programmable four-channel direct memory access (DMA) controller. It is designed to simplify high-speed transfers between peripheral devices and memories. Upon a peripheral request, the μPD8257 generates a sequential memory address, thus allowing the peripheral to read or write data directly to or from memory. Peripheral requests are prioritized within the μPD8257 so that the system bus may be acquired by the generation of a single HOLD command to the 8080A. DMA cycle counts are maintained for each of the four channels, and a control signal notifies the peripheral when the preprogrammed number of DMA cycles has occurred. Output control signals are also provided which allow simplified sectored data transfers and expansion to other μPD8257 devices for systems requiring more than four DMA channels.

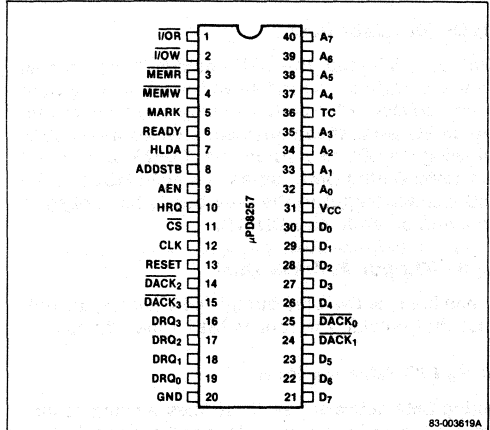
## Features

- Four-channel DMA controller
- Priority DMA request logic
- Channel inhibit logic
- Terminal count and modulo 128 outputs
- Automatic load mode
- Single TTL clock
- Single +5 V ± 10% power supply
- Expandable
- Available in extended temperature range

## Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD8257C-2	40-pin plastic DIP	5 MHz
μPD8257C-5	40-pin plastic DIP	3 MHz

## Pin Configuration



## Pin Identification

No.	Symbol	Function
1	I/OR	I/O read, control signal
2	I/OW	I/O write, control signal
3	MEMR	Memory read output
4	MEMW	Memory write output
5	MARK	Modulo 128 mark
6	READY	Ready input
7	HLDA	Hold acknowledge input (from 8080A)
8	ADDSTB	Address strobe output
9	AEN	Address enable output
10	HRQ	Hold request (to 8080A)
11	CS	Chip select input
12	CLK	Clock input
13	RESET	Reset input
14, 15, 24, 25	DACK <sub>2</sub> , DACK <sub>3</sub> , DACK <sub>1</sub> , DACK <sub>0</sub>	DMA acknowledge output
16-19	DRQ <sub>3</sub> -DRQ <sub>0</sub>	DMA request input
20	GND	Ground
21-23, 26-30	D <sub>7</sub> -D <sub>5</sub> , D <sub>4</sub> -D <sub>0</sub>	I/O data bus
31	V <sub>CC</sub>	+5 V power supply
32-35	A <sub>0</sub> -A <sub>3</sub>	I/O address bus
36	TC	Terminal count output
37-40	A <sub>4</sub> -A <sub>7</sub>	Output address bus

## Pin Functions

### D<sub>0</sub>-D<sub>7</sub> (I/O Data Bus)

During an I/O read, the CPU enables these lines as inputs, allowing it to read an address register, a word count register, or the status or temporary register. During an I/O write, these lines are enabled as outputs, allowing the CPU to program the μPD8257-2/-5 control registers. During DMA cycles, the eight MSBs of the address are output to the data bus to be strobed to an external latch via ADDSTB.

### A<sub>4</sub>-A<sub>7</sub> (Output Address Bus)

These lines, active only during DMA service, are outputs that provide the four MSBs of the address.

### A<sub>0</sub>-A<sub>3</sub> (I/O Address Bus)

During DMA active states, these lines are outputs that provide the 4 LSBs of the output address bus. During DMA idle states, these lines are inputs, allowing the CPU to load or examine control registers.

### DRQ<sub>0</sub>-DRQ<sub>3</sub> (DMA Request Input)

These are asynchronous channel request inputs used by peripherals to request DMA service. In a fixed priority scheme, DRQ<sub>0</sub> has the highest priority and DRQ<sub>3</sub> has the lowest. The polarity of these lines is programmable; however, reset initializes them to active high.

### HLDA (Hold Acknowledge)

Indicates that the CPU has relinquished control of the system busses.

### HRQ (Hold Request)

Requests control of the system bus. The μPD8257-2/-5 issues this signal in response to software requests or DRQ inputs from peripherals.

### $\overline{\text{DACK}}_0$ - $\overline{\text{DACK}}_3$ (DMA Acknowledge Output)

These lines indicate an active channel. They are sometimes used to select a peripheral. Only one DACK may be active at any time. All DACK lines are inactive unless DMA has control of the bus. The polarity of these lines is programmable; however, reset initializes them to active low.

### TC (Terminal Count)

When the terminal count occurs, TC goes high, informing the CPU that the data transfer is complete.

### RESET

Clears the command, status, request, and temporary registers, the first/last flip flop, and sets the mask register. The μPD8257-2/-5 is in idle state after a reset.

### $\overline{\text{CS}}$ (Chip Select)

The CPU uses  $\overline{\text{CS}}$  to select the μPD8257-2/-5 as an I/O device during an I/O read or write by the CPU. This provides CPU communication on the data bus.  $\overline{\text{CS}}$  may be held low during multiple transfers to or from the μPD8257-2/-5 as long as  $\overline{\text{I/OR}}$  or  $\overline{\text{I/OW}}$  is toggled following each transfer.

### READY

This signal can extend memory read and write pulses for slow memories or I/O peripherals.

### CLK (Clock)

Controls internal operations and data transfer rate.

### AEN (Address Enable)

This signal allows the external latch to output the upper address byte by disabling the system bus during DMA cycles. Use HLDA and AEN to deselect I/O peripherals that may be erroneously accessed during DMA transfers. The μPD8257-2/-5 deselects itself during DMA transfers.

### ADDSTB (Address Strobe)

This signal strobes the upper address byte from D<sub>0</sub>-D<sub>7</sub> into an external latch.

### $\overline{\text{MEMR}}$ (Memory Read)

This signal accesses data from a specified memory location during memory-to-peripheral or memory-to-memory transfers.

### $\overline{\text{MEMW}}$ (Memory Write)

This signal writes data to a specified memory location during peripheral-to-memory or memory-to-memory transfers.

### $\overline{\text{I/OR}}$ (I/O Read)

In the idle state, this signal is an input control line used by the CPU to read control registers. In the active state, the μPD8257-2/-5 uses  $\overline{\text{I/OR}}$  as an output control signal to access data from a peripheral during a DMA write.

## I/O Write

In the idle state, the CPU uses I/O as an input control signal to load information to the μPD8257-2/-5. In the active state, the μPD8257-2/-5 uses I/O as an output control signal to load data to a peripheral during a DMA read.

The rising edge of  $\overline{WR}$  must follow each data byte transfer in order for the CPU to write to the μPD8257-2/-5. Holding I/O low while toggling CS does not produce the same effect.

## MARK (Modulo 128 Mark)

This output notifies the selected peripheral that the current DMA cycle is the 128th cycle since the previous MARK output. MARK always occurs at 128 (and all multiples of 128) cycles from the end of the data block.

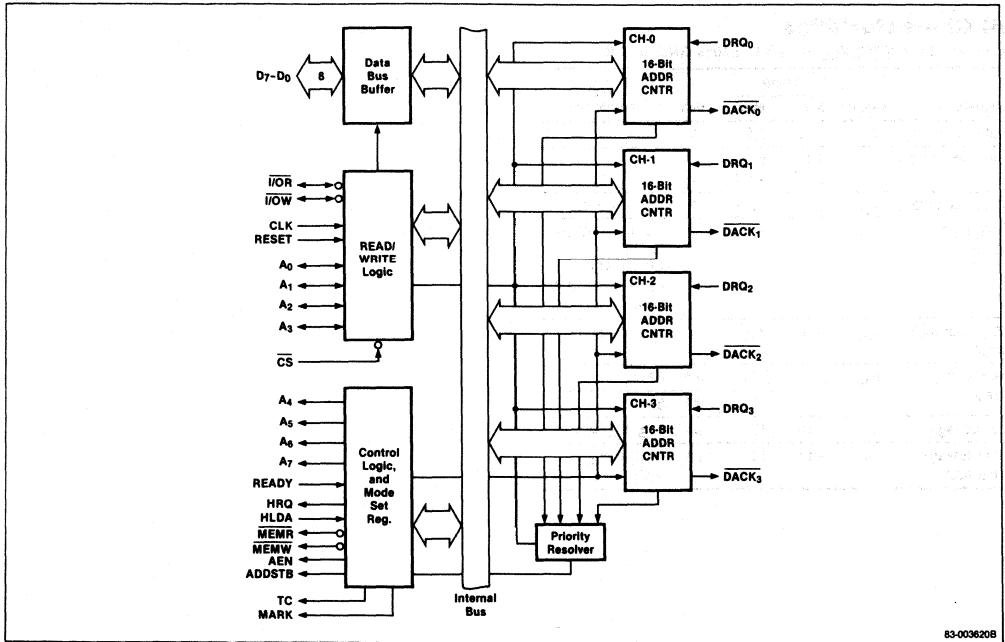
## V<sub>CC</sub>

Power supply.

## GND

Ground.

## Block Diagram



**Absolute Maximum Ratings**

$T_A = 25^\circ\text{C}$	
Operating temperature, $T_{\text{OPT}}$	$0^\circ\text{C}$ to $70^\circ\text{C}$
Storage temperature, $T_{\text{STG}}$	$-65^\circ\text{C}$ to $+150^\circ\text{C}$
Power supply voltage, $V_{\text{CC}}$	$-0.5\text{ V}$ to $+7\text{ V}$ (1)
Power dissipation	1 Watt

**Comment:** Stress above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Note:**

(1) With respect to Ground

**DC Characteristics**

$T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ;  $V_{\text{CC}} = +5\text{ V} \pm 10\%$  GND = 0 V

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input low voltage	$V_{\text{IL}}$	-0.5	0.8	V	
Input high voltage	$V_{\text{IH}}$	2.0	$V_{\text{CC}} + 0.5$	V	
Output low voltage	$V_{\text{OL}}$		0.45	V	$I_{\text{OL}} = 1.6\text{ mA}$
Output high voltage	$V_{\text{OH}}$	2.4	$V_{\text{CC}}$	V	$I_{\text{OH}} = -150\ \mu\text{A}$ for AB, DB and AEN $I_{\text{OH}} = -80\ \mu\text{A}$ for others
HRQ output high voltage	$V_{\text{HH}}$	3.3	$V_{\text{CC}}$	V	$I_{\text{OH}} = -80\ \mu\text{A}$
Power supply current	$I_{\text{CC}}$		100	mA	8257-2
			120	mA	8257-5
Input leakage	$I_{\text{IL}}$	-10	10	$\mu\text{A}$	$0 \leq V_{\text{IN}} \leq V_{\text{CC}}$
Output leakage during float	$I_{\text{OFL}}$	-10	10	$\mu\text{A}$	$0.45 \leq V_{\text{OUT}} \leq V_{\text{CC}}$

**Capacitance**

$T_A = 25^\circ\text{C}$ ;  $V_{\text{CC}} = \text{GND} = 0\text{ V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	$C_{\text{I}}$			10	pF	$f_c = 1\text{ MHz}$
I/O capacitance	$C_{\text{I/O}}$			20	pF	Unmeasured pins returned to GND

## AC Characteristics

T<sub>A</sub> = 0°C to +70°C; V<sub>CC</sub> = 5 V ±10%; GND = 0 V

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8257-2		μPD8257-5			
		Min	Max	Min	Max		
<b>Read</b>							
ADR or CS <sub>1</sub> Setup to RD <sup>†</sup>	t <sub>AR</sub>	0				ns	
ADR or CS → hold from RD <sup>†</sup>	t <sub>RA</sub>	0				ns	
Data Access from RD <sup>†</sup>	t <sub>RDE</sub>	0	140	0	170	ns	C <sub>L</sub> = 100 pF
DB → float delay from RD <sup>†</sup>	t <sub>RDF</sub>	10	85	20	100	ns	C <sub>L</sub> = 100 pF
RD width	t <sub>RW</sub>	200		250		ns	
<b>Write</b>							
ADR setup to WR <sup>†</sup>	t <sub>AW</sub>	20				ns	
ADR hold from WR <sup>†</sup>	t <sub>WA</sub>	0				ns	
Data setup to WR <sup>†</sup>	t <sub>DW</sub>	100		200		ns	
Data hold from WR <sup>†</sup>	t <sub>WD</sub>	0				ns	
WR width	t <sub>WWS</sub>	100		200		ns	
<b>Other timing</b>							
Reset pulse width	t <sub>RSTW</sub>	300		300		ns	
Power supply f(V <sub>CC</sub> ) setup to reset <sup>†</sup>	t <sub>RSTD</sub>	500		500		μs	
Signal rise & fall times	t <sub>r</sub> , t <sub>f</sub>		20		20		
Reset to first $\overline{\text{IOWR}}$	t <sub>RSTS</sub>	2		2		t <sub>CY</sub>	

### Note:

(1) All timing measurements are made at the following reference voltages unless specified otherwise: Input "1" at 2.0 V, "0" at 0.8 V, output "1" at 2.0 V, "0" at 0.8 V.

## AC Characteristics

T<sub>A</sub> = 0°C to +70°C; V<sub>CC</sub> = +5 V ±10%; GND = 0 V

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8257-2		μPD8257-5			
		Min	Max	Min	Max		
Cycle time (period)	t <sub>CY</sub>	0.200	4	0.320	4	μs	
Clock active (high)	t <sub>θ</sub>	80		80	.8t <sub>CY</sub>	ns	
DRQ <sup>†</sup> setup to $\theta$ (S1, S4)	t <sub>QS</sub>	50		120			
DRQ <sup>†</sup> hold from HLDA <sup>†</sup>	t <sub>QH</sub>	0		0			(4)
HRQ <sup>†</sup> or ↓ delay from $\theta$ (S1, S4) (measured at 2.0 V)	t <sub>DQ</sub>		160		160	ns	
HRQ <sup>†</sup> or ↓ delay from $\theta$ (S1, S4)	t <sub>HS</sub>		200	100	250	ns	(3)
HLDA <sup>†</sup> or ↑ setup to $\theta$ (S1, S4)	t <sub>HS</sub>	50		100		ns	
AEN <sup>†</sup> delay from $\theta$ (S1)	t <sub>AEL</sub>		150		300	ns	
AEN <sup>†</sup> delay from $\theta$ (S1)	t <sub>AET</sub>		150		200	ns	
ADR (AB) (active) delay from AEN <sup>†</sup> (S1)	t <sub>AEA</sub>	20		20		ns	(4)
ADR (AB) (active) delay from $\theta$ (S1)	t <sub>FAAB</sub>		200		250	ns	(2)
ADR (AB) (float) delay from $\theta$ (S1)	t <sub>AFAB</sub>		150		150	ns	(2)
ADR (AB) (stable) delay from $\theta$ (S1)	t <sub>ASM</sub>		200		250	ns	(2)
ADR (AB) (stable) hold from $\theta$ (S1)	t <sub>AH</sub>		t <sub>ASM</sub> - 50		t <sub>ASM</sub> - 50		(2)
ADR (AB) (valid) hold from RD <sup>†</sup> (S2, S1)	t <sub>AHR</sub>	60		60		ns	(4)



**AC Characteristics (cont)**

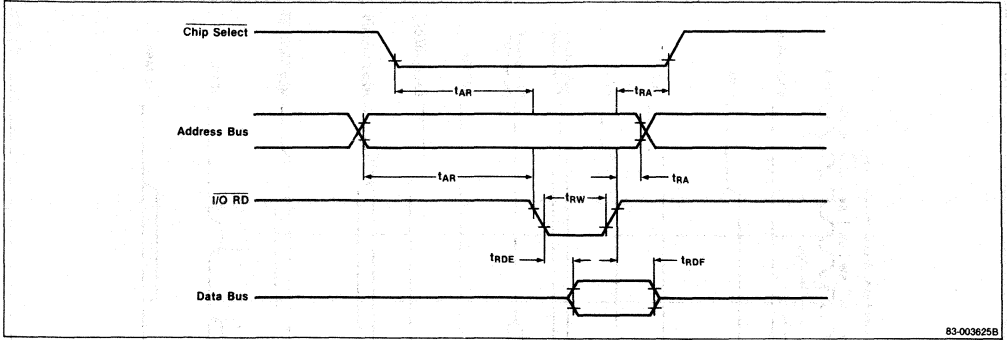
Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8257-2		μPD8257-5			
		Min	Max	Min	Max		
ADR (AB) (valid) hold from WRT <sup>†</sup> (S1, S1)	t <sub>AHW</sub>	100		300		ns	(4)
ADR (DB) (active) delay from θ <sup>†</sup> (S1)	t <sub>FADB</sub>		150		300	ns	(2)
ADR (DB) (float) delay from θ <sup>†</sup> (S2)	t <sub>AFDB</sub>	t <sub>STT</sub>	140	t <sub>STT</sub> + 20	170	ns	(2)
ADR (DB) setup to ADR STB <sup>↓</sup> (S1-S2)	t <sub>ASS</sub>	100		100		ns	(4)
ADR (DB) (valid) hold from ADR STB <sup>↓</sup> (S2)	t <sub>AHS</sub>	20		50		ns	(4)
ADR STB <sup>↓</sup> delay from θ <sup>†</sup> (S1)	t <sub>STL</sub>		150		200	ns	
ADR STB <sup>↓</sup> delay from θ <sup>†</sup> (S2)	t <sub>STT</sub>		140		140	ns	
ADR STB width (S1-S2)	T <sub>SW</sub>	t <sub>CY</sub> - 100		t <sub>CY</sub> - 100		ns	(4)
R <sup>↓</sup> D <sup>↑</sup> or W <sup>↑</sup> R (ext) <sup>↓</sup> delay from ADR STB <sup>↓</sup> (S2)	t <sub>ASC</sub>	20		70		ns	(4)
R <sup>↓</sup> D <sup>↑</sup> or W <sup>↑</sup> R (ext) <sup>↓</sup> delay from ADR (DB) (float) (S2)	t <sub>DBC</sub>	0		20		ns	(4)
DACK <sup>↑</sup> or <sup>↓</sup> delay from θ <sup>†</sup> (S2, S1) and TC/Mark <sup>↑</sup> delay from θ <sup>†</sup> (S3) and TC/Mark <sup>↑</sup> delay from θ <sup>†</sup> (S4)	t <sub>AK</sub>		200		250	ns	(5)
R <sup>↓</sup> D <sup>↑</sup> or W <sup>↑</sup> R (ext) <sup>↓</sup> delay from θ <sup>†</sup> (S2) and W <sup>↑</sup> R <sup>↑</sup> delay from θ <sup>†</sup> (S3)	t <sub>DCL</sub>		150		200	ns	(2) (6)
R <sup>↓</sup> D <sup>↑</sup> delay from θ <sup>†</sup> (S1, S1) and W <sup>↑</sup> R <sup>↑</sup> delay from θ <sup>†</sup> (S4)	t <sub>DCT</sub>		150		200	ns	(2) (7)
R <sup>↓</sup> D <sup>↑</sup> or W <sup>↑</sup> R (active) from θ <sup>†</sup> (S1)	t <sub>FAC</sub>		200		300	ns	(2)
R <sup>↓</sup> D <sup>↑</sup> or W <sup>↑</sup> R (float) from θ <sup>†</sup> (S1)	t <sub>AFC</sub>		150		150	ns	(2)
R <sup>↓</sup> D <sup>↑</sup> width (S2-S1 or S1)	T <sub>RWM</sub>	2t <sub>CY</sub> + t <sub>θ</sub> - 50		2t <sub>CY</sub> + t <sub>θ</sub> - 50		ns	(4)
W <sup>↑</sup> R width (S3-S4)	t <sub>WWM</sub>	t <sub>CY</sub> - 50		t <sub>CY</sub> - 50		ns	(4)
W <sup>↑</sup> R (ext) width (S2-S4)	t <sub>WWE</sub>	2t <sub>CY</sub> - 50		2t <sub>CY</sub> - 50		ns	(4)
READY set up time to θ <sup>†</sup> (S3, Sw)	t <sub>RS</sub>	30		30		ns	
READY hold time from θ <sup>†</sup> (S3, Sw)	t <sub>RH</sub>	30		30		ns	

**Note:**

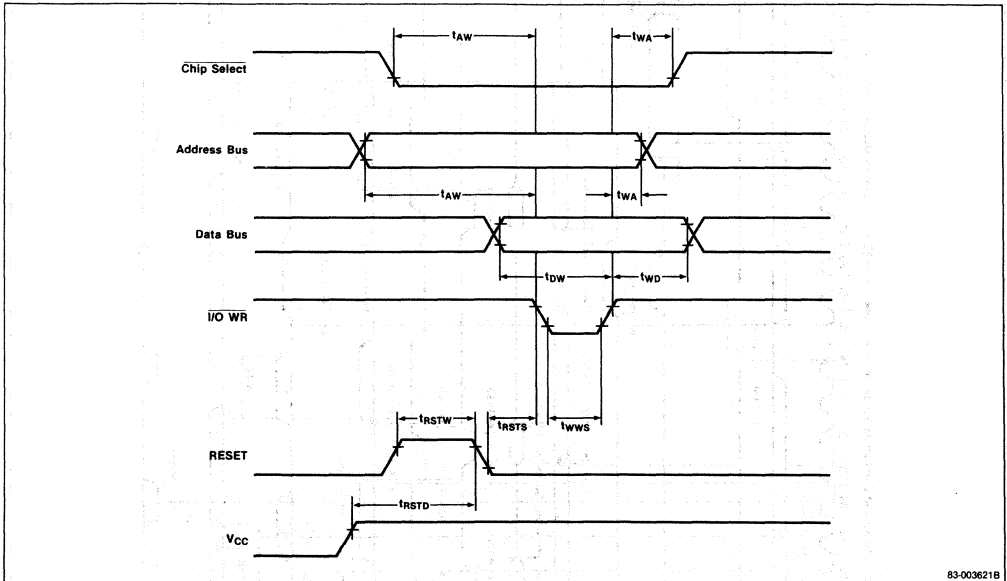
- (1) Load = 1 TTL
- (2) Load = 50 pF
- (3) Load = V<sub>OH</sub> = 3.3 V
- (4) Tracking specification
- (5) Δt<sub>AK</sub> ≤ 50 ns
- (6) Δt<sub>DCL</sub> ≤ 50 ns
- (7) Δt<sub>DCT</sub> ≤ 50 ns

## Timing Waveforms

### Read Timing

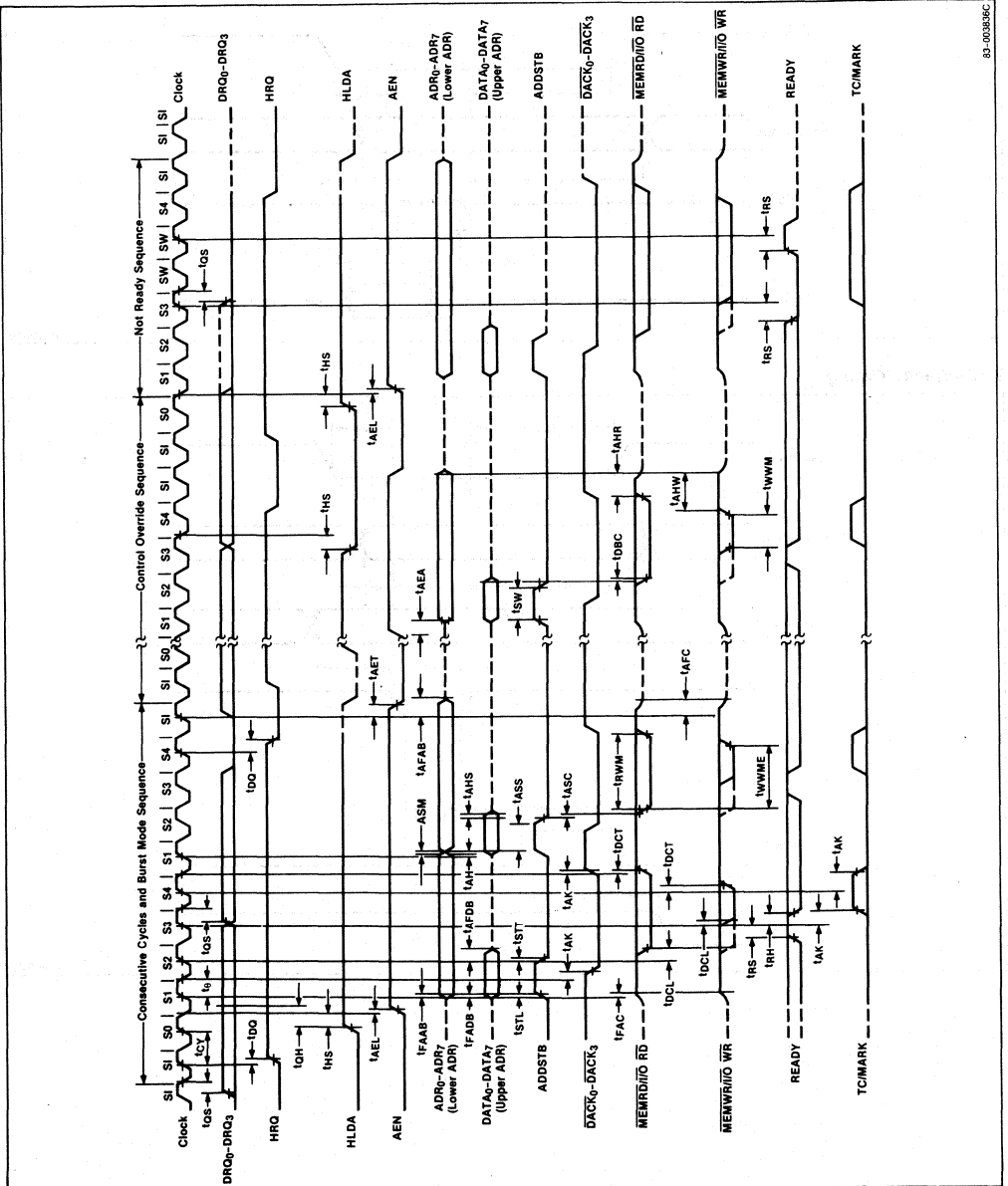


### Write/Reset Timing



Timing Waveforms (cont)

DMA (Master) Mode



83-00336C

## Functional Description

The μPD8257 is a programmable, direct memory Access (DMA) device. When used with an 8212 I/O port device, it provides a complete four-channel DMA controller for use in 8080A/8085A based systems. Once initialized by an 8080A/8085A CPU, the μPD8257 will block transfer up to 16,364 bytes of data between memory and a peripheral device without any attention from the CPU. It will do this on all 4-DMA channels. After receiving a DMA transfer request from a peripheral, the following sequence of events occurs within the μPD8257.

- (1) It acquires control of the system bus (placing 8080A/8085A in hold mode).
- (2) Resolves priority conflicts if multiple DMA requests are made.
- (3) A 16-bit memory address word is generated with the aid of an 8212 in the following manner:
  - (a) The μPD8257 outputs the least significant eight bits ( $A_0$ - $A_7$ ) which go directly onto the address bus.
  - (b) The μPD8257 outputs the most significant eight bits ( $A_8$ - $A_{15}$ ) onto the data bus where they are latched into an 8212 and then sent to the high order bits on the address bus.
- (4) The appropriate memory and I/O read/write control signals are generated allowing the peripheral to receive or deposit a data byte directly from or to the appropriate memory location.

Block transfer of data (e.g., a sector of data on a floppy disk) either to or from a peripheral may be accomplished as long as the peripheral maintains its DMA request ( $DRQ_n$ ). The μPD8257 retains control of the system bus as long as  $DRQ_n$  remains high or until the terminal count (TC) is reached. When the terminal count occurs, TC goes high, informing the CPU that the operation is complete.

There are three different modes of operation:

- (1) DMA read, which causes data to be transferred from memory to a peripheral;
- (2) DMA write, which causes data to be transferred from a peripheral to memory; and
- (3) DMA verify, which does not actually involve the transfer of data.

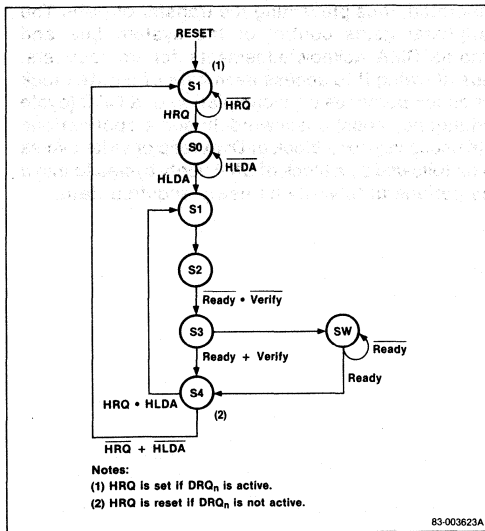
The DMA read and write modes are the normal operating conditions for the μPD8257. The DMA verify mode responds in the same manner as read/write except no memory or I/O read/write control signals are generated, thus preventing the transfer of data. The peripheral gains control of the system bus and obtains DMA acknowledgements for its requests, thus allowing it to access each byte of a data block for check purposes or accumulation of a CRC (cycle redundancy code) checkword. In some applications it is necessary for a block of DMA read or write cycles to be followed by a block of DMA verify cycles to allow the peripheral to verify its newly acquired data.

DMA Operation

As shown in figure 1, internally the μPD8257 contains six different states (S0, S1, S2, S3, S4 and SW). The duration of each state is determined by the input clock. In the idle state, (S1), no DMA operation is being executed. A DMA cycle is started upon receipt of one or more DMA requests (DRQ<sub>n</sub>). Then the μPD8257 enters the S0 state, during which a hold request (HRQ) is sent to the 8080A/8085A and the μPD8257 waits in S0 until the 8080A/8085A issues a hold acknowledge (HLDA) back. During S0, DMA requests are sampled and DMA priority is resolved (based upon either the fixed or priority scheme).

After receipt of HLDA, the DMA acknowledge line (DACK<sub>n</sub>) with the highest priority is driven low, selecting that particular peripheral for the DMA cycle. The DMA request line (DRQ<sub>n</sub>) must remain high until either a DMA acknowledge (DACK<sub>n</sub>) or both DACK<sub>n</sub> and TC (terminal count) occur, indicating the end of a block or sector transfer (burst model).

Figure 1. DMA Operation State Diagram



The DMA cycle consists of four internal states; S1, S2, S3, and S4. If the access time of the memory or I/O device is not fast enough to return a ready command to the μPD8257 after it reaches state S3, then a wait state is initiated (SW). One or more than one wait state occurs until a ready signal is received, and the μPD8257 is allowed to go into state S4. Either the extended write option or the DMA verify mode may eliminate any wait state.

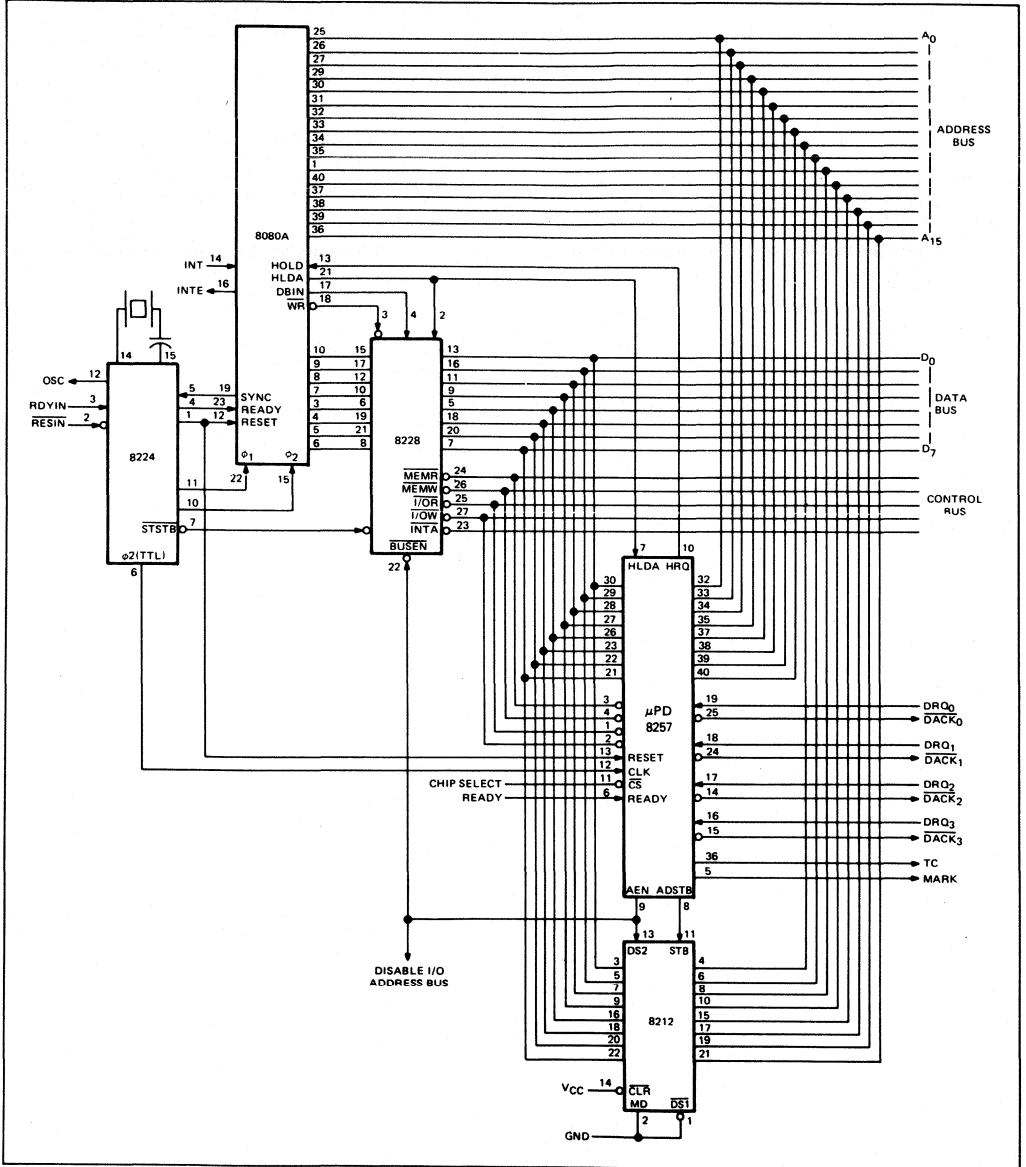
If the μPD8257 should lose control of the system bus, (i.e., HLDA goes low) then the current DMA cycle is completed; the device goes into the S1 state, and no more DMA cycles occur until the bus is reacquired. Ready setup time (t<sub>RS</sub>), write setup time (t<sub>DW</sub>), read data access time (t<sub>RD</sub>), and HLDA setup time (t<sub>QS</sub>) should all be carefully observed during the handshaking mode between the μPD8257 and the 8080A/8085A.

During DMA write cycles, the I/O Read (I/O R) output is generated at the beginning of state S2 and the memory write (MEMW) output is generated at the beginning of S3. During DMA read cycles, the memory read (MEMR) output is generated at the beginning of state S2 and the I/O write (I/O W) goes low at the beginning of state S3. No read or write control signals are generated during DMA verify cycles.

System Interface

Figure 2 is the schematic diagram of a μPD8257 system interface with the 8080A CPU, 8212 I/O Port, 8224 Clock Generator, and 8228 System Controller and Bus Driver.

Figure 2. Typical μPD8257 System Interface Schematic





## Description

The μPD8259A is a programmable interrupt controller directly compatible with the 8080A/8085A/8086/8088 microprocessors. It can service eight levels of interrupts and contains on-chip logic to expand interrupt capabilities up to 64 levels with the addition of other μPD8259A's. The user can choose a selection of priority algorithms to tailor the priority processing to meet his system requirements. These algorithms can be dynamically modified during operation, which expands the versatility of the system. The μPD8259A is completely upward compatible with the μPD8259-5, allowing software written for the μPD8259-5 to run on the μPD8259A/-2.

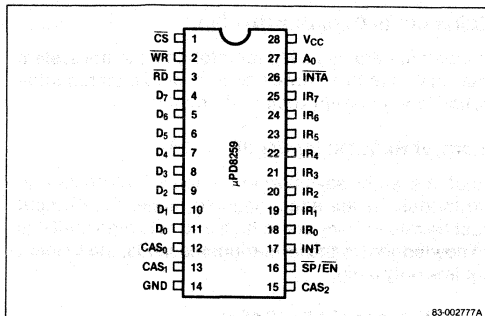
## Features

- Eight-level priority controller
- Programmable base vector address
- Expandable to 64 levels
- Programmable interrupt modes (algorithms)
- Individual request mask capability
- Single +5 V power supply (no clocks)
- Full compatibility with 8080A/8085A/8086/8088

## Ordering Information

Part Number	Package Type
μPD8259AC	28-pin plastic DIP
μPD8259AC-2	28-pin plastic DIP

## Pin Configuration



## Pin Identification

No.	Symbol	Function
1	CS	Chip select input
2	WR	Write input
3	RD	Read input
4-11	D <sub>7</sub> -D <sub>0</sub>	Bidirectional data bus
12, 13, 15	CAS <sub>0</sub> -CAS <sub>2</sub>	Cascade lines
14	GND	Ground
16	SP/EN	Slave program input / enable buffer output
17	INT	Interrupt output
18-25	IR <sub>0</sub> -IR <sub>7</sub>	Interrupt request inputs
26	INTA	Interrupt acknowledge input
27	A <sub>0</sub>	Command select address input
28	V <sub>CC</sub>	+5 V power supply



## Pin Functions

### Bidirectional Data Bus (D7-D0)

Three-state data bus used for interfacing to the system data bus. This bus carries control words, status information, and interrupt vector information.

### Interrupt Request Inputs (IR<sub>0</sub>-IR<sub>7</sub>)

These are eight asynchronous inputs that operate in two modes. In the edge-triggered mode, the IR input must be raised from low to high and held high until it is acknowledged. In the level-triggered mode, the IR input requires only a high.

### Cascade Lines (CAS<sub>0</sub>-CAS<sub>2</sub>)

These lines are used as a bus which controls multiple  $\mu$ PD8259As in a master/slave configuration. When an  $\mu$ PD8259A is a master, these lines are outputs. When a  $\mu$ PD8259A is used as a slave, the lines are inputs.

### Chip Select ( $\overline{CS}$ )

When  $\overline{CS}$  is low, the CPU can read and write to the  $\mu$ PD8259A. The INTA input operates independently of  $\overline{CS}$ .

### Command Select Address Input (A<sub>0</sub>)

The  $\mu$ PD8259A uses this input with  $\overline{CS}$  and  $\overline{WR}$  to decode command words written by the CPU. A<sub>0</sub> is used with  $\overline{CS}$  and  $\overline{RD}$  to decode controller status information for the CPU to read. Typically, A<sub>0</sub> is connected to the A<sub>0</sub> address lines on the CPU.

### Interrupt (INT)

When the  $\mu$ PD8259A receives a valid interrupt request, the INT output goes high to interrupt the CPU. This pin should be connected directly to the interrupt pin on the CPU.

### Interrupt Acknowledge ( $\overline{INTA}$ )

This input line goes active low to indicate that the CPU has received an interrupt request from the  $\mu$ PD8259A.  $\overline{INTA}$  enables interrupt vector data onto the data bus.

### Read Input ( $\overline{RD}$ )

When both  $\overline{RD}$  and  $\overline{CS}$  are low, the  $\mu$ PD8259A sends its status information to the data bus so the CPU can read it.

### Write Input ( $\overline{WR}$ )

The  $\mu$ PD8259A can receive command words from the CPU when both  $\overline{WR}$  and  $\overline{CS}$  are low.

### Slave Program Input/Enable Buffer Output ( $\overline{SP/EN}$ )

This is a dual function pin. In the buffered mode, the enable buffer output is used to enable the buffer transceivers. In the non-buffered mode, when the  $\overline{SP}$  input is high, the  $\mu$ PD8259A operates as a master and when the  $\overline{SP}$  input is low, the  $\mu$ PD8259A operates as a slave.

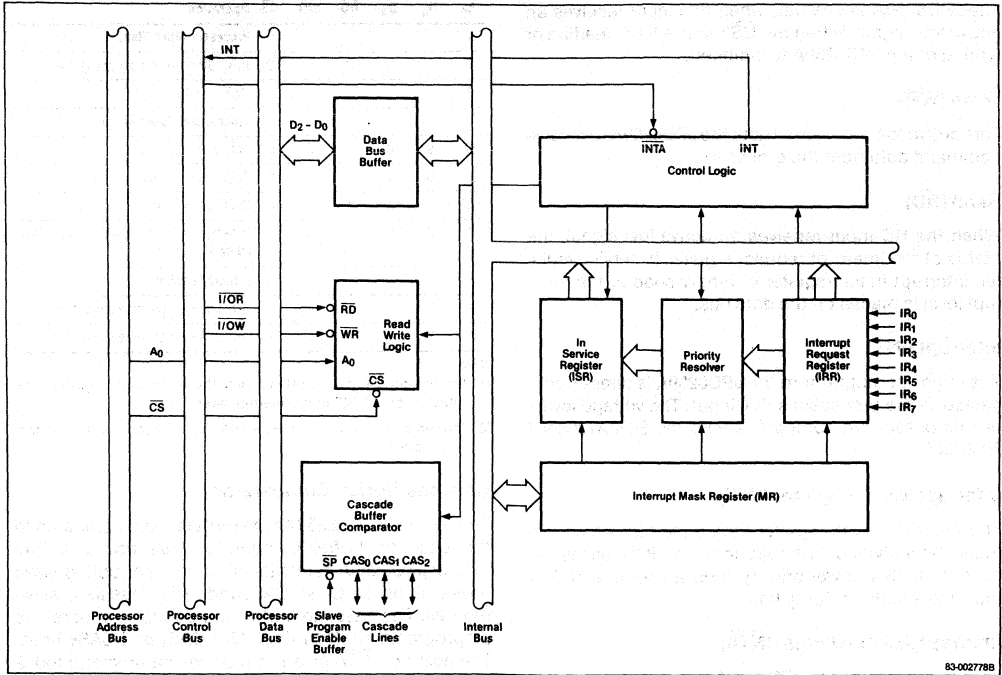
### Ground (GND)

Ground

### Power Supply (V<sub>CC</sub>)

Power supply input, +5 volts.

## Block Diagram



## Block Diagram Description

### Interrupt Request Register (IRR) and In-Service Register (ISR)

The interrupt request and in-service registers store the incoming interrupt request signals appearing on the IR<sub>0</sub>-IR<sub>7</sub> lines. The inputs requesting service are stored in the IRR while the interrupts actually being serviced are stored in the ISR. Refer to functional block diagram.

A positive transition on an IR input sets the corresponding bit in the interrupt request register. At the same time, the INT output of the μPD8259A is set high. The IR input line must remain high until the first INTA input has been received. Multiple non-masked interrupts occurring simultaneously can be stored in the IRR. The incoming INTA sets the appropriate ISR bit, which is determined by the programmed interrupt algorithm, and resets the corresponding IRR bit. The ISR bit stays active high during the interrupt service subroutine until it is reset by the programmed end of interrupt command (EOI).

### Priority Resolver

The priority resolver decides the priority of the interrupt levels in the IRR. When the highest priority interrupt is determined, it is loaded into the appropriate bit of the ISR by the first INTA pulse.

### Data Bus Buffer

The three state 8-bit bidirectional data bus buffer interfaces the μPD8259A to the systems data bus. It buffers the control word and status information being transferred between the μPD8259A and the processor.

### Read/Write Logic

The read/write logic accepts processor commands and stores them in its initialization command word (ICW) and operation command word (OCW) registers. This logic also controls the transfer of status information to the processor.

**Chip Select ( $\overline{CS}$ )**

The μPD8259A is enabled when this input receives an active low signal. When the  $\overline{CS}$  input is high, reading or writing of the μPD8259A is inhibited.

**Write ( $\overline{WR}$ )**

This active low signal instructs the μPD8259A to receive command data from the processor.

**Read ( $\overline{RD}$ )**

When the  $\overline{RD}$  input receives an active low signal, the status of the interrupt request register, in-service register, interrupt mask register or binary code of the interrupt level is placed on the data bus.

**Interrupt (INT)**

The interrupt output from the μPD8259A is directly connected to the processor's INT input. The voltage levels of this output are compatible with the 8080A/8085A/8086/8088.

**Interrupt Mask Register (IMR)**

The interrupt mask register stores the bits which will mask the individual interrupt lines. The IMR masks the data in the ISR. Lower priority lines are not affected by masking a higher priority line.

**Interrupt Acknowledge ( $\overline{INTA}$ )**

$\overline{INTA}$  pulses cause the μPD8259A to put vectoring information on the bus. The number of pulses depend upon whether the μPD8259A is in the μPD8085A mode or 8086/8088 mode.

**Command Select Address Input ( $A_0$ )**

$A_0$  is usually connected to the processor's data bus. Together with  $\overline{RD}$  and  $\overline{WR}$ , it signals the loading of data into the command register or the reading of status data. Table 1 illustrates the basic operations performed. Note that it is divided into three functions: input, output, and bus disable distinguished by the  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{CS}$  inputs.

**Table 1. μPD8259A Basic Operation**

$A_0$	$D_4$	$D_3$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Operation
<b>Processor Input (Read)</b>						
0			0	1	0	IRR, ISR or IR → data bus (Note 1)
1			0	1	0	IMR → data bus
<b>Processor Output (Write)</b>						
0	0	0	1	0	0	Data bus → OCW2
0	0	1	1	0	0	Data bus → OCW3
0	1	X	1	0	0	Data bus → ICW1
1	X	X	1	0	0	Data bus → OCW1, ICW2, ICW3, ICW4 (Note 2)
<b>Disable Function</b>						
X	X	X	1	1	0	Data bus → high impedance state
X	X	X	X	X	1	Data bus → high impedance state

**Note:**

- (1) The contents of OCW3 written prior to the read operation governs the selection of IRR, ISR or the interrupt level.
- (2) The sequencer logic on the μPD8259A aligns these commands in the proper order.

**Cascade Buffer/Comparator**

The IDs of all μPD8259As are buffered and compared in the cascade buffer/comparator. See figure 4. The master μPD8259A sends the ID of the interrupting slave device along the  $CAS_0$ ,  $CAS_1$  and  $CAS_2$  lines to all slave devices. The cascade buffer/comparator compares its preprogrammed ID to the  $CAS_0$ ,  $CAS_1$  and  $CAS_2$  lines. The next two  $\overline{INTA}$  pulses strobe the preprogrammed, 2 byte call routine address onto the data bus from the slave whose ID matches the code on the  $CAS_0$ ,  $CAS_1$  and  $CAS_2$  lines.

**Slave Program ( $\overline{SP}$ )**

The interrupt capability can be expanded to 64 levels by cascading multiple μPD8259As in a master plus slaves array. See figure 4. The master controls the slaves through the  $CAS_0$ ,  $CAS_1$  and  $CAS_2$  lines. The  $\overline{SP}$  input to the device selects the  $CAS_0$ ,  $CAS_1$  and  $CAS_2$  lines as either outputs ( $\overline{SP} = 1$ ) for the master or as inputs ( $\overline{SP} = 0$ ) for the slaves. If only one μPD8259A is used, the  $\overline{SP}$  input must be set to a logic 1, since it is functioning as a master.

## Absolute Maximum Ratings

T<sub>A</sub> = 25°C

Power supply voltage, V <sub>CC</sub>	-0.5 to +7.0 V (Note 1)
Input voltage, V <sub>I</sub>	-1.0 V to V <sub>CC</sub> + 1.0 V
Output voltage, V <sub>O</sub>	-0.5 V to V <sub>CC</sub> + 0.5 V
Operating temperature, T <sub>OPT</sub>	0 to +70°C
Storage temperature, T <sub>STG</sub>	-65 to +150°C
Power dissipation, P <sub>D</sub>	1.0 W

### Note:

(1) With respect to ground.

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## DC Characteristics

T<sub>A</sub> = 0 to +70°C, V<sub>CC</sub> = +5 V ± 10%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage low	V <sub>IL</sub>	-0.5		0.8	V	
Input voltage high	V <sub>IH</sub>	2.0		V <sub>CC</sub> + 0.5	V	
Output voltage low	V <sub>OL</sub>			0.45	V	I <sub>OL</sub> = 2.2 mA
Output voltage high	V <sub>OH</sub>	2.4			V	I <sub>OH</sub> = -400 μA
Interrupt output	V <sub>OH-INT</sub>	2.4			V	I <sub>OH</sub> = -400 μA
High voltage		3.5			V	I <sub>OH</sub> = -100 μA
Input leakage current (Note 1)	I <sub>LI</sub>	-10		10	μA	0 V ≤ V <sub>I</sub> ≤ V <sub>CC</sub>
Output leakage current	I <sub>LO</sub>	-10		10	μA	0.45 V ≤ V <sub>O</sub> ≤ V <sub>CC</sub>
V <sub>CC</sub> power supply current	I <sub>CC</sub>			85	mA	

### Note:

(1) For other inputs.

## AC Characteristics

### Timing Requirements

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = +5 V ± 10%

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8259A		μPD8259A-2			
		Min	Max	Min	Max		
A0 / CS setup to $\overline{RD}$ / $\overline{INTA}$ ↓	t <sub>AHRL</sub>	0		0		ns	
A0 / CS hold after $\overline{RD}$ / $\overline{INTA}$ ↑	t <sub>RHAX</sub>	0		0		ns	
$\overline{RD}$ pulse width	t <sub>RLRH</sub>	235		160		ns	
A0 / CS setup to $\overline{WR}$ ↓	t <sub>AHWL</sub>	0		0		ns	
A0 / CS hold after $\overline{WR}$ ↑	t <sub>RHAX</sub>	0		0		ns	
$\overline{WR}$ pulse width	t <sub>WLWH</sub>	290		190		ns	
Data setup to $\overline{WR}$ ↑	t <sub>DVWH</sub>	240		160		ns	
Data hold after $\overline{WR}$ ↑	t <sub>WHDX</sub>	0		0		ns	
Interrupt request width low	t <sub>JLJH</sub>	100		100		ns	(Note 1)
Cascade setup to second or third $\overline{INTA}$ ↓ (slave only)	t <sub>CVIAL</sub>	55		40		ns	
End of $\overline{RD}$ to next command	t <sub>RHRL</sub>	160		160		ns	
End of $\overline{WR}$ to next command	t <sub>WHRL</sub>	190		190		ns	
End of command to next command (different type)	t <sub>CHCL</sub>	500		500		ns	(Note 2)
End of $\overline{INTA}$ sequence to next $\overline{INTA}$ sequence	t <sub>CHCL</sub>	500		500		ns	(Note 2)

### Note:

(1) This is the low time required to clear the input latch in the edge-triggered mode.

(2) Worst case timing for t<sub>CHCL</sub> in an actual microprocessor system is typically much greater than 500 ns (8085A = 1 μs, 8085-2 = 1 μs, 8086 = 1 μs, 8086-2 = 625 ns).

**AC Characteristics (cont)**

**Timing Responses**

$T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = +5\text{V} \pm 10\%$

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8259A		μPD8259A-2			
		Min	Max	Min	Max		
Data valid from $\overline{\text{RD}}/\overline{\text{INTA}} \downarrow$	$t_{\text{RLDV}}$		200		120	ns	(Notes 1-5)
Data float after $\overline{\text{RD}}/\overline{\text{INTA}} \uparrow$	$t_{\text{RHZ}}$	10	100	10	85	ns	(Notes 1-5)
Interrupt output delay	$t_{\text{JHIH}}$		350		300	ns	(Notes 1-5)
Cascade valid from first $\overline{\text{INTA}} \downarrow$ (master only)	$t_{\text{IALCV}}$		565		360	ns	(Notes 1-5)
Enable active from $\overline{\text{RD}} \uparrow$ or $\overline{\text{INTA}} \uparrow$	$t_{\text{RLEL}}$		125		100	ns	(Notes 1-5)
Enable inactive from $\overline{\text{RD}} \uparrow$ or $\overline{\text{INTA}} \uparrow$	$t_{\text{RHEH}}$		150		150	ns	(Notes 1-5)
Data valid from stable address	$t_{\text{AHDV}}$		200		200	ns	(Notes 1-5)
Cascade valid to valid data	$t_{\text{CVDV}}$		300		200	ns	(Notes 1-5)

**Note:**

- (1) C of data bus = 100 pF
- (2) Max test C = 100 pF
- (3) Min test C = 15 pF
- (4)  $C_{\text{INT}} = 100$  pF
- (5)  $C_{\text{CASCADE}} = 100$  pF

**Capacitance**

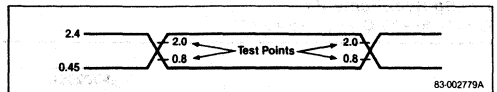
$T_A = 25^\circ\text{C}$ ,  $V_{CC} = \text{GND} = 0\text{V}$ ,  $f_c = 1.0\text{MHz}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	$C_i$			10	pF	(Note 1)
I/O capacitance	$C_{i/I/O}$			20	pF	(Note 1)

**Note:**

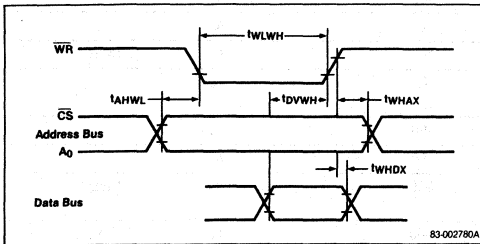
- (1) Unmeasured pins returned to  $V_{SS}$

**AC Test Input**

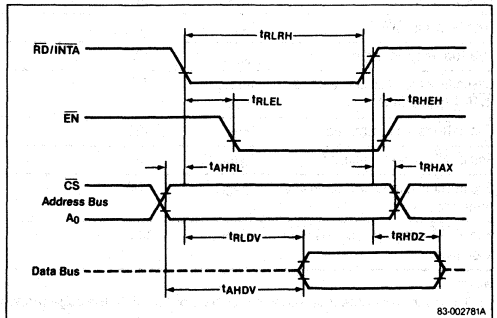


**Timing Waveforms**

**Write Mode**



**Read/INTA Mode**





## $\mu$ PD8259A

### Functional Description

The  $\mu$ PD8259A functions are described in following paragraphs under these major headings:

- Interrupt Sequence
- 8080/8085A Mode
- 8086/8088 Mode
- Initialization Command Words
- Operational Command Words
- Reading  $\mu$ PD8259A Status

### Interrupt Sequence

The  $\mu$ PD8259A derives its versatility from programmable interrupt modes and the ability to jump to any memory address through programmable CALL instructions.

The sequence used by the  $\mu$ PD8259A to handle an interrupt depends upon whether an 8080A/8085A or 8086/8088 CPU is being used.

The following sequence demonstrates how the  $\mu$ PD8259A interacts with the 8080A/8085A systems.

- (1) An interrupt(s) appearing on IR<sub>0</sub>-IR<sub>7</sub> sets the corresponding IR bit(s) high. This in turn sets the corresponding IRR bit(s) high.
- (2) Once the IRR bit(s) has been set, the  $\mu$ PD8259A will resolve priorities according to the preprogrammed interrupt algorithm. It then issues an INT signal to the processor.
- (3) When the processor receives an INT, it issues an  $\overline{\text{INTA}}$  to the  $\mu$ PD8259A.
- (4) The  $\overline{\text{INTA}}$  input to the  $\mu$ PD8259A from the processor group sets the highest priority ISR bit and resets the corresponding IRR bit. The  $\overline{\text{INTA}}$  also signals the  $\mu$ PD8259A to place an 8-bit CALL instruction opcode (11001101) onto its data bus lines.
- (5) The CALL instruction code instructs the processor group to issue two more  $\overline{\text{INTA}}$  pulses to the  $\mu$ PD8259A.

(6) The two  $\overline{\text{INTA}}$  pulses signal the  $\mu$ PD8259A to place its preprogrammed interrupt vector address onto the data bus. The first  $\overline{\text{INTA}}$  releases the low order 8 bits of the address and the second  $\overline{\text{INTA}}$  releases the high order 8 bits.

(7) The  $\mu$ PD8259As CALL instruction sequence is complete. A preprogrammed EOI command is issued to the  $\mu$ PD8259A at the end of the interrupt service routine. This resets the ISR bit and allows the  $\mu$ PD8259A to service the next interrupt.

The following sequence demonstrates how the  $\mu$ PD8259A interacts with the 8086/8088 systems.

- (1), (2), (3) Same as for 8080A/8085A.
- (4) During the first  $\overline{\text{INTA}}$  from the processor, the  $\mu$ PD8259A does not drive the data bus. The highest priority ISR bit is set and the corresponding IRR bit is reset.
- (5) The  $\mu$ PD8259A puts vector information onto the data bus on the second  $\overline{\text{INTA}}$  pulse from the 8086/8088.
- (6) There is no third  $\overline{\text{INTA}}$  pulse in this mode. In the AEOL mode the ISR bit is reset at the end of the second  $\overline{\text{INTA}}$  pulse, or it remains set until an EOI command is issued.

### 8080/8085A Mode

For these processors, the  $\mu$ PD8259A is controlled by three  $\overline{\text{INTA}}$  pulses. The first  $\overline{\text{INTA}}$  pulse will cause the  $\mu$ PD8259A to put the CALL opcode onto the data bus. See table 2. The second and third  $\overline{\text{INTA}}$  pulses will cause the upper and lower address of the interrupt vector to be released on the bus. See tables 3 and 4.

**Table 2. Contents of First Interrupt Vector Byte**

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	1	0	0	1	1	0	1

**Table 3. Contents of Second Interrupt Vector Byte**

IR	Interval = 4							
	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
7	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	1	1	0	0
6	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	1	0	0	0
5	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	0	1	0	0
4	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	0	0	0	0
3	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	0	1	1	0	0
2	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	0	1	0	0	0
1	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	0	0	1	0	0
0	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	0	0	0	0	0

## 8086/8088 Mode

In this mode only two  $\overline{INTA}$  pulses are sent to the μPD8259A. After the first  $\overline{INTA}$  pulse, the μPD8259A does not output a CALL but internally sets priority resolution. If it is a master, it sets the cascade lines. The interrupt vector is output to the data bus on the second  $\overline{INTA}$  pulse. See table 5.

## Initialization Command Words

### ICW1 and ICW2

**LTIM** If LTIM = 1, then the μPD8259A operates in the level interrupt mode. Edge detect logic on the interrupt inputs is disabled.

**ADI** CALL address interval. If ADI = 1 then the interval is four; if ADI = 0 then the interval is eight.

**SNGL** (Single) Indicates that there is only one μPD8259A in the system. If SNGL = 1, no ICW3 is issued.

**IC4** If this bit is set, ICW4 has to be read. If ICW4 is not needed, set IC4 to logic 0.

**A5-A15** Defines the page starting address of the service routines. In an 8085A system, the eight request levels generate CALLs to eight locations equally spaced in memory. These can be programmed to be spaced at intervals of four or eight memory locations, allowing eight routines to occupy a page of 32 or 64 bytes, respectively.

IR	Interval = 8							
	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
7	A <sub>7</sub>	A <sub>6</sub>	1	1	1	0	0	0
6	A <sub>7</sub>	A <sub>6</sub>	1	1	0	0	0	0
5	A <sub>7</sub>	A <sub>6</sub>	1	0	1	0	0	0
4	A <sub>7</sub>	A <sub>6</sub>	1	0	0	0	0	0
3	A <sub>7</sub>	A <sub>6</sub>	0	1	1	0	0	0
2	A <sub>7</sub>	A <sub>6</sub>	0	1	0	0	0	0
1	A <sub>7</sub>	A <sub>6</sub>	0	0	1	0	0	0
0	A <sub>7</sub>	A <sub>6</sub>	0	0	0	0	0	0

**Table 4. Contents of Third Interrupt Vector Byte**

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>

**Table 5. Contents of Interrupt Vector Byte, 8086/8088 Mode**

IR	Interval = 4							
	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
7	T <sub>7</sub>	T <sub>6</sub>	T <sub>5</sub>	T <sub>4</sub>	T <sub>3</sub>	1	1	1
6	T <sub>7</sub>	T <sub>6</sub>	T <sub>5</sub>	T <sub>4</sub>	T <sub>3</sub>	1	1	0
5	T <sub>7</sub>	T <sub>6</sub>	T <sub>5</sub>	T <sub>4</sub>	T <sub>3</sub>	1	0	1
4	T <sub>7</sub>	T <sub>6</sub>	T <sub>5</sub>	T <sub>4</sub>	T <sub>3</sub>	1	0	0
3	T <sub>7</sub>	T <sub>6</sub>	T <sub>5</sub>	T <sub>4</sub>	T <sub>3</sub>	0	1	1
2	T <sub>7</sub>	T <sub>6</sub>	T <sub>5</sub>	T <sub>4</sub>	T <sub>3</sub>	0	1	0
1	T <sub>7</sub>	T <sub>6</sub>	T <sub>5</sub>	T <sub>4</sub>	T <sub>3</sub>	0	0	1
0	T <sub>7</sub>	T <sub>6</sub>	T <sub>5</sub>	T <sub>4</sub>	T <sub>3</sub>	0	0	0

The address form is two bytes long (A<sub>0</sub>-A<sub>15</sub>). When the routine interval is four, A<sub>0</sub>-A<sub>4</sub> are automatically inserted by the μPD8259A, while A<sub>5</sub>-A<sub>15</sub> are programmed externally. When the routine interval is eight, A<sub>0</sub>-A<sub>5</sub> are automatically inserted by the μPD8259A, while A<sub>6</sub>-A<sub>15</sub> are programmed externally.

The eight-byte interval maintains compatibility with current software, while the four-byte interval is best for a compact jump table.

In an 8086/8088 system, T<sub>7</sub>-T<sub>3</sub> are inserted in the five most significant bits of the vectoring byte. The μPD8259A sets the three least significant bits according to the interrupt level.



**ICW3**

This word is read only when there is more than one μPD8259A in the system and cascading will be used. SNGL of ICW1 is programmed for logic 0. ICW3 will load the 8-bit slave register. The functions of this register are, in the master mode, when  $\overline{SP} = 1$  or  $BUF = 1$  and  $M/S = 1$  in ICW4, a 1 is set for each slave in the system. The master then releases byte 1 of the call sequence (for 8080A/8085A system) and enables the corresponding slave via the cascade lines to release vector bytes 2 and 3 (byte 2 only for 8086/8088).

In the slave mode, when  $\overline{SP} = 0$  or  $BUF = 1$  and  $M/S = 0$  in ICW4, bits  $ID_2-ID_0$  identify the slave. The slave compares its cascade input with these bits and if they are equal, vector bytes 2 and 3 of the call sequence (byte 2 only for 8086/8088) are released by the slave on the data bus.

**ICW4**

**SNFM**

If  $SNFM = 1$ , the special fully nested mode is programmed.

**BUF**

If  $BUF = 1$ , the buffered mode is programmed. In the buffered mode,  $\overline{SP}/\overline{EN}$  becomes an enable output and the master/slave determination is by  $M/S$ .

**M/S**

If the buffered mode is selected,  $M/S = 1$  means the μPD8259A is programmed to be a master,  $M/S = 0$  means the μPD8259A 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:  $\mu PM = 0$  sets the μPD8259A for 8085A system operation;  $\mu PM = 1$  sets the μPD8259A for 8086 system operation.

Figure 1 illustrates the command word initialization sequence.

**Figure 1. Initialization Sequence**

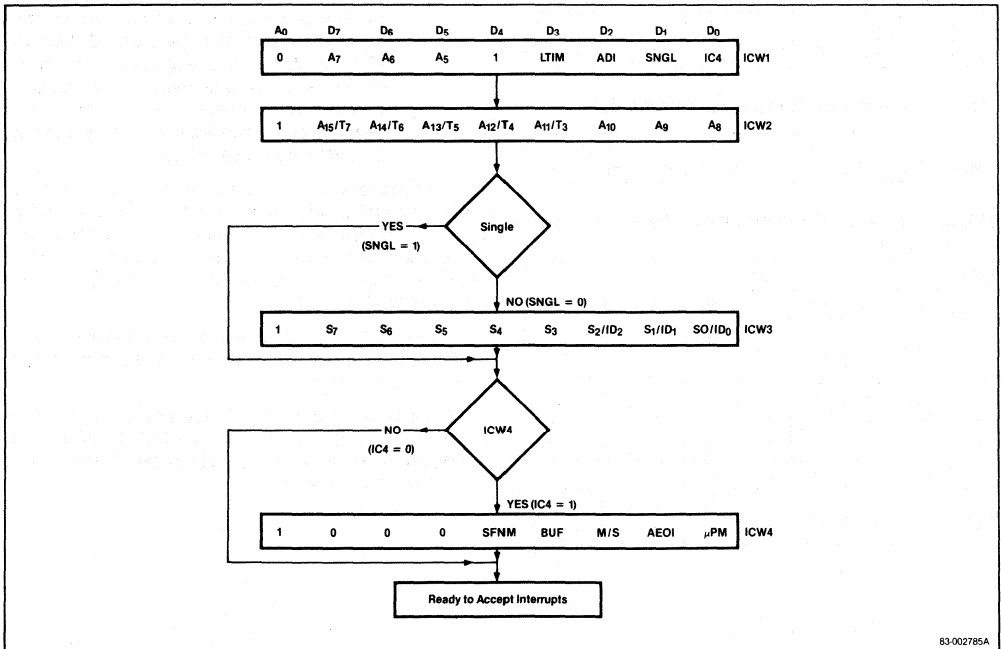
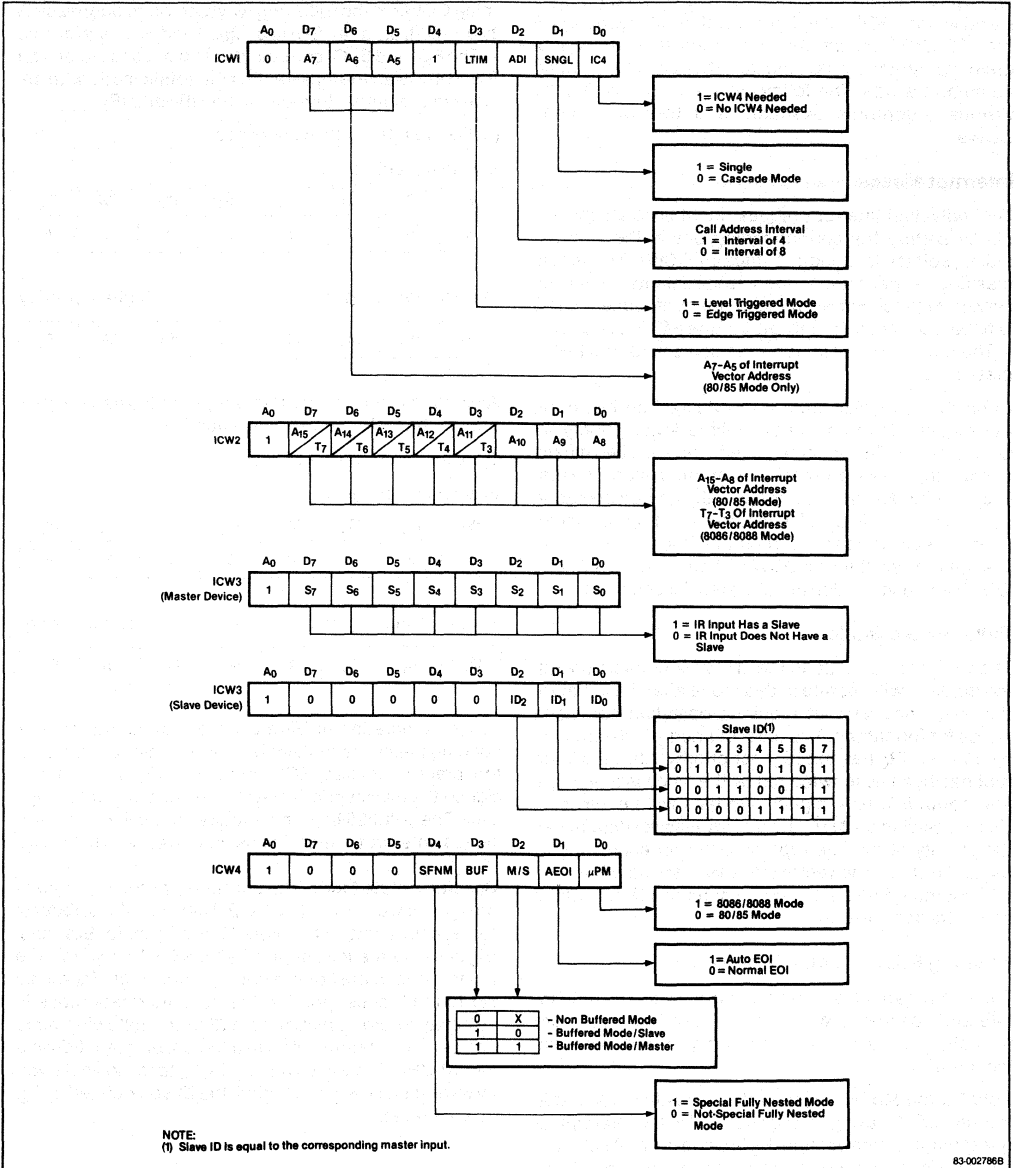


Figure 2 illustrates the initialization command word format.

**Figure 2. Initialization Command Word Format**



**Operational Command Words**

Once the μPD8259A has been programmed with initialization command words, it can be programmed for the appropriate interrupt algorithm by the operation command words (OCW). See figure 3. Interrupt algorithms in the μPD8259A can be changed at any time during program operation by issuing another set of operation command words. The following sections describe the various algorithms available and their associated OCWs.

**Interrupt Masks**

The individual interrupt request input lines are maskable by setting the corresponding bits in the interrupt mask register to a logic 1 through OCW1. The actual masking is performed upon the contents of the in-service register. For example, if interrupt request line 3 is to be masked, then only bit 3 of the IMR is set to logic 1. The IMR in turn acts upon the contents of the ISR to mask bit 3.

Once the μPD8259A has acknowledged an interrupt, the masked interrupt input inhibits lower priority requests from being acknowledged. There are two means of enabling these lower priority interrupt lines. The first is by issuing an end of interrupt (EOI) through operation command word 2 (OCW2), thereby resetting the appropriate ISR bit. The second approach is to select the special mask mode through OCW3. The special mask mode (SMM) and end of interrupt (EOI) are described later.

**Fully Nested Mode**

The fully nested mode is the μPD8259A's basic operating mode. It will operate in this mode after the initialization sequence without requiring operation command words for formatting. The order of priority is determined by IR<sub>0</sub>-IR<sub>7</sub>. IR<sub>0</sub> has the highest priority. After the interrupt has been acknowledged by the processor and system controller, only higher priorities will be serviced. Upon receiving an INTA, the priority resolver determines the priority of the interrupt, sets the corresponding IR bit, and outputs the vector address to the data bus. The EOI command resets the corresponding ISR bits at the end of its service routines.

**Rotating Priority Mode Commands**

The two variations of rotating priorities are the auto rotate and specific rotate modes. These two modes are typically used to service interrupting devices of equivalent priorities.

**Auto Rotate Mode.** Programming the auto rotate mode through OCW2 assigns priorities 0-7 to the interrupt request inputs. Interrupt line IR<sub>0</sub> is set to the highest priority and IR<sub>7</sub> to the lowest. Once an interrupt has been

serviced, it is automatically assigned the lowest priority. That same input must then wait for the devices ahead of it to be serviced before it can be acknowledged again. The auto rotate mode is selected by programming OCW2 in the following way: set rotate priority bit R to a logic 1, program EOI to a logic 1 and SECOI to a logic 0. The EOI and SEOI commands are discussed later. The following is an example of the auto rotate mode with devices requesting interrupts on line IR<sub>2</sub> and IR<sub>5</sub>.

(1) Before interrupts are serviced:

In-service register

IS <sub>7</sub>	IS <sub>6</sub>	IS <sub>5</sub>	IS <sub>4</sub>	IS <sub>3</sub>	IS <sub>2</sub>	IS <sub>1</sub>	IS <sub>0</sub>
0	0	1	0	0	1	0	0

Priority status register

highest priority

IR <sub>7</sub>	IR <sub>6</sub>	IR <sub>5</sub>	IR <sub>4</sub>	IR <sub>3</sub>	IR <sub>2</sub>	IR <sub>1</sub>	IR <sub>0</sub>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

According to the priority status register, IR<sub>2</sub> has a higher priority than IR<sub>5</sub> and will be serviced first.

(2) After interrupts are serviced:

In-service register

IS <sub>7</sub>	IS <sub>6</sub>	IS <sub>5</sub>	IS <sub>4</sub>	IS <sub>3</sub>	IS <sub>2</sub>	IS <sub>1</sub>	IS <sub>0</sub>
0	0	1	0	0	0	0	0

Priority status register

highest priority

IR <sub>2</sub>	IR <sub>1</sub>	IR <sub>0</sub>	IR <sub>7</sub>	IR <sub>6</sub>	IR <sub>5</sub>	IR <sub>4</sub>	IR <sub>3</sub>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

At the completion of IR<sub>2</sub>'s service routine, the corresponding in-service register bit (IS<sub>2</sub>) is reset to logic 0 by the preprogrammed EOI command. IR<sub>2</sub> is then assigned the lowest priority level in the priority status register. The μPD8259A is now ready to service the next highest interrupt, which, in this case, happens to be IR<sub>5</sub>.

**Specific Rotate Mode.** The priorities are set by programming the lowest level via OCW2. Then, the μPD8259A automatically assigns the highest priority. If, for example, IR<sub>3</sub> is set to the lowest priority (bits L<sub>2</sub>, L<sub>1</sub>, L<sub>0</sub> form the binary code of the bottom priority level), then IR<sub>4</sub> will be set to the highest priority. The specific rotate mode is selected by programming OCW2 in the following manner: set rotate priority bit R to a logic 1, program EOI to a logic 0, SEOI to a logic 1 and L<sub>2</sub>, L<sub>1</sub>, L<sub>0</sub> to the lowest priority level. If EOI is set to a logic 1, the ISR bit defined by L<sub>2</sub>, L<sub>1</sub>, L<sub>0</sub> is reset.

## End of Interrupt (EOI) and Specific End of Interrupt (SEOI)

The end of interrupt (EOI) or specific end of interrupt (SEOI) command must be issued to reset the appropriate in-service register bit before the completion of a service routine. Once the ISR bit has been reset to logic 0, the μPD8259A is ready to service the next interrupt.

Two types of EOI's are available to clear the appropriate ISR bit depending on the μPD8259A's operating mode.

**Non-Specific End of Interrupt (EOI).** When operating in interrupt modes where the priority order of the interrupt inputs is preserved, such as the fully nested mode, the particular ISR bit to be reset at the completion of the service routine can be determined. A non-specific EOI command automatically resets the highest priority ISR bit of those set. The highest priority ISR bit must necessarily be the interrupt being serviced and must necessarily be the service subroutine returned from.

**Specific End of Interrupt (SEOI).** When operating in interrupt modes where the priority order of the interrupt inputs is not preserved, such as the rotating priority mode, the last serviced interrupt level may not be known. In these modes, a specific end of interrupt must be issued to clear the ISR bit at the completion of the interrupt service routine. The SEOI is programmed by setting the appropriate bits in OCW2 to logic 1's. See figure 3. Both the EOI and SEOI bits of OCW2 must be set to a logic 1 with L<sub>2</sub>, L<sub>1</sub>, L<sub>0</sub> forming the binary code of the ISR bit to be reset.

## Special Mask Mode

Setting up an interrupt mask through the interrupt mask register by setting the appropriate bits in OCW1 to a logic 1 inhibits lower priority interrupts being acknowledged. In applications requiring that the lower priorities be enabled while the IMR is set, the special mask mode can be used. The SMM is programmed in OCW3 by setting the appropriate bits to a logic 1. Once the SMM is set, the μPD8259A remains in this mode until it is reset. The special mask mode does not affect the higher priority interrupts.

## Poll Mode

In poll mode, the processor must be instructed to disable its interrupt input (INT). Interrupt service is initiated through software by a poll command. Poll mode is programmed by setting the poll mode bit in OCW3 to logic 1 during a WR Pulse. The following RD pulse is then considered as an interrupt acknowledge. If an interrupt input is present, the RD pulse sets the appropriate ISR bit and reads the interrupt priority level. Poll mode is a one time operation and must be programmed through OCW3 before every read. The

word format which is strobed onto the data bus during the poll mode follows:

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
I	X	X	X	X	W <sub>2</sub>	W <sub>1</sub>	W <sub>0</sub>

where:

I = 1 if there is an interrupt requesting service

I = 0 if there are no interrupts

W<sub>2</sub>-W<sub>0</sub> forms the binary code of the highest priority level of the interrupts requesting service.

Poll mode can be used when an interrupt service routine is common to several interrupt inputs. The INTA sequence is no longer required; this saves ROM space. Poll mode can also be used to expand the number of interrupts beyond 64.

## Reading μPD8259A Status

The following major registers' status is available to the processor by appropriately formatting OCW3 and issuing RD command.

## Interrupt Request Register

The 8-bit interrupt request register stores the interrupt levels awaiting acknowledgement. The highest priority in-service bit is reset once it has been acknowledged. Note that the interrupt mask register has no effect on the IRR. Prior to the issuing of the RD command, a WR command must be issued with OCW3. Programmable logic bits RIS and ERIS of OCW3 determine whether the IRR or ISR register is to be read. To read the contents of the IRR, ERIS must be a logic 1, and RIS a logic 0.

## In-Service Register

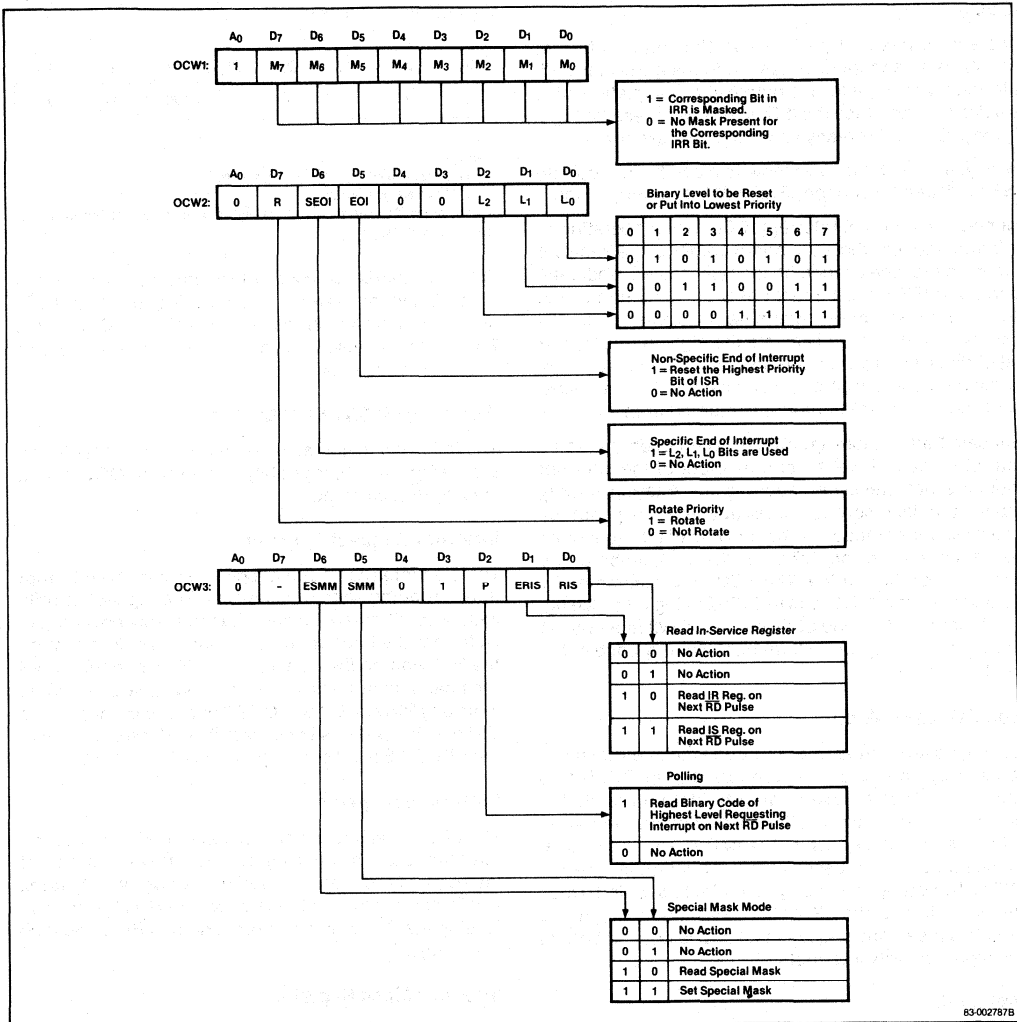
The 8-bit in-service register stores the priorities of the interrupt levels being serviced. Assertion of an end of interrupt (EOI) updates the ISR to the next priority level. A WR command must be issued with OCW3 prior to issuing the RD command. both ERIS and RIS should be set to logic 1.

## Interrupt Mask Register

The 8-bit interrupt mask register holds mask data modifying interrupt levels. A WR pulse preceding the RD is not necessary to read the IMR status. The IMR data is available to the data bus when RD is asserted with A<sub>0</sub> at logic 1.

A single OCW3 is sufficient to enable successive status reads providing it is of the same register. A status read is overridden by the poll mode when bits P and ERIS of OCW3 are set to logic 1.

Figure 3. Operation Command Word Format

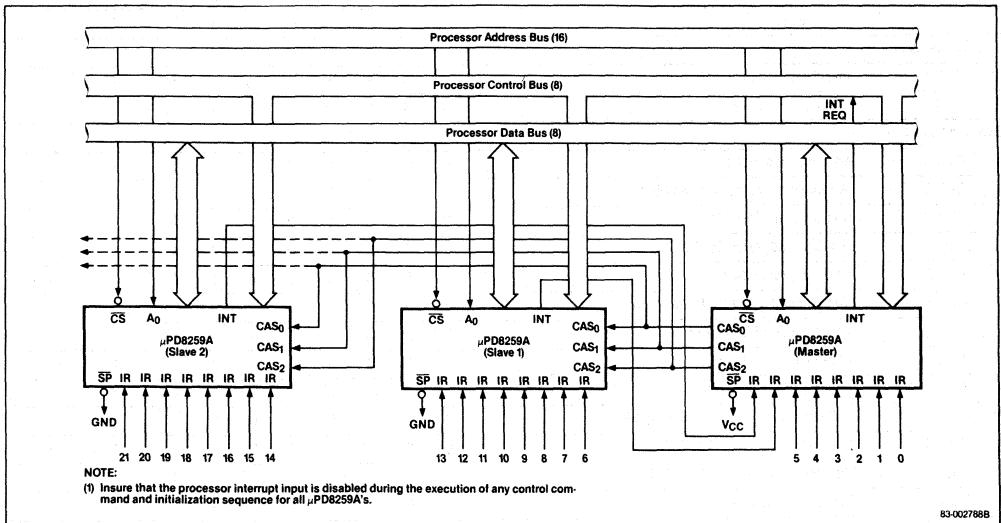


**Table 6. Summary of Operation Command Word Programming**

	A <sub>0</sub>	D <sub>4</sub>	D <sub>3</sub>				
OCW1	1	X	X	M <sub>7</sub> -M <sub>0</sub>	IMR (interrupt mask register) WR loads IMR data while RD reads status		
OCW2	0	0	0	R	SEOI	EOI	
	0	0	0	0	0	0	No action
	0	0	0	0	0	1	Non-specific end of interrupt
	0	0	0	0	1	0	No action
	0	0	0	0	1	1	Specific end of interrupt L <sub>2</sub> , L <sub>1</sub> , L <sub>0</sub> forms binary representation of level to be reset
	1	0	0	0	0	0	No action
	1	0	0	1	0	1	Rotate priority at end of interrupt (auto mode)
	1	1	0	0	0	0	Rotate priority, L <sub>2</sub> , L <sub>1</sub> , L <sub>0</sub> specifies bottom priority without end of interrupt
	1	1	1	0	0	1	Rotate priority at end of interrupt (specific mode). L <sub>2</sub> , L <sub>1</sub> , L <sub>0</sub> specifies bottom priority, and it is in-service register bit is reset.

	A <sub>0</sub>	D <sub>4</sub>	D <sub>3</sub>			
OCW3	0	0	1	ESMM	SMM	
	0	0	0	0	0	Special mask not affected
	0	0	0	0	1	Special mask not affected
	0	0	0	1	0	Reset special mask
	0	0	0	1	1	Set special mask
				ERIS	RIS	
	0	0	0	0	0	No action
	0	0	0	0	1	No action
	0	0	0	1	0	Read IR register status
	0	0	0	1	1	Read IS register status

**Figure 4. Cascading the μPD8259A**



**Instruction Set**

#	Mnemonic	Operation Description	Operation Code								A <sub>0</sub>	Format
			D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>		
<b>(Byte 1 Initialization, No ICW4 Required)</b>												
1	ICW1 A	Single, edge triggered	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	0	1	1	0	0	4
2	ICW1 B	Single, level triggered	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	1	1	1	0	0	4
3	ICW1 C	Not single, edge triggered	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	0	1	0	0	0	4
4	ICW1 D	Not single, level triggered	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	1	1	0	0	0	4
5	ICW1 E	Single, edge triggered	A <sub>7</sub>	A <sub>6</sub>	0	1	0	0	1	0	0	8
6	ICW1 F	Single, level triggered	A <sub>7</sub>	A <sub>6</sub>	0	1	1	0	1	0	0	8
7	ICW1 G	Not single, edge triggered	A <sub>7</sub>	A <sub>6</sub>	0	1	0	0	0	0	0	8
8	ICW1 H	Not single, level triggered	A <sub>7</sub>	A <sub>6</sub>	0	1	1	0	0	0	0	8
<b>(Byte 1 Initialization, ICW4 Required)</b>												
9	ICW1 I	Single, edge triggered	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	0	1	1	1	0	4
10	ICW1 J	Single, level triggered	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	1	1	1	1	0	4
11	ICW1 K	Not single, edge triggered	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	0	1	0	1	0	4
12	ICW1 L	Not single, level triggered	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	1	1	1	0	1	0	4
13	ICW1 M	Single, edge triggered	A <sub>7</sub>	A <sub>6</sub>	0	1	0	0	1	1	0	8
14	ICW1 N	Single, level triggered	A <sub>7</sub>	A <sub>6</sub>	0	1	1	0	1	1	0	8
15	ICW1 O	Not single, edge triggered	A <sub>7</sub>	A <sub>6</sub>	0	1	0	0	0	1	0	8
16	ICW1 P	Not single, level triggered	A <sub>7</sub>	A <sub>6</sub>	0	1	1	0	0	1	0	8
<b>(Byte 2 Initialization)</b>												
17	ICW2	Initialize byte 2	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	1	
<b>(Byte 3 Initialization)</b>												
18	ICW3 M	Initialize byte 3 (master)	S <sub>7</sub>	S <sub>6</sub>	S <sub>5</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	1	
19	ICW3 S	Initialize byte 3 (slave)	0	0	0	0	0	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	1	
<b>(Byte 4 Initialization)</b>												
20	ICW4 A	No action, redundant	0	0	0	0	0	0	0	0	1	
21	ICW4 B	Non-buffered, no AE01, 8086 / 8088	0	0	0	0	0	0	0	1	1	
22	ICW4 C	Non-buffered, AE01, 80 / 85	0	0	0	0	0	0	1	0	1	
23	ICW4 D	Non-buffered, AE01, 8086 / 8088	0	0	0	0	0	0	1	1	1	
24	ICW4 E	No action, redundant	0	0	0	0	0	1	0	0	1	
25	ICW4 F	Non-buffered, no AE01, 8086 / 8088	0	0	0	0	0	1	0	1	1	
26	ICW4 G	Non-buffered AE01, 80 / 85	0	0	0	0	0	1	1	0	1	
27	ICW4 H	Non-buffered, AE01, 8086 / 8088	0	0	0	0	0	1	1	1	1	
28	ICW4 I	Buffered, slave, no AE01, 80 / 85	0	0	0	0	1	0	0	0	1	
29	ICW4 J	Buffered, slave, no AE01, 8086 / 8088	0	0	0	0	1	0	0	1	1	
30	ICW4 K	Buffered, slave, AE01, 80 / 85	0	0	0	0	1	0	1	0	1	
31	ICW4 L	Buffered, slave, AE01, 8086 / 8088	0	0	0	0	1	0	1	1	1	
32	ICW4 M	Buffered, master, no AE01, 80 / 85	0	0	0	0	1	1	0	0	1	
33	ICW4 N	Buffered, master, no AE01, 8086 / 8088	0	0	0	0	1	1	0	1	1	

## Instruction Set (cont)

#	Mnemonic	Operation Description	Operation Code								A <sub>0</sub>	Format
			D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>		
<b>(Byte 4 Initialization) (cont)</b>												
34	ICW4 O	Buffered, master, AE01, 80 / 85	0	0	0	0	1	1	1	0	1	
35	ICW4 P	Buffered, master, AE01, 8086 / 8088	0	0	0	0	1	1	1	1	1	
36	ICW4 NA	Fully nested, non-buffered, no AE01, 8085A	0	0	0	1	0	0	0	0	1	
37	ICW4 NB	ICW4 NB-ICW4 ND are identical to ICW4 B-ICW4 D with the addition of fully nested mode	0	0	0	1	0	0	0	1	1	
38	ICW4 NC	ICW4 NB-ICW4 ND are identical to ICW4 B-ICW4 D with the addition of fully nested mode	0	0	0	1	0	0	1	0	1	
39	ICW4 ND	ICW4 NB-ICW4 ND are identical to ICW4 B-ICW4 D with the addition of fully nested mode	0	0	0	1	0	0	1	1	1	
40	ICW4 NE	Fully nested, non-buffered, no AE01, 80 / 85	0	0	0	1	0	1	0	0	1	
41	ICW4 NF	ICW4 NF-ICW4 NP are identical to ICW4 F-ICW4 P with the addition of fully nested mode	0	0	0	1	0	1	0	1	1	
42	ICW4 NG	ICW4 NF-ICW4 NP are identical to ICW4 F-ICW4 P with the addition of fully nested mode	0	0	0	1	0	1	1	0	1	
43	ICW4 NH	ICW4 NF-ICW4 NP are identical to ICW4 F-ICW4 P with the addition of fully nested mode	0	0	0	1	0	1	1	1	1	
44	ICW4 NI	ICW4 NF-ICW4 NP are identical to ICW4 F-ICW4 P with the addition of fully nested mode	0	0	0	1	1	0	0	0	1	
45	ICW4 NJ	ICW4 NF-ICW4 NP are identical to ICW4 F-ICW4 P with the addition of fully nested mode	0	0	0	1	1	0	0	1	1	
46	ICW4 NK	ICW4 NF-ICW4 NP are identical to ICW4 F-ICW4 P with the addition of fully nested mode	0	0	0	1	1	0	1	0	1	
47	ICW4 NL	ICW4 NF-ICW4 NP are identical to ICW4 F-ICW4 P with the addition of fully nested mode	0	0	0	1	1	0	1	1	1	
48	ICW4 NM	ICW4 NF-ICW4 NP are identical to ICW4 F-ICW4 P with the addition of fully nested mode	0	0	0	1	1	1	0	0	1	
49	ICW4 NN	ICW4 NF-ICW4 NP are identical to ICW4 F-ICW4 P with the addition of fully nested mode	0	0	0	1	1	1	0	1	1	
50	ICW4 NO	ICW4 NF-ICW4 NP are identical to ICW4 F-ICW4 P with the addition of fully nested mode	0	0	0	1	1	1	1	0	1	
51	ICW4 NP	ICW4 NF-ICW4 NP are identical to ICW4 F-ICW4 P with the addition of fully nested mode	0	0	0	1	1	1	1	1	1	
52	OCW1	Load mask and read mark registers	M <sub>7</sub>	M <sub>6</sub>	M <sub>5</sub>	M <sub>4</sub>	M <sub>3</sub>	M <sub>2</sub>	M <sub>1</sub>	M <sub>0</sub>	1	
53	OCW2 E	Non-specific EOI	0	0	1	0	0	0	0	0	0	
54	OCW2 SE	Specific EOI, L <sub>0</sub> -L <sub>2</sub> code of IS FF to be reset	0	1	1	0	0	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>	0	
55	OCW2 RE	Rotate on non-specific EOI	1	0	1	0	0	0	0	0	0	
56	OCW2 RSE	Rotate on specific EOI L <sub>0</sub> -L <sub>2</sub> code of line	1	1	1	0	0	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>	0	
57	OCW2 R	Rotate in auto EOI (set)	1	0	0	0	0	0	0	0	0	
58	OCW2 CR	Rotate in auto EOI (clear)	0	0	0	0	0	0	0	0	0	
59	OCW2 RS	Set priority command	1	1	0	0	0	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>	0	
60	OCW3 P	Poll mode	0	0	0	0	1	1	0	0	0	
61	OCW3 RIS	Read IS register	0	0	0	0	1	0	1	1	0	





### Description

The μPD8279 is a programmable keyboard and display input/output device providing the user with the ability to display data on alphanumeric segment displays or simple indicators. The display RAM can be programmed to function as a 16 x 8-bit or dual 16 x 4-bit memory and can be loaded or read by the host processor. The display can be loaded with right or left entry with an auto-increment of the display RAM address.

The keyboard interface provides a scanned signal to a 64 contact key matrix expandable to 128. General sensors or strobed keys may also be used. Keystrokes are stored in an 8 character FIFO and can be either 2 key lockout or N key rollover. Keyboard entries generate an interrupt to the processor.

### Features

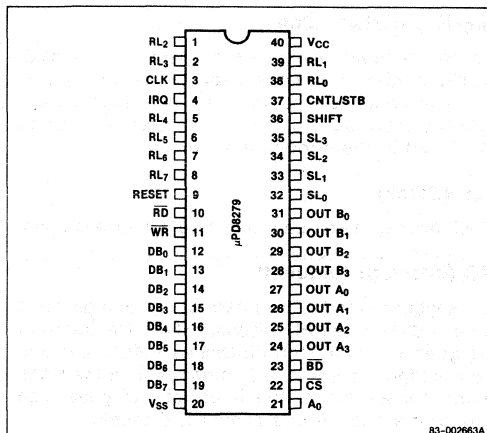
- Programmable by processor
- 32 hex or 16 alphanumeric displays
- 64 expandable to 128 keyboard
- Simultaneous keyboard and display
- 8 character keyboard—FIFO
- 2 key lockout or N key rollover
- Contact debounce
- Programmable scan timer
- Interrupt on key entry
- Single +5 V ±10% power supply
- Fully compatible with 8080A, 8085A, μPD780 (Z80®)

® Z80 is a registered trademark of Zilog, Inc.

### Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD8279C-2	40-pin plastic DIP	5 MHz
μPD8279C-5	40-pin plastic DIP	3 MHz

### Pin Configuration



### Pin Identification

No.	Symbol	Function
1,2,5,6,7,8,38,39	RL <sub>0</sub> -RL <sub>7</sub>	Return lines
3	CLK	Clock input
4	IRQ	Interrupt request
9	RESET	Reset input
10	RD	Read input
11	WR	Write input
12-19	DB <sub>0</sub> -DB <sub>7</sub>	Data bus
20	V <sub>SS</sub>	Ground reference
21	A <sub>0</sub>	Buffer address
22	CS	Chip select
23	BD	Blank display output
24-27	OUT A <sub>0</sub> -OUT A <sub>3</sub>	Display A outputs
28-31	OUT B <sub>0</sub> -OUT B <sub>3</sub>	Display B outputs
32-35	SL <sub>0</sub> -SL <sub>3</sub>	Scan lines
36	SHIFT	Shift input
37	CNTL/STB	Control/strobe input
40	V <sub>CC</sub>	+5 V input

**Pin Functions****RL<sub>0</sub>-RL<sub>7</sub> (Return Lines)**

Return line inputs which are connected to the scan lines through the keys or sensor switches. They have active internal pullups to keep them high until a switch closure pulls one low. They also serve as an 8-bit input in the strobed input mode.

**CLK (Clock)**

Clock from system used to generate internal timing.

**IRQ (Interrupt Request)**

In a keyboard mode, the interrupt line is high when there is data in the FIFO/sensor RAM. The interrupt line goes low with each FIFO/sensor RAM read and returns high if there is still information in the RAM. In the sensor mode, the interrupt line goes high whenever a change in a sensor is detected.

**RESET (Reset)**

A high signal on this pin resets the μPD8279.

 **$\overline{RD}$  (Read Input)**

Input read allows the data buffers to send data to the external bus.

 **$\overline{WR}$  (Write Input)**

Input write allows the data buffers to receive data from the external bus.

**DB<sub>0</sub>-DB<sub>7</sub> (Data Bus)**

Bidirectional data bus. All data and commands between the processor and the μPD8279 are transmitted on these lines.

**OUT A<sub>0</sub>-OUT A<sub>3</sub> (Display A Outputs)**

Output port for the 16 x 4 display refresh registers. The output data is synchronized to the scan lines (SL<sub>0</sub>-SL<sub>3</sub>) for multiplexed digit displays. Ports A and B may be blanked independently and may also be considered as one 8-bit port.

**OUT B<sub>0</sub>-OUT B<sub>3</sub> (Display B Outputs)**

Output port for the 16 x 4 display refresh registers. The output data is synchronized to the scan lines (SL<sub>0</sub>-SL<sub>3</sub>) for multiplexed digit displays. Ports A and B may be blanked independently and may also be considered as one 8-bit port.

**SL<sub>0</sub>-SL<sub>3</sub> (Scan Lines)**

Scan lines which are used to scan the key switch or sensor matrix and the display digits. These lines can be either encoded (1 of 16) or decoded (1 of 4).

**A<sub>0</sub> (Buffer Address)**

A high on this line indicates the signals in or out are interpreted as a command or status. A low indicates that they are data.

 **$\overline{CS}$  (Chip Select)**

A low on this pin enables the interface functions to receive or transmit.

 **$\overline{BD}$  (Blank Display Output)**

This output is used to blank the display during digit switching or by a display blanking command.

**SHIFT (Shift)**

The shift input status is stored along with the key position on key closure in the scanned keyboard modes. It has an active internal pullup to keep it high until a switch closure pulls it low.

**CNTL/STB (Control/Strobe Input)**

For keyboard modes this line is used as a control input and stored like status on a key closure. The line is also the strobe line that enters the data into the FIFO in strobed input mode (rising edge). It has an active internal pullup to keep it high until a switch closure pulls it low.

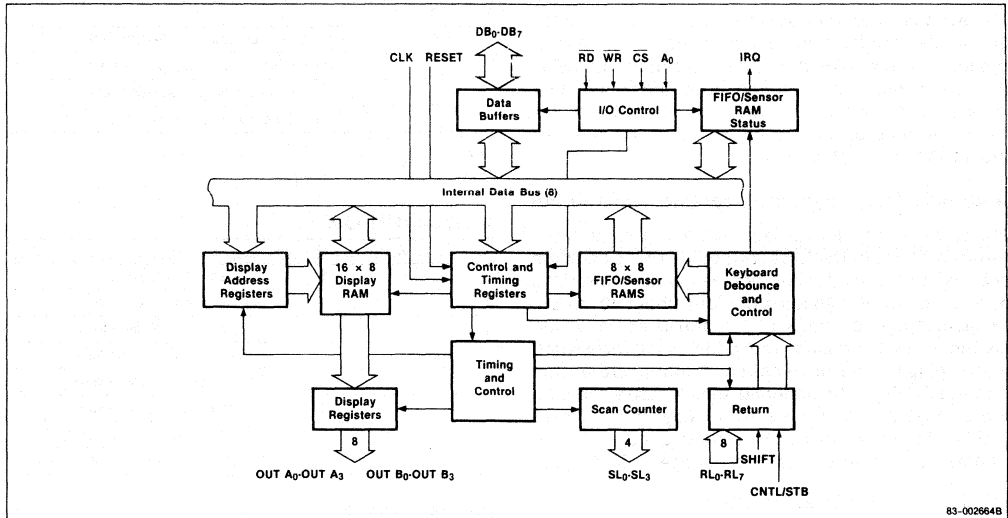
**V<sub>SS</sub> (Ground Reference)**

Ground.

**V<sub>CC</sub> (Power Supply)**

+5 V power supply input.

## Block Diagram



83-002664B

## Functional Description

The μPD8279 has two basic functions: 1) to control displays to output and 2) to control a keyboard for input. Its specific purpose is to unburden the host processor from monitoring keys and refreshing displays. The μPD8279 is designed to directly interface with the microprocessor bus. The microprocessor must program the operating mode to the μPD8279 as follows:

### Output Modes

- 8 or 16 character display
- Right or left entry display formats

### Input Modes

- Scanned keyboard with encoded (8 x 8 key keyboard) or decoded (4 x 8 key keyboard) scan lines.
- Scanned sensor matrix with encoded (8 x 8 matrix switches) or decoded (4 x 8 matrix switches) scan lines.
- Strobed input with data on return lines during control line strobe being transferred to FIFO.

## Block Diagram

Following is a description of each section of the μPD8279. See the block diagram for functional reference.

## I/O Control and Data Buffers

Communication to and from the μPD8279 is performed by selecting CS, A<sub>0</sub>, RD and WR. The type of information written or read by the processor is selected by A<sub>0</sub>. A logic 0 states that information is data while a 1 selects command or status. RD and WR select the direction by which the transfer occurs through the data buffers. When the chip is deselected (CS = 1) the bidirectional data buffers are in a high impedance state. This enables the μPD8279 to be tied directly to the processor bus.

## Timing Registers and Timing Control

The timing registers store the display and keyboard modes and other conditions programmed by the processor. The timing control contains the timing counter chain. One counter is a divide-by-N scaler, which may be programmed to match the processor cycle time. The scaler is programmed with a value between 2 and 31 to divide the external clock input by N to yield the internal clock frequency. A value which scales the internal frequency to 100 kHz gives a 5.1 ms scan time and 10.3 ms switch debounce. The other counters divide down to make key, row matrix, and display scans.

**Scan Counter**

The scan counter can operate in either the encoded or decoded mode. In the encoded mode, the counter provides a count which must be decoded to provide the scan lines. In the decoded mode, the counter provides a 1 out of 4 decoded scan. In the encoded mode, the scan lines are active high, and in the decoded mode, they are active low.

**Return Buffers, Keyboard Debounce and Control**

The eight return lines are buffered and latched by the return buffers. In the keyboard mode these lines are scanned to sample for key closures in each row. If the debounce circuit senses a closure, about 10 ms are timed out and a check is performed again. If the switch is still pressed, the address of the switch matrix plus the status of shift and control are written into the FIFO. In the scanned sensor mode, the contents of return lines are sent directly to the sensor RAM (FIFO) each key scan. In the strobed mode, the transfer takes place on the rising edge of CNTL/STB.

**FIFO/Sensor RAM and Status**

This section is a dual purpose 8 x 8 RAM. In strobe or keyboard mode it is a FIFO. Each entry is pushed into the FIFO and read in order. Status keeps track of the number of entries in the FIFO. Too many reads or writes to the FIFO will be treated as an error condition. The status logic generates an IRQ whenever the FIFO has an entry. In the sensor mode the memory is a sensor RAM which detects changes in the status of a sensor. If a change occurs, the IRQ is generated until the change is acknowledged.

**Display Address Registers and Display RAM**

The display address register contains the address of the word being read or written by the processor, as well as the word being displayed. This address may be programmed to autoincrement after each read or write. The display RAM may be read by the processor any time after the mode and address is set. Data entry to the display RAM may be set to either right or left entry.

**Command Operation**

The commands programmable to the μPD8279 via the data bus with CS active (0) and A<sub>0</sub> high are as follows:

**Keyboard/Display Mode Set**

0	0	0	D	D	K	K	K
---	---	---	---	---	---	---	---

**Display Mode:**

D	D	
0	0	Eight 8-bit character display—left entry
0	1 <sup>(1)</sup>	Sixteen 8-bit character display—left entry
1	0	Eight 8-bit character display—right entry
1	1	Sixteen 8-bit character display—right entry

**Note:**

(1) Power on default condition.

**Keyboard Mode:**

K	K	K	
0	0	0	Encoded scan—2 key lockout
0	0	0	Decoded scan—2 key lockout
0	1	0	Encoded scan—N key rollover
0	1	1	Decoded scan—N key rollover
1	0	0	Encoded scan—sensor matrix
1	0	1	Decoded scan—sensor matrix
1	1	0	Strobed input, encoded display scan
1	1	1	Strobed input, decoded display scan

**Program Clock**

0	0	1	P	P	P	P	P
---	---	---	---	---	---	---	---

Where P P P P P is the prescaler value between 2 and 31. This prescaler divides the external clock by P P P P P to develop its internal frequency. After reset, a default value of 31 is generated.

**Read FIFO/Sensor RAM**

0	1	0	A <sub>1</sub>	X	A	A	A	A <sub>0</sub> = 0
---	---	---	----------------	---	---	---	---	--------------------

A<sub>1</sub> is the autoincrement flag. AAA is the row to be read by the processor. The read command is accomplished with (CS • RD • A<sub>0</sub>) by the processor. If A<sub>1</sub> is 1, the row select counter will be incremented after each read. Note that autoincrementing has no effect on the display.

**Read Display RAM**

0	1	1	A <sub>1</sub>	X	A	A	A	A <sub>0</sub> = 0
---	---	---	----------------	---	---	---	---	--------------------

Where A<sub>1</sub> is the autoincrement flag and AAAA is the character which the processor is about to read.

**Write Display RAM**

1	0	0	A <sub>1</sub>	A	A	A	A
---	---	---	----------------	---	---	---	---

Where AAAA is the character the processor is about to write.

## Display Write Inhibit Blanking

1	0	1	X	IW	IW	BL	BL
				A	B	A	B

Where IWA and IWB are inhibit writing nibble A and B respectively, while BLA and BLB are used for blanking. When using the display as a dual 4-bit, it is necessary to mask one of the 4-bit halves to eliminate interaction between the two halves. This is accomplished with the IW flags. The BL flags allow the programmer to blank either half of the display independently. To blank a display formatted as a single 8-bit, it is necessary to set both BLA and BLB. Default after a reset is all zeros. All signals are active high (logic 1).

## Clear

1	1	0	C <sub>D</sub>	C <sub>D</sub>	C <sub>D</sub>	C <sub>F</sub>	C <sub>A</sub>
---	---	---	----------------	----------------	----------------	----------------	----------------

Where:

C <sub>D</sub>	C <sub>D</sub>	C <sub>D</sub>	
1	0	X	All zeros
1	1	0	AB = 20H
1	1	1	All ones
0	X	X	Disable clear display

This command is used to clear the display RAM, the FIFO, or both. The C<sub>D</sub> options allow the user the ability to clear the display RAM to either all zeros or all ones. Clearing the display takes one complete display scan. During this time the processor can't write to the display RAM.

If the C<sub>F</sub> bit is set to logic 1, the FIFO status is cleared, the FIFO empty flag is set, and IRQ is cleared. The sensor matrix mode RAM pointer will then be set to row 0.

C<sub>A</sub>, the clear all bit, has the combined effect of C<sub>F</sub> and C<sub>D</sub>; it uses the C<sub>D</sub> clearing code on the display RAM and also clears FIFO status. It also re-synchronizes the internal timing chain.

## End Interrupt/Error Mode Set

1	1	1	E	X	X	X	X
---	---	---	---	---	---	---	---

In the sensor matrix mode, this instruction clears IRQ and allows writing into RAM. In N key rollover, setting the E bit to 1 allow for operating in the special error mode. See description of FIFO status.

## FIFO Status

D <sub>U</sub>	S/E	0	U	F	N	N	N
----------------	-----	---	---	---	---	---	---

Where:

- DU = Display unavailable because a clear display or clear all command is in progress.
- S/E = Sense error flag due to multiple closure of switch matrix.
- O = FIFO overrun since an attempt was made to push too many characters into the FIFO.
- U = FIFO underrun. An indication that the processor tried to read an empty FIFO.
- F = FIFO full flag.
- NNN = The number of characters presently in FIFO.

The FIFO status is read with A<sub>0</sub> high and  $\overline{CS}$ ,  $\overline{RD}$  active low.

If the C<sub>D</sub> or C<sub>A</sub> command has not completed its clearing, the display is not available. The S/E flags are used to show an error in multiple closures has occurred. The O or U, overrun or underrun, flags occur when too many characters are written into the FIFO or the processor tries to read an empty FIFO. F is an indication that the FIFO is full and NNN is the number of characters in the FIFO.

## Data Read

Data can be read during A<sub>0</sub> = 0 and when  $\overline{CS}$ ,  $\overline{RD}$  are active low. The source of data is determined by the read display or read FIFO commands.

## Data Write

Data is written to the chip when A<sub>0</sub>,  $\overline{CS}$ , and  $\overline{WR}$  are active low. Data will be written into the display RAM with its address selected by the latest read or write display command.

## Data Format

CNTL	SH	SCAN	RET
------	----	------	-----

In the scanned key mode, the characters in the FIFO correspond to the above format where CNTL and SH are the most significant bits and the SCAN and return lines are the scan and column counters.

RL <sub>7</sub>	RL <sub>6</sub>	RL <sub>5</sub>	RL <sub>4</sub>	RL <sub>3</sub>	RL <sub>2</sub>	RL <sub>1</sub>	RL <sub>0</sub>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

In the sensor matrix mode, the data corresponds directly to the row of the sensor RAM being scanned. Shift and control (SH, CNTL) are not used in this mode.

### Command Word Summary

0	0	0	D	D	K	K	K	Keyboard display mode set
0	0	1	P	P	P	P	P	Load program clock
0	1	0	A <sub>1</sub>	X	A	A	A	Read FIFO/sensor RAM
0	1	1	A <sub>1</sub>	A	A	A	A	Read display RAM
1	0	0	A <sub>1</sub>	A	A	A	A	Write display RAM
1	0	1	X	IW	IW	BL	BL	Display write inhibit/blanking
1	1	0	C <sub>D</sub>	C <sub>D</sub>	C <sub>D</sub>	C <sub>F</sub>	C <sub>A</sub>	Clear
1	1	1	E	X	X	X	X	End interrupt/error mode set
D <sub>U</sub>	S/E	0	U	F	N	N	N	FIFO status

### Absolute Maximum Ratings

T<sub>A</sub> = 25°C

Power supply voltage, V <sub>DD</sub>	-0.5 V to +7.0 V (1)
Power dissipation, P <sub>D</sub>	1.0 W
Operating temperature, T <sub>OPT</sub>	0°C to +70°C
Storage temperature, T <sub>STG</sub>	-65°C to +150°C

#### Note:

(1) With respect to V<sub>SS</sub>.

**Comment:** Exposing the device to stresses above those listed in the absolute maximum ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum ratings for extended periods may affect device reliability.

### Capacitance

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	C <sub>I</sub>	5		10	pF	V <sub>I</sub> = V <sub>CC</sub>
Output capacitance	C <sub>O</sub>	10		20	pF	V <sub>O</sub> = V <sub>CC</sub>

### DC Characteristics

T<sub>A</sub> = 0°C to +70°C; V<sub>CC</sub> = 5 V ± 10%; V<sub>SS</sub> = 0 V

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input high voltage for return lines	V <sub>IH1</sub>	2.2		V	
Input high voltage for other lines	V <sub>IH2</sub>	2.0		V	
Input low voltage for return lines	V <sub>IL1</sub>	-0.5	1.4	V	
Input low voltage for other lines	V <sub>IL2</sub>	-0.5	0.8	V	
Output high voltage on interrupt line	IRQ pin	+3.5		V	I <sub>OH</sub> = -50 μA
	others	+2.4		V	I <sub>OH</sub> = -400 μA
Output low voltage	V <sub>OL</sub>	0.45		V	I <sub>OL</sub> = 2.2 mA
Input current on shift, control and return lines	I <sub>IL1</sub>	+10		μA	V <sub>I</sub> = V <sub>CC</sub>
		-100		μA	V <sub>I</sub> = 0 V
Input leakage current for other lines	I <sub>IL2</sub>	±10		μA	V <sub>I</sub> = V <sub>CC</sub> to 0 V
Output float leakage	I <sub>OF</sub>	±10		μA	V <sub>O</sub> = V <sub>CC</sub> to 0 V
Power supply current	I <sub>CC</sub>		120	mA	

## AC Characteristics

$T_A = 0^\circ\text{C to } +70^\circ\text{C}; V_{CC} = 5\text{ V } \pm 10\%; V_{SS} = 0\text{ V}$

Parameter	Symbol	μPD8279-5 Limits		μPD8279-2 Limits		Unit	Test Conditions
		Min	Max	Min	Max		
<b>Read</b>							
Address stable before read	$t_{AR}$	0		0		ns	
Address hold time for read	$t_{RA}$	0		0		ns	
Read pulse width	$t_{RR}$	250		200		ns	
Data delay from read	$t_{RD}$		150		140	ns	$C_L = 150\text{ pF}$
Address to data valid	$t_{AD}$		250		250	ns	$C_L = 150\text{ pF}$
Read to data floating	$t_{DF}$	10	100	10	100	ns	
Read cycle time	$t_{RCY}$	1000		200		ns	
<b>Write</b>							
Address stable before write	$t_{AW}$	0		0		ns	
Address hold time for write	$t_{WA}$	0		0		ns	
Write pulse width	$t_{WW}$	250		200		ns	
Data set up time for write	$t_{DW}$		150		150	ns	
Data hold time for write	$t_{WD}$	0		0		ns	
Write cycle time	$t_{WCY}$	1000		200		ns	
<b>Other</b>							
Clock pulse width	$t_{\phi W}$		120		70	ns	
Clock period	$t_{CY}$		320		200	ns	

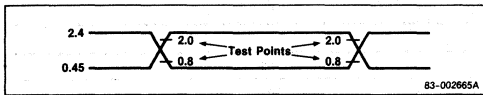


**General Timing**

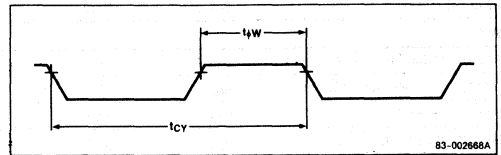
Keyboard scan time	5.1 ms
Keyboard debounce time	10.3 ms
Key scan time	80 μs
Display scan time	10.3 ms
Digit-on time	480 μs
Blanking time	160 μs
Internal clock cycle	10 μs

**Timing Waveforms**

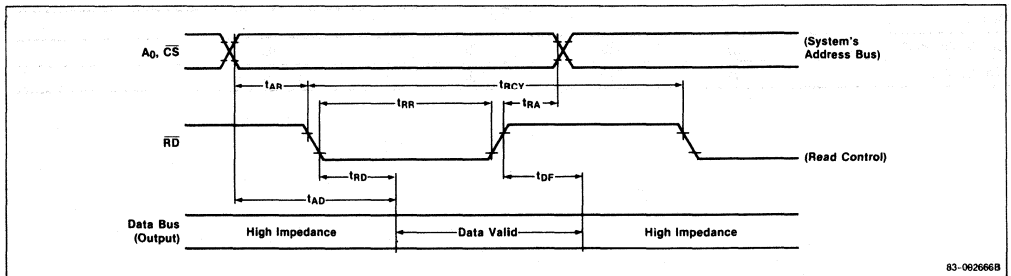
**AC Test Input**



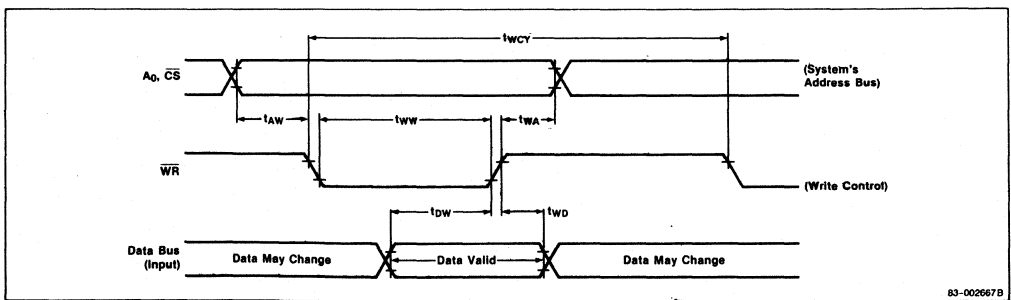
**Clock Input**



**Read**



**Write**



### Description

The  $\mu$ PB8282 and  $\mu$ PB8283 are 8-bit latches with three-state output buffers. The  $\mu$ PB8282 is non-inverting and the  $\mu$ PB8283 inverts the input data. These devices are ideal for demultiplexing the address/data buses on the 8085A/8086 microprocessors. The  $\mu$ PB8282/83 are fabricated using NEC's Schottky bipolar process.

### Features

- Support  $\mu$ PB8080, 8085A, 8048, 8086 family systems
- Transparent during active strobe
- Fully parallel 8-bit data register and buffer
- High output drive capability (32 mA) for driving the system data bus
- Three-state outputs

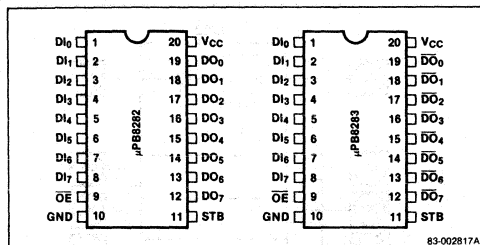
### Ordering Information

Part Number	Package Type	Output Drive Capability
$\mu$ PB8282C	20-pin plastic DIP	32 mA
$\mu$ PB8283C	20-pin plastic DIP	32 mA

### Pin Identification

No.	Symbol	Function
1-8	DI <sub>0</sub> -DI <sub>7</sub>	Data in
9	$\overline{OE}$	Output enable
10	GND	Ground
11	STB	Strobe
12-19	( $\mu$ PB8282) DO <sub>7</sub> -DO <sub>0</sub> ( $\mu$ PB8283) $\overline{DO}$ <sub>7</sub> - $\overline{DO}$ <sub>0</sub>	Data out
20	V <sub>CC</sub>	Power supply

### Pin Configurations



### Pin Functions

#### $\overline{OE}$ (Output Enable)

This active low input control signal enables the contents of the data latches onto the data output pins (B<sub>0</sub>-B<sub>7</sub>). When  $\overline{OE}$  goes high, the output buffers become high impedance.

#### STB (Strobe)

This input control pulse strobes data at input A<sub>0</sub>-A<sub>7</sub> into the data latches. Data is latched at STB's high to low transition. When active high, STB admits input data.

#### DI<sub>0</sub>-DI<sub>7</sub> (Data In)

When data that satisfies the STB strobe setup time requirements is input to these pins, it is latched into the data latches.

#### DO<sub>0</sub>-DO<sub>7</sub> ( $\mu$ PB8282) (Data Out) $\overline{DO}$ <sub>0</sub>- $\overline{DO}$ <sub>7</sub> ( $\mu$ PB8283)

When  $\overline{OE}$  is active (low), it outputs data to the DO<sub>0</sub>-DO<sub>7</sub> pins. When  $\overline{OE}$  is inactive high, DO<sub>0</sub>-DO<sub>7</sub> are high impedance. Enabling or disabling the output buffers will not cause negative-going transients to appear on the data output bus.

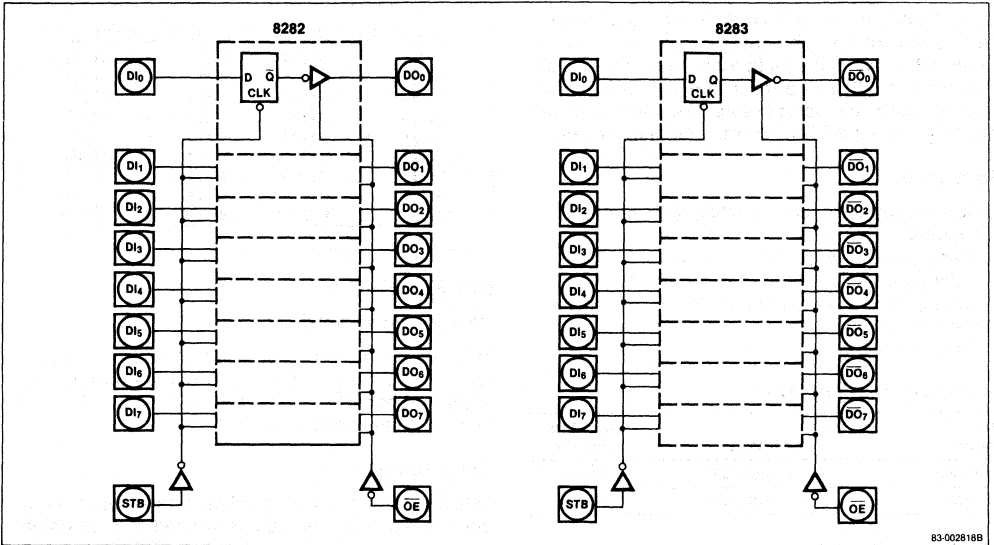
#### GND (Ground)

This is the ground.

#### V<sub>CC</sub> (Power Supply)

This is the +5 V power supply.

Block Diagrams



83-002818B

Functional Description

The μPB8282/83 are 8-bit latches with three-state output buffers. Data on the inputs is latched into the data latches on a high-to-low transition of the STB line. When STB is high, the latches appear transparent. The OE input enables the latched data to be transferred to the output pins. When OE is high, the outputs are put in the three-state condition. OE will not cause transients to appear on the data outputs.

Absolute Maximum Ratings

TA = 25°C

Operating temperature	0°C to +70°C
Storage temperature	-65°C to +150°C
All output and supply voltages	-0.5 to +7 V
All input voltages	-1.0 V to 5.5 V

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

TA = 0°C to +70°C; VCC = +5 V ±10%

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input clamp voltage	VC		-1	V	IC = -5 mA
Power supply current	ICC		160	mA	
Forward input current	IF		-0.2	mA	VF = 0.45 V
Reverse input current	IR		50	μA	VR = 5.25 V
Output low voltage	VOL		0.45	V	IOL = 32 mA
Output high voltage	VOH	2.4		V	IOH = -5 mA
Output off current	IOFF		±50	μA	VOFF = 0.45 to 5.25 V
Input low voltage	VIL		0.8	V	VCC = 5.0 V (1)
Input high voltage	VIH	2.0		V	VCC = 5.0 V (1)
Input capacitance	CIN		12	pF	VBIAS = 2.5 V, VCC = 5 V, TA = 25°C, F = 1 MHz

Note:

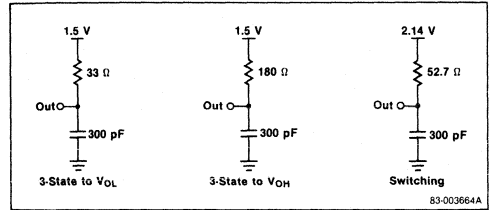
(1) Output loading IOL = 32 mA, IOH = -5 mA, CL = 300 pF

## AC Characteristics

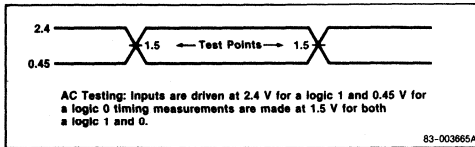
$T_A = 0^\circ\text{C to } +70^\circ\text{C}; V_{CC} = 5\text{ V} \pm 10\%$   
 $I_{OL} = 32\text{ mA}, I_{OH} = -5\text{ mA}, C_L = 300\text{ pF}$

Parameter	Symbol	Limits		Unit
		Min	Max	
Input to output delay	$t_{iVOV}$			
—Inverting		5	22	ns
—Non-inverting		5	30	ns
STB to output delay	$t_{SHOV}$			
—Inverting		10	40	ns
—Non-inverting		10	45	ns
Output disable time	$t_{EHOZ}$	5	22	ns
Output enable time	$t_{ELOV}$	10	30	ns
Input to STB setup time	$t_{iVSL}$	0		ns
Input to STB hold time	$t_{SLIX}$	25		ns
STB high time	$t_{SHSL}$	15		ns
Input, output rise time	$t_{LILH}, t_{OLOH}$		20	ns
Input, output fall time	$t_{HIL}, t_{HOL}$		12	ns

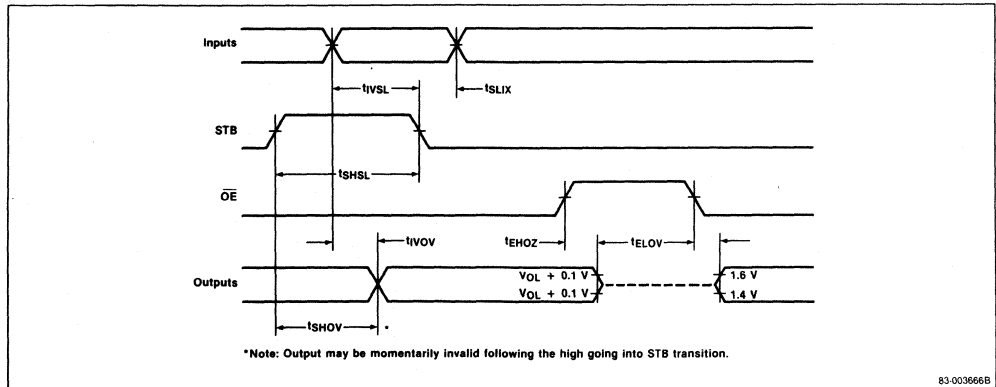
## Load Circuits



## AC Test Points



## Timing Waveform





### Description

The μPB8284 is a clock generator and driver for the 8086 and 8088 microprocessors. This bipolar driver provides the microprocessor with a reset signal and also provides properly synchronized READY timing. A TTL clock is also provided for peripheral devices.

### Features

- Generates system clock for the 8086 and 8088
- Frequency source can be a crystal or a TTL signal
- MOS level output for the processor
- TTL level output for the peripheral devices
- Power-up reset for the processor
- READY synchronization
- +5 V supply

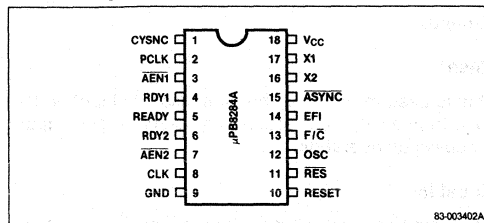
### Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPB8284AD	18-Pin cerdip	25 MHz ± 3

### Pin Identification

No.	Symbol	Function
1	CYSNC	Clock synchronization
2	PCLK	Peripheral clock
3, 7	AEN1, AEN2	Address enable
4, 6	RDY1, RDY2	Bus ready
5	READY	Ready
8	CLK	Processor clock
9	GND	Ground
10	RESET	Reset
11	RES	Reset in
12	OSC	Oscillator output
13	F/C	Frequency crystal select
14	EFI	External frequency in
15	ASYNCR	Asynchronous input
16, 17	X1, X2	Crystal in
18	V <sub>CC</sub>	V <sub>CC</sub>

### Pin Configuration



### Pin Functions

#### Clock Synchronization

An active high signal which allows multiple 8284s to be synchronized. When CYSNC is low, the internal counters count, and when high, the counters are reset. CYSNC should be grounded when the internal oscillator is used.

#### Peripheral Clock

A TTL level clock for use with peripheral devices. This clock is one-half the frequency of CLK.

#### Address Enable

This active low signal is used to qualify its respective RDY inputs. If there is only one bus to interface to, AEN inputs are to be grounded.

#### Bus Ready

This signal is sent to the 8284 from a peripheral device on the bus to indicate that data has been received or data is available to be read.

#### Ready

The READY signal to the microprocessor is synchronized by the RDY inputs to the processor CLK. READY is cleared after the guaranteed hold time to the processor has been met.

#### Processor Clock

This is the MOS level clock output of 33% duty cycle to drive the microprocessor and bipolar support devices (8288) connected to the processor. The frequency of CLK is one third of the crystal or EFI frequency.

**Ground**

Ground.

**Reset**

This is used to initialize the processor. Its input is derived from an RC connection to a Schmitt trigger input for power up operation.

**Reset In**

The Schmitt trigger input is used to determine the timing of RESET out via an RC circuit.

**Oscillator Output**

This TTL level clock is the output of the oscillator circuit running at the crystal frequency.

**Frequency Crystal Select**

F/C is a strapping option used to determine where CLK is generated. A high is for the EFI input, and a low is for the crystal.

**External Frequency In**

A square wave in at three times the CLK output. A TTL level clock to generate CLK.

**Asynchronous Input**

Ready Synchronization Select.  $\overline{ASYNC}$  is an input which defines the synchronization mode of the READY logic. When  $\overline{ASYNC}$  is low, two stages of READY synchronization are provided. When  $\overline{ASYNC}$  is left open or HIGH, a single stage of READY synchronization is provided.

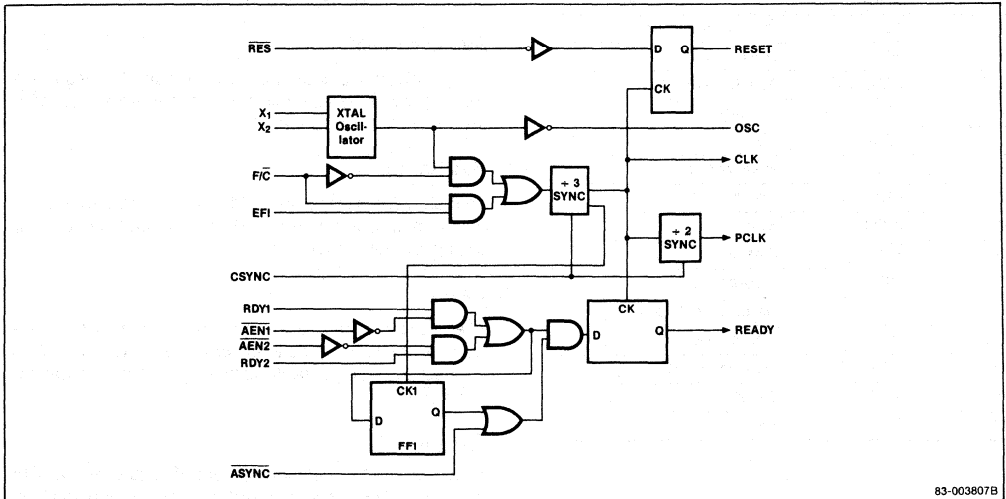
**Crystal In**

A crystal is connected to these inputs to generate the processor clock. The crystal frequency is three times the desired CLK output.

**VCC Supply Voltage**

+5 V supply.

**Block Diagram**



83-003807B

**Functional Description**

The clock generator can provide the system clock from either a crystal or an external TTL source. There is an internal divide-by-three counter which receives its input from either the crystal or TTL source (EFI pin) depending on the state of the F/C input strapping. There is also a clear input (C SYNC) which is used for

either inhibiting the clock, or synchronizing it with an external event (or perhaps another clock generator chip). Note that if the TTL input is used, the crystal oscillator section can still be used for an independent clock source, using the OSC output.

For driving the MOS output level, there is a 33% duty cycle MOS output (CLK) for the microprocessor, and a TTL output (PCLK) with a 50% duty cycle for use as a peripheral clock signal. This clock is at one-half of the processor clock speed.

Reset timing is provided by a Schmitt trigger input (RES) and a flip-flop to synchronize the reset timing to the falling edge of CLK. Power-on reset is provided by a simple RC circuit on the RES input.

### Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$

Power supply voltage, $V_{DD}$	-0.5 V to +7 V
Input voltage, $V_I$	-1.0 V to +5.5 V
Output supply voltage, $V_O$	-0.5 V to +7 V
Operating temperature, $T_{OP}$	-0°C to +70°C
Storage temperature, $T_{STG}$	-65°C to +150°C

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### DC Characteristics

$T_A = 0^\circ\text{C}$  to +70°C,  $V_{CC} = +5\text{ V} \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage low	$V_{IL}$			+0.8	V	$V_{CC} = 5.0\text{ V}$
Input voltage high	$V_{IH}$	2			V	$V_{CC} = 5.0\text{ V}$
Output voltage low	$V_{OL}$			+0.45	V	$5\text{ mA} = I_{OL}$
Output voltage high (CLK)	$V_{OH}$	4			V	$-1\text{ mA} = I_{OH}$
(Other outputs)		2.4			V	$-1\text{ mA} = I_{OH}$
Forward input current (ASYNC)	$I_F$			-1.3	mA	$V_F = 0.45\text{ V}$
(Other inputs)				-0.5	mA	$V_F = 0.45\text{ V}$
Reverse input current	$I_R$			50	μA	$V_R = 5.25\text{ V}$
Input forward clamp voltage	$V_C$			-1.0	V	$I_C = -5\text{ mA}$
Reset input high voltage	$V_{IHR}$	2.6			V	$V_{CC} = 5.0\text{ V}$
RES input hysteresis	$V_{IHR} - V_{ILR}$	0.25			V	$V_{CC} = 5.0\text{ V}$
Power supply current	$I_{CC}$			140	mA	

There are two READY inputs, each with its own qualifier (AEN1, AEN2). The unused AEN signal should be tied low.

The READY logic in the 8284A synchronizes the RDY1 and RDY2 asynchronous inputs to the processor clock to insure proper set up time, and to guarantee proper hold time before clearing the ready signal.

### AC Characteristics

$T_A = 0^\circ\text{C}$  to +70°C,  $V_{CC} = 5\text{ V} \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
<b>Timing Requirements</b>						
External frequency time high	$t_{EHEL}$	13			ns	90%–90% $V_{IN}$
External frequency time low	$t_{ELEH}$	13			ns	10%–10% $V_{IN}$
EFI period	$t_{EEL}$	(5)			ns	(Note 1)
XTAL frequency		12		25	MHz	
RDY1, RDY2 set-up to CLK	$t_{RIVCL}$	35			ns	
RDY1, RDY2 hold to CLK	$t_{CLRIX}$	0			ns	
AEN1, AEN2 set-up to RDY1, RDY2	$t_{A1VR1V}$	15			ns	
AEN1, AEN2 hold to CLK	$t_{CLAIX}$	0			ns	
CSYNC set-up to EFI	$t_{YHEH}$	20			ns	
CSYNC hold to EFI	$t_{EHYL}$	10			ns	
CSYNC width	$t_{YHYL}$	$2 t_{ELEL}$			ns	
RES set-up to CLK	$t_{IHLCL}$	65			ns	(Note 2)
RES hold to CLK	$t_{CL1H}$	20			ns	(Note 2)
RDY1, RDY2 active set-up to CLK	$t_{RIVCH}$	35			ns	ASYNC = Low
RDY1, RDY2 inactive set-up to CLK	$t_{RIVCL}$	35			ns	
ASYNC set-up to CLK	$t_{AYVCL}$	50			ns	
ASYNC hold to CLK	$t_{CLAYX}$	0			ns	
Input rise time	$t_{ILIH}$		20		ns	From 0.8 V to 2.0 V
Input fall time	$t_{ILIL}$		12		ns	From 2.0 V to 0.8 V



**AC Characteristics (cont)**

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5 V ± 10%

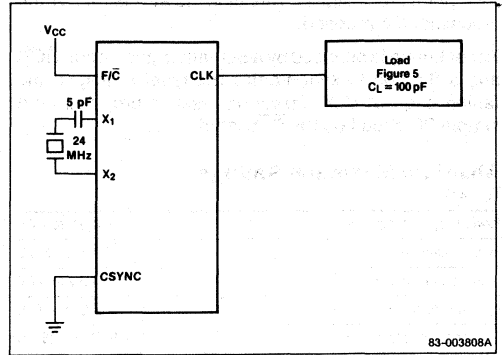
Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
<b>Timing Responses</b>						
CLK cycle period	t <sub>CLCL</sub>	125			ns	
CLK time high	t <sub>CHCL</sub>	(6)			ns	Figure 1 and figure 2
CLK time low	t <sub>CLCH</sub>	(7)			ns	Figure-1 and figure 2
CLK rise and fall time	t <sub>CH1CH2- CL2CL1</sub>		10		ns	1.0 V to 3.5 V
PCLK time high	t <sub>PHPL</sub>	(8)			ns	
PCLK time low	t <sub>PLPH</sub>	(8)			ns	
Ready inactive to CLK	t <sub>RYLCL</sub>	-8			ns	Figure 3 and figure 4, (Note 4)
Ready active to CLK	t <sub>RYHCH</sub>	(7)			ns	Figure 3 and figure 4, (Note 3)
CLK to reset delay	t <sub>CLIL</sub>		40		ns	
CLK to PCLK high delay	t <sub>CLPH</sub>		22		ns	
CLK to PCLK low delay	t <sub>CLPL</sub>		22		ns	
OSC to CLK high delay	t <sub>OLCH</sub>	-5	12		ns	
OSC to CLK low delay	t <sub>OLCL</sub>	2	22		ns	
Output rise time (except CLK)	t <sub>OLOH</sub>		20		ns	From 0.8 V to 2.0 V
Output fall time (except CLK)	t <sub>OHOL</sub>		12		ns	From 2.0 V to 0.8 V

**Note:**

- (1) δ = EFi rise (5 ns max) + EFi fall (5 ns max).
- (2) Set-up and hold only necessary to guarantee recognition at next clock.
- (3) Applies only to T3 and TW states.
- (4) Applies only to T2 states.
- (5) t<sub>EHCL</sub> + t<sub>ELCH</sub> + δ
- (6) (1/3 t<sub>CLCL</sub>) + 2.0
- (7) (2/3 t<sub>CLCL</sub>) - 15.0
- (8) t<sub>CLCL</sub> - 20

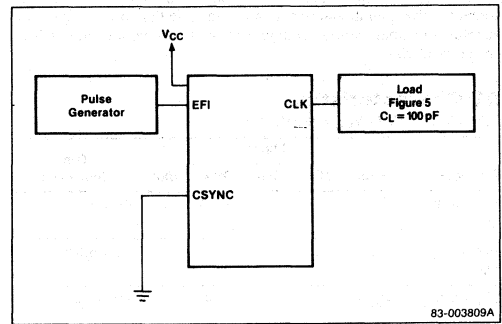
**AC Test Circuits**

**Figure 1. Clock High and Low Time**



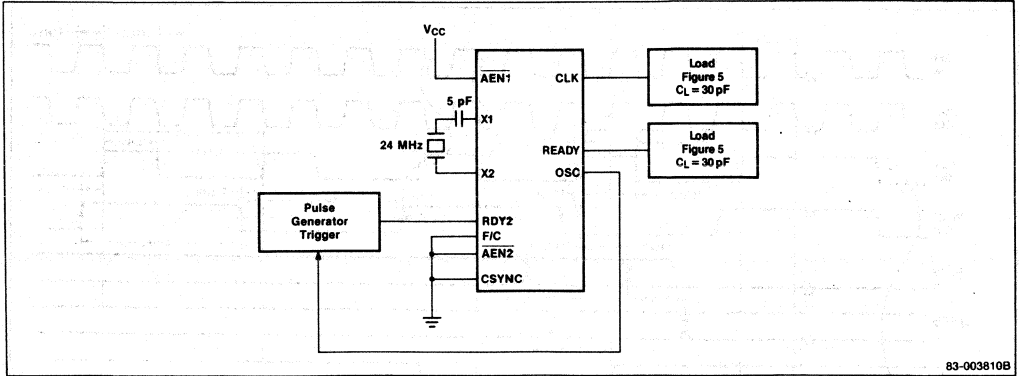
83-003808A

**Figure 2. Clock High and Low Time**

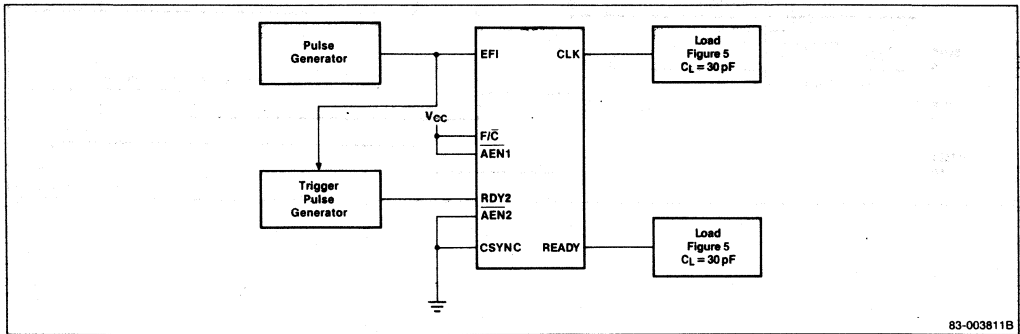


83-003809A

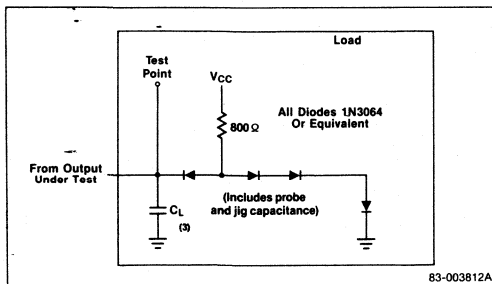
**Figure 3. Ready to CLK**



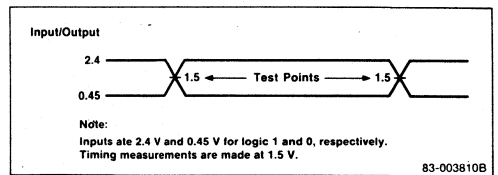
**Figure 4. Ready to CLK Output**



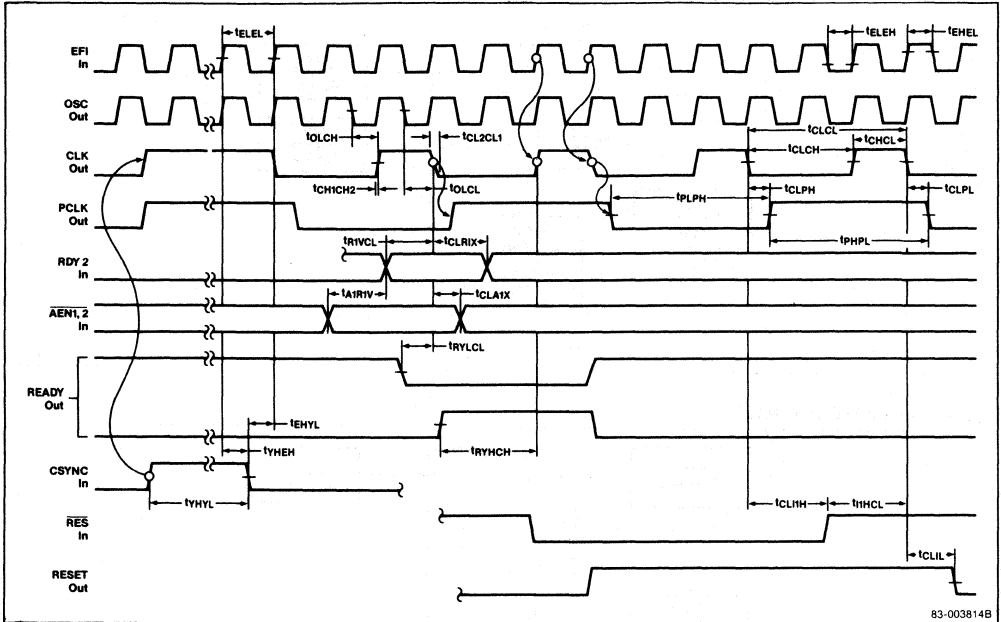
**Figure 5. AC Load**



**Figure 6. Timing Measurement Points**



Timing Waveform



83-003814B

### Description

The  $\mu$ PB8286 and  $\mu$ PB8287 are octal bus transceivers used for buffering microprocessor bus lines. Being bi-directional, they are ideal for buffering the data bus lines on 8- or 16-bit microprocessors. Each B output is capable of driving 32 mA low or 5 mA high.

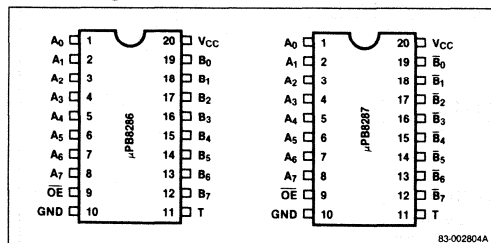
### Features

- Data bus buffer driver for  $\mu$ COM-8 (8080, 8085A, 780) and  $\mu$ COM-16 (8086) families
- Low input load current – 0.2 mA max
- High output drive capability for driving system data bus
- Three-state outputs

### Ordering Information

Part Number	Package Type	I/O Delay, Max
$\mu$ PB8286C	20-pin plastic DIP	22 ns
$\mu$ PB8287C	20-pin plastic DIP	30 ns

### Pin Configurations



### Pin Identification

No.	Symbol	Function
1-8	$A_0$ - $A_7$	Local data bus
9	$\overline{OE}$	Output enable
10	GND	Ground
11	T	Transmit
12-19	( $\mu$ PB8286) $B_7$ - $B_0$ ( $\mu$ PB8287) $\overline{B_7}$ - $\overline{B_0}$	System data bus
20	$V_{CC}$	Power supply

### Pin Functions

#### $\overline{OE}$ (Output Enable)

This active low input control signal enables the output drivers selected by T.

#### T (Transmit)

This input controls the direction of data through the transceivers. When high, data is transferred from the  $A_0$ - $A_7$  inputs to the  $B_0$ - $B_7$  outputs. When low, data is transferred from the  $B_0$ - $B_7$  inputs to the  $A_0$ - $A_7$  outputs.

#### $A_0$ - $A_7$ (Local Data Bus)

$A_0$ - $A_7$  are bidirectional drivers that, depending on the state of the transmit pin, accept data from or transfer data to the processor's local bus.

#### $B_0$ - $B_7$ (System Data Bus)

$B_0$ - $B_7$  are bidirectional drivers that, depending on the state of the transmit pin, accept data from or transfer data to the system bus.

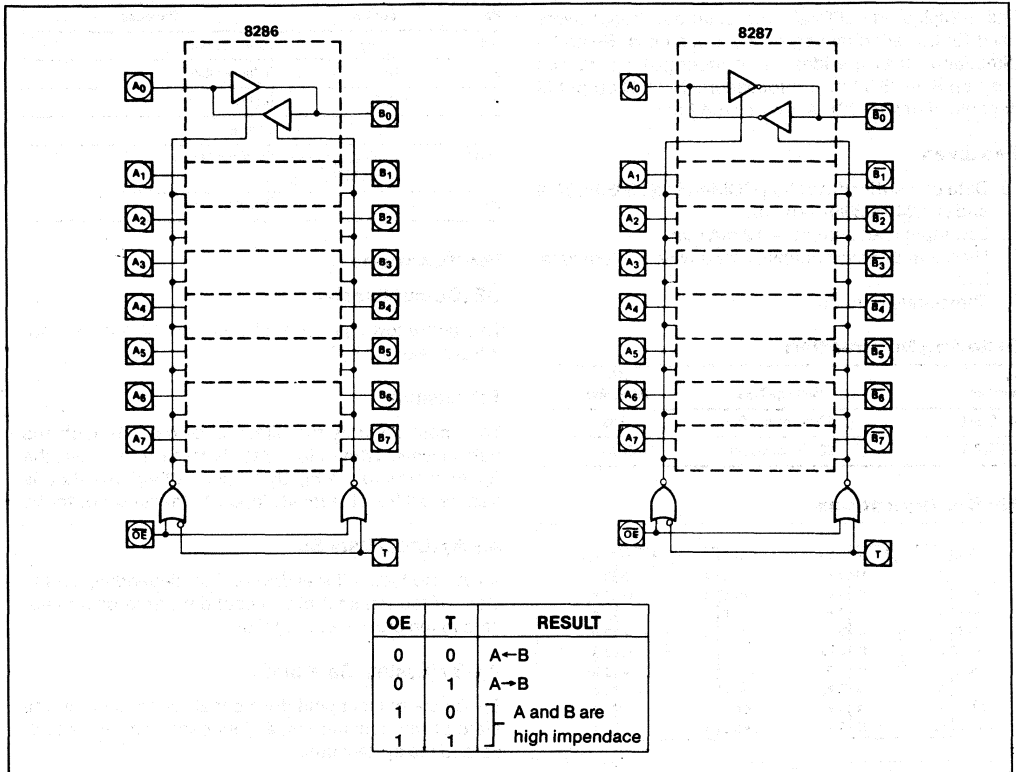
#### GND (Ground)

This is the ground.

#### $V_{CC}$ (Power Supply)

This is the +5V power supply.

**Block Diagram**



**Functional Description**

MOS microprocessors like the 8080/8085A/8086 are generally capable of driving a single TTL load. This also applies to MOS memory devices. While sufficient for minimum type small systems on a single PC board, it is usually necessary to buffer the microprocessor and memory signals when a system is expanded or signals go to other PC boards.

These octal bus transceivers are designed to do the necessary buffering.

**Bidirectional Driver**

Each buffered line of the octal driver consists of two separate three-state buffers. The B side of the driver is designed to drive 32 mA and interface the system side

of the bus to I/O, memory, etc. The A side is connected to the microprocessor.

**Control Gating,  $\overline{OE}$ , T**

The  $\overline{OE}$  (output enable) input is an active low signal used to enable the drivers selected by T on to the respective bus.

T is an input control signal used to select the direction of data through the transceivers. When T is high, data is transferred from the A<sub>0</sub>-A<sub>7</sub> inputs to the B<sub>0</sub>-B<sub>7</sub> outputs, and when low, data is transferred from B<sub>0</sub>-B<sub>7</sub> to the A<sub>0</sub>-A<sub>7</sub> outputs.

## Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$

Power supply voltage, $V_{CC}$	-0.5 V to +7 V
Input voltage, $V_I$	-1.0 V to +5.5 V
Output voltage, $V_O$	-0.5 V to +7 V
Operating temperature, $T_{OPT}$	$0^\circ\text{C}$ to $+70^\circ\text{C}$
Storage temperature, $T_{STG}$	$-65^\circ\text{C}$ to $+150^\circ\text{C}$

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## DC Characteristics

$T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = +5\text{ V} \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage low — A side	$V_{IL}$			+0.8	V	$V_{CC} = 5.0\text{ V}$ , (Note 1)
				+0.9	V	$V_{CC} = 5.0\text{ V}$ , (Note 1)
Input voltage high	$V_{IH}$		2		V	$V_{CC} = 5.0\text{ V}$ , (Note 1), $F = 1\text{ MHz}$
Output voltage low — B outputs	$V_{OL}$			+0.45	V	$I_{OL} = 32\text{ mA}$
				+0.45	V	$I_{OL} = 16\text{ mA}$
Output voltage high — B outputs	$V_{OH}$	2.4			V	$I_{OH} = -5\text{ mA}$
		2.4			V	$I_{OH} = -1\text{ mA}$
Input clamp voltage	$V_C$			-1	V	$I_C = -5\text{ mA}$
Input forward current	$I_F$			-0.2	$\mu\text{A}$	$V_F = 0.45\text{ V}$
Input reverse current	$I_R$		50		$\mu\text{A}$	$V_R = 5.25\text{ V}$
Power supply current	$I_{CC}$					
				130	$\text{mA}$	
				160	$\text{mA}$	
Output off current	$I_{OFF}$			$I_F$		$V_{OFF} = 0.45\text{ V}$
Output off current	$I_{OFF}$			$I_R$		$V_{OFF} = 5.25\text{ V}$

### Note:

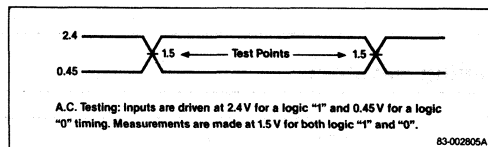
- (1) B outputs —  $I_{OL} = 32\text{ mA}$ ,  $I_{OH} = -5\text{ mA}$ ,  $C_L = 300\text{ pF}$   
 A outputs —  $I_{OL} = 16\text{ mA}$ ,  $I_{OH} = -1\text{ mA}$ ,  $C_L = 100\text{ pF}$

## AC Characteristics

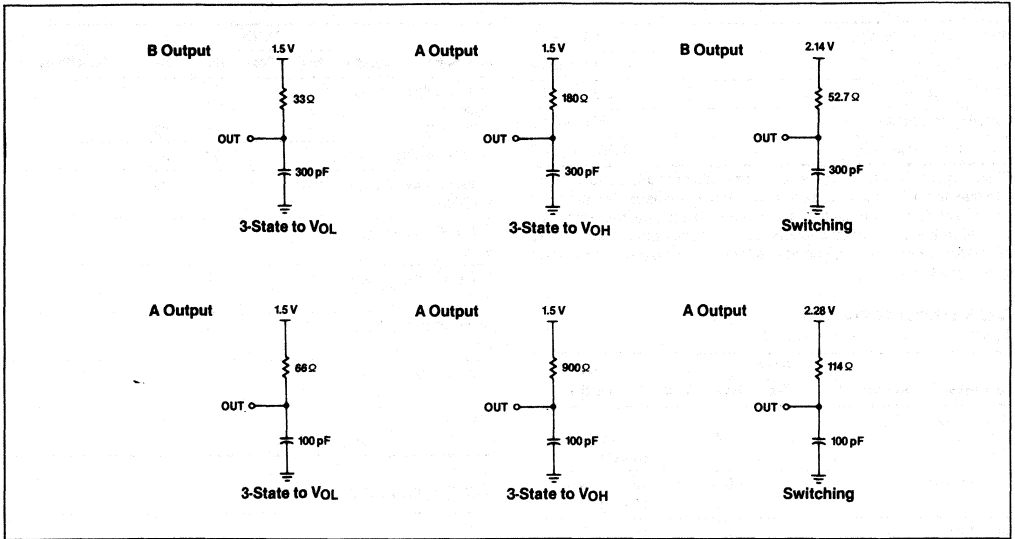
$T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5\text{ V} \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input to output delay	$t_{IVOV}$					
		Inverting	5		22	ns
		5		30	ns	
Transmit / receive hold time	$t_{EHTV}$					ns
Transmit / receive setup	$t_{VEL}$	10				ns
Output disable time	$t_{EHOZ}$	5		22		ns
Output enable time	$t_{ELOV}$	10		30		ns
I/O rise time	$t_{ILIH}$ $t_{OLOH}$				20	ns
I/O fall time	$t_{IHIL}$ $t_{OHOL}$				12	ns

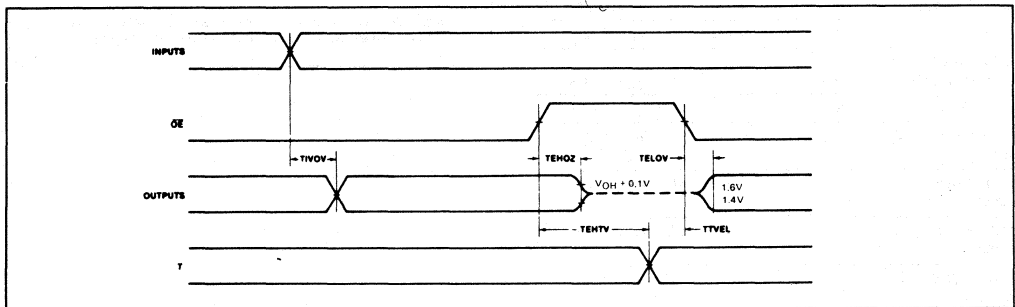
## AC Test Conditions



**Test Load Circuits**



**Timing Waveform**



## Description

The μPB8288 bus controller is used in medium to large μPD8086/μPD8088 systems. This 20-pin bipolar component provides command and control timing generation, as well as bipolar drive capability and optimal system performance. It provides both Multibus® command signals and control outputs for the microprocessor system. There is an option to use the controller with a multimaster system bus and separate I/O bus.

## Features

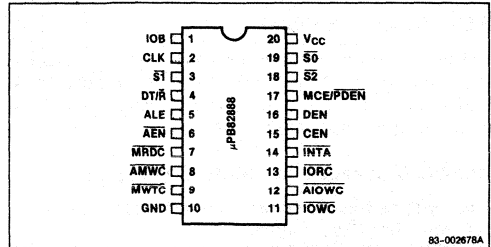
- System controller for μPD8086/μPD8088 systems
- Bipolar drive capability
- Provides advanced commands
- Three state output drivers
- Can be used with an I/O bus
- Enables interface to one or two multimaster buses

## Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPB8288D	20-pin cerdip	10 MHz

® Multibus is a registered trademark of Intel Corp.

## Pin Configuration



## Pin Identification

No.	Symbol	Function
1	IOB	I/O bus mode selector input
2	CLK	Clock input
3	S1	Status input
4	DT/R	Data transmit/receive
5	ALE	Address latch enable output
6	AEN	Address enable input
7	MRDC	Memory read command output
8	AMWC	Advanced memory write command output
9	MWTC	Memory write command output
10	GND	Ground
11	IOWC	I/O write command output
12	AIOWC	Advanced I/O write command output
13	IORC	I/O read command output
14	INTA	Interrupt acknowledge output
15	CEN	Command enable input
16	DEN	Data enable output
17	MCE/PDEN	Master cascade enable/peripheral data enable output
18	S2	Status input
19	S0	Status input
20	Vcc	+5 V power supply



**Pin Functions** **$\overline{\text{AEN}}$  (Address Enable)**

In the I/O system bus mode,  $\overline{\text{AEN}}$  enables the command outputs of the  $\mu\text{PB8288}$  105 ns after it becomes active. If  $\overline{\text{AEN}}$  is inactive, the command outputs become high impedance outputs.

 **$\overline{\text{AIOWC}}$  (Advanced I/O Write Command)**

This write command occurs earlier in the machine cycle than the  $\overline{\text{IOWC}}$  command.

 **$\overline{\text{ALE}}$  (Address Latch Enable)**

This signal is used for controlling transparent D-type latches ( $\mu\text{PB8282}/\mu\text{PB8283}$ ). It will strobe in the address on a high to low transition.

 **$\overline{\text{AMWC}}$  (Advanced Memory Write Command)**

This is an advanced write command which occurs early in the machine cycle, with timing the same as the read command.

 **$\overline{\text{CEN}}$  (Command Enable)**

This signal enables all command and control outputs. If  $\overline{\text{CEN}}$  is low, these outputs are inactive.

 **$\overline{\text{CLK}}$  (Clock)**

The clock signal from the  $\mu\text{PB8284}$  clock generator synchronizes the generation of command and control signals.

 **$\overline{\text{DEN}}$  (Data Enable)**

This signal enables the data transceivers onto the bus.

 **$\overline{\text{DT/R}}$  (Data Transmit/Receive)**

This signal is used to control the bus transceivers in a system; high for writing to I/O or memory and low for reading data.

 **$\overline{\text{INTA}}$  (Interrupt Acknowledge)**

$\overline{\text{INTA}}$  is used to signal an interrupting device to put the vector information on the data bus.

 **$\overline{\text{IOB}}$  (I/O Bus Mode)**

Sets mode of  $\mu\text{PB8288}$ ; high for the I/O bus mode and low for the system bus mode.

 **$\overline{\text{IORC}}$  (I/O Read Command)**

This signal enables the CPU to read data from an I/O device.

 **$\overline{\text{IOWC}}$  (I/O Write Command)**

This command is for transferring information to I/O devices.

 **$\overline{\text{MCE/PDEN}}$  (Master Cascade Enable/  
Peripheral Data Enable)**

Dual function pin system. ( $\overline{\text{MCE}}$ ) In the bus mode, this signal is active during an interrupt sequence to read the cascade address from the master interrupt controller onto the data bus. ( $\overline{\text{PDEN}}$ ) In the I/O bus mode, it enables the transceivers for the I/O bus just as  $\overline{\text{DEN}}$  enables bus transceivers in the system bus mode.

 **$\overline{\text{MRDC}}$  (Memory Read Command)**

This active low signal is for switching the data from memory to the data bus.

 **$\overline{\text{MWTC}}$  (Memory Write Command)**

This signal is used to transfer the data bus to memory, but not as early as  $\overline{\text{AMWC}}$ . (See timing waveforms).

 **$\overline{\text{GND}}$  (Ground)**

Ground.

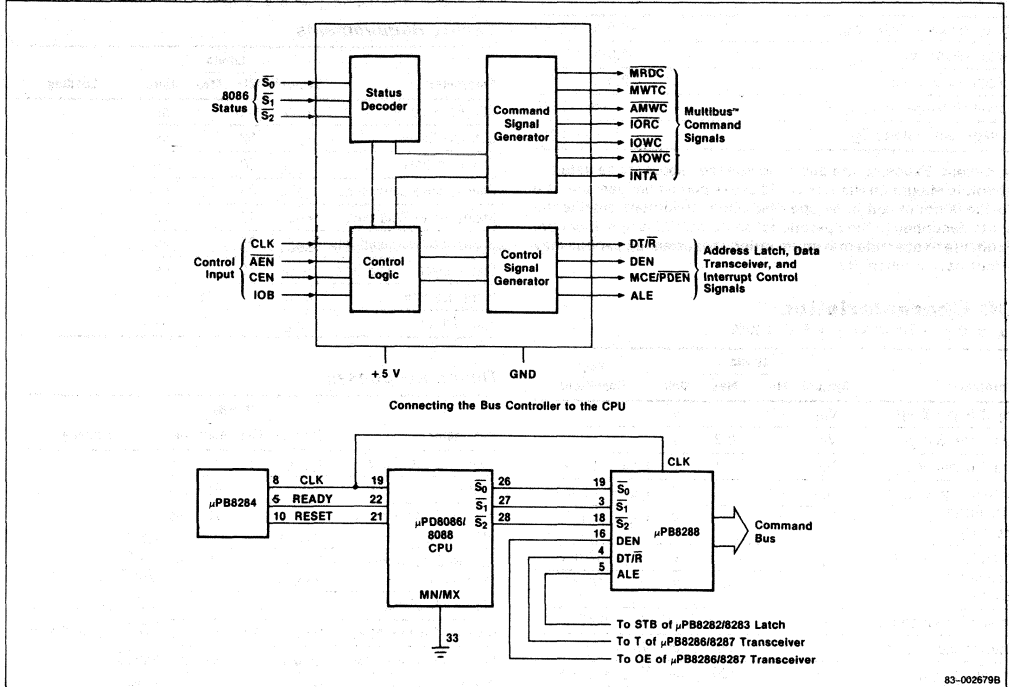
 **$\overline{\text{S0}}, \overline{\text{S1}}, \overline{\text{S2}}$  (Status Input Pins)**

The  $\mu\text{PB8288}$  decodes these status lines from the  $\mu\text{PD8086}$  to generate command and control signals. When not in use, these pins are high.

 **$\overline{\text{VCC}}$  (Power Supply)**

+5 V power supply.

## Block Diagram



### Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$

Power supply voltage, $V_{DD}$	-0.5 to +7.0 V
Input voltage, $V_I$	-1.0 to +5.5 V
Output voltage, $V_O$	-0.5 to +7.0 V
Operating temperature, $T_{OPR}$	0 to +70°C
Storage temperature, $T_{STG}$	-65 to +150°C

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum ratings for extended periods may affect device reliability.

### DC Characteristics

$T_A = 0$  to +70°C;  $V_{CC} = 5\text{ V} \pm 10\%$

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
Input high voltage	$V_{IH}$	2.0		V	
Input low voltage	$V_{IL}$		0.8	V	
Input clamp voltage	$V_C$		-1	V	$I_C = -5\text{ mA}$
Output high voltage (command)	$V_{OH}$	2.4		V	$I_{OH} = -5\text{ mA}$
Output high voltage (control)			2.4	V	$I_{OH} = -1\text{ mA}$
Output low voltage (command)	$V_{OL}$		0.5	V	$I_{OL} = 32\text{ mA}$
Output low voltage (control)			0.5	V	$I_{OL} = 16\text{ mA}$
Forward input current	$I_F$		-0.7	mA	$V_F = 0.45\text{ V}$
Reverse input current	$I_R$		50	μA	$V_R = V_{CC}$
Output off current	$I_{OFF}$		100	μA	$V_{OFF} = 0.4\text{ V}$ to 5.25 V
Power supply current	$I_{CC}$		230	mA	

### AC Characteristics

$T_A = 0$  to +70°C;  $V_{CC} = 5\text{ V} \pm 10\%$

#### Timing Requirements

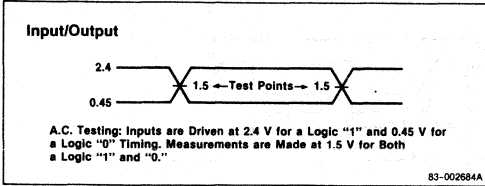
Parameter	Symbol	Limits		Unit	Loading
		Min	Max		
CLK cycle period	$t_{CLCL}$	100		ns	
CLK low time	$t_{CLCH}$	50		ns	
CLK high time	$t_{CHCL}$	30		ns	
Status active setup time	$t_{SVCH}$	35		ns	
Status active hold time	$t_{SHSV}$	10		ns	
Status inactive setup time	$t_{SHCL}$	35		ns	
Status inactive hold time	$t_{CLSH}$	10		ns	
Input rise time	$t_{ILIH}$		20	ns	
Input fall time	$t_{IHIL}$		12	ns	

#### Timing Responses

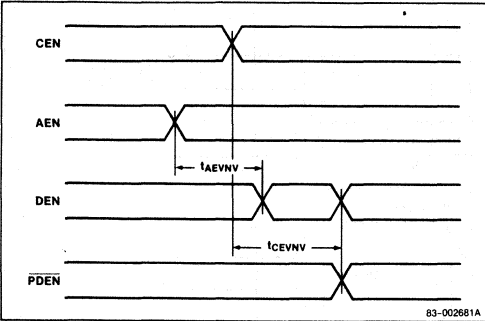
Parameter	Symbol	Limits		Unit	Loading
		Min	Max		
Control active delay	$t_{CVNV}$	5	45	ns	
Control inactive delay	$t_{CVNX}$	10	45	ns	
ALE MCE active delay (from CLK)	$t_{CLLH}/$ $t_{CLMCH}$		20	ns	
ALE MCE active delay (from status)	$t_{SVLH}$ $t_{SVMCH}$		20	ns	$I_{OL} = 32\text{ mA}$
ALE inactive delay	$t_{CHLL}$	4	15	ns	$I_{OH} = -5\text{ mA}$
Command active delay	$t_{CLML}$	10	35	ns	$C_L = 300\text{ pF}$
Command inactive delay	$t_{CLMH}$	10	35	ns	
Direction control active delay	$t_{CHDTL}$		50	ns	
Direction control inactive delay	$t_{CHDTH}$		30	ns	
Command enable time	$t_{AELCH}$		40	ns	
Command disable time	$t_{AECZ}$		40	ns	
Enable delay time	$t_{AELCV}$	115	200	ns	$I_{OL} = 16\text{ mA}$
AEN to DEN	$t_{AEVNV}$		20	ns	Other} $I_{OH} = -1\text{ mA}$
CEN to DEN, PDEN	$t_{CEVNV}$		25	ns	$C_L = 80\text{ pF}$
CEN to command	$t_{CELRH}$		$t_{CLML}$	ns	
Output rise time	$t_{OLOH}$		20	ns	
Output fall time	$t_{OHOL}$		12	ns	

## Timing Waveforms

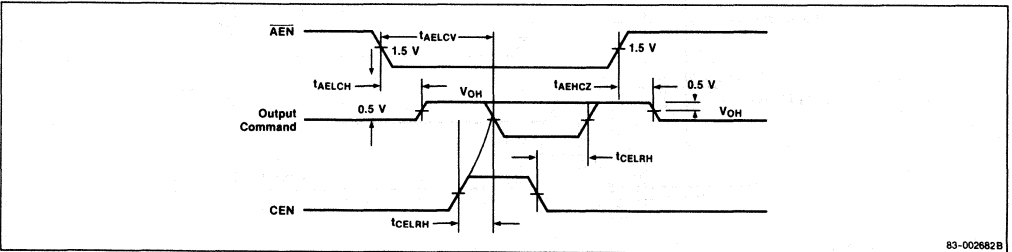
### Timing Measurement Points



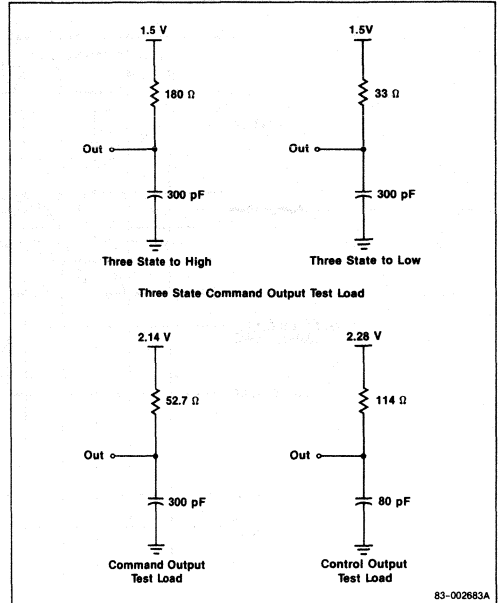
### DEN, PDEN Qualification Timing



### $\mu$ PD8288 Address Enable (AEN) Timing (Three State Enable/Disable)

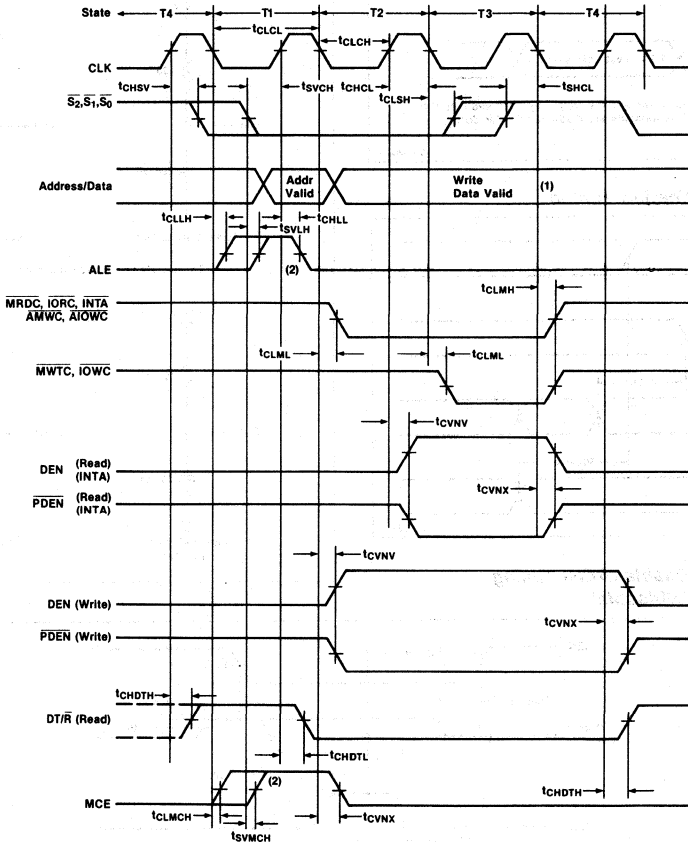


### Test Load Circuits



Timing Waveforms (cont)

State Timing



- Note:
- (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 1.5 V unless specified otherwise.

## Functional Description

The three status lines ( $\overline{S0}$ ,  $\overline{S1}$ ,  $\overline{S2}$ ) from the μPD8086 CPU are decoded by command logic within the μPB8288 to determine which command is to be issued. Table 1 below illustrates the decoding and command generation of the status lines.

**Table 1. Status Line Decoding**

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	μPD8086 State	μPB8288 Command
0	0	0	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

There are two ways the command is issued depending on the mode of the μPB8288.

The I/O bus mode is enabled if the IOB pin is pulled high. In this mode, all I/O command lines are always enabled and not dependent upon AEN. When the processor sends out an I/O command, the μPB8288 activates the command lines using  $\overline{PDEN}$  and  $\overline{DT/\overline{R}}$  to control any bus transceivers.

This mode is advantageous if I/O or peripherals dedicated to one microprocessor are in a multi-processor system, allowing the μPB8288 to control two external buses. No waiting is required when the CPU needs to access the I/O bus, as an  $\overline{AEN}$  low signal is needed to gain normal memory access.

If the IOB pin is tied to ground, the μPB8288 is in the system bus mode. In this mode, command signals are dependent upon the  $\overline{AEN}$  line. Thus the command lines are activated 105 ns after the  $\overline{AEN}$  line goes low. In this mode, there must be some bus arbitration logic to toggle the  $\overline{AEN}$  line when the bus is free for use. Here, both memory and I/O are shared by more than one processor, over one bus, with both memory and I/O commands waiting for bus arbitration.

Among the command outputs are some advanced write commands which are initiated early in the machine cycle and can be used to prevent the CPU from entering unnecessary wait states.

The  $\overline{INTA}$  signal acts as an I/O read during an interrupt cycle. This is to signal the interrupting device that its interrupt is being acknowledged, and to place the interrupt vector on the data bus.

The control outputs of the μPB8288 are used to control the bus transceivers in a system.  $\overline{DT/\overline{R}}$  determines the direction of the data transfer, and  $\overline{DEN}$  is used to enable the outputs of the transceiver. In the IOB mode the  $\overline{MCE/\overline{PDEN}}$  pin acts as a dedicated data enable signal for the I/O bus.

The MCE signal is used in conjunction with an interrupt acknowledge cycle to control the cascade address when more than one interrupt controller (such as a μPD8259A) is used. If there is only one interrupt controller in a system, MCE is not used because the  $\overline{INTA}$  signal gates the interrupt vector onto the processor bus. In multiple interrupt controller systems, MCE is used to gate the μPD8259A's cascade address onto the processor's local bus, where ALE strobes it into the address latches. This occurs during the first  $\overline{INTA}$  cycle. During the second  $\overline{INTA}$  cycle the addressed slave μPD8259A gates its interrupt vector onto the processor bus.

The ALE signal occurs during each machine cycle and is used to strobe data into the address latches and to strobe the status ( $\overline{S0}$ ,  $\overline{S1}$ ,  $\overline{S2}$ ) into the μPB8288. ALE also occurs during a halt state to accomplish this.

The  $\overline{CEN}$  (Command Enable) is used to control the command lines. If pulled high, the μPB8288 functions normally, and if grounded, all command lines are inactive.



## Description

The μPB8289 bus arbiter is used with the μPB8288 bus controller to interface 8086 and 8088 microprocessors to a multimaster system bus. The μPB8289 controls the μPB8288 bus controller and the bus transceivers and address latches, preventing them from accessing the system bus if the processor does not have use of the bus.

An external command sequence will cause the associated microprocessor to enter a wait state until the bus is ready. The processor remains in the wait state until the bus arbiter acquires use of the multimaster system bus. Then, the arbiter allows the bus controller, data transceivers, and address latches to access the system.

Once use of the bus has been acquired and data has been transferred, transfer acknowledge (XACK) is returned to the processor to indicate that the accessed slave device is ready. The processor may then complete its transfer cycle.

## Features

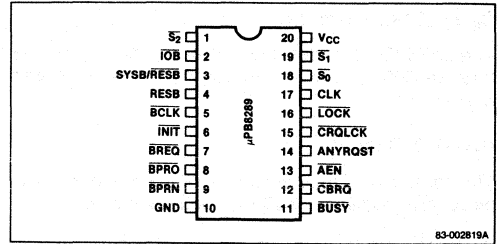
- Multimaster system bus protocol
- 8086 and 8088 processor synchronization with multimaster bus
- Simple interface with the 8288 bus controller and 8283/8282 address latches to a system bus
- Four operating modes for flexible system configuration
- Simplified interface to Multibus® systems
- Parallel, serial, and rotating priority resolution
- Bipolar buffering and drive capability

## Ordering Information

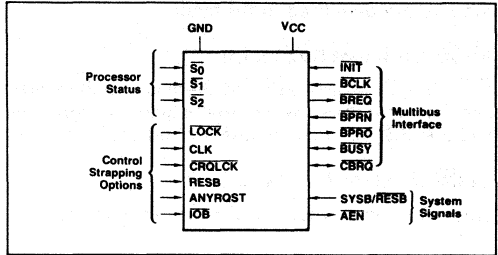
Part Number	Package Type	Max Frequency of Operation
μPB8289D	20-Pin Cerdip	8 MHz

Multibus is a registered trademark of Intel Corporation.

## Pin Configuration



## Functional Configuration



## Pin Identification

No.	Symbol	Function
1, 18, 19	S <sub>0</sub> -S <sub>2</sub>	Status inputs
2	IOB	I/O bus input
3	SYSB/RESB	System bus/resident bus inputs
4	RESB	Resident bus input
5	BCLK	System bus clock input
6	INIT	Initialize input
7	BREQ	Bus request output
8	BPRO	Bus priority output
9	BPRN	Bus priority input
10	GND	Ground
11	BUSY	Bus interface signal I/O
12	CBRQ	Common bus request I/O
13	AEN	Address enable output
14	ANYRQST	Any request input
15	CROLCK	Common request lock input
16	LOCK	Lock input
17	CLK	Clock input
20	Vcc	+5 V power supply



**Pin Functions** **$\overline{S_0}$ - $\overline{S_2}$  (Status Inputs)**

The μPB8289 decodes these status inputs from the 8086 or 8088 processor to begin bus requests and surrenders.

 **$\overline{IOB}$  (I/O Bus)**

This input signal tells the μPB8289 that there is an I/O peripheral bus and a multimaster system bus.

 **$\overline{SYSB}/\overline{RESB}$  (System Bus/Resident Bus)**

This input determines when bus requests and surrenders are permitted in SR mode.

 **$\overline{RESB}$  (Resident Bus Input)**

RESB tells the μPB8289 that there is a multimaster and resident bus. When the signal is high, the SYSB/RESB pin handles bus arbitration.

 **$\overline{BLCK}$  (System Bus Clock)**

This clock input synchronize all system bus interface signals.

 **$\overline{INIT}$  (Initialize)**

This active low-input resets all bus arbiters on the multimaster bus. No arbiters have use of the bus following INIT.

 **$\overline{BREQ}$  (Bus Request)**

An arbiter uses this output to request use of the multimaster system bus.

 **$\overline{BPRO}$  (Bus Priority Output)**

In serial priority resolving schemes, this output daisy-chains to  $\overline{BPRN}$  of the next lower priority arbiter.

 **$\overline{BPRN}$  (Bus Priority Input)**

This input tells the arbiter it may acquire the bus on the next falling edge at  $\overline{BCLK}$ .

 **$\overline{BUSY}$  (Bus Interface Signal)**

When the bus is available, this I/O signal notifies all arbiters on the bus. The highest requesting arbiter seizes the bus and pulls BUSY low to keep other arbiters off the bus.

 **$\overline{CBRQ}$  (Common Bus Request)**

This signal is an input from a lower priority arbiter requesting the bus. It is an output from arbiters that surrender the multimaster bus upon request.

 **$\overline{AEN}$  (Address Enable)**

This output tells the 8288 bus controller, 8284 clock driver, and the processor's address latches when to tri-state their output drivers.

 **$\overline{ANYRQST}$  (Any Request)**

This signal allows the multimaster bus to be surrendered to a lower priority arbiter.

 **$\overline{CRQLCK}$  (Common Request Lock)**

This input prevents the μPB8289 from surrendering the bus in response to a request on the  $\overline{CBRQ}$  input.

 **$\overline{LOCK}$  (Lock)**

This input prevents the arbiter from surrendering the multimaster system bus to any other bus arbiter, regardless of its priority.

 **$\overline{CLK}$  (Clock)**

This is the clock signal from the 8284 clock generator.

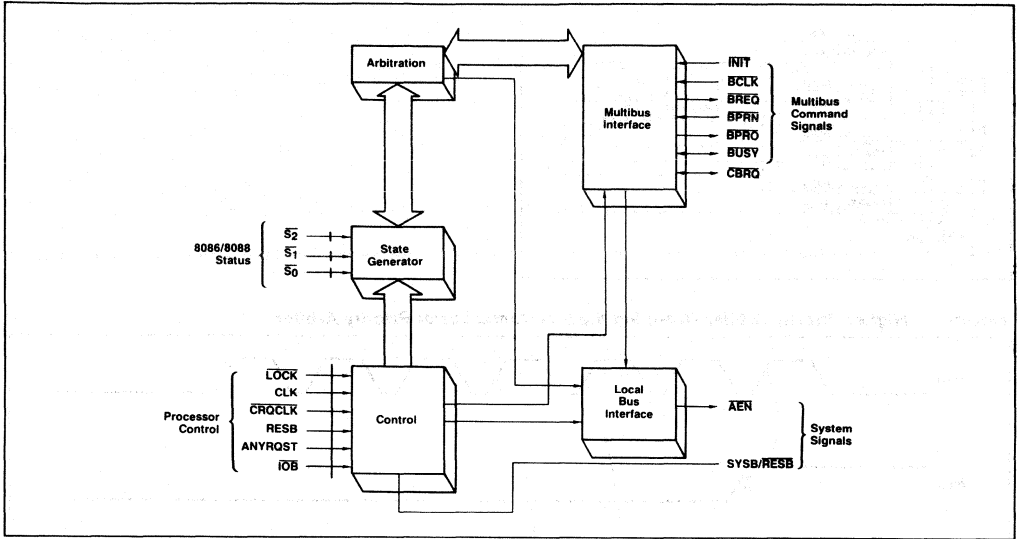
 **$\overline{GND}$  (Ground)**

This is the ground.

 **$\overline{V_{CC}}$  (Power Supply)**

This is the +5 V power supply.

## Block Diagram



## Functional Description

### Bus Master Arbitration

Higher priority masters generally acquire use of the bus when a lower priority master completes its present transfer cycle. Lower priority masters acquire the bus when no higher priority master is accessing the system bus. The ANYRQST strapping option allows the arbiter to surrender the bus to a lower priority master as if it were a higher priority master. The arbiter maintains the bus as long as no other bus masters are requesting the bus and its processor has not entered the halt state. The arbiter does not voluntarily surrender the bus and must be forced off by a request from another bus master, unless the arbiter's processor has entered the halt state. Additional strapping options allow for other sets of conditions.

### Priority Resolving Techniques

The μPB8289 provides several techniques for resolving priority between the many possible bus masters of a multimaster system bus. All of these techniques assume that one bus master will have priority over all others at any given time. You may use parallel, serial, or rotating priority resolving.

### Parallel Priority Resolving

This technique (figures 1, 2) uses a bus request line (BREQ) for each arbiter on the multimaster system bus. Each BREQ line goes to a priority encoder that generates the address of the highest priority active BREQ line. This binary address is decoded to select the bus priority in line (BPRN) that is returned to the highest priority arbiter. The arbiter that receives priority (BPRN true) allows its bus master onto the multimaster system bus as soon as the bus becomes available. An arbiter that gets priority over another arbiter cannot immediately seize the bus, but must wait until the current bus transaction is complete. When the transaction is complete, the current occupant of the bus surrenders the bus by releasing BUSY. BUSY is an active low OR tied line which goes to every arbiter on the system bus. When BUSY goes high (inactive), the priority arbiter seizes the bus and brings BUSY low to keep other arbiters off the bus. Note that all multimaster system bus transactions are synchronized to the bus clock (BCLK).

7

Figure 1. Parallel Priority Resolving

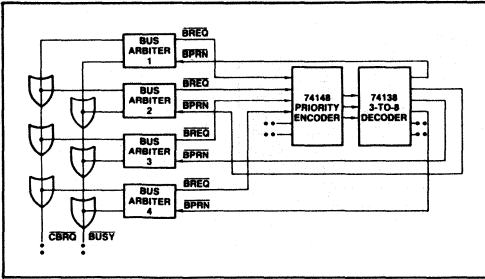
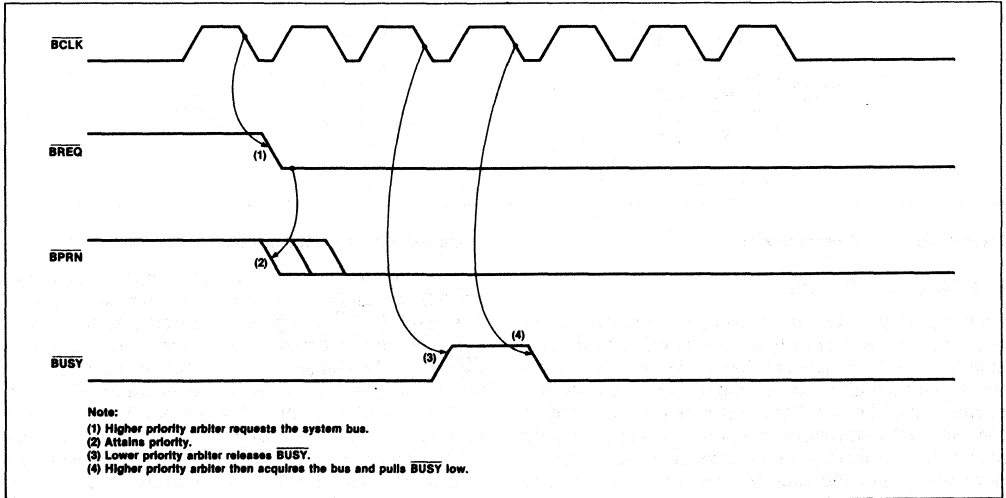


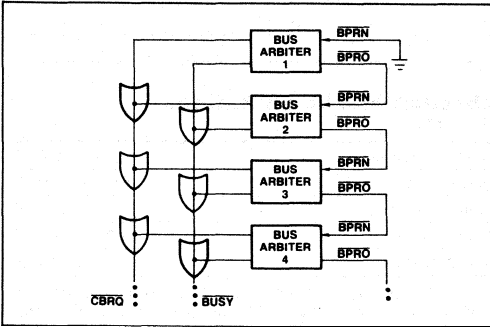
Figure 2. Higher Priority Arbiter Obtaining the Bus from a Lower Priority Arbiter



### Serial Priority Resolving

The serial priority resolving technique (figure 3) daisy-chains the bus arbiters together by connecting the higher priority arbiter's  $\overline{BPRQ}$  output to the  $\overline{BPRN}$  of the next lowest priority arbiter. This eliminates the need for the priority encoder-decoder arrangement. The number of arbiters that may be daisy-chained together is a function of  $\overline{BCLK}$  and the propagation delay from arbiter to arbiter. At 10 MHz, only 3 arbiters may be daisy-chained.

**Figure 3. Serial Priority Resolving**



### Rotating Priority Resolving

This technique resembles the parallel priority resolving technique except that priority is dynamically reassigned. The priority encoder is replaced by a circuit that rotates priority between arbiters to allow each arbiter an equal chance to use the system bus.

### Modes of Operation

The  $\mu$ PB8289 has two basic operating modes: I/O peripheral bus mode ( $\overline{IOB}$  mode), and resident bus mode (RESB mode). The  $\overline{IOB}$  strapping option configures the  $\mu$ PB8289 into  $\overline{IOB}$  mode and the RESB strapping option configures it to RESB mode. If both options are strapped false, the arbiter interfaces the processor to a multimaster system bus only. If both options are strapped true, the arbiter interfaces the processor to a multimaster system bus, a resident bus, and an I/O bus. Figure 4 shows the  $\mu$ PB8289 in a typical CPU system.

### $\overline{IOB}$ Mode

$\overline{IOB}$  mode allows the processor to access both an I/O peripheral bus and a multimaster system bus. On an I/O peripheral bus, all devices on the bus, including memory, are treated as I/O devices and addressed by I/O commands. All memory commands are directed to the multimaster system bus. In  $\overline{IOB}$  mode, the processor communicates with and controls peripherals over the peripheral bus and communicates with system memory over the system memory bus. Figure 5 shows the  $\mu$ PB8289 in this mode.

### RESB Mode

RESB mode allows the processor to communicate over both a resident bus and a multimaster system bus. A resident bus can issue memory and I/O commands, but it is separate from the multimaster system bus. The resident bus has one master and is dedicated to only that master. The 8086 and 8088 can communicate with a resident bus and a multimaster system bus. The processor can access the memory and peripherals of both buses. Memory mapping selects which bus is accessed. The  $\overline{SYSB}/\overline{RESB}$  input on the arbiter instructs the arbiter on which bus to access. The signal connected to  $\overline{SYSB}/\overline{RESB}$  also enables and disables commands from one of the bus controllers. Figure 6 shows the  $\mu$ PB8289 in this mode.

### Mode Summary

	Status Lines From 8086 or 8088 or 8089			IOB Mode Only	RESB (Mode) Only IOB = High RESB = High		IOB Mode RESB Mode IOB = Low RESB = High		Single Bus Mode IOB = High RESB = Low	
	S2	S1	S0		IOB = Low	SYSB/RESB =		SYSB/RESB =		
						High	Low	High		Low
I/O commands	0	0	0	x	✓	x	x	x	✓	
	0	0	1	x	✓	x	x	x	✓	
	0	1	0	x	✓	x	x	x	✓	
Halt	0	1	1	x	x	x	x	x	x	
Memory commands	1	0	0	✓	✓	x	✓	x	✓	
	1	0	1	✓	✓	x	✓	x	✓	
	1	1	0	✓	✓	x	✓	x	✓	
Idle	1	1	1	x	x	x	x	x	x	

**Notes:**

- (1) x = Multimaster system bus is allowed to be surrendered.
- (2) ✓ = Multimaster system bus is requested.

### Multimaster System Bus

Mode	Pin Strapping	Requested (1)	Surrendered (2)
Single bus multimaster mode	IOB = high RESB = low	When the processor's status lines go active	HLT + TI • HPBRQ†
RESB mode only	IOB = high RESB = high	SYSB/RESB = High 2 active	(SYSB/RESB = low + TI) CBRQ + HLT + HPBRQ
IOB mode only	IOB = low RESB = low	Memory commands	(I/O status + TI) • CBRQ + HLT + HPBRQ
IOB mode • RESB mode	IOB = low RESB = high	(Memory command) • (SYSB/RESB = high)	(I/O status commands) + (TI) (SYSB/RESB = low) • CBRQ + HPBRQ† + HLT)

**Note:**

- (1) Except for HALT and idle status.
  - (2) LOCK prevents surrender of bus to any other arbiter. CRQLCK prevents surrender of bus to a lower priority arbiter.
  - (3) HLT = processor halt; S2S0 = 011.
  - (4) TI = processor idle; S2S0 = 111.
  - (5) + means OR.
  - (6) • means AND.
- † HPBRQ = higher priority bus request or BPRN = 1.

### Absolute Maximum Ratings

TA = 25°C

Operating temperature	0°C to 70°C
Storage temperature	-65°C to +150°C
Voltage on any pin	-0.5 V to +7 V
All input voltages	-1.0 V to +5.5 V
Power dissipation	1.5 W

**Comment:** Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### DC Characteristics

TA = 0°C to +70°C; VCC = 5 V ± 10%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input low voltage	VIL			0.8	V	
Input high voltage	VIH	2.0			V	
Input clamp voltage	VC			-1.0	V	VCC = 4.50 V, IC = -5 mA
Input forward current	IF			-0.5	mA	VCC = 5.50 V, VF = 0.45 V
Reverse input leakage current	IR			60	μA	VCC = 5.50 V, VR = 5.50 V
Output low voltage	VOL					
BUSY, CBRQ				0.45	V	IOL = 20 mA
AEN				0.45	V	IOL = 16 mA
BPRO, BREQ				0.45	V	IOL = 10 mA
Output high voltage	VOH	Open collector				
BUSY, CBRQ		2.4			V	I0H = 400 μA
All other outputs						
Power supply current	ICC			165	mA	

## Capacitance

Parameter	Symbol	Limits			Test Conditions
		Min	Typ	Max	
Input capacitance	C <sub>IN</sub> Status			25	pF
Input capacitance	C <sub>IN</sub> (others)			12	pF

**Note:**

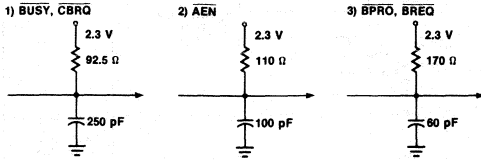


Figure 4. Typical CPU System Using the μPD8289 Bus Arbiter

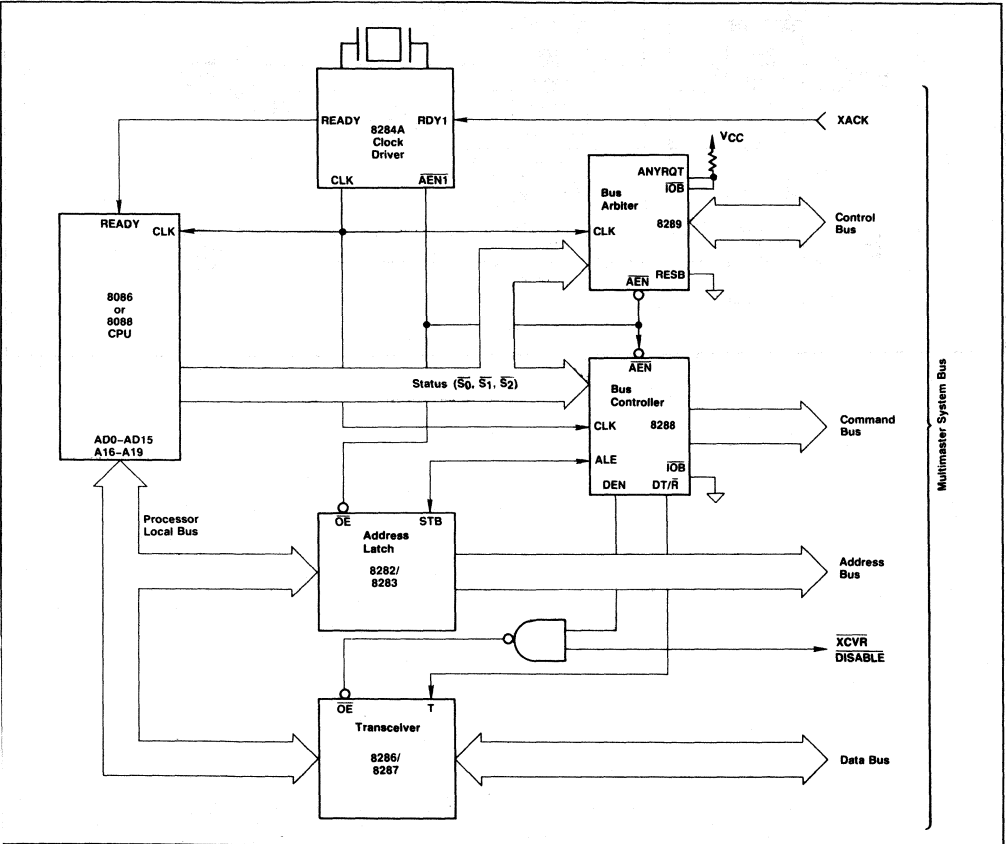


Figure 5. Typical Medium-Complexity IOB System

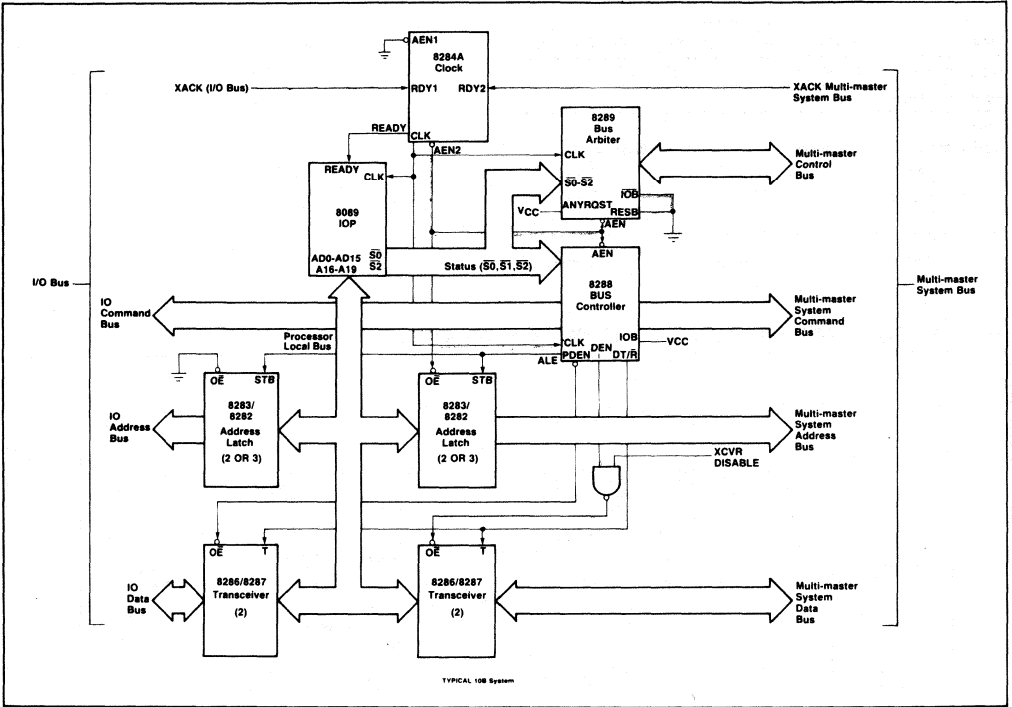
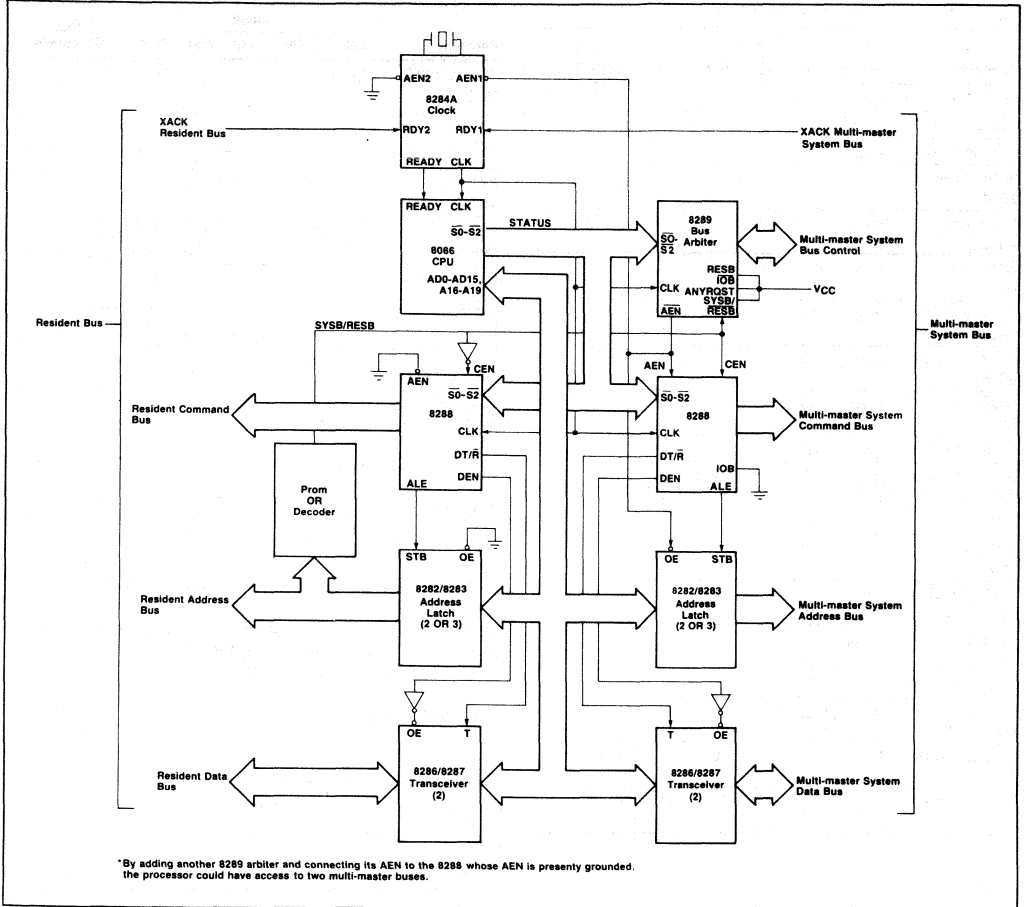


Figure 6. Typical System, Resident Bus Configuration





**AC Characteristics**

T<sub>A</sub> = 0°C to +70°C; V<sub>CC</sub> = 5V ± 10%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
CLK cycle period	t <sub>CLCL</sub>	125			ns	
CLK low time	t <sub>CLCH</sub>	65			ns	
CLK high time	t <sub>CHCL</sub>	35			ns	
Status active setup	t <sub>SVCH</sub>	65		t <sub>CLCL</sub> - 10	ns	
Status inactive setup	t <sub>SHCL</sub>	50		t <sub>CLCL</sub> - 10	ns	
Status active hold	t <sub>HVCH</sub>	10			ns	
Status inactive hold	t <sub>HVCL</sub>	10			ns	
BUSY↑ setup to BCLK↓	t <sub>BYSBL</sub>	20			ns	
CBRQ↑ setup to BCLK↓	t <sub>CBSBL</sub>	20			ns	
BCLK cycle time	t <sub>BLBL</sub>	100			ns	
BCLK high time	t <sub>BHCL</sub>	30	0.65 (t <sub>BLBL</sub> )		ns	
LOCK inactive hold	t <sub>CLLL1</sub>	10			ns	
LOCK active setup	t <sub>CLLL2</sub>	40			ns	
BPRN↑ to BCLK setup time	t <sub>PNBL</sub>	15			ns	
YSB/RESB setup	t <sub>CLSR1</sub>	0			ns	
YSB/RESB hold	t <sub>CLSR2</sub>	20			ns	
Initialization pulse width	t <sub>VIH</sub>	3 t <sub>BLBL</sub> + 3 t <sub>CLCL</sub>			ns	
Input rise time	t <sub>ILIH</sub>			20	ns	From 0.8 V to 2.0 V
Input fall time	t <sub>IHL</sub>			12	ns	From 2.0 V to 0.8 V

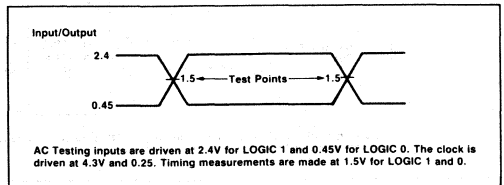
**Timing Response**

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
BCLK to BREQ delay (1)	t <sub>BLBRL</sub>			35	ns	
BCLK to BPRO (1)(2)	t <sub>BLPOH</sub>			40	ns	
BPRN↑ to BPRO↑ delay (1)(2)	t <sub>PNPO</sub>			25	ns	
BCLK to BUSY low	t <sub>BLBYL</sub>			60	ns	
BCLK to BUSY float (3)	t <sub>BLBYH</sub>			35	ns	
CLK to AEN high	t <sub>CLAEH</sub>			65	ns	
BCLK to AEN low	t <sub>BLAEL</sub>			40	ns	
BCLK to CBRQ low	t <sub>BLCBL</sub>			60	ns	
BCLK to CBRQ float (3)	t <sub>RLCRH</sub>			35	ns	
Output rise time	t <sub>OLOH</sub>			20	ns	From 0.8 V to 2.0 V
Output fall time	t <sub>OHL</sub>			12	ns	From 2.0 V to 0.8 V

**Note:**

- (1) Denotes that the spec applies to both transitions of the signal.
- (2) BCLK generates the first BPRO. Subsequent changes of BPRO are generated through BPRON.
- (3) Measured at 0.5 V above GND.

**AC Test Condition**



AC Testing inputs are driven at 2.4V for LOGIC 1 and 0.45V for LOGIC 0. The clock is driven at 4.3V and 0.25. Timing measurements are made at 1.5V for LOGIC 1 and 0.

## Timing Waveforms

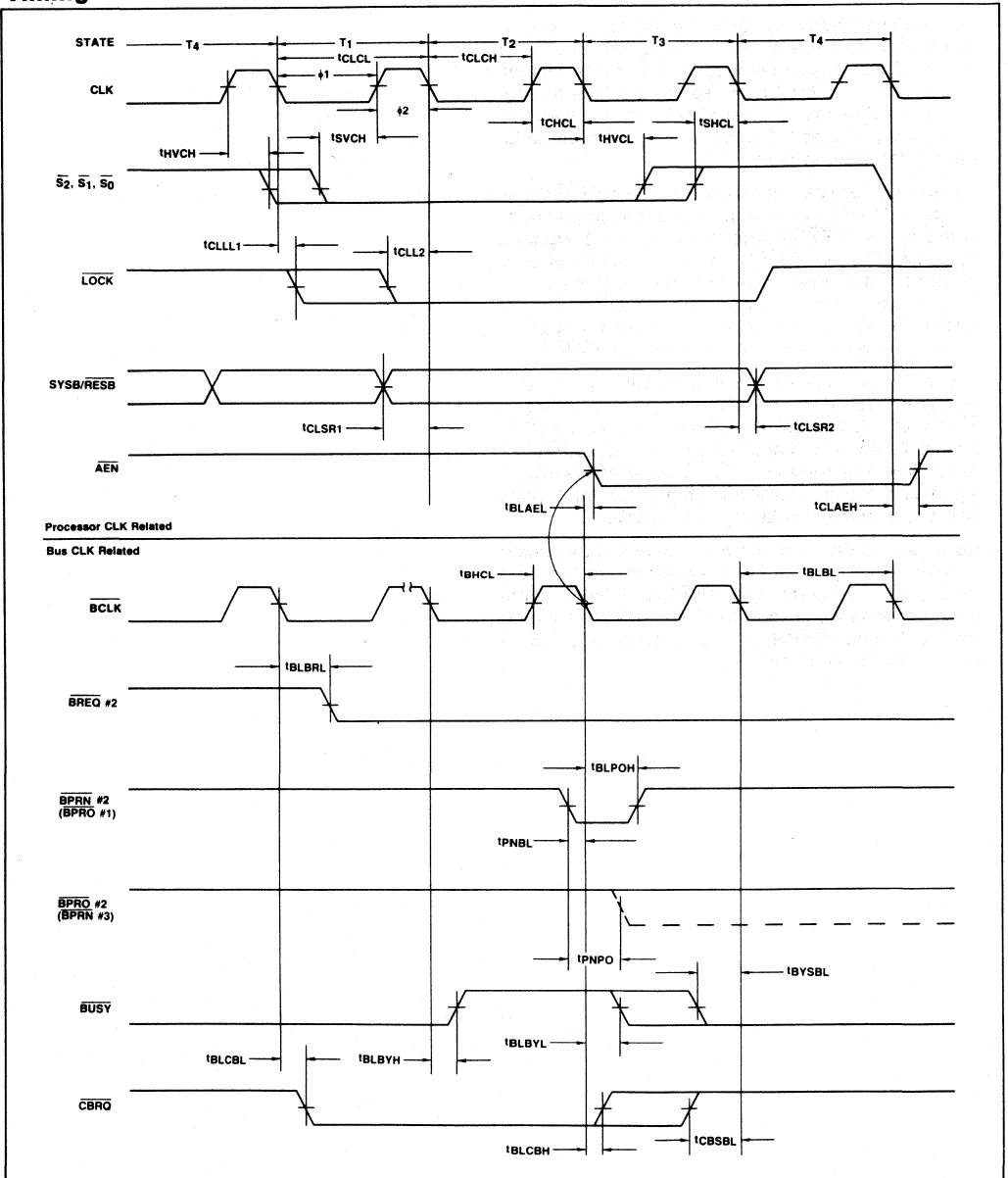
The signals related to CLK are typical processor signals and do not relate to the depicted sequence of events of the signals referenced to BCLK. The signals shown related to the BCLK represent a hypothetical sequence of events for illustration. Assume three bus arbiters of priorities 1, 2, and 3 configured in the serial priority resolving scheme.

Assume arbiter 1 has the bus and is holding  $\overline{\text{BUSY}}$  low. Arbiter 2 detects its processor wants the bus and pulls  $\overline{\text{BREQ}} \#2$  low. If  $\overline{\text{BPRN}} \#2$  is high (as shown), arbiter 2 pulls  $\overline{\text{CBRQ}}$  low.  $\overline{\text{CBRQ}}$  signals to higher-priority arbiter 1 that a lower-priority arbiter wants the bus. A higher-priority arbiter would be given  $\overline{\text{BPRN}}$  when it makes the bus request rather than having to wait for another arbiter to release the bus through  $\overline{\text{CBRQ}}$ .

Arbiter 1 relinquishes the multimaster system bus when it enters a state of not requiring it, by lowering its  $\overline{\text{BPRO}} \#1$  (tied to  $\overline{\text{BPRN}} \#2$ ) and releasing  $\overline{\text{BUSY}}$ . Arbiter 2 now sees that it has priority from  $\overline{\text{BPRN}} \#2$  being low and releases  $\overline{\text{CBRQ}}$ . As soon as  $\overline{\text{BUSY}}$  signifies the bus is available (high), arbiter 2 pulls  $\overline{\text{BUSY}}$  low on the next falling edge of BCLK.

Note that if arbiter 2 didn't want the bus at the time it received priority, it would pass priority to the next lower priority by lowering its  $\overline{\text{BPRO}} \#2$  (TPNPO). Note also that even a higher-priority arbiter which is acquiring the bus through  $\overline{\text{BPRN}}$  will momentarily drop  $\overline{\text{CBRQ}}$  until it has acquired the bus.

Timing Waveforms



## DEVELOPMENT TOOLS

## Section 8 – Development Tools V-Series Microprocessors

RA70116	Relocatable Assembler for V20/V30/V40/V50 .....	8.3
RA70320	Relocatable Assembler for V25/V35 .....	8.7
CC70116	C-compiler for V20/V30/V40/V50/V25 and V35 .....	8.11
RC70116	Relocatable Object Format (ROF) Converter .....	8.13
NECTOMS	Converter to change relocatable object code from NEC format into Microsoft (MS) format .....	8.15
MSTONEC	Converter to change relocatable object code from Microsoft (MS) format into NEC format .....	8.17
PKG70616	Software Generation Tool package for V60/V70 .....	8.19
IE-700K	Stand alone In-Circuit Emulator for V-Series .....	8.23
Mini-IE	Mini emulator for V40/V50 and V25 .....	8.43
EBIBM-70108/70116	Design-/Evaluation board for V20 and V30 .....	8.47
EBIBM-70208/70216	Design-/Evaluation board for V40 and V50 .....	8.49
EBIBM-7061UNIX	V60 coprocessor slave board with UNIX for PC-XT/AT .....	8.51
EBIBM-70616SBC	V60 single board computer for Multibus 1 .....	8.53

### Intelligent Peripherals and Dataflow Processors

EBIBM-72001	IBM-XT/AT board for $\mu$ PD72001 .....	8.55
EBIBM-7220A	IBM-XT/AT board for $\mu$ PD7220A .....	8.57
EBIBM-72120	IBM-XT/AT board for $\mu$ PD72120 .....	8.59
EBIBM-72067	IBM-XT board for $\mu$ PD72067 .....	8.61
EBIBM-7261A	IBM-XT board for $\mu$ PD7261A and $\mu$ PD2306A .....	8.65
EBIBM-ATDISK	IBM-AT board for $\mu$ PD7261A, $\mu$ PD9306A, $\mu$ PD72065 and $\mu$ PD71066 .....	8.67
EBIBM-7262	IBM-AT board for $\mu$ PD7262 and $\mu$ PD72067 .....	8.71
Functional Assembler $\mu$ PD7281	Software package for $\mu$ PD7281, functional Assembler, Simulator and Object Converter .....	8.73
SX-7281	Multibus boards for $\mu$ PD7281 .....	8.75
EBIBM-7281	IBM-XT/AT board for $\mu$ PD7281 and $\mu$ PD9305 .....	8.77
CDMSD-I5DD-7229	Character generation software for $\mu$ PD7229 LCD controller .....	8.79

## Description

The RA70116 Relocatable Assembler package converts symbolic source code for the V20/V30/V40/V50 ( $\mu$ PD70108/116/208/216) microprocessors into executable absolute address object code. The V20-V50 Relocatable Assembler package consists of four separate programs: an assembler (RA70116), a linker (LK70116), a hexadecimal format object code converter (OC70116), and a librarian (LB70116).

RA70116 translates a symbolic source module into a relocatable object module. The assembler verifies that each instruction assembled is valid for the V20-V50 microprocessors, and produces a listing file and a relocatable object module.

LK70116 combines relocatable object modules and absolute load modules and converts them into an absolute load module. OC70116 converts an absolute object module or an absolute load module to an ASCII hexadecimal format object file.

LB70116 allows commonly used relocatable object modules to be stored in one file and linked into multiple programs, greatly increasing programming efficiency. When a library file is included in the input of the linker, the linker extracts only those modules required to resolve external references from the file and relocates and links them.

## Features

- Absolute address object code output
- User-selectable and directable output files
- Extensive error reporting
- Powerful Librarian
- Runs under a variety of operating systems
  - CP/M-86™
  - MS-DOS™
  - ISIS III™
  - VAX/VMS™ and VAX/UNIX™

## PROGRAM SYNTAX

An RA70116 source module consists of a series of code, data, or stack segments. Each segment consists of statements composed of up to four fields: symbol, mnemonic, operand, and comment.

The symbol field may contain a label, whose value is the instruction or data address; or a name which represents an instruction address, data address, or constant. The mnemonic field may contain an instruction or assembler directive. The operand field contains the data or expression for the specified instruction or directive. The comment field allows you to add explanatory comments to your programs.

Character constants are translated into 7-bit ASCII codes. Numeric constants may be specified as binary, octal, decimal, or hexadecimal. Arithmetic expressions may include the operators +, -, \*, /, MOD, OR, AND, NOT, XOR, EQ, NE, LT, LE, GT, GE, SHR, SHL, LOW, HIGH, PTR, SHORT, THIS, SEG, OFFSET, TYPE, LENGTH, SIZE, MASK, WIDTH, (), [], ., .:, and <>.

## ASSEMBLER DIRECTIVES

Assembler directives give instructions to the assembler but are not translated into machine code during assembly. Basic assembler directives include: storage definition and allocation directives (DB, DW, DD, DBS, DWS, DDS, STRUC/ENDS, RECORD); symbol directives (EQU, LABEL, PURGE); and location counter control directives (ORG, EVEN, ALIGN). Program control directives include: segment definition and control directives (SEGMENT/ENDS, PROC/ENDP, ASSUME, GROUP, END); linkage directives (NAME, PUBLIC, EXTRN); and the PARITY directive.

The relocation types for SEGMENT/ENDS directives are specified in the operand column. These include: BYTE, WORD, PARA, PAGE and INPAGE. The combination types of PUBLIC, COMMON, AT, STACK and MEMORY, also specified in the operand column, define how to link segments and groups of the same name.

V20, V30, V40 and V50 are trademarks of NEC Corporation. CP/M-86 is a registered trademark of Digital Research. MS-DOS is a registered trademark of Microsoft.

VAX and VMS are registered trademarks of Digital Equipment Corporation. UNIX is a trademark of AT & T. ISIS is a trademark of Intel Corporation.

**ASSEMBLER CONTROLS**

There are two types assembler controls available for RA70116. Basic controls, which are specified in the assembler command line, are:

- File specification
- Output file selection
- Output file destination
- Listing format controls
- Date specification

General controls, which are specified in the source program, are:

- Inclusion of other source files
- Page eject
- Generation/suppression of listing
- Listing titles

The listing file may contain the complete assembly listing or only lines with errors, and a symbol or cross-reference table. The symbol table shows all defined symbols in alphabetical order, their types, attributes, and the values initially assigned to them. The cross-reference table contains all defined symbols and the numbers of all statements that refer to them.

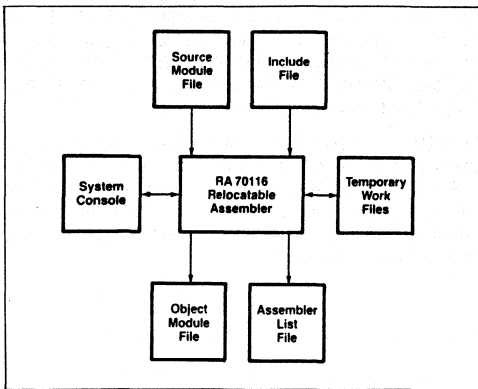
The object file contains the relocatable object module. The format of this module is an NEC proprietary relocatable object module format. Figure 1 is a functional diagram of the assembler.

**LINKER**

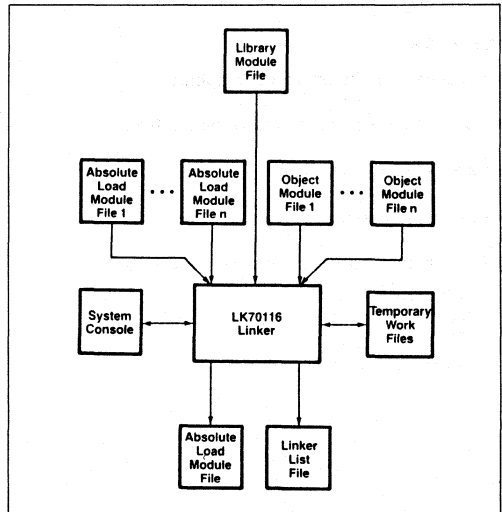
The linker, LK70116, combines relocatable object modules and absolute load modules and produces one absolute load module. Figure 2 is the functional diagram of the linker. The controls for LK70116 may be specified in either the command line or in a parameter file. You may specify the date, the absolute load module name, and the starting address and order for code/data/stack segments; you may also protect areas of memory from being assigned. Further, you may specify that a list file be created containing a link map, a local symbol table, or a public symbol table. The absolute load module contains symbol information for the symbolic debugger and absolute object code.

**Hexadecimal Format Object Converter**

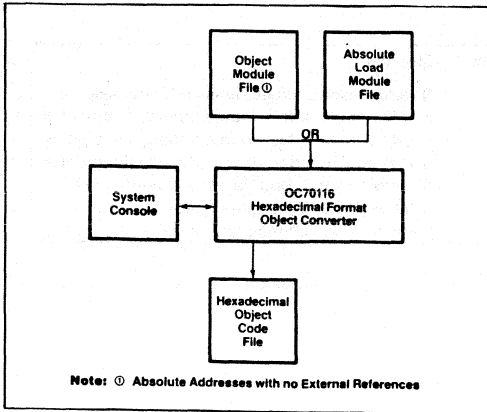
The object code converter, OC70116, translates an absolute load module file into an expanded hexadecimal format (7-bit ASCII) file, which may be downloaded to a PROM programmer. You may specify that addresses be output in the order in which they were input or be output in ascending order. Figure 3 is a functional diagram of the hexadecimal format object code converter.



**Figure 1. Relocatable Assembler Functional Diagram**



**Figure 2. Linker Functional Diagram**



**Figure 3. Hexadecimal Format Object Code Converter Functional Diagram**

## Librarian

The librarian LB70116, creates and maintains files containing relocatable object modules. This reduces on the number of files which need to be linked together by allowing several modules to be kept in a single file, and provides an easy way to link frequently-used modules into programs. Modules may be added to, deleted from, or replaced within a library file.

## Operating Environment

The NEC RA70116 package can be supplied to run under a variety of operating systems. Versions are available to run on a CP/M-86 or MS-DOS system with one or more disk drives and at least 128K of system memory. Another version is available to run on an Intel Intellec (tm) Series III development system with the ISIS operating system, one or more disk drives and at least 128K of installed system memory. Other versions are available to run on a Digital Equipment Corporation VAX computer under both the UNIX 4.2 BSD and the VMS (Version 4.1 or later) operating systems.

## Emulator Communication Program

V20 through V50 absolute load modules produced by the RA70116 Relocatable Assembler package can be debugged by using the appropriate NEC stand-alone In-Circuit Emulator: IE-70108, IE-70116, IE-70208, IE-70216. Communication between these emulators and your host system is through an RS-232C serial line utilizing the KERMIT communications protocol developed at Columbia University. With the appropriate version of the KERMIT Communication Program running on both the emulator and your host system, V20 through V50 absolute load modules or hexadecimal format object code files may be transferred between machines. Depending upon your development environment, the emulator may be controlled directly from the console of your host system or from a user supplied terminal.

A version of the KERMIT Communication Program is supplied with each of these emulators. NEC Electronics supplies versions of the KERMIT to run of the following systems: the IBM PC™ under MS-DOS or CP/M-86, the VAX under VMS or UNIX and on the Intel Intellec Series II/III ISIS-II based development system. Also for NEC's APC under CP/M-86 and for the Siemens PCD under MS-DOS one version each is available. An appropriate version is provided with each relocatable assembler package at no extra charge. Versions of KERMIT for other host systems may be available directly from Columbia University.

## Documentation

For further information on source program formats, assembler operation, and actual program examples, NEC Electronics provides the following documentation:

μPD70108, μPD70116, μPD70208, μPD70216 Relocatable Assembler User's Manual.

This documentation is provided with purchased copies of this product. Additional copies may be obtained from NEC Electronics.

## License Agreement

RA70116 is sold under terms of a license agreement which is included with purchased copies of the assembler. The accompanying card must be completed and returned to NEC Electronics (Europe) GmbH to register the license. Software updates are provided to registered users only.



### Ordering Information

Product No.	Host System	Operating System	Medium
RAC86-08SS-70116	APC	CP/M-86	8" single sided single density floppy disk
RAC86-I5DD-70116	IBM-XT	CP/M-86	5 1/4" double sided double density floppy disk
RAMSD-I5DD-70116	IBM-XT/AT	PC-DOS	5 1/4" double sided double density floppy disk
RASPD-I5DD-70116	PCD (Siemens)	MS-DOS	5 1/4" double sided double density floppy disk
RAIS3-08SD-70116	INTEL MDS	ISIS III	8" single sided double density floppy disk
RAUNX-0T16-70116	VAX	ULTRIX	9-track, 1600 BPI magnetic tape
RAVMS-0T16-70116	VAX	VMS	9-track, 1600 BPI magnetic tape

## Description

The RA70320 Relocatable Assembler package converts symbolic source code for the V25/V35/ $\mu$ PD79011/ $\mu$ PD79021 ( $\mu$ PD70320/70330/79011/79021) microcomputers into executable absolute address object code. The V25/V35/ $\mu$ PD79011/ $\mu$ PD79021 Relocatable Assembler package consists of four separate programs: an assembler (RA70320), a linker (LK70320), a hexadecimal format object code converter (OC70320), and a librarian (LB70320).

RA70320 translates a symbolic source module into a relocatable object module. The assembler verifies that each instruction assembled is valid for the V25/V35/ $\mu$ PD79011/ $\mu$ PD79021 microcomputers, and produces a listing file and a relocatable object module.

LK70320 combines relocatable object modules and absolute load modules and converts them into an absolute load module. The load module contains the program symbols and can be down loaded to incircuit emulator for symbolic debugging. OC70320 converts an absolute object module or an absolute load module to an ASCII hexadecimal format object file.

LB70320 allows commonly used relocatable object modules to be stored in one file and linked into multiple programs, greatly increasing programming efficiency. When a library file is included in the input of the linker, the linker extracts only those modules required to resolve external references from the file and relocates and links them.

## FEATURES

- Absolute address object code output
- User-selectable and directable output files
- Extensive error reporting
- Powerful Librarian
- Runs under a variety of operating systems
  - CP/M-86™
  - MS-DOS™
  - ISI-III™
  - VAX/VMS™ and VAX/UNIX™

V25 and V35 are trademarks of NEC Corporation  
CP/M-86 is a registered trademark of Digital Research.  
MS-DOS is a registered trademark of Microsoft.  
ISI is a trademark of Intel Corporation.  
VAX and VMS are registered trademarks of Digital Equipment corp.  
UNIX is a trademark of AT&T.

## Program Syntax

An RA70320 source module consists of a series of code, data, or stack segments. Each segment consists of statements composed of up to four fields: symbol, mnemonic, operand, and comment.

The symbol field may contain a label, whose value is the instruction or data address; or a name which represents an instruction address, data address, or constant. The mnemonic field may contain an instruction or assembler directive. The operand field contains the data or expression for the specified instruction or directive. The comment field allows you to add explanatory comments to your programs.

Character constants are translated into 7-bit ASCII codes. Numeric constants may be specified as binary, octal, decimal, or hexadecimal. Arithmetic expressions may include the operators +, -, \*, /, MOD, OR, AND, NOT, XOR, EQ, NE, LT, LE, GT, GE, SHR, SHL, LOW, HIGH, PTR, SHORT, THIS, SEG, OFFSET, TYPE, LENGTH, SIZE, MASK, WIDTH, (), [], ., : and <>.

## Assembler Directives

Assembler directives give instructions to the assembler but are not translated into machine code during assembly. Basic assembler directives include: storage definition and allocation directives (DB, DW, DD, DBS, DWS, DDS, STRUC/ENDS, RECORD); symbol directives (EQU, LABEL, PURGE); and location counter control directives (ORG, EVEN, ALIGN). Program control directives include: segment definition and control directives (SEGMENT/ENDS, PROC/ENDP, ASSUME, GROUP, END); linkage directives (NAME, PUBLIC, EXTRN); and the PARITY directive.

The relocation types for SEGMENT/ENDS directives are specified in the operand column. These include: BYTE, WORD, PARA, PAGE and INPAGE. The combination types of PUBLIC, COMMON, AT, STACK and MEMORY, also specified in the operand column, define how to link segments and groups of the same name.

The RA70320 supports also two V25/V35/ $\mu$ PD79011/ $\mu$ PD79021 device specific assembler directives e.g. the ASGNSFR and SETFDB directives. Where as the ASGNSFR directive tells assembler which segment the 'Internal RAM' and special function registers 'SFR' are assigned to, the 'SETFDB' directive sets the 'Internal RAM' and 'SFR' base address.

Note:  $\mu$ PD79011 is a V25 with Real-time-operating-system kernel on chip.

$\mu$ PD70021 is a V35 with Real-time-operating-system kernel on chip.

**Assembler Controls**

There are two types of assembler controls available for RA70320. Basic controls, which are specified in the assembler command line, are:

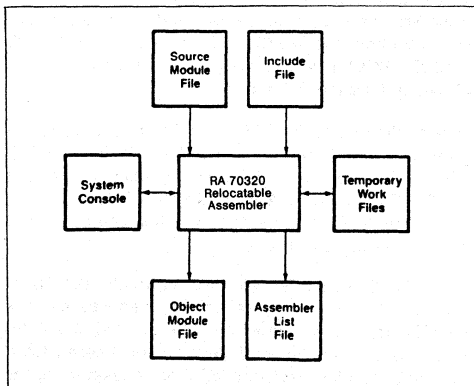
- File specification
- Output file selection
- Output file destination
- Listing format controls
- Date specification

General controls, which are specified in the source program, are:

- Inclusion of other source files
- Page eject
- Generation/suppression of listing
- Listing titles

The listing file may contain the complete assembly listing or only lines with errors, and a symbol or cross-reference table. The symbol table shows all defined symbols in alphabetical order, their types, attributes, and the values initially assigned to them. The cross-reference table contains all defined symbols and the numbers of all statements that refer to them.

The object file contains the relocatable object module. The format of this module is an NEC proprietary relocatable object module format. Figure 1 is a functional diagram of the assembler.



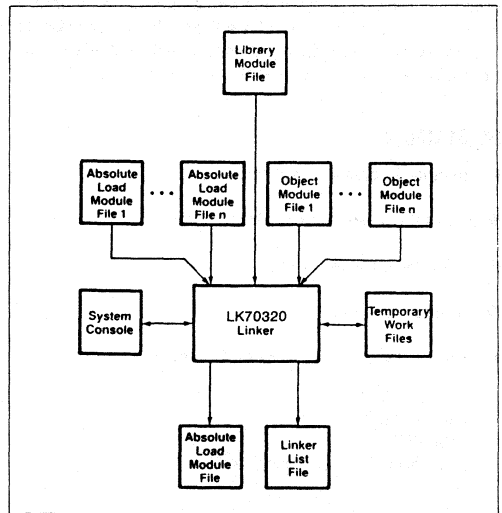
**Figure 1. Relocatable Assembler Functional Diagram**

**Linker**

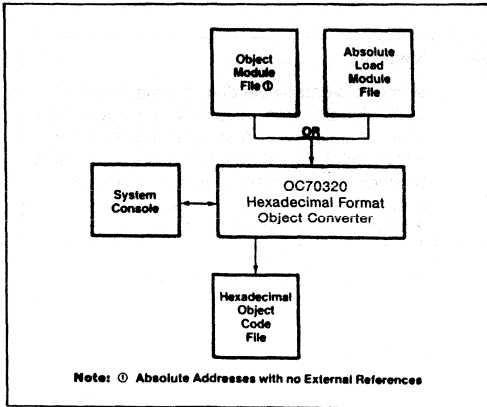
The linker, LK70320, combines relocatable object modules and absolute load modules and produces one absolute load module. Figure 2 is the functional diagram of the linker. The controls for LK70320 may be specified in either the command line or in a parameter file. You may specify the date, the absolute load module name, and the starting address and order for code/data/stack segments; you may also protect areas of memory from being assigned. Further, you may specify that a list file be created containing a link map, a local symbol table, or a public symbol table. The absolute load module contains symbol information for the symbolic debugger and absolute object code.

**Hexadecimal Format Object Converter**

The object code converter, OC70320, translates an absolute load module file into an expanded hexadecimal format (7-bit ASCII) file, which may be downloaded to a PROM programmer. You may specify that addresses be output in the order in which they were input or be output in ascending order. Figure 3 is a functional diagram of the hexadecimal format object code converter.



**Figure 2. Linker Functional Diagram**



**Figure 3.**  
**Hexadecimal Format Object Code Converter Functional Diagram**

## Librarian

The librarian, LB70320, creates and maintains files containing relocatable object modules. This reduces on the number of files which need to be linked together by allowing several modules to be kept in a single file, and provides an easy way to link frequently-used modules into programs. Modules may be added to, deleted from, or replaced within a library file.

## Operating Environment

The NEC RA70320 package can be supplied to run under a variety of operating systems. Versions are available to run on a CP/M-86 or MS-DOS system with one or more disk drives and at least 128K of system memory. Another version is available to run on an Intel Inteltec (tm) Series III development system with the ISIS operating system, one or more disk drives and at least 128K of installed system memory. Other versions are available to run on a Digital Equipment Corporation VAX computer under both the UNIX 4.2 BSD and the VMS (Version 4.1 or later) operating systems.

## Emulator Communication Program

V25 and V35 absolute load modules produced by the RA70320 Relocatable Assembler package can be debugged by using the appropriate NEC stand-alone In-Circuit Emulator: IE-70320-A008 and IE-70330-A008. Communication between these emulators and your host system is through an RS-232C serial line utilizing the KERMIT communications protocol developed at Columbia University. With the appropriate version of the KERMIT Communication Program running on both the emulator and your host system, V25 and V35 absolute load modules or hexadecimal format object code files may be transferred between machines. Depending upon your development environment, the emulator may be controlled directly from the console of your host system or from a user supplied terminal.

A version of the KERMIT Communication Program is supplied with each relocatable assembler package and each of these emulators at no extra charge. NEC Electronics supplies versions of the KERMIT to run on the following systems: the IBM PC™ under MS-DOS or CP/M-86, the VAX under VMS or UNIX and on the Intel Inteltec Series II/III ISIS-II based development system. Also for NEC's APC under CP/M-86 and for the Siemens PCD under MS-DOS one version each is available. Versions of KERMIT for other host systems may be obtained directly from Columbia University.

## Documentation

For further information on source program formats, assembler operation, and actual program examples, NEC Electronics provides the following documentation:

μPD70320, μPD70330 Relocatable Assembler User's Manual.

This documentation is provided with purchased copies of this product. Additional copies may be obtained from NEC Electronics.

## License Agreement

RA70320 is sold under terms of a license agreement which is included with purchased copies of the assembler. The accompanying software registration card must be completed and returned to NEC Electronics (Europe) GmbH to register the license. Software updates are provided to registered users only.

### Ordering Information

Product No.	Host System	Operating System	Medium
RAC86-08SS-70320	APC	CP/M-86	8" single sided single density floppy disk
RAC86-I5DD-70320	IBM-XT	CP/M-86	5 1/4" double sided double density floppy disk
RAMSD-I5DD-70320	IBM-XT/AT	PC-DOS	5 1/4" double sided double density floppy disk
RASPD-I5DD-70320	PCD (Siemens)	MS-DOS	5 1/4" double sided double density floppy disk
RAIS3-08SD-70320	INTEL MDS	ISIS III	8" single sided double density floppy disk
RAUNX-0T16-70320	VAX	ULTRIX	9-track, 1600 BPI magnetic tape
RAVMS-016-70320	VAX	VMS	9-track, 1600 BPI magnetic tape

### Description

The CC70116 C-compiler Package converts standard C source code into relocatable object modules for the V20/V30 ( $\mu$ PD70108/70116) microprocessors as well as the V40/V50 ( $\mu$ PD70208/70216) microprocessors. These modules are compatible with the ones produced by the RA70116 Relocatable Assembler package and may be linked with other modules using LK70116, the linker provided with the RA70116 Package.

The CC70116 C-compiler can also be used to compile C-Source modules for NEC's V-Series CMOS 16-bit microcomputer V25 and V35 ( $\mu$ PD70320/70322 and  $\mu$ PD70330/70332).

The CC70116 C-compiler Package is available for use on all NEC Development Systems and many other manufacturers' microcomputer development systems, personal computers, and minicomputers.

### Features

- Standard Kernighan and Ritchie C
  - Defined in UNIX™ System III
- NEC Enhancements
  - Small and medium memory model support
  - Enumeration data type support
  - Assignment of all members by a structure name
  - Ability to use identical names in identifiers of different types in different structures
  - Addition of a void type to declare functions with no return value
  - Addition of char as a data type for which unsigned can be specified
  - Ability to initialize structures with bit fields
- CC70116 Library contains 73 of the UNIX System III library functions
- User-selectable object code optimizer
- Runs under a variety of operating systems
  - CP/M-86™
  - MS™-DOS
  - ISIS™ III
  - VAX/VMS™ and VAX/UNIX™

Unix is a trademark of AT&T  
CP/M-86 is a trademark of Digital Research  
MS of MS-DOS is a trademark of Mirosoft Corporation  
VAX and VMS are trademarks of Digital Equipment Corporation  
ISIS is a trademark of Intel Corporation

### Extensions to Standard C

**Small and medium memory models:** Programs using the small memory model are more efficient in terms of space utilization, and so the small model is recommended when the program will fit into 64K of data and code space. When more space is needed, however, it is available through the medium memory model, which allows access to the full 1MB of address space.

**Enumeration data type:** Similar to the enumerated type in Pascal, this allows the definition of a limited set of (mnemonic) values for a variable, thus preventing assignment of invalid values to the variable.

**Assignment of all members by a structure name:** Allows assignment of all the members of a structure from another structure of the same type simply by using the names of the structures.

**Ability to use identical names for identifiers of different types in different structures:** Allows the same member name in two different structures to have a different name in each.

**Addition of a void type:** Permits the use of functions that return no value in expressions where a return value is required.

**Addition of char as a data type for which unsigned can be specified:** Allows the full 8 bits of the char data type to be used as a value, rather than only the low-order 7 bits.

**Ability to initialize structures with bit fields:** Allows structure with bit-field members to be initialized via the `|||C` construct.

### Library Functions

The CC 70116 C-compiler library contains 73 of the UNIX System III standard library functions. Also supplied are two start-up routines, one for execution under the CP/M-86 operating system and one for stand-alone startup in embedded applications. Some library routines are operating system dependent and may only be used to create programs that run under CP/M-86; other functions do not call the operating system and may be used in any program. The library functions are listed in table 1

**Table 1. CC70116 Library Functions**

abort (1)	fputc	ispunct	setbuf
abs	fputs	isspace	sprintf
access (1)	fread	isupper	srand
atoi	freopen	isxdigit	scanf
atol	fscanf	lseek (1)	strcat
close (1)	fseek	mktemp	strchr
creat (1)	ftell	open (1)	strcmp
ctype	*fwrite	perror	strcpy
exit (1)	getc	printf	strlen
fclose	getchar	putc	strncat
fdopen	gets	putchar	strncmp
feof	getw	puts	strncpy
ferror	isalnum	putw	swab
fflush	isalpha	qsort	ungetc
fgetc	isascii	rand	unlink (1)
fgets	isctrl	read (1)	write (1)
fileno	isdigit	rewind	
fopen	islower	sbrk (1)	
fprintf	isprint	scanf	

**Note:**

(1) CP/M-86 dependent

**Operating Environment**

The NEC CC70116 package is available to run on any CP/M-86, MS-DOS, or Intel® Series III system with one or more disk drives and at least 128K of system memory. To produce executable programs, the LK70116 (linker) and OC70116 (object converter) programs are required and the LB70116 (librarian) program is useful. These programs are available separately from NEC Electronics as part of the RA70116 Relocatable Assembler Package.

**Ordering Information**

Product No.	Host System	Operating System	Medium
CCC86-08SS-70116	APC	CP/M-86	8" single sided single density floppy disk
CCC86-15DD-70116	IBM-XT	CP/M-86	5 1/4" double sided double density floppy disk
CCMSD-15DD-70116	IBM-XT/AT	PC-DOS	5 1/4" double sided double density floppy disk
CCSPD-15DD-70116	PCD (Siemens)	MS-DOS	5 1/4" double sided double density floppy disk
CCIS3-08SD-70116	INTEL MDS	ISIS III	8" single sided double density floppy disk
CCUNIX-0T16-70116	VAX	ULTRIX	9-track, 1600 BPI magnetic tape
CCVMS-0T16-70116	VAX	VMS	9-track, 1600 BPI magnetic tape

LK70116 produces an absolute load module that contains both symbol and source code line number information and the absolute object code. This module may be loaded directly into an IE-70108/70116 or an IE-70208/70216 for execution with full symbolic debug capabilities.

**Documentation**

For further information on source program format, compiler operation, and actual program examples, NEC Electronics provides the following documentation:

CC70116, V-series C-compiler User's Manual

This documentation is provided with purchased copies of the compiler. Additional copies may be obtained from NEC Electronics.

**License Agreement**

CC70116 is sold under terms of a license agreement. A signed agreement is a pre-condition for the purchase of the CC70116 package. The shipped copy will be accompanied by a 'Registration Card' which must be completed and returned to NEC Electronics (Europe) GmbH to ensure that software updates are provided later on.

### Description

Customers whose designs incorporates the V20, V30, V40, V50 or V25 and V35 microprocessors/microcomputers, and who already have software packages developed with 8086/88 or 80186/188 tools, can use a simple converter program to port the existing software to the new application.

The absolute object format converter (AOF converter) RC converts INTEL absolute object files as generated by the utility LOC86 into the corresponding NEC object file format. This format is defined in the object program module specification IPD 300167 and loadable with the in-circuit emulators IE-70K or IE-700K as well as the Mini-Emulators (EBIBM-VxxMINI-IE).

During the conversion process all segmentation and symbolic debugging information as generated by Intel language processors is transferred into the appropriate NEC representation in order to provide the user of IE-70K, IE-700K and Mini-Emulators with the full symbolic debugging environment. The figure below shows where AOF fits into the system configuration.

The converted files can be relinked (optionally) with NEC format binary files generated with Relocatable Assemblers or C-Compiler (see Figure below). The AOF converter offers an optimisation feature switch so that the conversion overhead in terms of code length is reduced to a minimum.

This converter program was developed by AID\*<sup>1</sup> and is also supported by this company. There are versions for IBM-PC\*<sup>2</sup> (MS-DOS\*<sup>3</sup>), MDS-III (ISIS-III\*<sup>4</sup>) and VAX (VMS/ULTRIX)\*<sup>5</sup> — in other words for most commonly available machines.

The AOF package can also be ordered directly from NEC Electronics.

The AOF is supplied on flexible magnetic disk (floppies) for PCs which contains the machine-executable program. For VAX the software is supplied on magnetic tape, 1600 bpi, EBCDIC code.

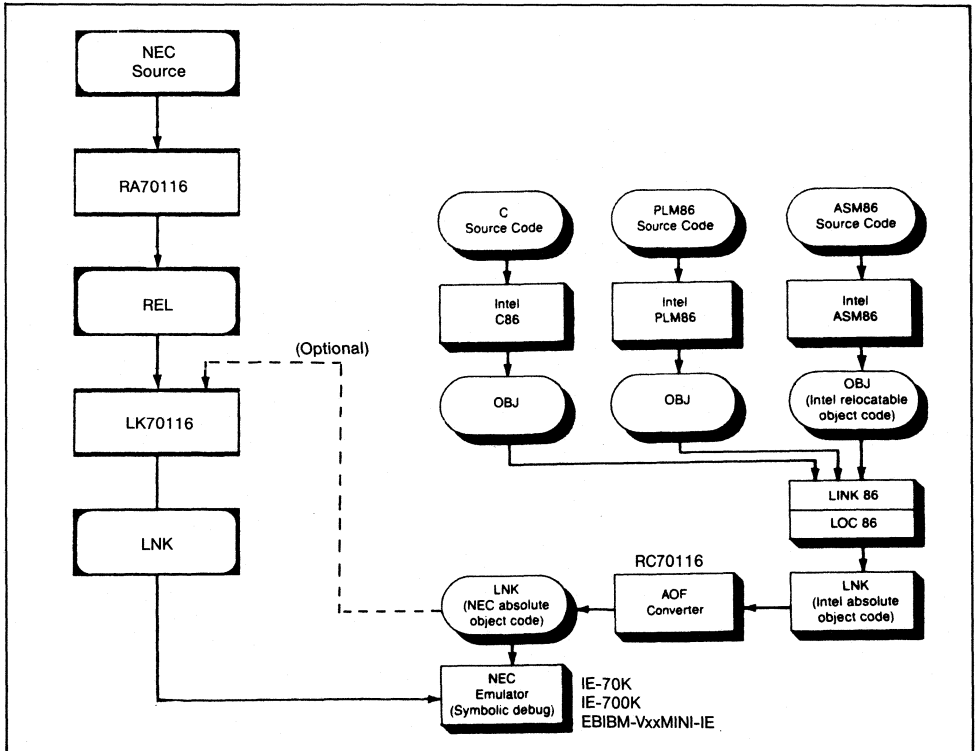


Figure: V-Series development support object code conversion



### Ordering Information

Product No.	Host System	Operating System	Medium
RCMSD-I5DD-70116	IBM-PC™ or any compatible	MS-DOS™	5¼" double sided double density floppy disk
RCIS3-08SS-70116	Intel MDS-III™	ISIS-III™	8" single sided double density floppy disk
RCVMS-OT16-70116	VAX™	VMS™	9-track, 1600 BPI magnetic tape
RCUNX-OT16-70116	VAX™	ULTRIX™ Berkley 4.2	9-track, 1600 BPI magnetic tape

IBM is a trademark of International Business Machines Corp.

MS-DOS is a trademark of Microsoft Corp.

ISIS-III is a trademark of Intel Corp.

VAX/VMS and ULTRIX are trademarks of Digital Equipment Corp.

Note \* AID Automatisierung Informatik Datentechnik, Ing. ges.  
f. Microcomputer Hard- u. Software mbH.  
Marienbergstr. 78, 8500 Nürnberg 10, Tel.: (09 11) 52 40 16

## Description

This utility was developed for programmers using the relocatable assembler RA70116 and C-compiler supplied by NEC but who want to run the code (as far as application independent) on the IBM-PC. NECTOMS converts NEC relocatable object format to Microsoft Linker input format to

generate .EXE files. This converter is delivered together with all RA70116 and CC70116 package purchased from NEC Electronics. The Figure below shows the general usage of this converter.

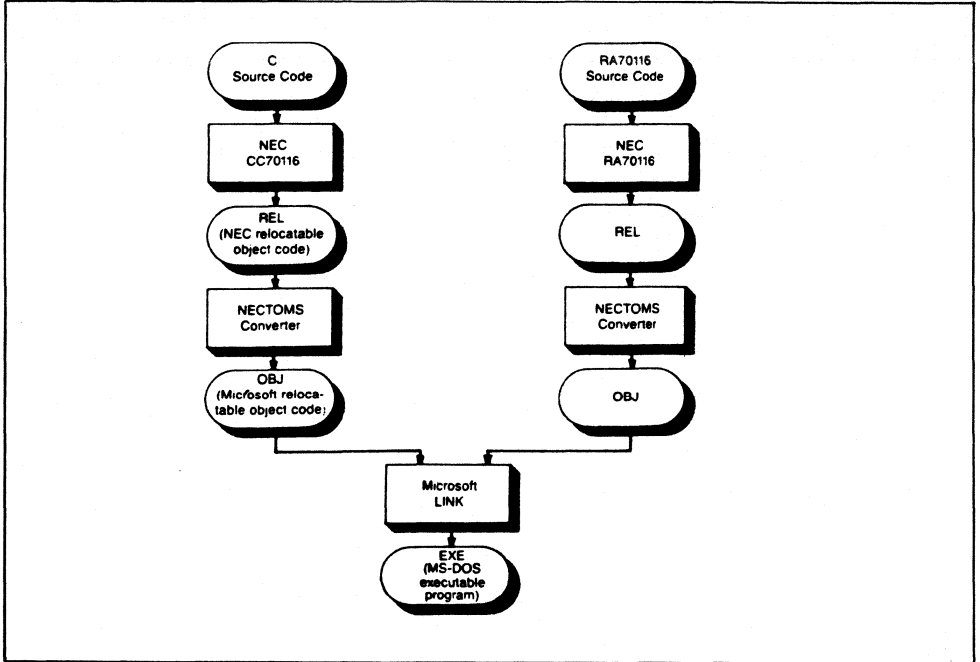


Figure: V-Series development support NECTOMS converter program



## Description

Customers who are designing in V20, V30, V40, V50, V25, or V35 and who are developing the software with Microsoft Assembler and C-compiler, they can have full support for symbolic debugging. The modules generated by MASM and C-Compiler (Microsoft) have their own binary file format and can be changed into NEC binary format using this "Micro-

Soft TO NEC" (MSTONEC) format converter. The code modules that have passed this converter can be linked with NEC linker (LK70116) to generate executable files including all symbol information. The linker output file can be downloaded from IBM-PC to the incircuit emulator or the so called mini emulators for doing symbolic debugging.

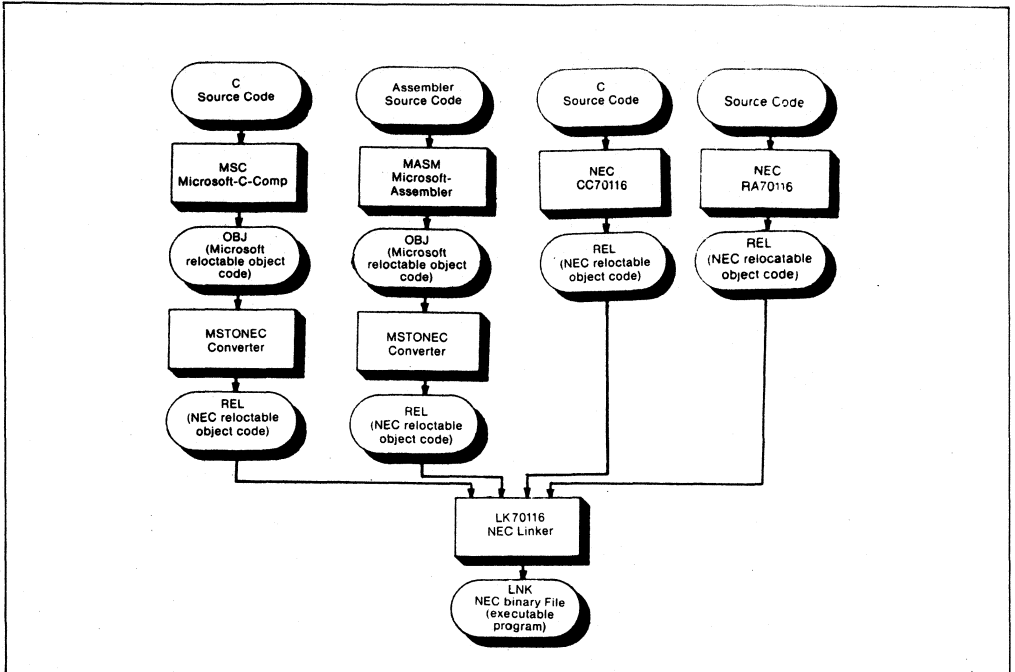


Figure: V-Series development support MSTONEC converter program



## Description

The PKG70616 Software Generation Tools package is a collection of software tools that are used to generate object code modules for the V60<sup>TM</sup> ( $\mu$ PD70616) and V70<sup>TM</sup> ( $\mu$ PD70632) microprocessors. These tools include:

cc70616	(C-compiler)
as70616	(Assembler)
ld70616	(Link Editor)
ar70616	(Archiver)
hx70616	(Object Converter)
cf70616	(Configurator)

The C-compiler assembler, and link editor are all derived from the standard UNIX<sup>TM</sup> software development tools. In particular, the C-compiler is derived from AT&T's portable C-compiler. These tools are available from NEC on a 1600 bpi tar format tape in object code for a VAX-11<sup>TM</sup> Series 780 running either UNIX 4.2 BSD or ULTRIX<sup>TM</sup>. In combination with a V60 based processor board this package (PKG70616) can also be used with PC-XT/AT. The V60 board has to be placed in IBM Expansion bus and will run UNIX V, then the software can be developed in UNIX environment even on a PC.

## Operational Flow

A user of this software development package can prepare a C source program using an editor (such as vi or emacs) which is available on the UNIX system. The user can also prepare some V60 assembly language source files. These files are processed by the C-compiler and assembler or by the assembler alone to produce relocatable object files in common object file format (COFF). These relocatable object files are then combined by the link editor which resolves symbolic names to create an absolute object file. The user has the option of preparing a memory description file that serves as input to the link editor to gain detailed control over its operation. The relocatable object files can be combined in a library using the archiver, and such a library can serve as input to the link editor.

V60/V70 is a registered trademark of NEC Corporation  
 VAX-11 is a trademark of Digital Equipment Corporation  
 UNIX is a registered trademark of AT&T Bell Laboratories  
 ULTRIX is a registered trademark of Digital Equipment Corporation

The configurator can prepare translation tables. It can be invoked by an operating system for this purpose, but it can also be used in stand-alone applications when a debugging monitor or an in-circuit emulator is used to invoke only individual tasks. In this case the user prepares a configuration file to describe how memory is to be used. This configuration file, together with an absolute object file, serves as the input for the configurator.

The object converter allows various alternative object file formats to be created to facilitate the use of PROM burners and other such tools which cannot accept COFF format files.

## C-compiler (cc70616)

The V60 C-compiler accepts C syntax as described in "The C Programming Language" by B. Kernighan and D. Ritchie (Prentice-Hall, 1978). In addition to that syntax, this C-compiler supports several additional data types. These include enumeration, void, and several unsigned integer types. The compiler also supports labels up to 100 characters as well as structure assignment, assembler escapes, and the initialization of structures with bit fields.

The cc70616 C-compiler follows the standard UNIX file-naming conventions. Specifically, the following table shows these conventions:

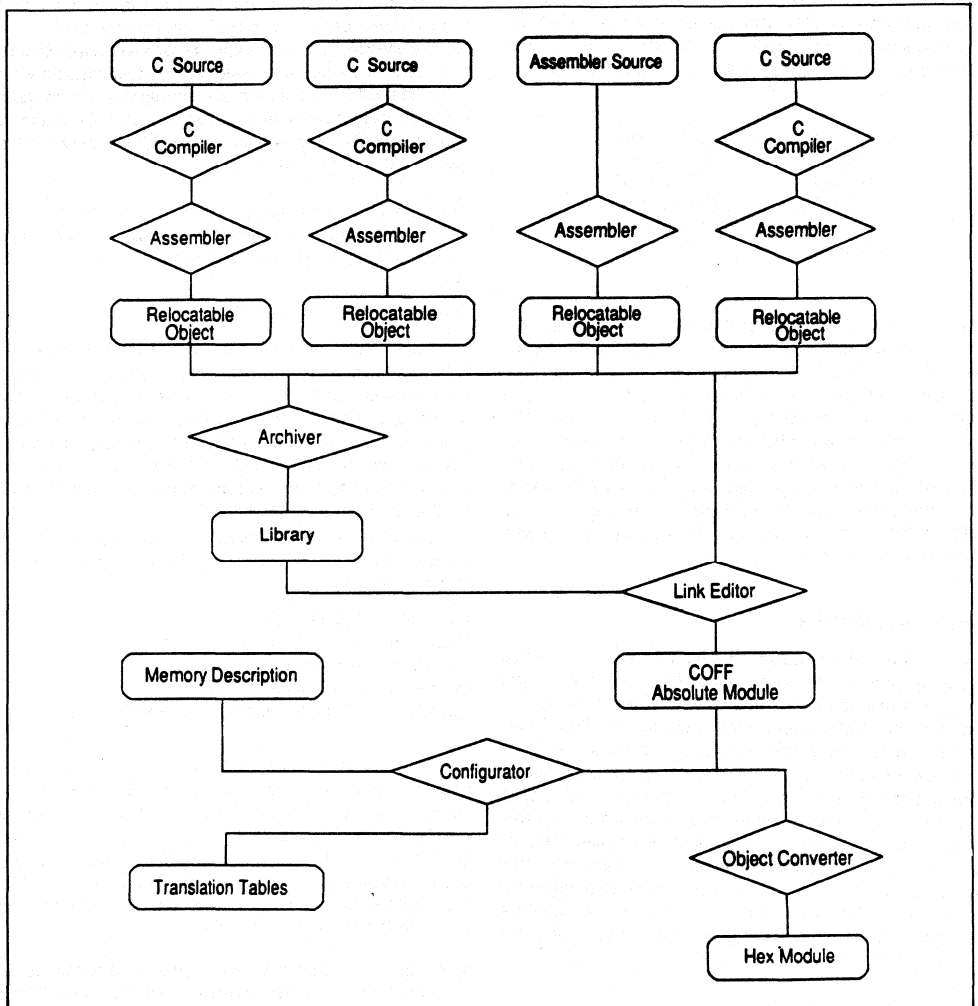
file_name.c	C Source File
file_name.i	C Pre-Processor Output
file_name.s	Assembler Source File
file_name.o	Relocatable Object File
a.out	Absolute Object File (Default Name)

The C-compiler is based on the AT&T portable C-compiler and accepts the usual command line options of that compiler. Because the V60 and V70 processors have so many general purpose registers, the C-compiler supports options to grant considerable control over the allocation of these registers. To facilitate the generation of code to be put in ROM, the compiler supports a command line option to put variables into the text segment.

All C-compilers support certain simple data types, but unfortunately there is no standard for their precision. For the cc70616 C-compiler, the following precision is supported:

char	8 bits
short	16 bits
int	32 bits
long	32 bits
float	32 bits, IEEE 754 format with 8-bit exponent
double	64 bits, IEEE 754 format with 23-bit exponent

Operational Flow Diagram



## Assembler (as70616)

The as70616 assembler takes assembly language programs for the  $\mu$ PD70616 or  $\mu$ PD70632 microprocessors and produces relocatable object files in the UNIX COFF (common object file format). Assembly language source files are usually named with a.s extension (e.G., name.s) and the corresponding COFF relocatable object file is usually named name.o.

A statement in an as70616 source program consists of up to four fields, with the label and comment fields being optional. Typically, there is one statement on each line of an assembly language program, but multiple statements can occur on a single line, separated by a semicolon. Symbols of unrestricted length are supported with upper and lower case being distinguished. The assembler recognizes binary, octal, decimal, hexadecimal, and floating numeric constants as well as character and string constants. In addition, the assembler can compute operands expressed symbolically using conventional arithmetic and logical operations.

The as70616 assembler supports a variety of pseudo-instructions as listed below:

.align	.base	.bss	.byte	.comm	.data
.dword	.equ	.file	.globl	.hword	.lcomm
.float	.ln	.org	.set	.sfloat	.space
.str	.strz	.text	.type	.word	

Normally, an object file created by the assembler includes text, data, and bss sections. To facilitate generating object code to be placed in ROM, a command line option of the as70616 assembler is provided to force initialized data into the text section. Other options are included for controlling both optimization and the relocation displacement size. There is an option for invoking the m4 pre-processor (not supplied with the software generation package, but generally available on UNIX systems) and for deleting the assembly source after it is preprocessed by m4.

## Link Editor (ld70616)

Distinct C and assembler source files can be compiled and assembled separately to obtain object files, name1.0, name2.0, etc. The ld70616 link editor is provided to combine such relocatable object files (which may also be drawn from a library created by the ld70616 archiver) into an absolute or relocatable object file. The resulting object file will be absolute if there are no references that remain unresolved after the linking is complete. The default name for an absolute object file created by the link editor is a.out, but this name may also be specified by a command line option (-o objname).

The link editor normally combines like-named sections together and assigns sections with those same names to be placed consecutively in virtual memory starting at address 0. Normally, the linker is invoked implicitly by the compiler or by the assembler. The user does not have the option, however of explicitly invoking the loader with or without a configuration file. A configuration file is used, the user can specify explicitly how to combine the different sections of the different relocatable object files and where to place the resulting combined section (such as to accommodate a memory system with gaps in its address space). Finally, the configuration file can specify read, write, and execute permission to be enforced by the MMU.

## Archiver (ar70616)

The ar70616 archiver provides another way to combine multiple relocatable object files into a single file. In this case, the purpose is to create a library file, typically named with a .a extension, such as mathlib.a. The ld70616 is equipped to search such a library for object files that it finds referenced in other relocatable object files. The link editor will do such a search of a library only once, however, so the order that object files occur in the library is significant (if a.o occurs before b.o in a library the b.o should not reference a.o). For that reason, the ar70616 archiver provides a variety of command line options for moving, inserting, and deleting object modules from a library.

Because modules can be updated from time to time, the ar70616 archiver also provides command line options to replace modules in a single operation. In addition, there are options for extracting modules from a library and for listing the library.



### Object Converter (hx70616)

It may not be convenient to use the UNIX COFF object file format with existing hardware tools, such as PROM burners. The hx70616 can be used in these cases to convert a COFF object file to any of the following formats:

1. Intel extended HEX format
2. Extended Tektronix HEX format
3. Motorola S-record format

Through command line options, the hx70616 object file converter allows one to convert only specified sections of the COFF file. While converting symbol table information (local, global or both), it can be specified through command line options. There are options to select explicit expansion of bss sections (in a COFF file, the bss section appears only as a byte count), and to select the block size for the resulting HEX file.

### Configurator (cf70616)

The  $\mu$ PD70616 and  $\mu$ PD70632 microprocessors have on-chip MMU's which translate virtual addresses to physical memory addresses or to I/O port addresses. The details of this translation must be specified through values placed in the area-table registers and through area tables and page tables in memory. Refer to the  $\mu$ PD70616 programmer's reference manual for the details of how these tables and the data in these registers are constructed.

### Ordering Information:

Product No.	Host System	Operating System	Medium
PKGUX-OT16-70616	VAX	ULTRIX	9-track, 1600 BPI magnetic tape
PKGVS-OT16-70616	VAX	VMS	9-track, 1600 BPI magnetic tape

Like the object files, the translation tables, can be tedious to construct manually. Just as compilers, assemblers, and link editors allows object files to be built from symbolic specifications, the cf70616 configurator allows translation tables, and area table register definitions to be built from a symbolic specification. This symbolic specification is presented as input to the configurator in a configuration file. This input may contain descriptions of multiple virtual spaces as well as names of object files that are to be loaded into those virtual spaces.

The output of the configurator has the same format as a library created by the ar70616 archiver. The default name for this file is cfg.out. The files in the archive describe the virtual spaces as well as provide loading detail for the object files.

### License Agreement

PKG70616 is sold under terms of a license agreement. A signed agreement is a pre-condition for the purchase of the PKG70616. The shipped copy will be accompanied by a 'Registration card' which must be completed and returned to NEC Electronics (Europe) GmbH to ensure that software updates are provided later on.

### Description

The IE-700k is a portable stand-alone In-Circuit Emulator, providing both hardware emulation and software debug capabilities for NEC's V-Series (V20-V70) microprocessors. Realtime and single-step emulation in both native and 8080 emulation modes (for V20, V30, V40, V50), coupled with sophisticated memory mapping, user programmable breakpoints and trace qualifiers, symbolic debug and macro file command facilities, create a powerful development environment. Command entry is simplified by the use of eight softkeys. Softkeys are a context sensitive method of command entry that the IE uses to prompt the user for the next input using dynamically labelled softkeys. The softkeys are located at the bottom of the display and correspond to one of eight function keys on the keyboard. To select the next command, the function key corresponding to the desired softkey is entered and the softkeys are automatically re-labelled with the next set of valid commands.

User programs can be uploaded/downloaded from a variety of host systems via a serial link or loaded directly from an MS-DOS™ disk.

### Features

- Portable Stand-Alone In-Circuit Emulator
  - 9.5 inch amber CRT display
  - Two 5 inch 640 Kilobyte floppy disk drives
  - ASCII Keyboard with eight function keys
  - EPROM programmer: 2732, 2764, 27128, 27256, 27512
  - Can be configured to support V20, V30, V25, V35, V40, V50, V60, V70
  - Two RS-232 serial ports
  - parallel printer port
- Precise real-time and single step emulation
  - Programmable internal clock: 2 to 16 MHz in 1KHz steps
  - External clock: 2 to 16 MHz
- Memory and I/O space mappable in 1 KB blocks
- 256 KB of memory for prototype memory emulation
  - Expandable to 768 KB (with optional board of 512 KB)
- Eight User-programmable Breakpoints
  - Real-time break on address, data, CPU status, external probes
  - Break on pass count and register, memory, or I/O values
  - Selectable as execution or code fetch break
- 1K trace buffer: Machine cycle, mnemonic and jump trace display
  - Programmable trigger point and trace qualifiers

- Eight optional probes for tracing of target system signals.
- Full symbolic debug capabilities
- Symbolic line assembler and disassembler
- Macro file command capability
- Dual window display in emulation mode
- Softkey and menu driven user input
- Power-on diagnostics

MS-DOS is trademark of Microsoft Corporation

### Hardware Description

The IE-700K consists of a system chassis with a detachable ASCII keyboard and an emulation pod unit. The chassis houses a 9.5 inch amber CRT, two 5 1/4 inch 640KB floppy disk drives, an EPROM programmer, a card cage, a power supply and three control boards (main CPU, emulation control and trace-emulation memory).

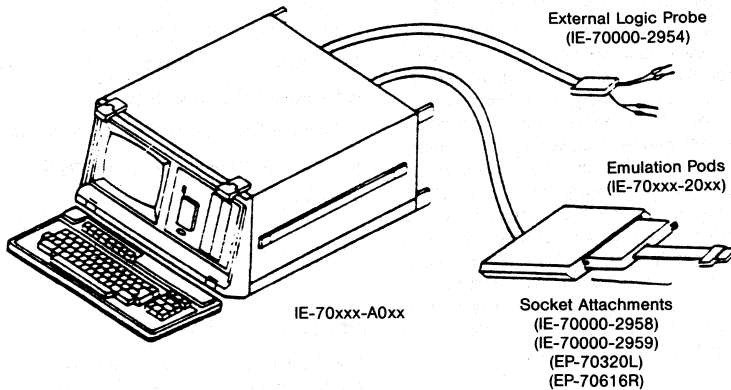
The main CPU board contains the supervisor CPU, 512KB of system memory and the peripheral interfaces. The emulation control board controls the memory mapping, event detection, break and emulation CPU status circuitry. The trace-emulation board contains the trace buffer and 256KB of external emulation memory (expandable by optional board).

The external emulation pod unit houses the emulation CPU, high speed buffers and clock selection logic. The pod units for V60 and V70 also contain the trace memory and a high speed emulation memory to ensure real time trace and emulation. The emulation pod unit can be connected to the target system via an attachment for the corresponding package (see Fig. 1 and Tab. 1). These socket attachments are sold separately.

The IE-700K supports the following external interfaces: two RS-232C serial ports, one Centronics parallel printer port, an RGB video output and eight optional external logic probes. These probes (IE-70000-2954) can be used for tracing and/or breaking on signals from the target system.

Ease of migration within the V-Series family of emulators is provided in the system design. To alternate between V40 and V50 emulation simply requires changing the emulation pod unit. The emulator can be converted to support any of the V20, V25, V30, V35, V40, V50, V60 and V70 CMOS microprocessors/microcomputers from NEC by exchanging the appropriate control boards and the emulation pod unit with those for the processor to be emulated.

Figure 1: IE-700K System Configuration



## Memory and I/O Mapping Capabilities

The IE-700K contains 256KB (expandable to 768KB) of emulation RAM (0 wait states) for emulating external RAM or ROM. The complete 1MB memory space of the  $\mu$ PD70208/70216 must be mapped into one of the following categories of memory:

- ROM – External ROM emulation memory (Read Only)
- RAM – External RAM emulation memory (Read/Write)
- Target – Memory resident in target system (Read/Write)
- Locked – Access inhibited memory (remaining unmapped addresses)

All memory mapping is done in 1K byte blocks (4K byte with V60, V70) using the CONFIGure and MEMOry softkey commands. If an address which is mapped as "locked" is accessed, a break in emulation will occur.

The complete 64K byte (16M byte with V60, V70) I/O space of the must be mapped in 1KB (4KB) blocks to the emulation memory, the target system or as "locked" memory.

## Emulation

The IE-700K executes user programs in real-time in four different modes: break, trace, count and time. In break emulation mode, the user's program is executed in realtime or in single-step until a break point is encountered. In trace emulation mode, the user's program is executed until the trace buffer is filled. During the count emulation mode, the emulator counts the number of times a particular trigger point is reached within a given set of conditions. In the time emulation mode, the emulator times execution between the specified enable and disable points. The measurable time range is from 0 to 72 minutes (in microseconds).

Once emulation is stopped in either break or trace mode, the trace is automatically displayed showing data up to the last instruction executed. In count or time mode, the current count or elapsed time is displayed. At this point, the user can display the contents of memory, general purpose and special registers, symbol tables or directories by themselves or simultaneously with the trace display. Each window may be scrolled independently.

Prior to the start of emulation, the user can specify whether an external or internal clock will be used for emulation, whether the internal or external ready signal is used and whether the NMI signal from the target system should be enabled or disabled. If the internal clock is used, it can be set from 2 MHz to 16 MHz in 1 KHz steps.

## Break Capabilities

The IE-700K has eight breakpoints. Seven can be set to occur on a real-time event or a non real-time condition. The remaining one is reserved for setting a real-time address breakpoint in the "Go" command.

A real-time breakpoint can be set to occur on an address, a data value, a CPU state and the external probe status. A non real-time breakpoint can be set to occur after satisfying an address/condition setting a certain number of times (maximum 4096). Conditions pertaining to the general purpose registers, memory locations, I/O locations or the external probes can be defined. For non real-time breakpoints, the user program is executed in real time until it reaches the address, then emulation pauses while the conditions are checked. If the conditions are not satisfied, emulation will continue in this manner until they are met.

In order to distinguish between the condition occurring at an op-code fetch or at the execution of an instruction, each breakpoint can be tagged with either a non-execution or execution flag.

## Trace Capabilities

The trace buffer is 1KB (4KB for V60, V70) frames by 64 bits wide and sampling is done on every machine cycle in round-robin fashion. The emulator traces the external address and data bus, the CPU and queue status and the eight external logic probes.

The IE-700K has eight trace specifications points. One of these is reserved for setting a trigger point in the "Go" command. The other seven points can be specified as trace trigger, enable, disable, qualify or check points. Checkpoints are used to display register, memory or I/O contents each time a certain event or address occurs. The trace buffer can be split into a maximum of 64 partitions to allow tracing of particular segments of the user program (ie. a subroutine).

The tracer data may be displayed in one of three modes: machine, disassemble, jump. In machine display mode, all bus activity is displayed in machine code. In disassemble mode, all instructions are disassembled. In jump mode, only instructions which alter program flow are displayed.

## System Software

The IE-700K controlled by the MIOS/U propriety operating system. Command input is simplified by using the eight function keys to choose up to 24 soft keys. The dynamically reprogrammed softkeys visually prompt the user the next valid set of commands. The softkeys are located at the bottom of the display and correspond to one of the eight function keys. To select the next command, the function key corresponding to the desired softkey is entered and the softkeys are automatically re-labelled with the next set of commands.

The following utilities are provided with the emulator (depending on type that is ordered):

Utilities	Function	
EMUV2030	IE-70108/70116-A	} device specific emulator software for: V20/V30 V/40/V50 V25/V35 V60 and (V70 in the future)
EMUV4050	IE70208/70216 A	
EMUV2535	IE-70320/70330-A	
EMUV6070	IE-70616/70632-A	

KERMIT	Communication program for file transfer
FILESERV	File management for the system disks
EDITOR	Full screen editor
FORMAT	Floppy disk formatter
PROM	Built-in EPROM programmer control program
TERMINAL	Terminal utility program for file transfer between emulator and another intelligent device
SYMBOL	Symbol Table Converter: converts non-SROC symbol formats to SROC format.
OBJCONV	Object File Converter: converts object files to and from the Motorola SROC format.
TIMESSET	Internal battery backed-up clock and calendar setting
DEFINE	Soft Key definition
MDEVICE	Disk format specification

## Connecting Host Systems

Host systems may be connected to the IE-700K via the RS-232-C connectors at the rear of the machine. Parameters such as baud rates, character length, parity and number of stop bits are software programmable to suit the system being attached. The communications program KERMIT is supplied with the emulator and can be used for uploading and downloading of files. Currently, NEC provides KERMIT for the VAX under VMS and UNIX, the IBM PC and PC/XT under both PC-DOS and CP/M-86, the IBM PC/AT under PC-DOS, and the Intel Series III development system.

Files may also be transferred to the emulator via the RS-232C ports by using the TERMINAL utility. The emulator acts as a terminal for data transfer.

## Symbolic Debug and Line Assembly/Disassembly

The IE-700K supports complete symbolic debug utilizing the link modules produced by LK70116 from the RA70116 relocatable assembler package (by ID70616 from PKG 70616 software generation tool package for V60 and V70). The symbols can be used as address and data constants in break, trace and emulation control commands and are displayed during disassembly. A symbolic line assembler is also available to make modifications to existing programs.

CP/M is a registered trademark of Digital Research.

VAX and VMS are registered trademark of Digital Equipment Corp.

UNIX is a trademark of AT&T.

Intel is trademark of Intel Corporation

IBM, IBM PC, XT and PC-DOS are trademarks of International Business Machines Corporation

This stand-alone in-circuit emulator is a very flexible system, and it can be configured for each device out of the V-Series microprocessors/microcomputers by varying the composition of internal boards and PODs. Table 1 shows a

chart that contains all the different parts with their order numbers and indicates very clearly which parts have to be ordered/exchanged when modifying an emulator from one device to another.

**Table 1: Structure/Part Number of IE-700K Series**

CPU		V20/V30	V40/V50	V25/V35	V60
Order Numbers		IE-70108-A010 IE-70116-A010	IE-70208-A008 IE-70216-A008	IE-70320-A008 IE-70330-A008	IE-70616-A016
C O M P O S I T I O N	Emulation Main Frame (with 512KB System Memory)	IE-70000-2950			
	Trace & Emulation Memory	IE-70116-2410 (256KB)		IE-70320-2408 (128KB)	(128KB included in IE-70616-2016)
	Emulation Controller	IE-70116-2310	IE-70216-2310	IE-70320-2308	IE-70616-2316
	POD	IE-70108-2010 IE-70116-2010	IE-70208-2008 IE-70216-2008	IE-70320-2008 IE-70330-2008	IE-70616-2016
PGA Attachment			IE-70000-2959		EP-70616R
PLCC Attachment			IE-70000-2958	EP-70320L	
Expansion System Memory (512KB)		IE-70000-2960 (optional)		IE-70000-2960	IE-70000-2960
Expansion Emulation Memory (512KB)		IE-70000-2957 (optional)			IE-70000-2957
Logic Probe		IE-70000-2954 (optional)			

- V20 =  $\mu$ PD70108
- V30 =  $\mu$ PD70116
- V40 =  $\mu$ PD70208
- V50 =  $\mu$ PD70216
- V60 =  $\mu$ PD70616
- V25 =  $\mu$ PD70320/322
- V35 =  $\mu$ PD70330/332

In case the IE-700K needs to be modified for emulating another V-Series product, there are various possibilities when a number of different boards and pods have to be exchanged as described in Table 2:

**Table 2:**

Modification	Item to be ordered and attached to, or plugged into IE-700K	
from V20 to V30	IE-70116-2010	- a V30 POD (Buffer Box)
from V30 to V20	IE-70108-2010	- a V20 POD (Buffer Box)
from V20/V30 to V40	IE-70216-2310 + IE-70208-2008 + IE-70000-2959 or IE-70000-2958	- Emulation Controller - V40 POD (Buffer Box) - Attachment for 68 pin PGA or - Attachment for 68 pin PLCC
from V20/V30 to V50	IE-70216-2310 + IE-70216-2008 + IE-70000-2959 or IE-70000-2958	- Emulation Controller - V50 POD (Buffer Box) - Attachment for 68 pin PGA or - Attachment for 68 pin PLCC
from V20/V30 to V25	IE-70320-2408 + IE-70320-2308 + IE-70320-2008 + IE-70000-2960 + EP-70320L	- Trace & Emulation memory - Emulation controller - V25 POD (Buffer Box) - Expansion system memory - Attachment for 84 pin PLCC
from V20/V30 to V35	IE-70320-2408 + IE-70320-2308 + IE-70330-2008 + IE-70000-2960 + EP-70320L	- Trace & Emulation memory - Emulation controller - V35 POD (Buffer Box) - Expansion system memory - Attachment for 84 pin PLCC
from V20/V30 to V60	IE-70616-2316 + IE-70616-2016 + IE-70000-2960 + IE-70000-2957 + EP-70616R	- Emulation Controller - V60 POD (Buffer Box plus Trace & Emulation memory) - Expansion system memory - Expansion emulation memory - Attachment for 68 pin PGA
from V25/V35 to V20	IE-70116-2410 + IE-70116-2310 + IE-70108-2010	- Trace & Emulation memory - Emulation controller - V20 POD (Buffer Box)
from V25/V35 to V30	IE-70116-2410 + IE-70116-2310 + IE-70116-2010	- Trace & Emulation memory - Emulation controller - V30 POD (Buffer Box)
from V25/V35 to 40	IE-70116-2410 + IE-70216-2310 + IE-70208-2008 + IE-70000-2959 or IE-70000-2958	- Trace & Emulation memory - Emulation Controller - V40 POD (Buffer Box) - Attachment for 68 pin PGA or - Attachment for 68 pin PLCC
from V25/V35 to V50	IE-70116-2410 + IE-70216-2310 + IE-70216-2008 + IE-70000-2959 or IE-70000-2958	- Trace & Emulation memory - Emulation controller - V50 POD (Buffer Box) - Attachment for 68 pin PGA or - Attachment for 68 pin PLCC
from V25/V35 to V60	IE-70616-2316 + IE-70616-2016 + IE-70000-2960 + IE-70000-2957 + EP-70616R	- Emulation controller - V60 POD (Buffer Box plus Trace & Emulation memory) - Expansion system memory - Expansion emulation memory - Attachment for 68 pin PGA
from V40/V50 to V20	IE-70116-2310 IE-70108-2010	- Emulation controller - V20 POD (Buffer Box)
from V40/V50 to V30	IE-70116-2310 IE-70116-2010	- Emulation controller - V30 POD (Buffer Box)

**IE-700K  
In-Circuit Emulator**



Modification	Item to be overed and attached to, or plugged into IE-700K
from V40/V50 to V25	<ul style="list-style-type: none"> <li>IE-70320-2408</li> <li>+ IE-70320-2308</li> <li>+ IE-70320-2008</li> <li>+ IE-70000-2960</li> <li>+ EP-70320L</li> </ul> <ul style="list-style-type: none"> <li>- Trace &amp; Emulation memory</li> <li>- Emulation controller</li> <li>- V25 POD (Buffer Box)</li> <li>- Expansion system memory</li> <li>- Attachment for 84 pin PLCC</li> </ul>
from V40/V50 to V35	<ul style="list-style-type: none"> <li>IE-70320-2408</li> <li>+ IE-70320-2308</li> <li>+ IE-70330-2008</li> <li>+ IE-70000-2960</li> <li>+ EP-70320L</li> </ul> <ul style="list-style-type: none"> <li>- Trace &amp; Emulation memory</li> <li>- Emulation controller</li> <li>- V35 POD (Buffer Box)</li> <li>- Expansion System memory</li> <li>- Attachment for 84 pin PLCC</li> </ul>
from V40V50 to V60	<ul style="list-style-type: none"> <li>IE-70616-2316</li> <li>+ IE-70616-2016</li> <li>+ IE-70000-2960</li> <li>+ IE-70000-2957</li> <li>+ EP-70616R</li> </ul> <ul style="list-style-type: none"> <li>- Emulation controller</li> <li>- V60 POD (Buffer Box plus Trace &amp; Emulsion memory)</li> <li>- Expansion system memory</li> <li>- Expansion emulation memory</li> <li>- Attachment for 68 pin PGA</li> </ul>
from V60 to V20	<ul style="list-style-type: none"> <li>IE-70116-2410</li> <li>+ IE-70116-2310</li> <li>+ IE-70116-2010</li> </ul> <ul style="list-style-type: none"> <li>- Trace &amp; Emulation memory</li> <li>- Emulation controller</li> <li>- V20 POD (Buffer Box)</li> </ul>
from V60 to V30	<ul style="list-style-type: none"> <li>IE-70116-2410</li> <li>+ IE-70116-2310</li> <li>+ IE-70108-2010</li> </ul> <ul style="list-style-type: none"> <li>- Trace &amp; Emulation memory</li> <li>- Emulation controller</li> <li>- V30 POD (Buffer Box)</li> </ul>
from V60 to V25	<ul style="list-style-type: none"> <li>IE-70320-2408</li> <li>+ IE-70320-2308</li> <li>+ IE-70320-2008</li> <li>+ EP-70320L</li> </ul> <ul style="list-style-type: none"> <li>- Trace &amp; Emulation memory</li> <li>- Emulation controller</li> <li>- V25 POD (Buffer Box)</li> <li>- Attachment for 84 pin PLCC</li> </ul>
from V60 to V35	<ul style="list-style-type: none"> <li>IE-70320-2408</li> <li>+ IE-70320-2308</li> <li>+ IE-70330-2008</li> <li>+ EP-70320L</li> </ul> <ul style="list-style-type: none"> <li>- Trace &amp; Emulation memory</li> <li>- Emulation controller</li> <li>- V35 POD (Buffer Box)</li> <li>- Attachment for 84 pin PLCC</li> </ul>
from V60 to V40	<ul style="list-style-type: none"> <li>IE-70116-2410</li> <li>+ IE-70216-2310</li> <li>+ IE-70208-2008</li> <li>+ IE-70000-2959</li> <li>or IE-70000-2958</li> </ul> <ul style="list-style-type: none"> <li>- Trace &amp; Emulation memory</li> <li>- Emulation controller</li> <li>- V40 POD (Buffer Box)</li> <li>- Attachment for 68 pin PGA or</li> <li>- Attachment for 68 pin PLCC</li> </ul>
from V60 to V40	<ul style="list-style-type: none"> <li>IE-70116-2410</li> <li>+ IE-70216-2310</li> <li>+ IE-70216-2008</li> <li>+ IE-70000-2959</li> <li>or IE-70000-2958</li> </ul> <ul style="list-style-type: none"> <li>- Trace &amp; Emulation memory</li> <li>- Emulation Controller</li> <li>- V50 POD (Buffer Box)</li> <li>- Attachment for 68 pin PGA or</li> <li>- Attachment for 68 pin PLCC</li> </ul>

### Ordering Information:

for ordering complete system supporting full functionality following ordering Guide gives you necessary ordering numbers:

Processor	Package	Kit to be ordered
V20	DIP	IE-70108-A010 + IE-70000-2954
V30	DIP	IE-70116-A010 + IE-70000-2954
V25	PLCC	IE-70320-A008 + IE 70000-2954 + EP-70320L
V35	PLCC	IE-70330-A008 + IE-70000-2954 + EP-70320L
V40	PGA/PLCC	IE-70208-A008 + IE-70000-2954 + IE-70000-2959/2958
V50	PGA/PLCC	IE-70216-A008 + IE-70000-2954 + IE-70000-2959/2958
V60	PGA	IE-70616-A016 + IE-70000-2954 + EP-70616 R

### Documentation:

For further information on the IE-700K, NEC Electronics provides the user's manual with the In-circuit emulator:

μPD70208/70216/70320/70322 (IE-700K) Development Tools

Additional copies may be obtained from NEC Electronics.



**IE-70208/216 Demonstration**

This demonstration shows some typical operating modes for the IE-70208-A008/IE-70216-A008 or just IE-70208/216 from NEC. While demonstrating some of the more common features, it is by no means a complete tutorial on the capabilities of the IE.

When the IE-70208/216 mainframe is powered up and a system disk inserted, the softkeys at the bottom of the display are labelled with the defined commands. To start the emulator

software, press the softkey labelled "EMUV4050". This action causes the emulator software to load and prepare the system for use. When the V40/V50 emulation software is loaded, the IE display will be similar to the display in screen 1.

This demo requires that commands be entered using the softkeys. If the command is not available in the current set of softkey, it can be found by using the "etc." softkey.

**Screen 1. Initial Start-up Screen**

```

<CPU-COMPOSITION>                                     14-OCT-86  11:04:33
      NAME      CPU      MEMORY      VER.
1  USED  V50    70216    256k(bytes) v1.7

                                                    v1.7  (COMMON)

*****
<CONFIGURATION>      CPU : 70216                                     14-OCT-86  11:04:33
  CLOCK : INT      8.000MHz  READY : INT      EXT-DATA : YES      NMI : DSB
<MEMORY MAP> NO.   TYPE      ADDRESS      <I/O MAP> NO.   TYPE      ADDRESS
                1  LOCKED  00000-FFFFF      1  LOCKED  0000-FFFF
                unused emulation memory 249k(bytes)
  
```

The following legend is used to interpret the various fields in the initial configuration display.

**MEMORY: 256K**  
The IE-70208/216 has 256K of emulation memory installed.

**VER.: V1.7**  
Version 1.7 of IE-70208/216 emulator software is installed.

**CLOCK: INT 8.000 MHz**  
The V40/V50 in the IE pod will use an internally generated 8.000 MHz clock.

**READY: INT**  
The READY signal is generated internally.

**EXT-DATA: YES**  
The external data inputs are enabled and available for use.

**NMI: DSB**  
Non-maskable interrupts are disabled.

## IE-70208/216 Configure Command

The first step is to configure the IE-70208/216 to the target system. The following steps assume that the IE will be used in the stand-alone mode, without any target hardware. If the IE is being used with a target such as the EBIBM-70208/70216 the configuration must be slightly modified to map in the physical memory and I/O devices of the EBIBM-70208/70216

(Target system):

Define SETUP.MAC

Since this setup is to be used again, record the setup as a macro file for quick recall at a later time.

CONFigur

Enter the configuration mode of the IE-70208/216 to setup the emulator to the target system characteristics.

Memory RAM 00000 To 03FFF

This target has 16K of RAM from addresses 00000h to 03FFFh but the RAM command maps this memory into the internal RAM.

Memory ROM 0FC000 To 0FFFFF

The target also has 16K of ROM from addresses FC000h to FFFFFh but the ROM command maps this memory into the internal RAM but write-protects it.

I/O Emulate 0000 To 00FF

Setup the IE to emulate the lower 256 bytes of the I/O space.

Clock Int 7.315

The target system clock is set to 7.135 MHz. The clock can be adjusted from 2 MHz to 10 MHz in 1 KHz steps.

End

The emulator configuration is complete so exit the Config mode and return to the command level.

End-Def

Close the macro definition file. This setup can be recalled by simply entering the file name or by assigning it to a softkey as we will see next.

## Screen 2. Configuration Screen

```
<CONFIGURATION>          CPU : 70216          15-OCT-86   7:44:59
  CLOCK : INT 7.135MHz  READY : INT  EXT-DATA : YES  NMI : DSB
<MEMORY MAP> NO.  TYPE  ADDRESS  <I/O MAP> NO.  TYPE  ADDRESS
      1  RAM    00000-03FFF  1  EMULATE  0000-03FF
      2  LOCKED 04000-FBFFF  2  LOCKED   0400-FFFF
      3  ROM    FC000-FFFF
      unused emulation memory 216k (bytes)
```

Since we defined the setup as a macro, it is a simple matter to install it as a softkey for instant recall. Macros can also be invoked from by entering the macro file name after the macro command. When the softkey assigned to the setup macro is pressed, the file will be opened and the commands executed in rapid succession.

Macro Softkey SETUP.MAC

Assign the macro file SETUP.MAC to a softkey.

SETUP

Execute the setup macro file.

With the IE-70208/216 configuration complete, we are ready to load some software to be debugged.

**IE-70208/216 Program Command**

The Program Mode is used to load, save and verify programs being debugged on the IE-70208/216.

**Program**

The program command is used to load and save programs used on the IE-70208/216. The Load command can also be used to verify the contents of memory or clear it in preparation for beginning debug of new software.

Load DEMO. LNK Both

The target software is loaded from the default floppy disk. The Both command specifies that both code and symbol information is to be loaded. When debugging large programs, the Load command can specify that only symbol information from certain modules be loaded.

Following the execution of the Load command, the IE display contains the modules loaded along with the segment and group names. On the right-hand side of the display, the address ranges and segment types are displayed.

**Screen 3. Program Load Screen**

<PROGRAM> 70216		14-OCT-86	11:24:56
No	Module name / Segment name / Group name	Used area	Type
1	DEMO		
	??SEG	0020:???? - ????	
	CODE	0100:0000 - 0075	[ROM]
	DATA	0200:0000 - 008A	[ROM]
	STACK	0300:0000 - 00C7	[ROM]
	BOOT	FFFF:0000 - 0004	[ROM]

Next, take a look at the symbol information for this module.

Display Symbol

Display the symbol information for this module.

End

Exit the Program mode and return to the command level.

## Screen 4. Symbol Display

```

<SYMBOL> CPU : 70216                               14-OCT-86 11:25:30
<ADDRESS SYMBOL>  Module Name = DEMO
MAIN              0100:0000 FAR   RESET              FFFF:0000 NEAR
COUNT           0200:0000 W     S1                0200:0020 B
TRYNXT           0100:0051 NEAR  TABLE START      0200:0002 B
DONE             0100:0073 NEAR  NOTFND           0100:0055 NEAR
END_BENCH       0100:0075 NEAR  BENCH_E         0100:0048 NEAR
AGAIN           0100:0018 NEAR  ID              0100:0039 B
TABLE END       0200:007A B     STRING END       0200:008A B
*****
<TRACE> CPU : 70216   LINE : 1/9                               14-OCT-86 11:25:30

```

All symbols or only symbols in a specified module can be displayed. Each symbol is displayed along with its assigned address and the type (listed in the following table).

- FAR  
Far procedure (located in a different segment)
- NEAR  
Near procedure (located in the same segment)
- B  
Byte data
- W  
Word data

Since more symbols are defined than can be displayed in the upper window, the additional symbols can be displayed by using the Shift-RollUp or Shift-Next Page keys.

**IE-70208/216 Time Mode**

Now it's time to begin some serious debugging work. The program loaded in the IE contains one of the benchmarks from the EDN benchmark suite. Since we are interested in observing how long it takes one or more iterations of the benchmark to execute, the Time mode of the IE-70208/216 will be used.

**MOde Time**

Select the Time mode.

**Condit**

Start by setting the conditions for measuring the execution time.

**Add Event START Address = BENCH\_E Status Fetch**

Because it is tedious to enter triggers, enables and disables for each point, we can define an event that defines a certain condition. In this example, the event is named START and it occurs when the symbol BENCH\_E (the entry point of the benchmark) is fetched. Note that there are a wide variety of conditions which can be selected for determining the event.

Add Event STOP Address = END\_BENCH Status Fetch  
Next using the symbol at the end of the benchmark, an event is defined marking the completion of the benchmark.

**Add Enable Event START**

Using the event START, an enable is defined to enable the timing measurement. It becomes Enable 1 in screen 5.

**Add Disable Event STOP**

Likewise, the STOP measurement is defined to enable the timing measurement. It becomes Enable 1 in screen 5.

End

All conditions are configured so return to the command line.

**Screen 5. Time Mode Condition Display**

<TIME-CONDITION>		CPU : 70216	USED POINT : 2	14-OCT-86	11:26:52
NO	ENABLE (nex)	DISABLE (nex)			
1	START (ev)	STOP (ev)			
-----					
<EVENT>	ADDRESS	DATA	STATUS	EXT-DATA	EXECUTE
START	= 01048	= XXXX	FETCH	= XXXXXXXX	nex
	[BENCH_E]				
STOP	= 01075	= XXXX	FETCH	= XXXXXXXX	nex
	[END_BENCH]				

Up to eight points can be specified as enables or disables. This allows complex timing measurements to be made involving multiple regions of the program.

**Go Continue Enable 1 Disable 1**

To begin measurements, a Go command is used. The Continue qualifier tells the IE-70208/216 to make repeated timing measurements. When this command is executed, the IE-70208/216 clears the window and the measurement times are displayed. This would be useful if the execution time varies depending on the input data. Since the current,

maximum and minimum times are displayed, it is easy to determine the best case, worst case and typical timing of the module. It also possible to vary the clock to determine the effect that clock rate has on the program.

## Screen 6. BENCH\_E Measurement

```

<TIME> CPU : 70216                                14-OCT-86 11:28:15

    MAX   =           268 (usec.)
    MIN   =           267 (usec.)

    TIME  =           268 (usec.)
    
```

### Display Condit

Even while the timing measurements are taking place, other commands can be executed to display memory, registers or change a configuration. In this example, the time display is moved to the bottom window and the time conditions are displayed in the upper window.

### Stop

That's enough so terminate the Go command with the Stop softkey.

## Screen 7. Time Conditions and Measurement

```

<TIME-CONDITION> CPU : 70216 USED POINT : 2          14-OCT-86 11:28:30
NO ENABLE (nex)  DISABLE (nex)
1  START (ev)   STOP (ev)
-----
<EVENT> ADDRESS DATA STATUS EXT-DATA EXECUTE
START   = 01048   = XXXX   FETCH   = XXXXXXXX nex
        [BENCH_E]
STOP    = 01075   = XXXX   FETCH   = XXXXXXXX nex
        [END_BENCH]
*****
<TIME> CPU : 70216                                14-OCT-86 11:28:41

    MAX   =           268 (usec.)
    MIN   =           267 (usec.)

    TIME  =           267 (usec.)
    
```

**IE-70208/216 Break Mode**

We have seen the Time mode and how it helps in the measurement of the timing between events. Next we will explore the Break mode which allows program execution to be slowed down or interrupted when certain events take place.

**Reset**

Begin by resetting the IE-70208/216 program to the initial state.

**MOMode Break**

Select the Break mode.

**Condit**

Begin by defining the conditions for breakpoints.

**Add Event ENTRY Address = BENCH\_E EXECUTE**

Let's define an event which will occur when the symbol BENCH\_E (the entry point) is executed. Because the bus status and instruction execution are de-coupled and independent, both execute and non-execute qualifiers are available.

**Add Break Event ENTRY Times 100**

Define the breakpoint to be when event ENTRY has occurred 100 times. This breakpoint is assigned to breakpoint 1.

**End**

Return to the command mode.

**Go Once Until 1**

Instruct the IE-70208/216 to execute instructions until breakpoint 1 is reached. When the breakpoint is reached the program where the breakpoint occurred is displayed.

**Display Register**

Display the registers to show the PS:PC of the stopped program. Note that there are registers labelled @PS, @DS20, @SS and @DS1. These are virtual registers and can be used when the V40/V50 is in the run state and access to the physical segment registers is unavailable.

**Screen 8. Break Mode Display**

```

<REGISTER>          CPU : 7C216                      14-OCT-86  11:29:43
EXECUTE MODE
AW : 0300  BW : 000F  CW : 0078  DW : 0000
SP : 00C4  BP : 0000  IX : 007B  IY : 0002
PS : 0100  DS0 : 0200  SS : 0300  DS1 : 0200
PC : 0049
                                m  vdibsz a p c
                                PSW : 111100000010110

@PS : 0000 @DS0 : 0000 @SS : 0000 @DS1 : 0000

*****
<TRACE> CPU : 70216  LINE : 1043/1051                14-OCT-86  11:29:32
01027 90          NOP                               11111111
01028 BB8A00     MOV          BW,008A                11111111
0102B 90          NOP                               11111111
0102C 81EB7B00   SUB          BW,007B                11111111
01030 E81500     CALL         BENCH_E                11111111
                                030C6 0033  SS: d_write  11111111
                                code fetch  Lost  11111111
01033 FF0600     BENCH_E: PUSH     AW                11111111
01048 50          030C4 0300  SS: d_write  11111111

```

In the Break mode, one of the softkeys is labelled Step and can be used to single-step the V40/V50 an instruction at a time to allow close observation of program execution. After each instruction is executed, the windows are automatically updated to reflect the new processor state.

**Step**

Each time the step key is pressed a single instruction is executed.

**Step**

Each time the step key is pressed a single instruction is executed.

## Screen 9. Trace Mode Register Display

```

<REGISTER>          CPU : 70216                      14-OCT-86  11:30:09

AW : 0300  BW : 000F  CW : 0078  DW : 0000
SP : 00C0  BP : 0000  IX : 007B  IY : 0002
PS : 0100  DS0 : 0200  SS : 0300  DS1 : 0200
PC : 004B
                                m  vdibsz a p c
                                PSW : 1111000000010110

@PS : 0000 @DS0 : 0000 @SS : 0000 @DS1 : 0000

*****
<TRACE> CPU : 70216  LINE : 1021/1029  14-OCT-86  11:30:09
01030 E81500          CALL  BENCH_E          11111111
                                030C6 0033          SS: d_write      11111111
                                code fetch          Lost 11111111
01033 FF0600          PUSH  AW          11111111
01048 50             BENCH_E: 030C4 0300          SS: d_write      11111111
                                PUSH  DW          11111111
01049 52             030C2 0000          SS: d_write      11111111
                                PUSH  CW          11111111
0104A 51             030C0 0078          SS: d_write      11111111

```

A variable named COUNT is declared in the benchmark to be incremented each time an iteration of the benchmark was completed. The Mem/reg command can be used to display COUNT and examine the current contents.

### Mem/reg COUNT

Enter the memory/register level and display the variable COUNT.

### Size Word

Set the display mode to word since COUNT was declared to be a word variable.

### Display Memory COUNT

Display the variable count which should have the value 99 (63H). This is one less than 100 because it is incremented after the execution of the function and the breakpoint occurred on the 100th entry into BENCH.E.

### End

Return to the command level.

### Condit

Enter the Condition mode and program the next breakpoint.

### Add Event IY\_REG Address = TRYNEXT EXECute

Add an event that occurs when the instruction at symbol TRYNEXT is executed.

### Add Break Event IY\_REG Times 3 Condit IY = 10

Add a breakpoint that occurs on the 3rd time that event IY\_REG occurs with the contents of IY = 10.

### End

Return the command level.

### Display Memory COUNT

Display the variable COUNT and remember its value so we can see where the breakpoint actually occurred.

### Go Once Until 2

Execute the program up to (and including) the 2nd breakpoint.

### Display Register

Display the registers to we can see the contents of IY after the breakpoint is reached.



**Screen 10. Break Mode Display**

```

<REGISTER>          CPU : 70216                      14-OCT-86  11:38:22
      NON EXECUTE MODE
AW : 0348  BW : 000F  CW : 005C  DW : 0000
SP : 00C0  BP : 0000  IX : 007B  IY : 0010
PS : 0100  DSO : 0200  SS : 0300  DS1 : 0200
PC : 0051
                                m  vdibsz a p c
                                PSW : 111100000000110

@PS : 0000 @DS0 : 0000  @SS : 0000 @DS1 : 0000

*****
<TRACE> CPU : 70216      LINE : 784/792                14-OCT-86  11:38:15
01052 AE                      CMPMB                      11111111
                                0200E **30 lower DS1: d_read  11111111
01053 7406                      BE      MATCH1      Lost 11111111
01055 BFFFFF      NOTFND:  MOV      IY,0FFFF      Lost 11111111
01051 F2          TRYXNT:  REPNE                      11111111
01052 AE                      CMPMB                      11111111
                                0200F 30** upper DS1: d_read  11111111
01053 7406                      BE      MATCH1      Lost 11111111
01055 BFFFFF      NOTFND:  MOV      IY,0FFFF      Lost 11111111

```

When the processor is stopped, COUNT can be examined to see how many iterations passed before the breakpoint was reached. If one needs to count events, the Count mode is available to accurately determine the number of times a trigger event has occurred.

**Display Memory COUNT**

Display of the variable COUNT reveals that the count has been incremented by 2.

**Condit**

Let's setup another breakpoint.

Add Event TOUT Address >= BENCH.E & <= END.BENCH  
EXExecute

Define the event to be when executing between the addresses BENCH.E (the entry point) and END.BENCH (the exit). This address range will allow the breakpoint only to occur within this address range.

Add Break Event TOUT Condit CW = 0

Using the event TOUT (timeout) we set up the breakpoint to occur when the condition CW = 0 is satisfied. This event should occur if the block transfer instruction terminates due to a count of zero.

**End**

Return to the command level.

## Screen 11. Break Mode Conditions

```

<BREAK-CONDITION> CPU : 70216 EXCEPT : NO USED POINT: 4 14-OCT-86 11:39:46
NO BREAK-POINT(ex)TIMES CONDITION
1 ENTRY (ev) 100
2 IY REG (ev) 3 IY = #10
3 TOUT (ev) 1 CW = #0
-----
<EVENT> ADDRESS DATA STATUS EXT-DATA EXECUTE
ENTRY = 01048 = XXXX OFF = XXXXXXXX ex
      [BENCH_E]
IY_REG = 01051 = XXXX OFF = XXXXXXXX ex
      [TRYNKT]
TOUT >= 01048 = XXXX OFF = XXXXXXXX ex
      [BENCH_E]
      & <= 01075
      [END_BENCH]
  
```

Next we can add the register display to the upper window and execute to the third breakpoint.

Display Register  
Display the register contents.

Go Once Current Until 3  
Start from the current position and continue until breakpoint 3 is reached. Note that the registers indicate that the breakpoint occurred when CW was decremented to zero.

## Screen 12. Break Mode (Register CW = 0)

```

<REGISTER> CPU : 70216 14-OCT-86 11:40:22
NON EXECUTE MODE
AW : 0348 BW : 000F CW : 0000 DW : 000F
SP : 00BC BP : 0000 IX : 008A IY : 006B
PS : 0100 DS0 : 0200 SS : 0300 DS1 : 0200
PC : 0064
m vdibaz a p c
PSW : 1111000001000110

@PS : 0000 @DS0 : 0000 @SS : 0000 @DS1 : 0000

*****
<TRACE> CPU : 70216 LINE : 762/770 14-OCT-86 11:40:21
01064 5E 02088 **43 lower DS0: d_read 11111111
01065 5F POP IX Lost 11111111
01066 8BCA POP IY Lost 11111111
01068 7404 MOV CW,DW Lost 11111111
01062 F3 BE FOUND Lost 11111111
01063 A6 REPE 11111111
CMPBKB 11111111
0206A **48 lower DS1: d_read 11111111
02089 48** upper DS0: d_read 11111111
  
```

**IE-70208/216 Trace Mode**

Next the demonstration will explore the use of the Trace mode. The Trace mode is used to capture state information during the real-time execution of a program.

Reset

Reset the program and prepare for using the Trace mode.

MOde Trace Arm Or

Select the trace mode with triggers OR'ed.

Condit

Select the conditions and triggers for the trace mode.

Add Event BENCHE Address  $\geq$  BENCH\_E  $\leq$  END BENCH

Define the event to be between the addresses BENCH.E (the entry point) and END.BENCH (the exit). This address range will allow the breakpoint only to occur within this address range.

Add Event R\_STK Status D-Read Ss

Define the event R\_STK (read stack) to be a data read using the stack segment register. Note that reads using the BP register as well as the SP will satisfy this conditions.

Add Event W\_STK Status D-Write Ss

Define the event W\_STK (write stack) to be a write read using the stack segment register. Note that writes using the BP register as well as the SP will satisfy this condition.

Add Trigger Event R\_STK

Define event R\_STK to be a trigger.

Add Trigger Event W\_STK

Define event W\_STK to be a trigger

Add Enable Event BENCHE

Use the event BENCHE as an enable for the triggers.

End

Return to the command level.

**Screen 13. Trace Mode Conditions**

<TRACE-CONDITION>		CPU : 70216		14-OCT-86 11:42:02	
NO	TRIGGER (nex)	ENABLE (nex)	DISABLE (nex)	QUALIFY (nex)	USED POINT : 4
1	R_STK (ev)	BENCHE (ev)			CONDITION
2	W_STK (ev)				
-----					
<EVENT>	ADDRESS	DATA	STATUS	EXT-DATA	EXECUTE
BENCHE	$\geq$ 01048 [BENCH_E]	= XXXX	OFF	= XXXXXXXX	nex
	$\leq$ 01075 [END BENCH]				
R_STK	= XXXXX	= XXXX	D-READ (SS)	= XXXXXXXX	nex
W_STK	= XXXXX	= XXXX	D-WRITE (SS)	= XXXXXXXX	nex

The next command instructs the IE to perform at trace when trigger 1 is reached. Traces can be performed before, about or after the trigger point to allow a variety of events to be observed.

Go Once ABOUT 1

Start executing the program until the triggers are reached. Because the trigger OR mode was selected, any read/write using the SS register inside the BENCH.E procedure will cause the trigger. The ABOUT command specifies the captured data be centered around the trigger.

Stop

Demonstration complete, all done.

### Screen 14. Trace Mode Display

```

<TRACE> CPU : 70216      LINE : 19/37      14-OCT-86 11:42:39
01024 BE7B00      MOV    IX,007B      11111111
01027 90          NOP                    11111111
01028 BB8A00     MOV    BW,008A     11111111
0102B 90          NOP                    11111111
0102C 81EB7B00   SUB    BW,007B     11111111
01030 E81500     CALL  BENCH_E     11111111
                   030C6 0033      SS: d_write 11111111
                   code fetch
01033 FF0600     BENCH_E: PUSH  AW      Lost 11111111
01048 50         030C4 0300      SS: d_write 11111111
01049 52         PUSH  DW      11111111
                   030C2 0000      SS: d_write 11111111
0104A 51         PUSH  CW      11111111
                   030C0 0078      SS: d_write 11111111
0104B FC        CLR1  DIR      11111111
0104C 8A04      MOV    AL,BYTE PTR [IX] 11111111
                   0207B 48** upper DS0: d_read 11111111
0104E 2BCB     SUB    CW,BW      11111111
01050 41        INC    CW         11111111

```

Because of the large number of powerful features, not all of the capabilities of the IE-70208/216 have been explored in this mini-tutorial. For additional information on the IE-700k contact your local NEC sales office.



### Description:

The EBIBM-V25MINI-IE, EBIBM-V40MINI-IE and EBIBM-V50MINI-IE are powerful development tools which provide software debug capabilities for the NEC  $\mu$ PD70320 (V25), 16 bit microcomputers as well as  $\mu$ PD70208 (V40) and  $\mu$ PD70216 (V50) microprocessors. In the following it will just be called MINI-IE. The concept is based on PC-XT/AT and Advanced Fullscreen Debugger (AFD-Sym)\* to support symbolic debugging. Additionally the hardware provided consists of a parallel I/O interface (modified Centronics card) plugged into PC slot plus an emulation pod for either the V25, V40 or V50 microprocessor. They combine real-time and single step emulation with user-programmable break points, on-line help function, command file facilities and a user-friendly interface to produce a powerful debugging tool. User programs. DPS EXE-files and COM-files as well as NEC. LNK-files can be down-loaded into target system via 8 bit parallel bus. In case there is a system with more than one V25, V40 or V50 to be developed, up to three Mini-IEs can be attached to one PC in parallel.

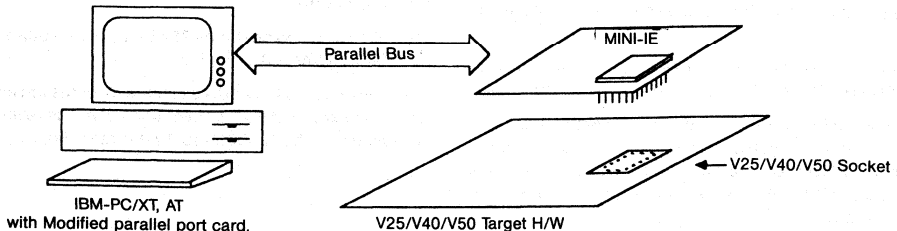
### Features:

- Easy to use with high functionality
- Precise real-time and single-step emulation with 8 MHz internal or up to 8 MHz external clock
- Full screen display of all relevant information, updated after each command
- All register and memory by full-screen overtyping
- Automatic setup and execution from batch/macro file
- One line HELP screens and user guidance for command entries
- Command retrieve function
- 32 bit arithmetic expression evaluation
- Directory display and file select capability for download
- Browse function for ASCII files
- On line assembler and disassembler (Native mode)
- Single Step and 'Procedure Step'

- Running programs can be stopped by:
  - Keyboard (Ctrl-Esc)
  - conditional breakpoints
  - software interrupt (INT3)
  - NMI signal
- Up to 8 conditional breakpoints with occurrence counter
- Fast conditional TRACE, recording instructions and register contents
- One additional immediate address stop
- Register bank switching (V25 only)
- Patch, Fill, Search and I/O commands
- Printer output facility for disassembled code, data or trace buffer
- Resident operation. Memory can be freed when AFD is no longer required
- Up and down load facility for files through 8 bit parallel bus

\*Note: AFD-Sym is a licensed product from Fa. AdTec and AFD-Sym is a modified version of the well known AFD-Pro

**Fig. 1: Configuration of EBIBM-VxxMINI-IE**



**Ordering Information:**

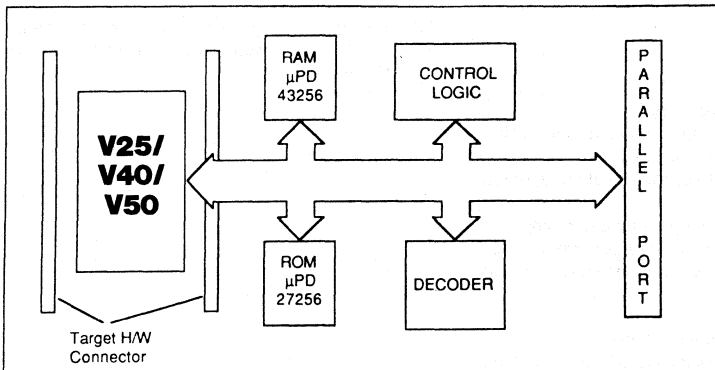
Part Number	Description
EBIBM-V25MINI-IE	Mini ICE for V25 including AFD
EBIBM-V40MINI-IE	Mini ICE for V40 including AFD
EBIBM-V50MINI-IE	Mini ICE for V50 including AFD

**Hardware Description:**

The MINI-IE is based on quite a simple hardware that consists of two boards. The first one is a modified printer adapter card which is hooked on the PC bus and supports bi-directional data transfer between the PC and the second

board or the target hardware respectively. The second board is a single Eurocard (see Fig. 2) including V25, V40 or V50 as target CPU plus connector to be plugged into user/target system, furthermore it incorporates 8k (or 2x8k for V50) byte EPROM, 64k byte CMOS RAM (32k byte for V25), Control- and Decode Logic and a parallel interface. For all three types the memory and I/O address mapping is given in Fig. 3. All three boards are linked through an 8-bit parallel cable.

**Fig. 2: Block Diagram of the personality pod**



**Memory Mapping**

The MINI-IE provides 8/16k ROM and 32/64k RAM which are mapped into the 1 Mbyte address space as indicated in Fig. 3.

The ROM/EPROM contains a small part of 5k byte of AFD-Sym software, the rest is free for user specific purpose.

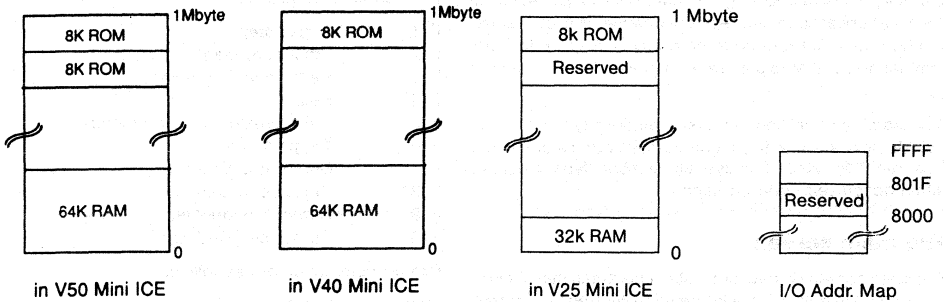
The 32/64k on board RAM is located at address 0 and first 1k byte is used for interrupt table, following 10k bytes are used by AFD-Sym for internal variables and trace buffer. Remaining 21/53k byte are free as Emulation memory for application programs.

Additional Memory if required, can be implemented in the target hardware.

The AFD-Sym also occupies 75k bytes of PC RAM and runs on MS-DOS Ver. 2.0 or higher.

From the 64k byte I/O Address space 32 bytes are internally used and excluded from applications use. The reversed I/O addresses are from 8000h to 801Fh (see Fig. 3).

**Fig 3: Address Map-I/O Map**



## Emulation Mode

The AFD-Sym (Mini-ICE) works in EMULATION mode which requires the hardware as described above. This is different to the ordinary AFD and AFD-Pro which loads the user program just into PC RAM for debugging. But AFD-Sym loads any application program via the parallel interface down into the Mini-ICE RAM or the target system RAM for execution and debugging respectively. After first load AFD-Sym is kept resident in the PC RAM with the big advantage to leave the debugger for some operation under MS-DOS and to come back to AFD-Sym afterwards to carry on debugging from exactly the same address and with the same status as it has been stopped before switching to MS-DOS. Because more than one Mini ICE can be connected to the PC the command for this mode is used with an additional number (E1, ..., E3) to select a dedicated Mini ICE.

In EMULATION mode programs can be executed in both real-time and single-step mode.

## Break Capabilities

To do efficient program debugging full control on program execution is required, therefore powerful break capabilities have been implemented.

Up to eight break points can be specified combined with various conditions and actions to be taken, also occurrence counters can be set which allow to control the execution very dedicated. For testing repetitively executed parts (ie. program loops) there is an extremely useful break function implemented called "snapshot", this reports all register contents during a short interrupt at the dedicated break location and continues program execution automatically. Later then after a final break all recorded results can be listed to analyze program functionality.

The breakpoint specification is supported by the display of a disassembled code area, the start address of this area needs to be given in the same menu. Also there is the possibility to set the breakpoints 1 to 8 by using a key combination "ALT-1, ..., 8" without going into this menu, this means when "ALT-1" is pressed breakpoint 1 is set at the address represented by PS:PC content.

## Trace Capabilities

To analyze critical code sections the trace function is really helpful. This facility, if activated records all truly executed instructions in combination with corresponding Register- and Flag values as well as the top four Stack words to show clearly programming errors.

Both breakpoint handling and trace operation is software controlled and not supported by dedicated hardware, this implies that breakpoints can be set to RAM locations only and trace is performed by consecutive single step operations.

## Symbolic Debugging:

For efficient software debugging it is commonly expected to support this at symbol level rather than at address level only. Symbols can be loaded from .MAP and .LST files, also optional code and symbol loading from NEC LK70116 generated files is supported. The symbols can be used as address and data constants.



## Line Assembly/Disassembly

The line assembler can be involved during debugging stage to modify or add some program that can be executed immediately afterwards to see result of change. This convenient function does not require to go always back to Editor, Assembler and Linker to generate a new version of executable code.

Disassembly operation can disassemble any code loaded into Memory starting from a dedicated address or symbol. This function also displays all symbols/labels defined within displayed code partition.

### The main screen

The main screen display consists of six areas that show simultaneously quite a lot of information such as the contents of all processor registers, the top four stack words, a command line followed by an area showing the code at location PS:PC in disassembled form, then two fields that can display two different memory areas in hexadecimal form and the bottom line indicates possible commands with corresponding function key.

### Help function

Generally this tool is very easy to be used because the command syntax is constructed in such a way that commands are identified just by the first character. Additionally AFD-Sym uses the line above the command line as a permanent help line which shows after typing first character the possible command(s) plus the choice of parameters. Further on-line help support at any stage is given by pressing function key F4. Four help menus are available and give a short explanation on all possible commands which eliminates basically the need of a User's Manual.

## List of important commands

Basic operations/commands are already assigned to the function keys as follows:

F 1	Single step
F 2	Procedure step
F 3	Retrieve last command
F 4	Help
F 5	Enter breakpoint entry menu
F 6	Toggle scree
F 7	Cursor one field up
F 8	Cursor one field down
F 9	Cursor one field left
F10	Cursor one field right

Further commands are as follows:

Command	Function
L, W	Load from or write into a file
R, D	Display register or memory contents
G	Execute program
Quit	Quit and return to DOS
A	Enter line assembler mode
F	Fill memory
S	Search for data in memory
C	Compare to memory areas
CO	Copy one area to another
I/O	Input and display or output I/O data

## Installation

The AFD-Sym runs on IBM-PC, DOS version 2.0 or above. Up to three Mini ICEs can be attached as described above by using printer ports LPT1...LPT3.

## Summary

These Mini ICEs provide a powerful, user friendly and cost effective tool to the customer fulfilling most of the requirements a user may have during development stage.

## Description

The V20/V30 Design / Development Kit (EBIBM-70108/70116) is a convenient tool to develop both Hardware and Software using the  $\mu$ PD70108/ $\mu$ PD70116 Microprocessors from NEC. The EBIBM-70108/70116 design kit can be connected via a dedicated  $\mu$ PD71051 to a Host (eg. IBM PC), this serial link can be used to download programs (object code) from Host into the RAM on the design kit. These programs then can be executed / debugged under monitor control. A second serial interface the EBIBM-70108/70116 is providing has to be connected to a pure terminal that is used to control the monitor and to work with the design kit respectively. The so called XLINK command of the monitor puts the design kit into a transparent mode that permits to work with the host by controlling it just from the terminal.

The EBIBM-70108/70116 monitor provides a full range of features for debugging such as instruction trace, breakpoints, register- and memory-control in native mode as well as 8080 emulation mode and code disassembly. A wide range of other functions expected in a single board microprocessor monitor are also available to the user.

To ease the design of application specific hardware based on the V/20V/30, the EBIBM-70108/70116 design kit contains a large uncommitted area available for user prototyping of application hardware. To further increase the ease of prototype design and development, an expansion bus capable of supporting a variety of IBM PC expansion bus compatible cards is also provided. This permits fast, low cost upgrade of memory and I/O devices to support a wide range of potential user applications with maximum flexibility.

The EBIBM-70108/70116 design kit has been designed to accept either a V20 or a V30 microprocessor through the use of programmable bus width logic. By means of a pair of jumpers, the EBIBM-70108/70116 hardware adjust the bus width to either 8 or 16 bits. The EBIBM-70108/70116 design kit monitor software has also been designed to work with either microprocessor.

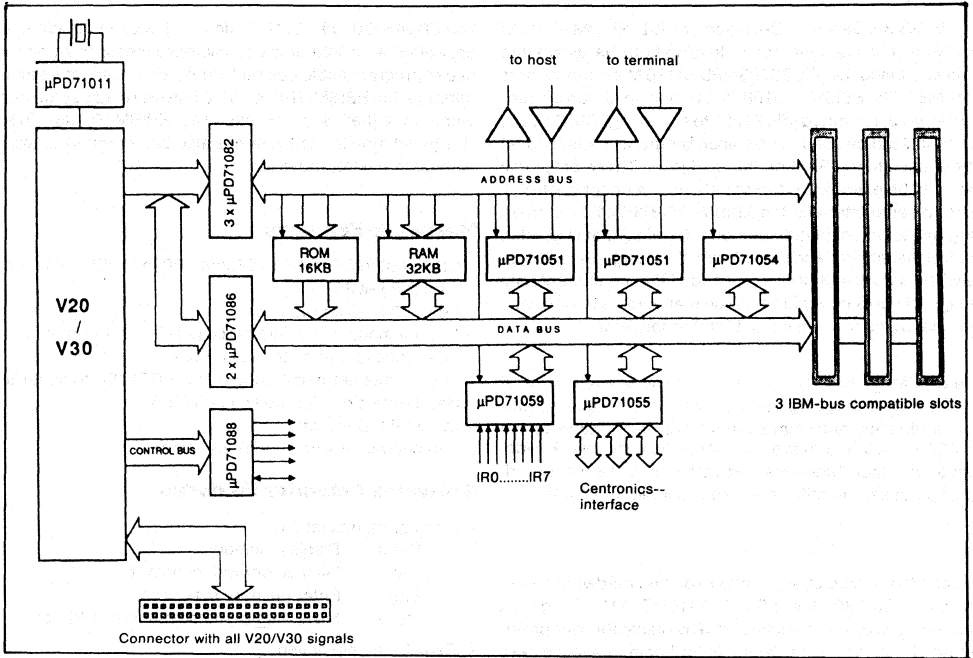
## Hardware Features

- CPU (V20 in EBIBM-70108 and V30 in EBIBM-70116)
- 64 k Byte ROM
- up to 128 k Byte Ram
- two additional serial interfaces  $\mu$ PD71051, to provide interfaces to a terminal and a Host
- one additional Timer/Counter Unit  $\mu$ PD71054, to generate Baud Rate clock for the two  $\mu$ PD71051
- three IBM-Bus slots
- one connector with all V20/V30 signals

## Software / Monitor Features

- Memory manipulation
  - Dump: Display memory
  - Fill: Fill memory with constant
  - Enter: Enter memory byte-wise
  - Move: Move memory to destination address
- Register manipulation
  - Display registers
  - Set registers
- Help command
- Programm execution in real time (RUN)
- Programm execution in single steps (TRACE)
- I/O manipulation
- Disassembling in
  - Native Mode and
  - Emulation mode
- Load Object Code from Host into on-board RAM
- XLINK to perform a transparent link between the terminal and the Host

**Blockdiagram of EBIBM-70108/70116**



## Description

The V40/V50 Design / Development Kit (EBIBM-70208/70216) is a convenient tool to develop both Hardware and Software using the  $\mu$ PD70208/ $\mu$ PD70216 Microprocessors from NEC. The EBIBM-70208/70216 design kit can be connected via a dedicated  $\mu$ PD71051 to a Host (eg. IBM PC), this serial link can be used to download programs (object code) from Host into the RAM on the design kit. These programs then can be executed / debugged under monitor control. A second serial interface the EBIBM-70208/70216 is providing has to be connected to a pure terminal that is used to control the monitor and to work with the design kit respectively. The so called XLINK command of the monitor puts the design kit into a transparent mode that permits to work with the host by controlling it just from the terminal.

The EBIBM-70208/70216 monitor provides a full range of features for debugging such as instruction trace, breakpoints, register- and memory-control in native mode as well as 8080 emulation mode and code disassembly. A wide range of other functions expected in a single board microprocessor monitor are also available to the user.

To ease the design of application specific hardware based on the V40/V50, the EBIBM-70208/70216 design kit contains a large uncommitted area available for user prototyping of application hardware. To further increase the ease of prototype design and development, an expansion bus capable of supporting a variety of IBM PC expansion bus compatible cards is also provided. This permits fast, low cost upgrade of memory and I/O devices to support a wide range of potential user applications with maximum flexibility.

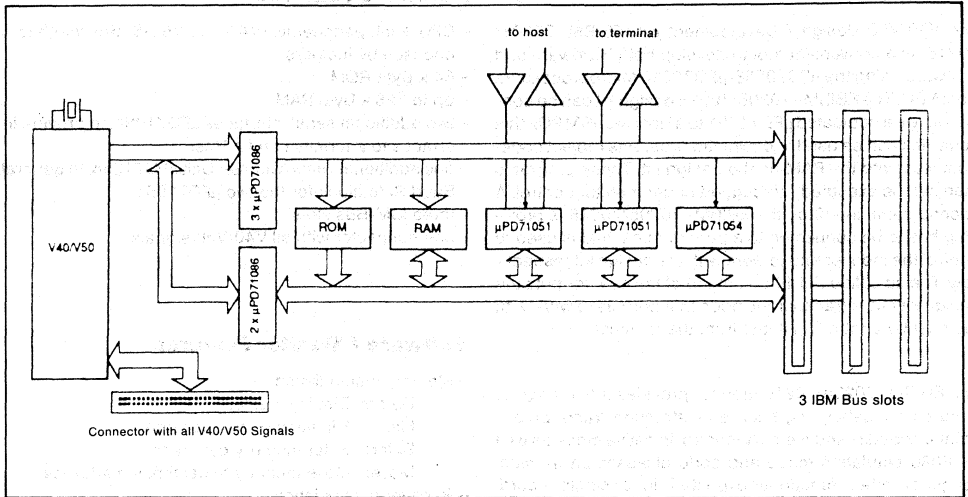
The EBIBM-70208/70216 design kit has been designed to accept either a V40 or a V50 microprocessor through the use of programmable bus width logic. By means of a pair of jumpers, the EBIBM-70208/70216 hardware adjusts the bus width to either 8 or 16 bits. The EBIBM-70208/70216 design kit monitor software has also been designed to work with either microprocessor. The design kit requires no  $\mu$ PD70208/216 internal peripherals. They are free to be used by the application software. The onboard monitor software shares the CPU with the application software on a mutually exclusive basis and to support slow and dynamic memories on the EBIBM-70208/70216 motherboard, the Refresh Control Unit and Wait Control Unit can be optionally programmed when the EBIBM-70208/70216 is reset.

## Hardware Features

- CPU + int. peripherals (V40-8 or V50-8) Bus width is chosen by jumpers
- 64 k Byte ROM
- up to 128 k Byte RAM
- two additional serial interfaces  $\mu$ PD71051, to provide interfaces to a terminal and a Host
- one additional Timer/Counter Unit  $\mu$ PD71054, to generate Baud Rate clock for the two  $\mu$ PD71051
- three IBM-Bus slots
- one connector with all V40/V50 signals

## Software / Monitor Features

- Memory manipulation
  - Dump: Display memory
  - Fill: Fill memory with constant
  - Enter: Enter memory byte-wise
  - Move: Move memory to destination address
- Register manipulation
  - Display registers
  - Set registers
- Help command
- Programm execution in real time (RUN)
- Programm execution in single steps (TRACE)
- I/O manipulation
- Disassembling in
  - Native Mode and
  - Emulation mode
- Load Object Code from Host into on-board RAM
- XLINK to perform a transparent link between the terminal and the Host



## Description

The EBIBM-V60UNIX board is a coprocessor board for the PC-XT/AT™ personal computers, incorporating an NEC high performance 32-bit microprocessor  $\mu$ PD70616R-16 (V60™). This board can be used for demonstration and evaluation purposes of the V60 microprocessor, also to develop on a PC software for V60/V70 applications this board can run the V60/V70 software generation package PKG70616. A third possibility to make use of is to change very simple a PC-XT or -AT™ into a UNIX™ machine to run available UNIX™ software. The board is delivered with a ready-to-run UNIX V™ and 4.2 BSD including the software generation package (with C-Compiler, Rel. Assembler, Linker, Configurator and Librarian), see also data sheet of PKG70616 within this section.

## Hardware Features

- V60 CPU, with 16MHz operation
- 4M bytes dynamic RAM (32 x 1M bit)
- eight Interrupt inputs ( $\mu$ PD71059)
- three 16 bit timer channels  
(2 channels as interval times; 1 channel as baud rate generator for RS-232)
- one serial channel (RS-232 by a  $\mu$ PD71051)
- applicable to PC-XT/AT and PS/2 model 30 extension slots

TM: UNIX, UNIX V are trademarks of AT&T  
 IBM PC-XT, AT and PS/2 model 30 are trademarks of International Business Machines Corporation  
 V60 is a registered trademark by NEC Corporation

## Ordering Information

Product No.	Host System	Operating System	Medium
EBIBM-V60UNIX	IBM-PC-XT/AT PS/2 model 30	UNIX	V5 1/4" double sided double density floppy disk

## Software Features

- UNIX V and 4.2BSD is ported to
- 280 UNIX commands are implemented
- four virtual terminals are supported
- communication with Kermit
- File conversion UNIX/MS-DOS

## Documentation

For further information on the EBIBM - V60UNIX, NEC Electronics provides the following manual, describing how to install and how to use the board:

### EBIBM - V60UNIX User's Manual

This document is provided with the purchased package (board plus Floppies). Additional copies may be obtained from NEC Electronics.

## Licence Agreement

The EBIBM - V60unix (with UNIX) operating system is sold under terms of a Licence agreement. A signed agreement is a pre-condition for the purchase of the EBIBM - V60UNIX. The shipped product will be accompanied by a "Registration card" which must be completed and returned to NEC Electronics (Europe) GmbH to ensure that software updates are provided later on.

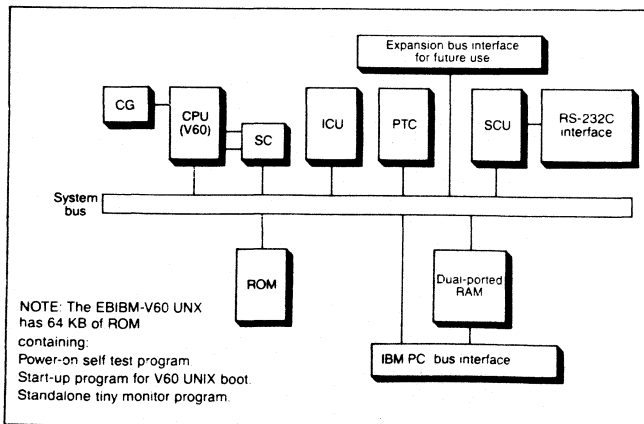


Figure EBIBM-V60 UNIX Block Diagram



## Description

The EBIBM-V60SBC is a single board computer with Multibus I™ interface. The board is equipped with V60™ micro-processor operating at 16 MHz, with 256 Kbyte ROM and 2 Mbyte RAM in a two part configuration. The on board RAM controller has also parity check function. For high speed numerical calculations the CPU is accompanied by the numerical coprocessor AFPP (μPD72691) and will give a significant performance improvement to an existing Multibus I system being used for industrial/process control purpose. The board features complete monitoring of critical areas which include data validity, bus access, software (program) execution, clock oscillation, CPU execution state and power failure, allowing easy and reliable integration in existing Multibus I system.

## Features

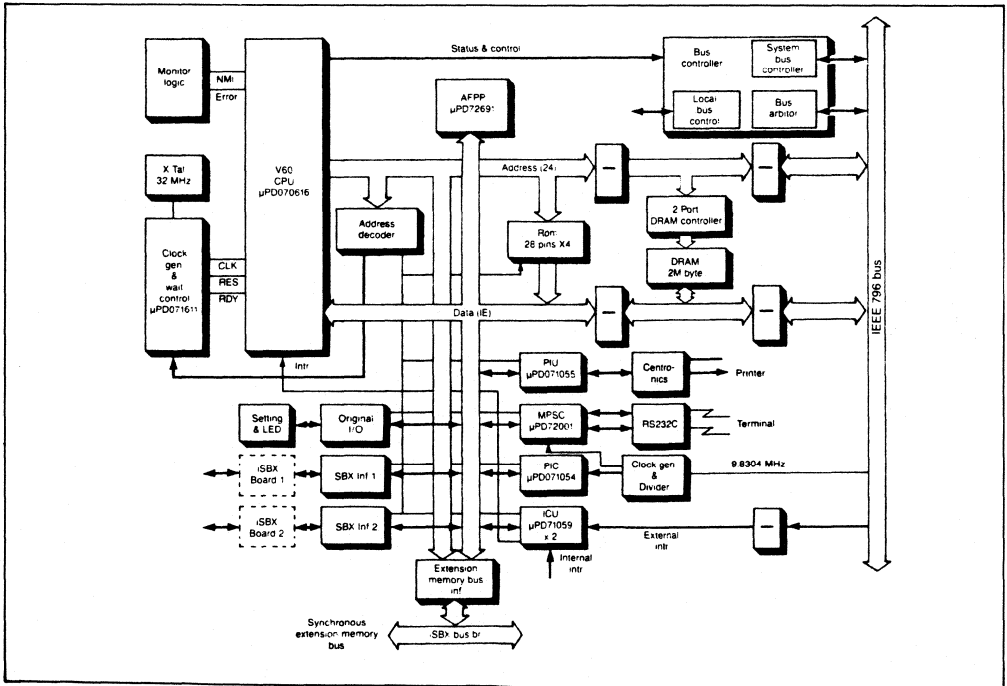
- V60 CPU with 16 MHz clock
- AFPP (μPD72691) on board for numeric support
- 4 Mbyte RAM and 256k ROM
- two RS-232 channels (μPD72001)
- a Centronics interface (μPD71055)
- two iSBXTM bus interfaces to I/O as well as extension memory
- three 16 bit times channels (μPD71054)
- 6 ext. interrupts (μPD71059)

TM: Multibus I and iSBX are registered trademark's of Intel corp. V60 is a registered trademark of NEC Corporation

## Ordering Information:

Product No.	Host System	Operating System
EBIBM-V60SBC	Multibus I	no specific (i.e. RX616)

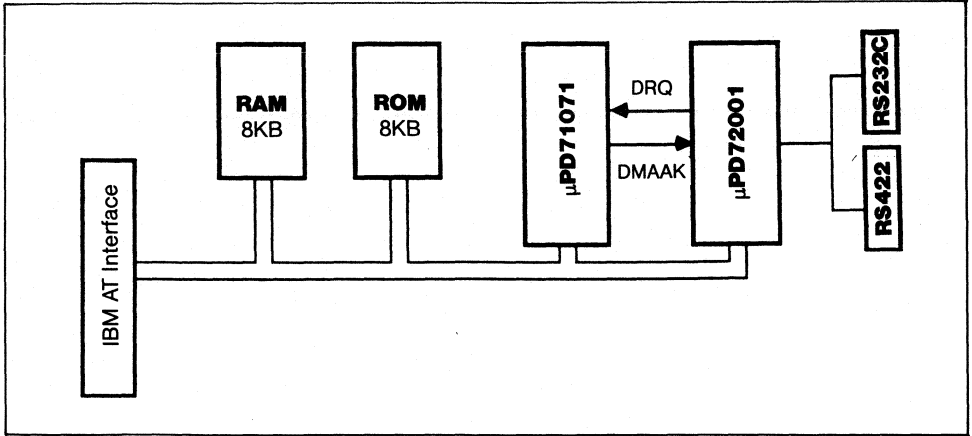
Figure EB-V60 SBC Block Diagram







The EBIBM-72001 is an evaluation tool for the advanced multi protocol serial communications controller  $\mu$ PD72001 (AMPSC). The blockdiagram of the IBM XT compatible board is shown in the following figure.



All communication protocols supported by the AMPSC can be evaluated using this board. Line drivers are implemented for HDLC/SDLC-, MONO SYNC/BISYNC- and ASYNC protocols. The line drivers are software selectable to support RS232c or RS422/449 standards. By this, the board can be configured for a wide range of applications.

A DMA controller  $\mu$ PD71071 allows high speed data transfer between the  $\mu$ PD72001 and the personal computer.

The board provides sockets for 8K Byte of EPROM and 8K Byte of RAM. For evaluation purposes it might be desirable to replace the EPROM by RAM. Therefore the EBIBM-72001 provides the possibility to configure the memory configuration by jumper. The on-board memory is mapped into the

IBM XT memory, allowing to easily load programs into the local memory. However, there is no need to write low level software. The board comes with a complete set of software drivers for the various protocols (HDLC, BISYNC, ASYNC). The user can call these routines, which are written in assembly language, from his application software in high level language. As the software drivers are fully documented, it is a simple matter to modify them for individual needs.

A supervisor software in BASIC controls the EBIBM-72001 and provides a menu-driven evaluation tool to examine all internals of the  $\mu$ PD72001.

Therefore the EBIBM-72001 can be used as an education-tool to become familiar with the AMPSC as well as development tool to implement higher level protocols.



## IBM-Board for $\mu$ PD7220A and $\mu$ PD41264 Dual Port VRAM

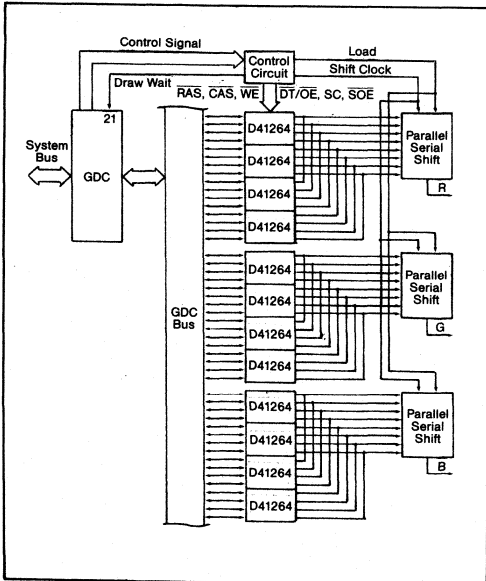
The board is dedicated for the demonstration of the extreme high drawing speed of the GDC  $\mu$ PD7220A together with the  $\mu$ PD41264 Video RAMs. It cannot be used to replace existing IBM color boards because it is not 100% IBM compatible (E.g. The  $\mu$ PD7220A draws a vector by simply specifying the start address, direction and length whereas the original IBM color board needs each pixel of a vector to be calculated by the microprocessor).

Nevertheless it is possible to use this board as an IBM color board after modifying the graphic drawing driver procedures of graphics programs.

Note: There are 100% compatible IBM color board available on the market which make use of the  $\mu$ PD7220A. These boards are nearly standard color cards with the additional feature that the on board GDC can also access the image memory for faster drawings. The EBIBM-7220A board design differs from these boards. (e.g. no 6845 CRT controller has been used; no direct image memory access by the microprocessor is available)

The board is useful to familiar with the GDC and to test certain features. For this purpose there is a menu driven evaluation/demonstration software which comes along with the board.

### Block Diagram EBIBM-7220A



It has been written in PASCAL (Turbo Pascal = Trade Mark of Borland Inc.) to enable the user to modify the program very easily. This evaluation software runs on all IBM color or monochrome displays.

On the provided diskette this program is stored in source code and executable file format for two different color monitor resolutions.

### Hardware Description

The GDC-VRAM provides a resolution of 1024 dots per line by 1024 lines for each of three color planes. These three color planes can therefore store each pixel in one of 8 colors. (E.g. a yellow pixel is stored by setting its position in the red and green plane and resetting it in the blue plane). Because of the 16 bit data bus of the  $\mu$ PD7220A and the 64 K x 4 bit organisation of the  $\mu$ PD41264 Video RAM four devices have been used for each plane to simplify the design. Therefore the GDC controls 12 Video RAMs which make a 384K bytes image memory or 3.072 million different pixels.

The selection of the color planes is done by the GDC address bits A16 and A17 which can be selected by the CURS (=Cursor Set) command of the  $\mu$ PD7220A.

### Software Description

The supplied evaluation/demo programs (GDC 14 and GDC 21) on the diskette can be used in two different operation modes. One time it is the education mode where it is used to test certain features and commands of the Graphics Display Controller  $\mu$ PD7220A. This mode is also quite useful to become familiar with the controller.

The other operation is the demonstration mode. Here the GDC will perform a lot of small graphics drawing procedures. This is useful to demonstrate the high speed drawing available with the GDC and Video RAMS and the special drawing features of the GDC. This demonstration will be done in an endless loop which can be left when pressing ESCAPE and waiting a short time.

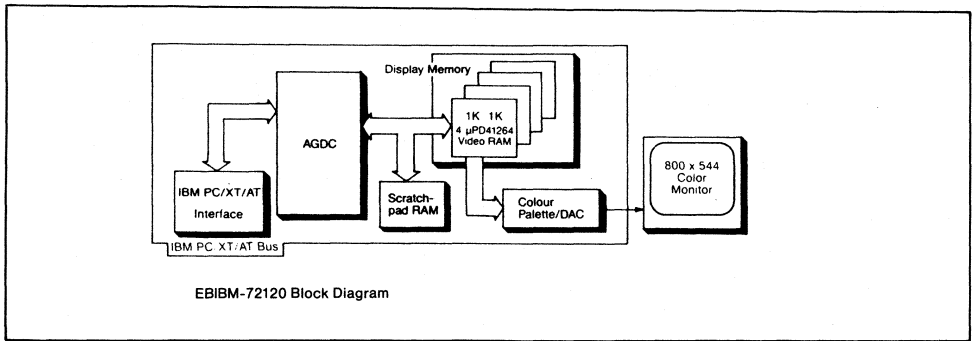
The evaluation/demonstration program can be started by simply pressing GDC 14 or GDC 21 and carriage return. Then the following menu will occur on the screen.



## Advanced Graphic Display Controller

### Description

The new AGDC  $\mu$ PD72120 will be supported by the EBIBM-72120 with a 1024 x 1024 x 4 planes display memory (dual-port Video RAM  $\mu$ PD41264). Mapped into display memory is a 64K x 16 bit scratchpad RAM for character fonts, tiling pattern storage etc. The display interface includes a colour palette and digital to analog converter for connection to a 800 x 544 colour monitor, eg. the NEC Multisync. Fast demonstrations, but also interesting programming examples and the comprehensive documentation are of immediate benefit to system designers.





### EBIBM-72067 AFDC Board for the PC-XT

The following description is divided into two sections, the hardware and software. The hardware part describes the board and how to install it. The software part explains the features of the supplied evaluation program.

#### Hardware Description

The essential part of the board is the  $\mu$ PD72067 Advanced Floppy Disk Controller (AFDC).

This CMOS device incorporates

- the main FDC like  $\mu$ PD72065
- window signal generation VFO circuit
- write precompensation circuit
- system clock generation circuit

which all were external peripheral circuits in conventional FDC designs. Usage of this single VLSI chip enables reduction of system size and improvement of reliability.

An additional feature of the AFDC is the internal mode which allows you to set up the transfer speed, the write precompensation value and the format (ECMA/IBM) per software.

#### Installation

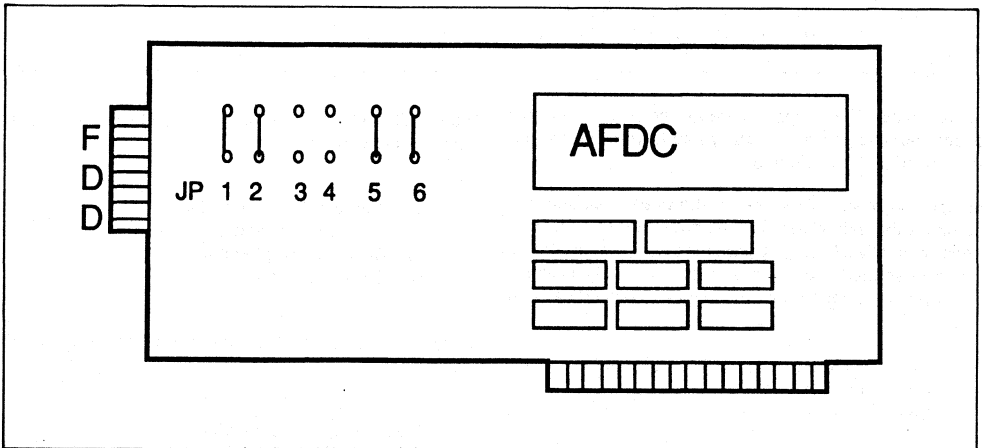
The installation is very easy because you simply remove the FDC-Board out of your PC and plug in the AFDC-Board. After power on the AFDC is in the external mode, that means, that the data rate, precompensation value and IBM/ECMA format is selected by some jumpers. The standard jumper configuration for IBM 5 1/4" DD Diskette is as follow:

DRO: JP1 set (0) = 250 Kbps transfer speed  
DR1: JP2 set (0)

PCSO JP3 not set (1) = 250ns write precompensation  
PCS1: JP4 not set (1)

FMT: JP5 set (0) = IBM format  
EXT: JP6 set (0) = internal PLL used

With this configuration you can use the board as a normal IBM FDC board.





**Software Description**

With the board there comes a special evaluation program named AFDC. This program is quite useful to become familiar with the AFDC because it allows you to issue commands directly to the controller and to get back the result

parameters.

The Turbo-Pascal Source is also available on the Disk. Insert the disk and type AFDC <CR>. The following menu will occur on the screen.

RE	REcalibrate	SR	Spec: Step Rate time	8ms
SE	SEek	HU	Spec: Head Unload time	480ms
FO	FOrmat	HL	Spec: Head Load time	4ms
RI	Read Id	DM	Spec: Dma mode	yes
RD	Read Data	BS	Form: Bytes/Sector	512
RL	Read deLeted data	ST	Form: Sector/Track	9
SQ	Scan eQual	TC	Code: Tracks/Cylinder	2
SL	Scan Low or equal	G3	Form: Gap 3 length	42
SH	Scan High or equal	DP	Form: Data Pattern	F6H
DI	read Dlagnostic	MT	Code: Multi Track	yes
WD	Write Data	MF	Code: MFm/fm mode	yes
WL	Write deLeted data	SK	Code: SKip del. address	no
SI	Sense Int. status	UN	Code: Unit Number	0
SD	Sense Drive status	AS	menu: A F D C Special commands	
SP	SPecify	X	eXit eXit	
MSR: RQM DIO NDM CB D3B D2B D1B D0B		Input:		
ST0: IC IC SE EC NR HD US1 US0		Preset:		
ST1: EN - DE OR - ND NW MA		Result:		
ST2: - CM DD NC SH SN BC MD		DMA:		
ST3: FT WP RY TO TS HD US1 US0				

The left side of the upper screen shows the available commands which are executed by simply typing the two letters in front of the command.

The upper right side of the screen shows some parameters. They have the default value for a normal IBM 5 1/4" Double density disk drive. Changing parameters is the same procedure as described above. The second column of this part of the screen gives you the command for which the parameter is used.

- spec: used for the SPECIFY command
- form: used for disk-formatting
- code: used for various commands

On the lower part of the screen you see the MSR (Main Status Register) and the result parameters (ST0-ST3) which are encoded by their symbols. An inverse printed symbol means that the bit is set to 1. Beside there the preset and the result parameters are shown in hex-format. DMA gives you the number of transferred bytes after a Read- or Write Data command.

You usually will start a session with the program by typing the following commands:

- SP : sent out SPECIFY-command
- RE : RECALIBRATE (seek to track 0)
- SI : Sense Interrupt status
- SD : Sense Drive status

If everything is o.k. the following bits are set (inverse printed):

- RQM : Request for Master
- SE : Seek End
- T0 : Track 0
- RY : Drive Ready





## Hard Disk Controller Board

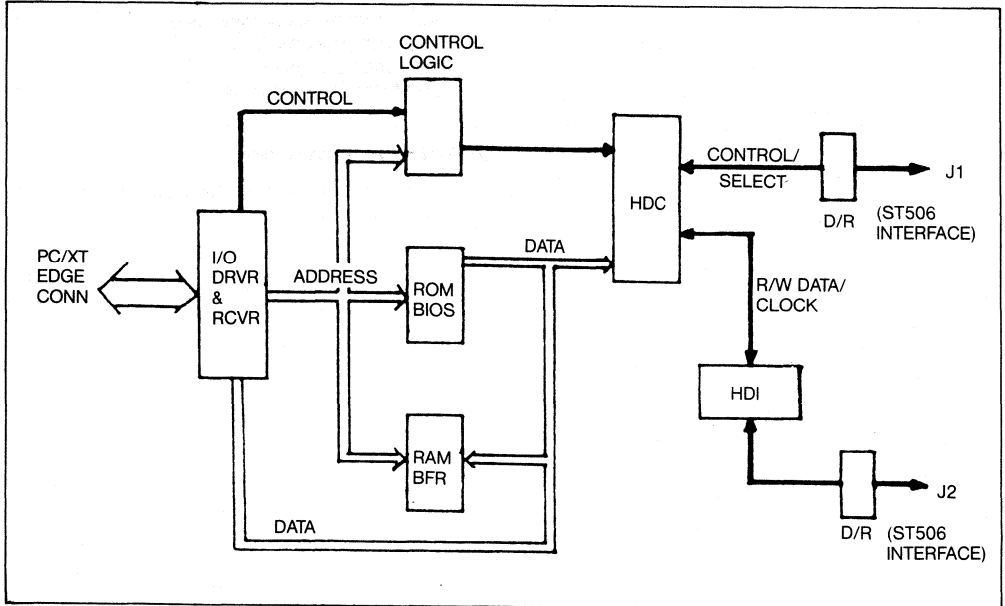
### Description

Originally designed as an application tool, the Hard Disk Controller Board (HDCB) provides a simple interface for an ST506-type winchester in a typical PC environment: specifically the IBM-PC or-PC/XT and look-alikes. This single board controller achieves all the functions as called out in IBM's PC/XT specification for its Fixed Disk Drive Adapter using NEC's  $\mu$ PD7261A HDC chip and the  $\mu$ PD9306, the Hard Disk Interface chip along with necessary drivers and logic for drive selection.

Necessary control firmware is provided in the on-board ROM to utilize the system DMA for data transfers between the disk and memory, to handle interrupts and to provide device level control for the hard disk controller so that the hardware differences are completely transparent to the system CPU and to the user.

The HDCB attaches to the winchester via standard ST506 interface cables and to the PC system board through its I/O channel interface.

### Hard Disk Controller Board Block Diagram



The  $\mu$ PD7261A is an intelligent microprocessor which acts as command processor, format controller and microprocessor interface for the HDCB. Some of the functions it performs are:

- To decode command received from the host over the 8-bit data bus;
- To execute seek and recalibrate commands;
- To interface to the drive and to read the drive status lines;

- Serial-to-parallel and parallel-to-serial data conversion;
- CRC and ECC generation and checking;
- MFM data decoding and encoding;
- Generation of Write Precompensation signals;
- Address mark detection and generation;
- ID field search;
- DMA data transfer control during read/write operations

The  $\mu$ PD9306 performs data separation using a digital phase-locked loop, and contains the necessary logic for write precompensation and clock generation for the  $\mu$ PD7261A.

The ROM BIOS that is provided on the board contains routines which handle interrupts from the  $\mu$ PD7261A, translate the IBM codes for the  $\mu$ PD7261A, set-up the system DMAC as specified by IBM, and sets up the  $\mu$ PD7261A for the following softsector format:

4 heads  
17 sectors/trak  
512 bytes/sector  
615 cylinders

**Specifications:****Physical Characteristics**

Gold-plated edge fingers mount into standard IBM-PC or -PC/XT system board I/O interface slot  
Height — 4.375 in.  
Width — 6.625 in.  
Depth — 0.5 in.

**Electrical Characteristics**

DC Power Requirements — 5V 0.7A typ.

**Reference Literature**

IBM PC/XT Technical Reference  
 $\mu$ PD7261A Data Sheet  
 $\mu$ PD9306 Data Sheet  
 $\mu$ PD7261A User's Manual

**Ordering Information:**

EBIBM-7261A Hard Disk Controller Board for PC-XT

## 1. ATDISK-Controller System Overview

Originally designed as an application tool, this Hard Disk Controller Board provides a simple interface for harddisks and normal as well as high density floppydisks in a IBM PC-AT environment: specifically the IBM AT02, AT03, XT286 and look-alikes.

This single board controller contains a Harddisk-Unit based on the NEC  $\mu$ PD7261 AD Hard Disk Controller and the  $\mu$ PD9306 AC Hard Disk Interface. The Floppydisk-Unit of the controller board is based on the NEC  $\mu$ PD72065C and the  $\mu$ PD71066 CMOS PLL; the Floppydisk-Unit can be switched from normal to high density diskdrives.

The controller board provides connectors for two floppy- and for two harddisk-drives. Furthermore a connector for external floppydrives or streamer is placed at the rear edge of the board. The compact design-layout of the board allows the usage in the IBM XT286 and compatible systems.

The ROM-BIOS that is provided on the board contains all necessary routines to interface properly with the IBM AT-BIOS and DOD Versions 3.XX. There is no need to perform any patches; after preformatting with the delivered program called "HDFORMAT" all the DOS-routines like "FDISK", "FORMAT" etc. can be used. The performance of the NEC  $\mu$ PD7261 Controller-Chip will cause a disk-access with higher speed.

Furthermore it is possible to connect different harddisk-types to the controller. The harddisk-type can be selected via jumper settings. The possible usage of the  $\mu$ PD7261B-23 at higher clockrates will cause an increased performance too.

## 2. System Installation

### 2.1 Board Connectors and Jumper Settings

Use the board connectors as follows:

Connector I1:  
Floppy drives A,B

Connector I2:  
Floppy drives C,D

Connector J1:  
Hard disk "in-use" LED of the AT

Connector J2:  
Hard disk drives

Connector J3:  
Hard disk drive 1

Connector J4:  
Hard disk drive 2

The 37-pin connector at the rear edge of the board may be used for external floppy disk drives (C,D) or a floppy type streamer tape.

Please confirm, that all the on-board jumpers are in the default position (factory set). The meaning of the different jumpers is as follows: (\* - default setting)

JP1: Address select	o	o
	l primary (*)	
	o	o
		l secondary
	o	o
JP2: RAM Type	o o-o Type 256 (*)	
	o-o o Type 64	
JP3: $\mu$ PD9306 Type	o o o (*)	
JP4: clock rate 7261	o-o o 10 MHz (*)	
	o o-o 20 MHz	
JP5: clock rate 9306	o-o o 10 MHz (*)	
	o o-o 20 MHz	

### 2.2 Selecting the Harddisk Type

JX - Hard disk drive select jumpers:

The on board hard disk BIOS will determine the hard disk drive by the setting of these jumpers. Please note the meaning of this jumpers:

- JX1 - Bit 1 (Jumper set --> Bit = 1)
- JX2 - Bit 2
- JX3 - Bit 3
- JX4 - Bit 4
- JX5 - Bit 5
- JX6 - Bit 6

The Jumpers JX4, JX5, JX6 are used for drive 0, the jumpers JX1, JX2, JX3 are used for drive 1.

By setting these jumpers the user can select the following hard disk drive types:

Bits	1	2	3	4	5	6	Type	Tracks	Heads	Capacity
0	0	0	0	0	0	0	no hard	disk drive		
0	0	1	0	0	1	1		306	4	10 MB
0	1	0	0	1	0	2		615	4	20 MB
0	1	1	0	1	1	3		615	6	30 MB
1	0	0	1	0	0	4		665	4	23 MB (NEC)
1	0	1	1	0	1	5		665	6	35 MB (NEC)
1	1	0	1	1	0	6		940	6	50 MB
1	1	1	1	1	1	7		940	8	65 MB

**Notes:**

The user cannot define more than two floppy-disk drives with the IBM-AT Setup program. For the use of more than two disk drives an additional BIOS driver is needed. This BIOS driver is supplied with the external IBM diskette drives and called "EXDSKBIO.DRV". Please follow the installation procedure given in the IBM external diskette user's manual. The external drive connector of the ATDISK-board supports XT-compatible drives (360 KB) or floppy-tapes only.

The original IBM-BIOS (located on the mainboard) dated "01/10/84" will have as a consequence several patches by the PC-DOS while booting. Therefore the ATDISK-Controller board will display the following message:

"17XX System- and Controller BIOS are incompatible"

The user can fix this problem by changing the mainboard-BIOS to a new version.

**3. System Hardware Description**

**3.1 Floppy Disk Interface**

prim. sec.

3F2H	372H	Handling Register	write only
3F4H	374H	FDC Status Register	read/write
3F5H	375H	FDC Data Register	read/write
3F7H	377H	Control Register	read/write

**Signal Description**

Handling Register:

Bit 7	Drive D Motor Enable	active high
Bit 6	Drive C Motor Enable	active high
Bit 5	Drive B Motor Enable	active high
Bit 4	Drive A Motor Enable	active high
Bit 3	Enable Disk Interrupt + DMA	active high
Bit 2	Disk Function Reset	active low
Bit 1	Drive Select	
Bit 0	Drive Select	

Meaning of the Drive Select Bits:

Bit 1	Bit 0	
0	0	Drive A
0	1	Drive B
1	0	Drive C
1	1	Drive D

Note: A channel reset clears all bits.

**Control Register (Output):**

Bit 7	unused
Bit 6	unused
Bit 5	unused
Bit 4	unused
Bit 3	unused
Bit 2	unused
Bit 1	Data Transfer Rate
Bit 0	Data Transfer Rate

Meaning of the Data Transfer Rate Bits:

Bit 1	Bit 0	
0	0	500 Kbps
0	1	300 Kbps
1	0	250 Kbps
1	1	reserved

Note: A channel reset clears all bits.

**Control Register (Input):**

Bit 7	Diskette Change	active high
Bit 6	reserved	
Bit 5	reserved	
Bit 4	reserved	
Bit 3	reserved	
Bit 2	reserved	
Bit 1	reserved	
Bit 0	reserved	

**3.2 Harddisk Interface**

prim. sec.

1F0H	170H	Data FIFO	read/write
		16 Bit. The lower	
		Byte first.	
1F2H	172H	Status Register	read/write
1F4H	174H	same as 176H	
1F5H	175H	same as 177H	
1F6H	176H	HCD Data FIFO	read/write
1F7H	177H	HDC Command Register	read/write

## Signal Description

Status Register (Output):

Bit 7 DMA Direction 0 = From HDC to FIFO  
1 = From FIFO to HDC

Bit 6 unused

Bit 5 unused

Bit 4 unused

Bit 3 Enable HD Interrupts active high

Bit 2 DMA Function Reset active low

Bit 1 HDC Function Reset active high

Bit 0 DMA Address Counter Clear active high

Note: A channel reset clears all bits.

Status Register (Input):

Bit 7 Type Drive 1

Bit 6 Type Drive 1

Bit 5 Type Drive 1

Bit 4 Type Drive 0

Bit 3 Type Drive 0

Bit 2 Type Drive 0

Bit 1 INT; latched HDC Interrupt active high

This bit will be cleared

by the next HDC-CS.

Bit 0 DREQ; HDC DMA-Request active high

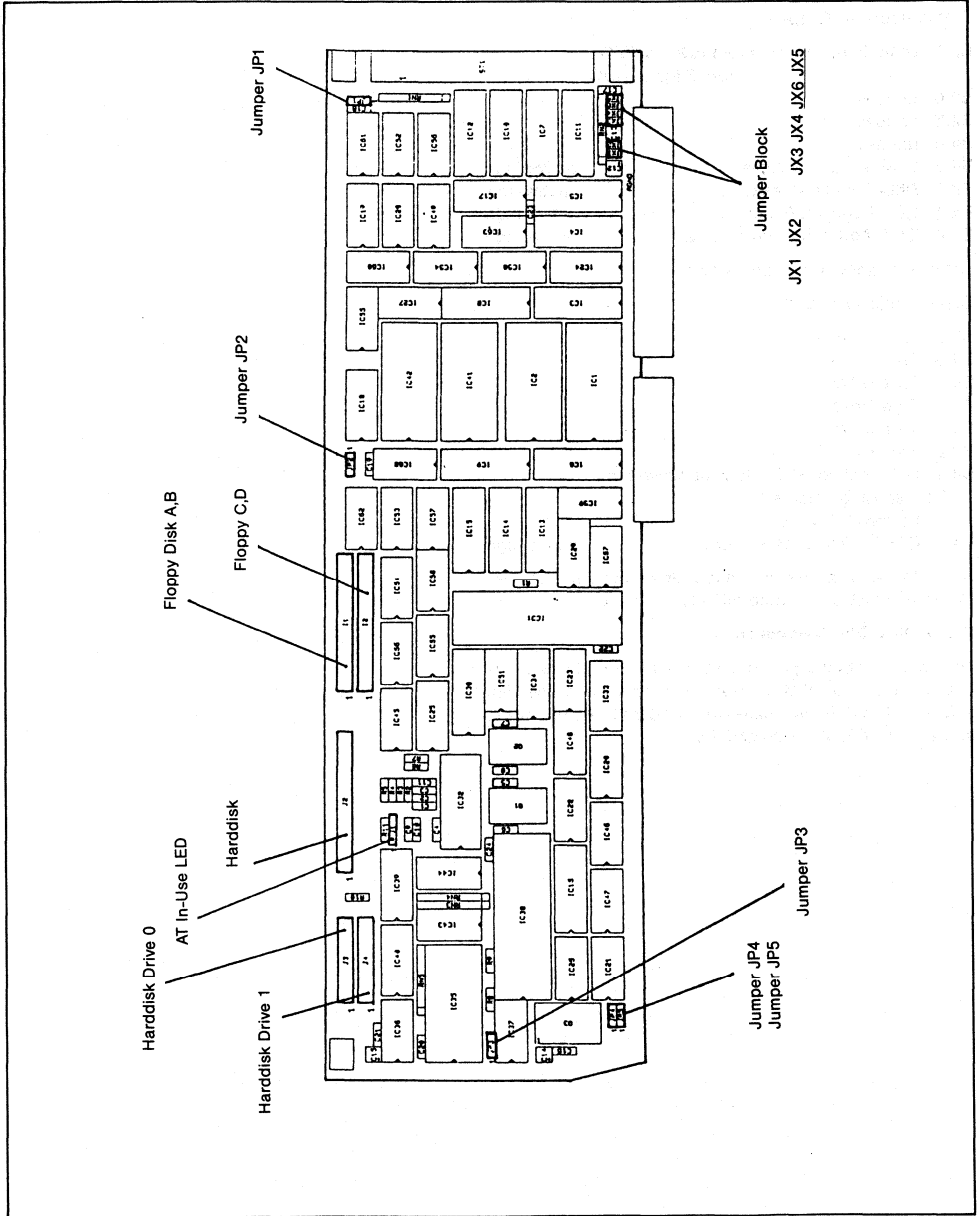
Note: These status bits are valid even though a DMA transfer between HDC and data-FIFO takes place.

### 3.3 BIOS ROM Expansion

The controller board provides special BIOS routines in on-board expansion ROM to interface properly with the IBM software. The BIOS is located in 8 x 16 Bit EPROMS at the addresses C8000H up to CBFFFH.



ATDISK Board Layout



### 1. ESDI-Controller System Overview

Originally designed as an application tool, this Hard Disk Controller Board provides a simple interface for harddisks and normal, as well as high density floppydisks in a IBM PC-AT environment: soecifically the IBM AT02, AT03, XT286 and look-alikes.

This single board controller consists of a Harddisk-Unit for two harddisks with the Enhanced Small Device Interface (ESDI) and a Floppydisk-Unit for four double/high density floppydisks. Additionally the use of an external floppystreamer-tape is possible. Therefore a 37-pin connector is provided at the rear edge of the board. Furthermore a special interface-connector may be used to join four additional harddisks to the controller board (this option is for development purposes only; additional hardware and software has to be developed by the user).

The harddisk-unit of the system is based on the NEC  $\mu$ PD7262 ESDIC Hard Disk Controller and consists of an interface-circuit for two harddisks, a 64 KByte RAM, and a 16 KByte Firmware-EPROM to expand the System-Bios. All harddisk-drives, complying to the ESDI-Specification Rev.F may be used with the EBIBM-7262 ESDI controller board. To meet this specification a Data Transfer Rate of 10-15MBit per Second is needed. The Data Transfer Rate used by the system depends on the harddisk-drive connected to the controller board. However, the EBIBM-7262ESDI board is able to perform transfer rates up to 18 MBit per Second.

The floppydisk-interface of the system is fully IBM-BIOS compatible and supports all formats of IBM-PC/AT floppydisks. The user can connect up to four floppy drives (40 track double-density, 80 track high-density, mixed mode). This unit of the controller board is based on the NEC  $\mu$ PD72067 AFDC (Advanced Floppy Disk Controller); this new controller-chip incorporates a window signal generation VFO circuit, a write precompensation circuit, a write clock generation circuit, a system clock generation circuit. All these circuits were peripheral circuits to conventional FDC's (Floppy Disk Controllers). Thus, the number of parts could be reduced substantially. The  $\mu$ PD72067 supports all commands of conventional FDC's such as  $\mu$ PD765A/7265 and  $\mu$ PD72065/72066 and handles both IBM and ECMA/ISO formats.

### 2. System Installation Procedure

For use of the ESDI controller board an IBM AT or compatible system is required

At first it may be desirable to make a backup of the system's hardisk to prevent loss of the software on it.

#### Step 1:

Run the AT setup-program and tell the system that there is no harddisk installed.

#### Step 2:

Now turn off your system and remove plug from mains. The unit can be opened now (refer to IBM User's Guide).

Remove the actual harddisk controller board.

Install the HDAT controller board and make sure that all cables are connected correctly (an ESDI-type harddisk may be connected only!). The unit may be closed again now.

#### Step 3:

Insert a DOS system diskette in drive A and power-up the AT. The system will now boot from drive A.

After successful installation and boot procedure you should insert now the disk delivered with the board in drive A.

After calling the HDFORMAT performing program you can determine interleaving-factor. Now the harddisk will be preformatted.

A>hdformat

Fixed disk format program for NEC 7262 ESDI disk controller

Enter drive (0-1): 0

Enter interleave factor (1-16): 1

Set bad cylinder flags (y/n): n  
(if there are no bad cyl.)

#### Step 4:

Remove the disk from drive A and insert the DOS-diskette again. Now run the FDISK program to create a DOS-Partition.

#### Step 5:

Then type

A>format c:/s

WARNING, ALL DATA ON NON-REMOVABLE DISK DRIVE C: WILL BE LOST!

Proceed with Format (Y/N)?y

Formatting ... Format complete

#### Step 6:

Copy all DOS utilities to the Harddisk by typing

A>copy \*.\* C: (copies all DOS utilities)

Now try rebooting from the harddisk (remove the DOS diskette from drive A).

The system is now ready for using.

**3. System Hardware Description**

**3.1 Floppy Disk Interface**

prim. sec.

3F2H 372H	Handling Register AFDC	write only
3F4H 374H	AFDC Status Register	read/write
3F5H 375H	AFDC Data Register	read/write
3F7H 377H	Control Register	read/write

**Signal Description**

Handling Register:

Bit 7	Drive D Motor Enable	active high
Bit 6	Drive C Motor Enable	active high
Bit 5	Drive B Motor Enable	active high
Bit 4	Drive A Motor Enable	active high
Bit 3	Enable Disk Interrupt + DMA	active high
Bit 2	Disk Function Reset	active low
Bit 1	Drive Select	
Bit 0	Drive Select	

Meaning of the Drive Select Bits:

Bit 1	Bit 0	
0	0	Drive A
0	1	Drive B
1	0	Drive C
1	1	Drive D

Note: A channel reset clears all bits.

Control Register (Output):

Bit 7	unused
Bit 6	unused
Bit 5	unused
Bit 4	unused
Bit 3	unused
Bit 2	unused
Bit 1	Data Transfer Rate
Bit 0	Data Transfer Rate

Meaning of the Date Transfer Rate Bits:

Bit 1	Bit 0	
0	0	500 Kbps
0	1	300 kbps
1	0	250 Kbps
1	1	reserved

Note: A channel reset clears all bits.

Control Register (Input):

Bit 7	Diskette Change	active high
Bit 6	reserved	
Bit 5	reserved	
Bit 4	reserved	
Bit 3	reserved	
Bit 2	reserved	
Bit 1	reserved	
Bit 0	reserved	

**3.2 Harddisk Interface**

prim. sec.

1F0H 170H	Data FIFO	read/write
	16 Bit.	
	The lower Byte first.	
1F2H 172H	Status Register	read
1F2H 172H	Handling Register ESDIC	write
1F4H 174H	same as 176H	
1F5H 175H	same as 177H	
1F6H 176H	ESDIC Data FIFO	read/write
1F7H 177H	ESDIC Command Register	read/write

**Signal Description**

Handling Register ESDIC (Output):

Bit 7	DMA Direktion 0 = From ESDIC to FIFO	
	1 = From FIFO to ESDIC	
Bit 6	unused	
Bit 5	unused	
Bit 4	unused	
Bit 3	Enable ESDIC Interrupts	active high
Bit 2	DMA Function Reset	active low
Bit 1	unused	active high
Bit 0	DMA Address Counter Clear	active high

Note: A channel reset clears all bits.

Status Register (Input):

Bit 7	INT; latched ESDIC Interrupt	active high
	This bit will be cleared by the next ESDIC-READ.	
Bit 6	Type Drive 1	
Bit 5	Type Drive 1	
Bit 4	Type Drive 1	
Bit 3	unused	
Bit 2	Type Drive 0	
Bit 1	Type Drive 0	
Bit 0	Type Drive 0	

Note: These status bits are valid even though a DMA transfer between ESDIC and data-FIFO takes place.

**3.3 BIOS ROM Expansion**

The controller board provides special BIOS routines in on-board expansion ROM to interface properly with the IBM software. The BIOS is located in 8K x 16 Bit EPROMS at the addresses C8000H up to CBFFFH.

### Image Pipelined Processor

#### Description

To support the Image Pipelined Processor  $\mu$ PD7281 IMPP NEC has developed a complete support tool package consisting of both Software and Hardware.

#### Software

A software package containing a functional assembler, a simulator and an object converter is provided for NEC's development system as well as for many other personal computers and minis.

#### Features

- Complete software development system for  $\mu$ PD7281
- Assembler for generating  $\mu$ PD7281 tokens
- Software simulator provides
  - Program debugging capabilities
  - System simulation and evaluation
- Runs on a variety of operating systems
  - CP/M-86\*)
  - MS-DOS\*)
  - VAX/VMS\*) and VAX/UNIX\*)

#### FA7281 Assembler

FA7281 translates symbolic source programs for the  $\mu$ PD7281 into object modules which serve as input to either the software simulator or the object code conversion programs.

Features are as follows:

- Automatic generation of all required tokens
- Extensive error reporting
- Command line controls
- User-selectable and directable output files

#### OH7281 Object Code Converter

OH7281 converts object modules produced by the assembler into hexadecimal format object module files for input to a HEX-loader or into ASCII data format object module files for use as data within a source module. A symbol table file may be produced for use with the software simulator.

#### SM7281 Software Simulator

SM7281 accepts object code modules produced by the assembler and simulates the user program under specified system parameters. The simulator can fully simulate and entire processing subsystem, providing the user the tools to fully debug his program and to evaluate system performance without having to actually build the hardware. Features are as follows:

- Support simulation of three system models
  - One or more cascaded  $\mu$ PD7281s,  $\mu$ PD9305 and image memory
  - One or more cascaded  $\mu$ PD7281s and image memory
  - One or more Cascaded  $\mu$ PD7281s only
- Continuous/single-step execution
- Set/display input data timing
- Display/modify contents of memory, latch or registers
- Sophisticated breakpoint capabilities
- Sophisticated trace capabilities
  - Define/display items to be traced
  - Define/display trace start/stop conditions
- Supports full symbolic debug
  - Define/delete/modify symbols
  - Display symbols
- On-line assembler and disassembler
- Macro command file capability
- Save/load simulator setup to/from disk
- Save console input commands and execution results on disk
- Display LT, PU, IM operating ratios for program evaluation

#### Ordering Information for the complete software package

FAMSD-05DD-7281	MS-DOS, <sup>TM</sup> 5 1/4" double-density floppy diskette
FACPM-05DD-7281	CP/M-86, <sup>TM</sup> 5 1/4" double-density floppy diskette
FAMPM-08SS-7281	CP/M-86, <sup>TM</sup> 8" single-density floppy diskette
FAVMS-OT16-7281	VAX/VMS, <sup>TM</sup> 9-track 1600 BPI magnetic tape
FAUNI-OT16-7281	VAX/UNIX, <sup>TM</sup> 9-track 1600 BPI magnetic tape

CP/M-86 – is a registered trademark of Digital Research Corporation  
MS-DOS – is a registered trademark of Microsoft Corporation  
VAX and VMS – are registered trademarks of Digital Equipment Corporation  
UNIX – is a trademark of AT&T



### Hardware

The method of program debugging for the  $\mu$ PD7281 largely differs from debugging method for typical microcomputers. This is because the  $\mu$ PD7281 employs a data flow-type microprocessor.

In debugging typical microcomputers, the focus is put mainly on data transfers between the microcomputer and its I/O devices, especially the memory. That is, when a desired data is given to the microcomputer, the operation of the microcomputer is temporarily interrupted to check its status. Modification is then made if necessary. Then, execution is again assumed. This method is very effective for typical microcomputers, because the number of instructions being executed when operation is interrupted is only one. Therefore, even if operation is interrupted, the execution can be easily resumed from the same status after checking the internal status.

This means that the basic unit of the debugging cycle is instruction execution.

However, the  $\mu$ PD7281 employs the data flow method. Therefore the concept of the instruction unit (fetch--decode--execution) cannot be used for the  $\mu$ PD7281. The number of instructions being executed may not be only one at a point. Usually, two or more instructions are being executed.

Additionally, it is known whether or not the instruction sent to the pipeline first is being executed. Therefore, if execution is interrupted once and the internal status is sent outside, the internal status changes completely and the original status cannot be resumed. Therefore, because the  $\mu$ PD7281 employs data flow method, its debugging method differs significantly from other development tools commonly used (such as EVAKT or IE). In debugging with the SX-7281-S, the main focus is put on efficient control of the tokens of the  $\mu$ PD7281.

### SX-7281

SX-7281-S hardware units allow real time testing of user's programs for binary image processing. SX-7281-M is a two multibus board system containing on one board 4 times  $\mu$ PD7281, the support chip  $\mu$ PD9305 MAGIC and a Graphic display controller  $\mu$ PD7220. On the second board the image memory of 256K x 18 bits for 16 bit of data plus sign and control bits. For image information the memory can be configured up to 1024 x 1024 dots in 4 planes. Access is provided by the IEEE796 Bus (Multibus™).

By adding a third Multibus board containing a 8086 Processor and a serial interface RS232 into a 3 slot Multibus frame the system called SX-7281-S becomes a stand-alone unit for connection to any host computer. Commands and functions are the same for both Multibus and stand-alone system.

### Ordering Information

SX-7281-M	Hardware system for binary image processing, Multibus unit
SX-7281-S	Hardware system for binary image processing, stand alone unit



### Description

A new dataflow processor  $\mu$ PD7281 is the base of an image processing board for standard PCs. All parts needed are implemented. An input multiplexer selects one of 16 CCIR sources (camera or VTR) with either internal or external synchronization. This signal is converted to digital with an 8 bit flash-converter and can be manipulated by an input look-up-table. Four image memory planes are available with each  $512 \times 512 \times 8$  bit. Two of them can be connected to a  $512 \times 512 \times 16$  bit plane advantageous for special applications. Three analog composite video outputs with output look-up-tables allow pseudo color representation of the selected memory plane.

Four data-flow processors ( $\mu$ PD7281) and a chip for memory and host interface ( $\mu$ PD9305) provide up to 20 MIPS. An internal bus transfers the data between memory and dataflow processors to avoid any performance degradation.

Pipelining and parallel processing together with the efficient instructions of the dataflow processors result in high speed image processing. An on-board 16 bit  $\times$  64 k RAM or ROM contains programs for the dataflow processors. By simple commands part of the information can be loaded into the four processors. Initialisation and downloading is done by the PC or other logic in stand-alone applications typically for industrial use.

A software package supports the board. Via menus the system parameters (input channel select, memory plane configuration etc.) and program modules for image processing (different types of filter, edge detection . . .) can be selected from the host PC. Another software package

contains an assembler and simulator to generate user software for the dataflow processors. So the board is also a good tool for familiarisation and design of dataflow based hardware and software development.

The board was developed in close co-operation with Electronique Lyonnaise who proposed the nick name "PC Oeil" what means "PC Eye".

### Software Organization

PC Oeil is destined to run the image library developed for the PC Oeil board in an IBM-PC environment.

This library proposes around 60 functions that enable one to deal with image acquisition, processing, analysis or manipulation.

However this software remains open and the number of the functions is not limited.

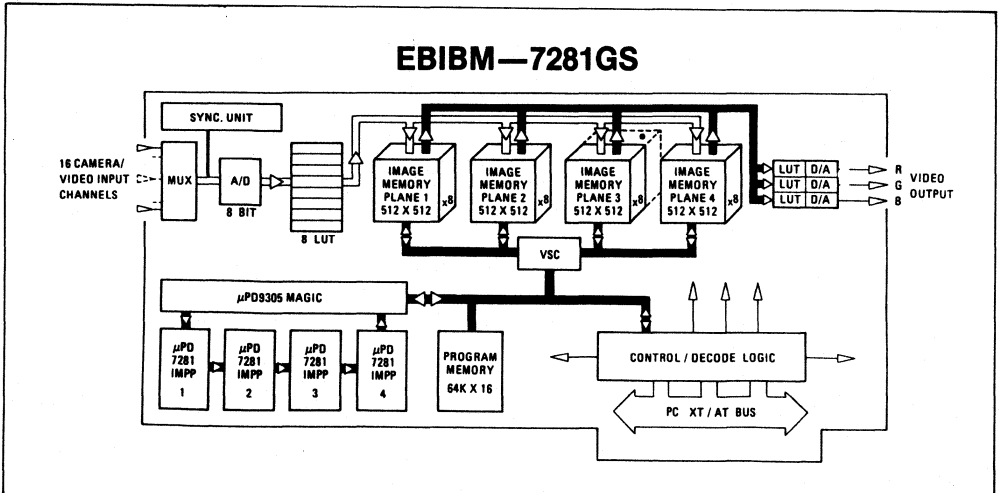
To suppress for the operator the difficulties due to data-mation, the image library primitives are open in evolving menus. This software is conversational. For advancing in the menus, the operator can use either a mouse or certain keys on the keyboard. The primitives are organized around three important classes which appear in the main menu like this:

- Image Management Utilities
- Operations on grey level images
- Operations on binary images.

### Ordering Information

EBIBM-7281GS PC board for grey scale image processing

### EBIBM-7281GS Block Diagram







### LCD Controller Support

#### Description

The  $\mu$ PD7229G LCD controller is supported by the CDMSD-I5DD-7229 software development package. This product assists in generating the characters and symbols for customized character sets in an interactive PC session. Starting from the standard character set of the NEC  $\mu$ PD7228G, the user can control cursor movement, set or reset dots to be displayed, and the program will generate the code necessary for mask pattern design.



## **SENTRY TEST TAPE OVERVIEW**

**9**

**Section 9 – Sentry Test Tape Overview**

Sentry Test Tape ..... 9.3

### Sentry Test Tape

NEC offers a number of sentry test tapes for customer entrance testing. These test tapes run on sentry VII and check all AC and DC parameters. Test tapes are available for the following devices:

μPD70008  
μPD70108  
μPD70116  
μPD70208  
μPD70216  
μPD71051  
μPD71054  
μPD71055  
μPD71059  
μPD71071  
μPD780  
μPD7201A  
μPD72001  
μPD7220A  
μPD7260  
μPD7261A  
μPD765AC-2  
μPD7265  
μPD72065  
μPD7281



## **PACKAGING INFORMATION**

**10**



## Section 10 – Packaging Information

Package/Device Cross Reference .....	10.3
16-Pin Plastic DIP (300 mil) .....	10.5
18-Pin Plastic DIP (300 mil) .....	10.5
18-Pin Cerdip (300 mil) .....	10.6
20-Pin Plastic DIP (300 mil) .....	10.6
20-Pin Cerdip (300 mil) .....	10.7
20-Pin Plastic SO (Small Outline) .....	10.7
24-Pin Plastic DIP (600 mil) .....	10.8
24-Pin Plastic Skinny DIP (400 mil) .....	10.8
28-Pin Plastic DIP (600 mil) .....	10.9
28-Pin Plastic SO (Small Outline) .....	10.9
28-Pin PLCC (Plastic Leaded Chip Carrier) .....	10.10
30-Pin Plastic Shrink DIP (400 mil) .....	10.10
40-Pin Plastic DIP (600 mil) .....	10.11
40-Pin Ceramic DIP (600 mil) .....	10.11
40-Pin Cerdip (600 mil) .....	10.12
44-Pin Plastic Flat Package .....	10.12
44-Pin PLCC (Plastic Leaded Chip Carrier) .....	10.13
48-Pin Plastic DIP (600 mil) .....	10.13
48-Pin Ceramic DIP (600 mil) .....	10.14
52-Pin Plastic Flat Package .....	10.14
52-Pin PLCC (Plastic Leaded Chip Carrier) .....	10.15
64-Pin Plastic Flat Package .....	10.16
68-Pin PLCC (Plastic Leaded Chip Carrier) .....	10.17
69-Pin Ceramic PGA (Pin Grid Array) .....	10.17
80-Pin Plastic Flat Package .....	10.18
84-Pin PLCC (Plastic Leaded Chip Carrier) .....	10.19
94-Pin Plastic Flat Package .....	10.20
132-Pin Ceramic PGA (Pin Grid Array) .....	10.21

### Package / Device Cross Reference

Package	Device	Package	Device
16-Pin Plastic DIP (300 mil)	$\mu$ PB8216C $\mu$ PB8226C	28-Pin Plastic Leaded Chip Carrier (PLCC)	$\mu$ PD71051L $\mu$ PD71054L $\mu$ PD71059L
18-Pin Plastic DIP (300 mil)	$\mu$ PD71011C $\mu$ PD71084C	30-Pin Plastic Shrink DIP (400 mil)	$\mu$ PD71066CT
18-Pin Cerdip (300 mil)	$\mu$ PB8284AD	40-Pin Plastic DIP (600 mil)	$\mu$ PD765AC $\mu$ PD765AC-2 $\mu$ PD780C $\mu$ PD780C-1 $\mu$ PD780C-2 $\mu$ PD7201AC $\mu$ PD7210C $\mu$ PD7265C $\mu$ PD7265AC-2 $\mu$ PD8085AC-2 $\mu$ PD8085AHC $\mu$ PD8085AHC-2
20-Pin Plastic DIP (300 mil)	$\mu$ PB8282C $\mu$ PB8283C $\mu$ PB8286C $\mu$ PB8287C $\mu$ PD71082C $\mu$ PD71083C $\mu$ PD71086C $\mu$ PD71087C $\mu$ PD71088C $\mu$ PD71611C $\mu$ PD71613C		$\mu$ PD8155C $\mu$ PD8155C-2
20-Pin Cerdip (300 mil)	$\mu$ PB8288D $\mu$ PB8289D		$\mu$ PD8155HC $\mu$ PD8155HC-2 $\mu$ PD8156C $\mu$ PD8156C-2 $\mu$ PD8156HC $\mu$ PD8156HC-2 $\mu$ PD8237AC-5 $\mu$ PD8255AC-2 $\mu$ PD8255AC-5 $\mu$ PD8257C-2 $\mu$ PD8257C-5 $\mu$ PD8279C-2 $\mu$ PD8279C-5 $\mu$ PB9201C $\mu$ PD70008C
20-Pin Plastic SO (Small Outline) (300 mil)	$\mu$ PD71011G $\mu$ PD71082G $\mu$ PD71083G $\mu$ PD71084G $\mu$ PD71086G $\mu$ PD71087G $\mu$ PD71088G		$\mu$ PD70008AC-4 $\mu$ PD70008AC-6 $\mu$ PD70108C-5 $\mu$ PD70108C-8 $\mu$ PD70108C-10 $\mu$ PD70116C-5 $\mu$ PD70116C-8 $\mu$ PD70116C-10 $\mu$ PD71055C $\mu$ PD72001C $\mu$ PD72065C $\mu$ PD72066C
24-Pin Plastic DIP (600 mil)	$\mu$ PD71054C $\mu$ PD8243C $\mu$ PD8243HC $\mu$ PD82C43C $\mu$ PD8253C-2 $\mu$ PD8253C-5		
24-Pin Plastic Skinny DIP (400 mil)	$\mu$ PD82C43CX		
28-Pin Plastic DIP (600 mil)	$\mu$ PD8251AC $\mu$ PD8251AFC $\mu$ PD8259AC $\mu$ PD8259AC-2 $\mu$ PD9306AC $\mu$ PD71051C $\mu$ PD71059C		
28-Pin Plastic SO (Small Outline) (375 mil)	$\mu$ PD71065G		

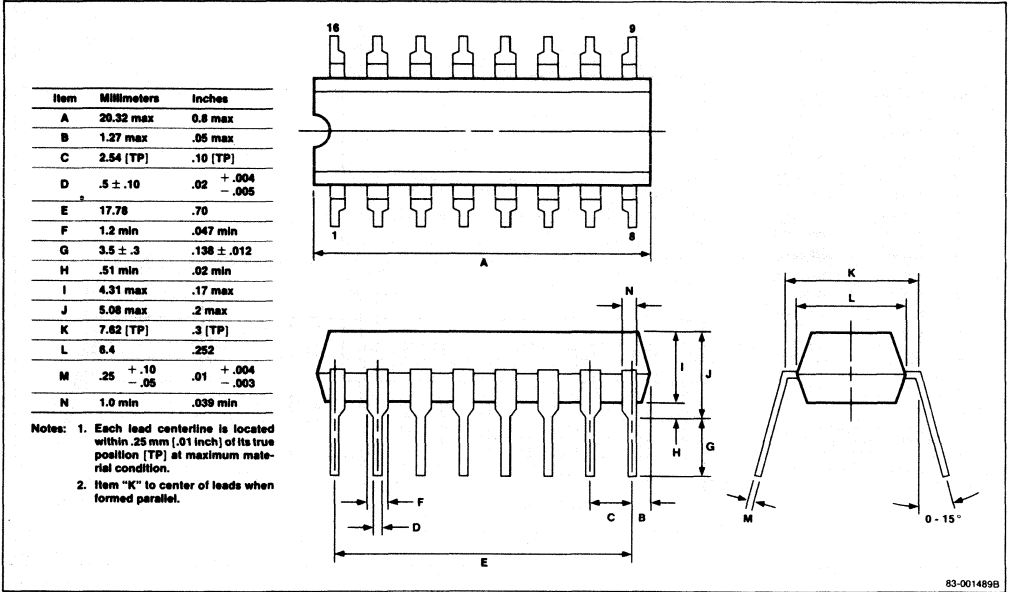
## PACKAGING INFORMATION

### Package/Device Cross Reference

Package	Device	
40-Pin Ceramic Dip (600 mil)	μPD7201AD	
	μPD7220AD	
	μPD7220AD-1	
	μPD7220AD-2	
	μPD7260D	
	μPD7261AD	
	μPD7261BD-18	
	μPD7261BD-23	
	μPD7262D-12	
	μPD7262D-18	
	μPD7281D	
	μPD8086D	
	μPD8088D	
	μPD8088D-2	
	40-Pin Cerdip (600 mil)	μPD8086D
		μPD8086D-2
μPD70108D-5		
μPD70108D-8		
μPD70108D-10		
μPD70116D-5		
μPD70116D-8		
44-Pin Plastic Flat Pack	μPD70008AG-4	
	μPD70008AG-6	
	μPD71051GB	
	μPD71054GB	
	μPD71055GB	
	μPD71059GB	
44-Pin Plastic Leaded Chip Carrier (PLCC)	μPD70008AL-6	
	μPD70108L-5	
	μPD70108L-8	
	μPD70116L-5	
	μPD70116L-8	
	μPD71055L	
	μPD72065L	
	μPD72066L	
48-Pin Plastic Dip (600 mil)	μPD71071C	
	μPD72105C	
	μPD72067C	
48-Pin Ceramic Dip (600 mil)	μPD71071D	
52-Pin Plastic Flat Pack	μPD70108GC-5	
	μPD70108GC-8	
	μPD70116GC-5	
	μPD70116GC-8	
	μPD71071GC	
	μPD72065G	
	μPD72066G	
	μPD72067G	
	μPD7225G-00	
	μPD7225G-01	

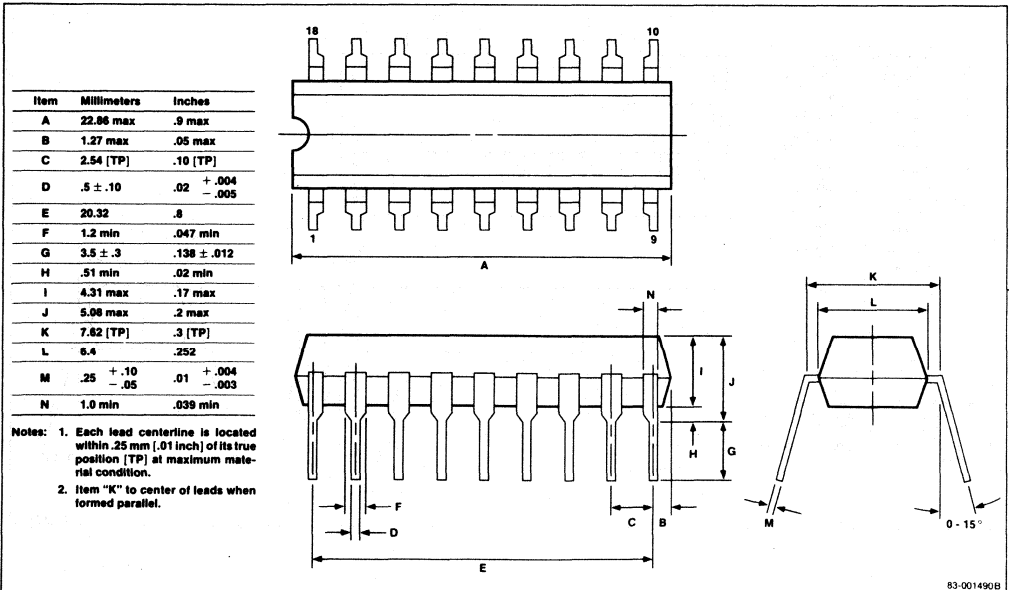
Package	Device
52-Pin Plastic Leaded Chip Carrier (PLCC)	μPD71071L
	μPD72001L
	μPD72105L
	μPD72067L
64-Pin Plastic Flat Pack	μPD7227G-11 μPD7227G-12
68-Pin Plastic Leaded Chip Carrier (PLCC)	μPD72030G
	μPD70208L-8
	μPD70208L-10
	μPD70216L-8
	μPD70216L-10 μPD72022L
68-Pin Ceramic PGA	μPD70208R-8
	μPD70208R-10
	μPD70216R-8
	μPD70216R-10
	μPD70616R-16
80-Pin Plastic Flat Pack	μPD70208GF-8
	μPD70208GF-10
	μPD70216GF-8
	μPD70216GF-10
	μPD72068GF
	μPD7228AG
	μPD7228G μPD7229G
84-Pin Plastic Leaded Chip Carrier (PLCC)	μPD70320L
	μPD70322L
	μPD70320L-8
	μPD70322L-8
	μPD70330L
	μPD70332L
	μPD70330L-8
	μPD70332L-8
	μPD72120L
	μPD72068L
	μPD79011L
94-Pin Plastic Package	μPD70320GJ
	μPD70322GJ
	μPD70320GJ-8
	μPD70322GJ-8
	μPD70330GJ
	μPD70332GJ
	μPD70330GJ-8
	μPD70332GJ-8 μPD72120GJ μPD79011GJ
132-Pin Ceramic PGA	μPD9305R μPD72120R

### 16-Pin Plastic DIP (300 mil)



83-001489B

### 18-Pin Plastic DIP (300 mil)



83-001490B

# PACKAGING INFORMATION

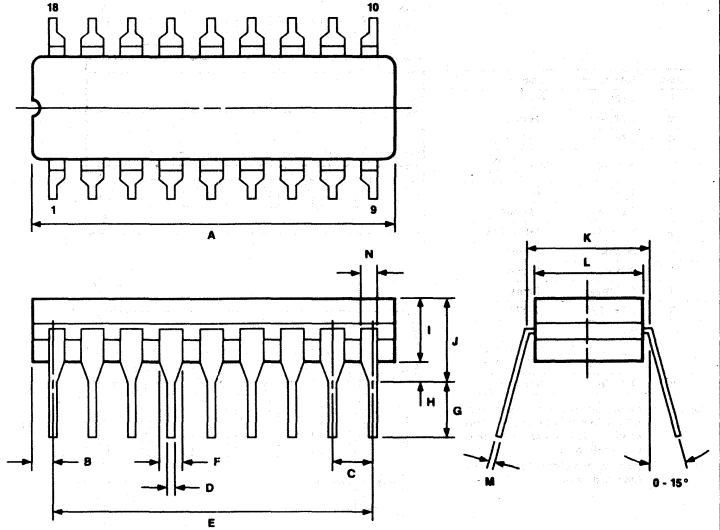


## 18-Pin Cerdip (300 mil)

Item	Millimeters	Inches
A	22.86 max	.900 max
B	1.27 max	.050 max
C	2.54 [TP]	.100 [TP]
D	.46 ± .05	.018 ± .002
E	20.32	.800
F	1.42 min	.055 min
G	3.5 ± .3	.138 ± .012
H	.51 min	.020 min
I	3.95	.156
J	5.08 max	.200 max
K	7.62 [TP]	.300 [TP]
L	6.60	.260
M	.25 ± .05	.010 <sup>+0.002</sup> -.003
N	.89 min	.035 min

**Notes:**

- [1] Each lead centerline is located within .25 mm [.010 inch] of its true position [TP] at maximum material condition.
- [2] Item "K" to center of leads when formed parallel.



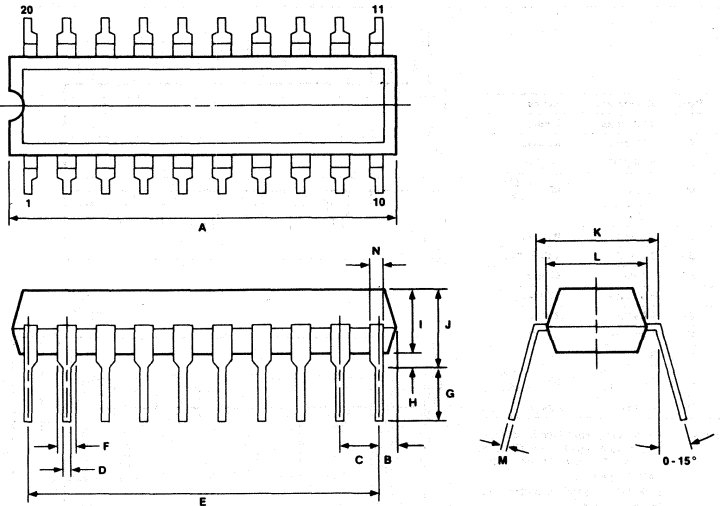
83-003629B

## 20-Pin Plastic DIP (300 mil)

Item	Millimeters	Inches
A	25.40 max	1.000 max
B	1.27 max	.050 max
C	2.54 [TP]	.100 [TP]
D	.50 ± .10	.020 <sup>+0.004</sup> -.005
E	22.86	.900
F	1.1 min	.043 min
G	3.5 ± .30	.138 ± .012
H	.51 min	.020 min
I	4.31 max	.170 max
J	5.08 max	.200 max
K	7.62 [TP]	.300 [TP]
L	6.4	.252
M	.25 <sup>+0.10</sup> -.05	.010 <sup>+0.004</sup> -.003
N	.9 min	.035 min

**Note:**

- [1] Each lead centerline is located within .25 mm [.010 inch] of its true position [TP] at maximum material condition.
- [2] Item "K" to center of leads when formed parallel.



83-001491B

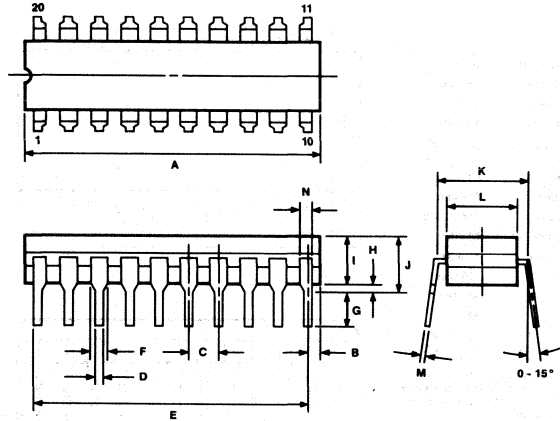
### 20-Pin Cerdip (300 mil)

Item	Millimeters	Inches
A	25.40 max	1.000 max
B	1.27 max	.050 max
C	2.54 [TP]	.100 [TP]
D	.46 ±.06	.018 ±.002
E	22.86	.900
F	1.42 min	.055 min
G	3.5 ±.3	.138 ±.012
H	.51 min	.020 min
I	3.95	.156
J	5.08 max	.200 max
K	7.62 [TP]	.300 [TP]
L	7.32	.288
M	.25 ±.05	.010 <sup>+ .002</sup> -.003
N	.89 min	.035 min

**Note:**

[1] Each lead centerline is located within .25 mm (.01 inch) of its true position [TP] at maximum material condition.

[2] Item "K" to center of leads when formed parallel.



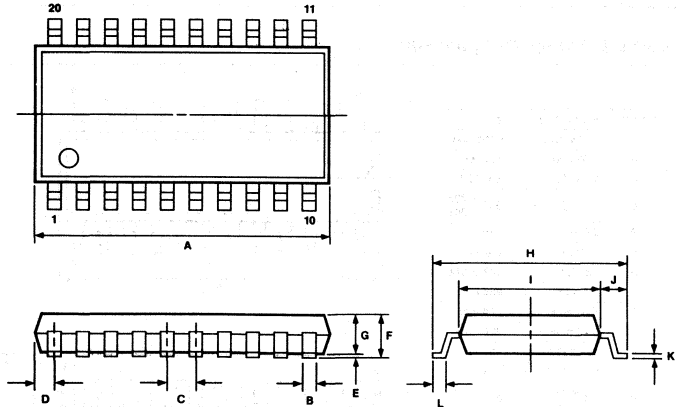
83-003849B

### 20-Pin Plastic SO (Small Outline) (300 mil)

Item	Millimeters	Inches
A	13.00 max	.512 max
B	.76 max	.031 max
C	1.27 [TP]	.050 [TP]
D	.40 <sup>+ .10</sup> -.05	.016 <sup>+ .004</sup> -.003
E	.1 ±.1	.004 ±.004
F	1.8 max	.071 max
G	1.50	.059
H	8.2 ±.3	.323 ±.012
I	6.0	.236
J	1.1	.043
K	.15 <sup>+ .10</sup> -.05	.006 <sup>+ .004</sup> -.002
L	.6 ±.2	.024 <sup>+ .008</sup> -.009
M	.12	.005

**Note:**

[1] Each lead centerline is located within .12 mm (.005 inch) of its true position [TP] at maximum material condition.



83-003850B

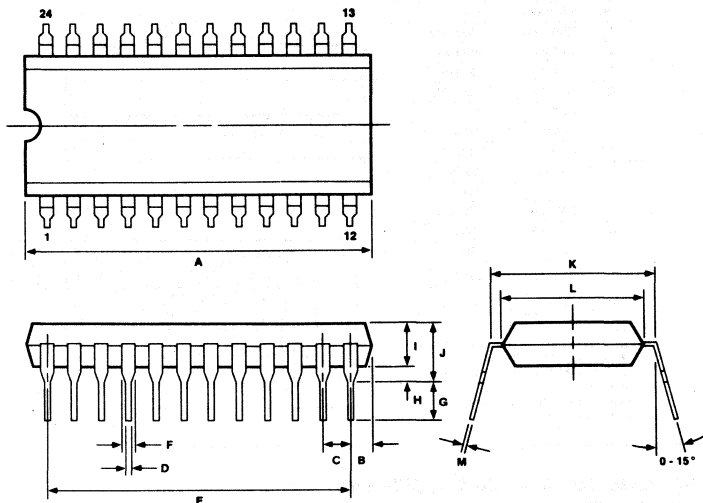
# PACKAGING INFORMATION

# NEC

## 24-Pin Plastic DIP (600 ml)

Item	Millimeters	Inches
A	33.02 max	1.3 max
B	2.54 max	.10 max
C	2.54 [TP]	.10 [TP]
D	.5 ± .10	.02 +.004 -.005
E	27.94	1.1
F	1.2 min	.047 min
G	3.6 ± .3	.138 ± .012
H	.51 min	.02 min
I	4.31 max	.17 max
J	5.72 max	.226 max
K	15.24 [TP]	.60 [TP]
L	13.2	.52
M	.25 +.10 -.05	.01 +.004 -.003

- Notes: 1. Each lead centerline is located within .25 mm [.01 inch] of its true position [TP] at maximum material condition.  
2. Item "K" to center of leads when formed parallel.

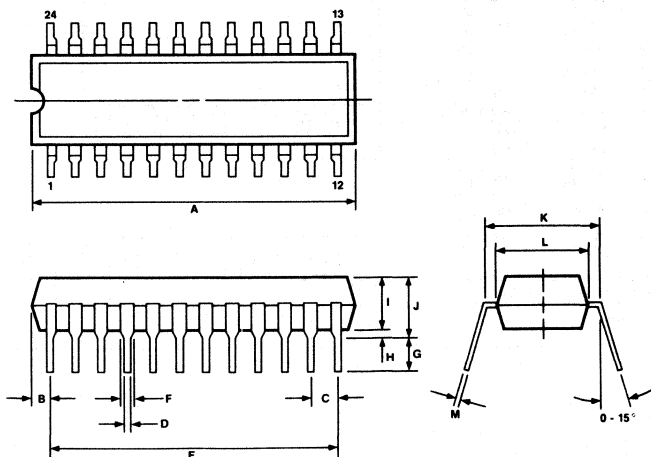


83-001492B

## 24-Pin Plastic Skinny DIP (400 ml)

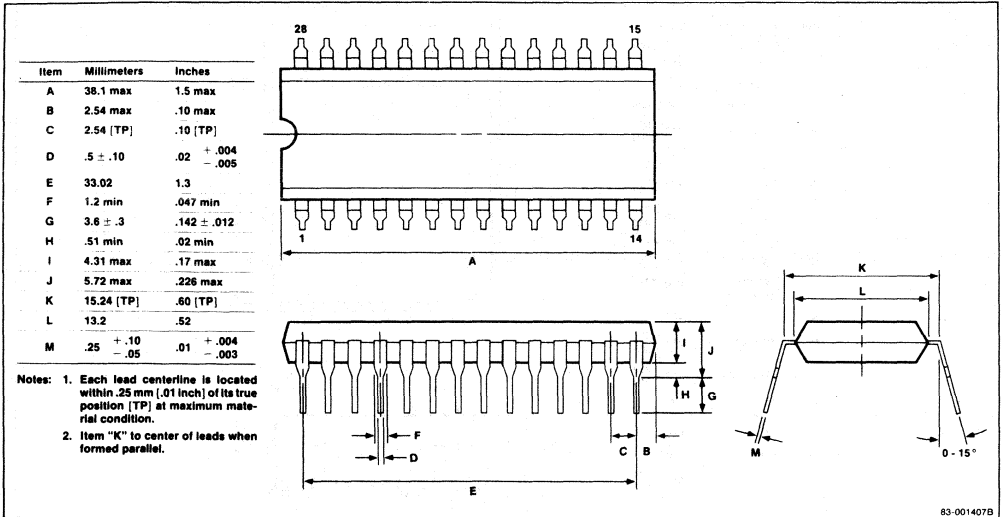
Item	Millimeters	Inches
A	35.56 max	1.400 max
B	3.81 max	.150 max
C	2.54 [TP]	.100 [TP]
D	.50 ± .1	.020 +.004 -.005
E	27.94	1.100
F	1.1 min	.043 min
G	3.6 ± .3	.142 ± .012
H	.51 min	.020 min
I	4.31 max	.170 max
J	5.72 max	.226 max
K	10.16 [TP]	.400 [TP]
L	8.6	.339
M	.25 +.10 -.05	.010 +.004 -.003

- Note:  
[1] Each lead centerline is located within .12 mm [.005 inch] of its true position [TP] at maximum material condition.  
[2] Item "K" to center of leads when formed parallel.

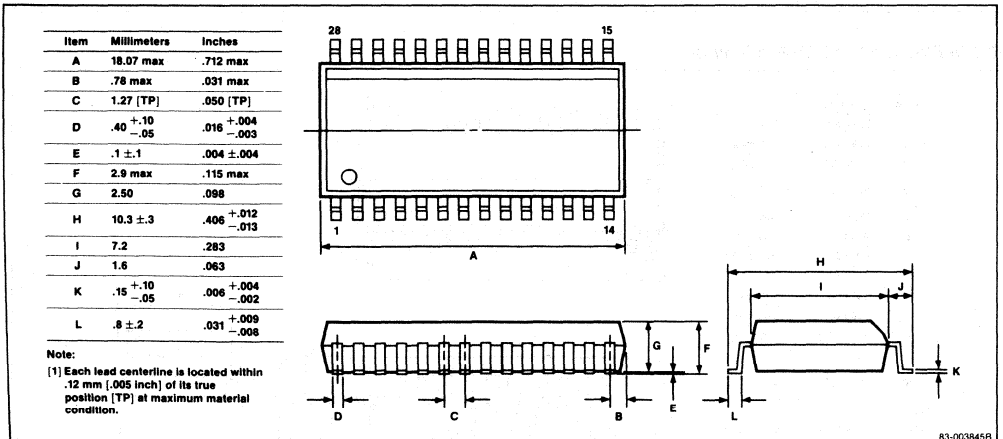


83-003887B

### 28-Pin Plastic DIP (600 mil)



### 28-Pin Plastic SO (Small Outline) (375 mil)





# PACKAGING INFORMATION

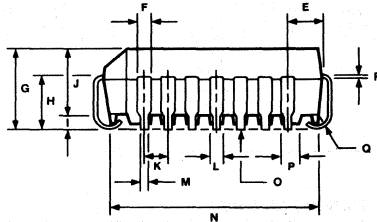
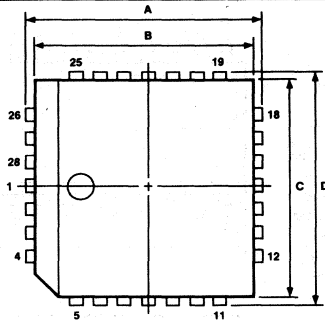
# NEC

## 28-Pin Plastic Leaded Chip Carrier (PLCC)

Item	Millimeters	Inches
A	12.45 ±.2	.490 ±.008
B	11.50	.453
C	11.50	.453
D	12.45 ±.2	.490 ±.008
E	1.94 ±.15	.076 <sup>+.007</sup> -.006
F	.6	.024
G	4.4 ±.2	.173 <sup>+.009</sup> -.008
H	2.8 ±.2	.110 <sup>+.009</sup> -.008
I	0.7 min	.028 min
J	3.6	.142
K	1.27 [TP]	.050 [TP]
L	.7	.028
M	.40 ±.10	.016 <sup>+.004</sup> -.005
N	10.42 ±.20	.410 <sup>+.009</sup> -.008
O	.15	.006
P	1.0	.040
Q	R .8	.031
R	.20 <sup>+.10</sup> -.05	.008 <sup>+.004</sup> -.002

**Note:**

- [1] Each lead centerline is located within .12 mm [.005 inch] of its true position [TP] at maximum material condition.
- [2] Flat within .15 mm [.006 inch] total.



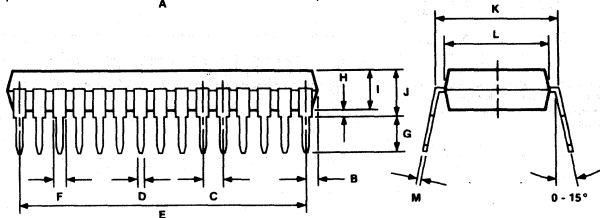
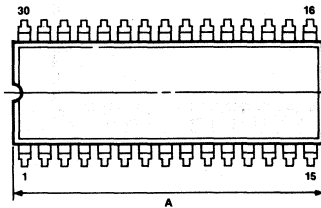
83-003789B

## 30-Pin Plastic Shrink DIP (400 mil)

Item	Millimeters	Inches
A	28.46 max	1.121 max
B	1.78 max	.070 max
C	1.778 [TP]	.070 [TP]
D	.50 ±.10	.020 <sup>+.004</sup> -.005
E	24.89	.980
F	.85 min	.033 min
G	3.2 ±.3	.126 ±.012
H	.51 min	.020 min
I	4.31 max	.170 max
J	5.08 max	.200 max
K	10.16 [TP]	.400 [TP]
L	8.6	.339
M	.25 <sup>+.10</sup> -.05	.010 <sup>+.004</sup> -.003

**Note:**

- [1] Each lead centerline is located within .17 mm [.007 inch] of its true position [TP] at maximum material condition.
- [2] Item "K" to center of leads when formed parallel.



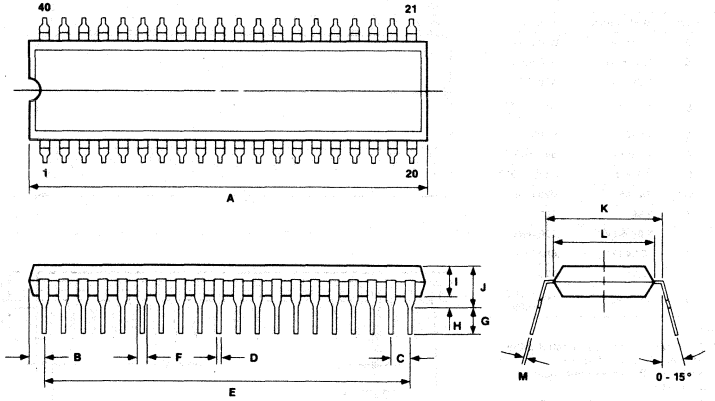
83-003848B

### 40-Pin Plastic DIP (600 mil)

Item	Millimeters	Inches
A	53.34 max	2.100 max
B	2.54 max	.100 max
C	2.54 [TP]	.100 [TP]
D	.50 ± .10	.020 +.004 -.005
E	48.26	1.900
F	1.2 min	.047 min
G	3.8 ± .3	.142 ± .012
H	.51 min	.020 min
I	4.31 max	.170 max
J	5.72 max	.226 max
K	15.24 [TP]	.600 [TP]
L	13.2	.520
M	.25 +.10 -.05	.010 +.004 -.003

**Notes:**

- [1] Each lead centerline is located within .25 mm [.010 inch] of its true position [TP] at maximum material condition.
- [2] Item "K" to center of leads when formed parallel.



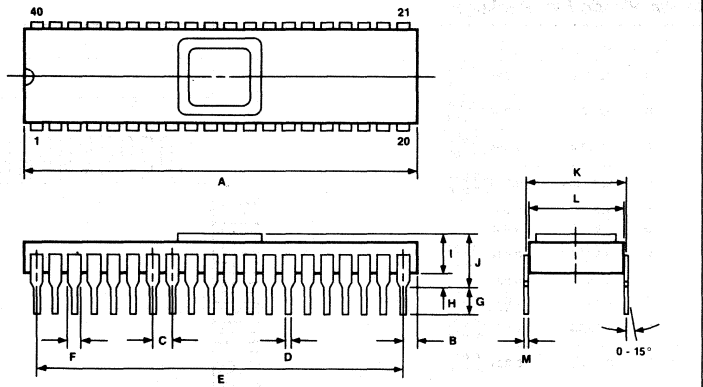
83-001399B

### 40-Pin Ceramic DIP (600 mil)

Item	Millimeters	Inches
A	53.34 max	2.100 max
B	2.54 max	.100 max
C	2.54 [TP]	.100 [TP]
D	.46 ± .05	.018 ± .002
E	48.26	1.900
F	.92 min	.036 min
G	3.5 ± .03	.0138 ± .012
H	1.0 min	.039 min
I	2.64	.104
J	4.57 max	.180 max
K	15.24 [TP]	.600 [TP]
L	14.93	.588
M	.25 ± .05	.010 +.002 -.003

**Note:**

- [1] Each lead centerline is located within .25 mm (.001 inch) of its true position [TP] at maximum material condition.
- [2] Item "K" to center of leads when formed parallel.

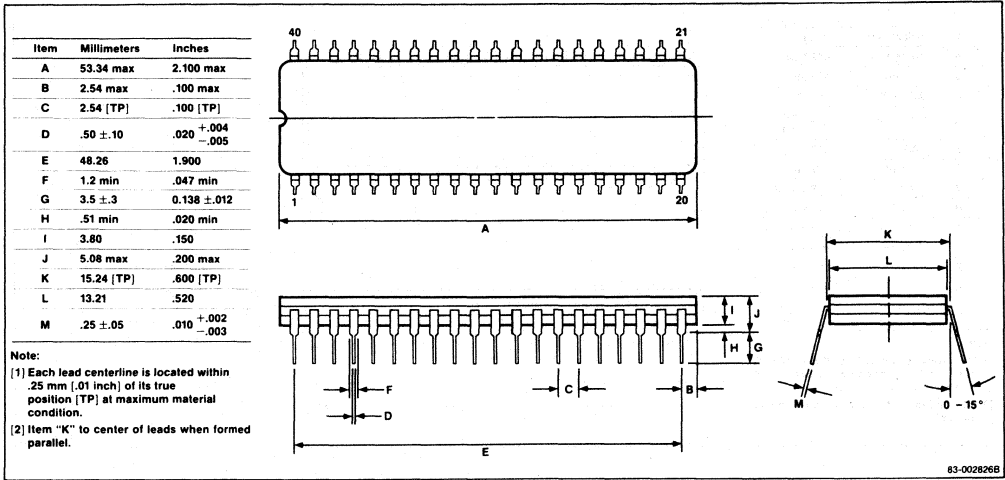


83-003786B

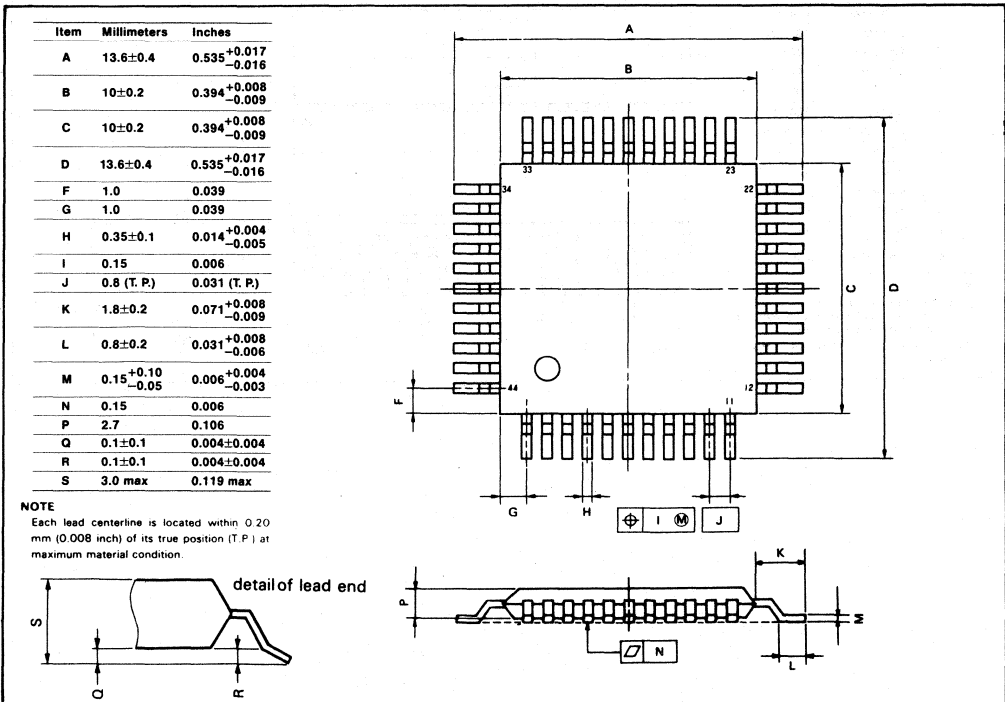
# PACKAGING INFORMATION



## 40-Pin Cerdip (600 mil)



## 44-Pin Plastic Flat Package

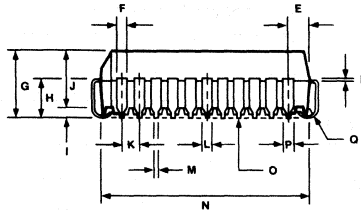
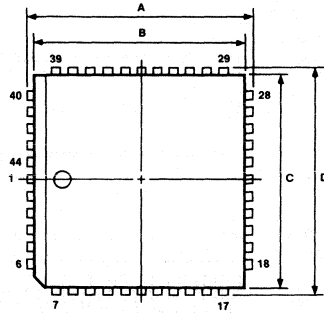


## 44-Pin Plastic Leaded Chip Carrier (PLCC)

Item	Millimeters	Inches
A	17.5 ±.2	.689 ±.008
B	16.58	.653
C	16.58	.653
D	17.5 ±.2	.689 ±.008
E	1.94 ±.15	.076 +.007 -.006
F	.6	.024
G	4.4 ±.2	.173 +.009 -.008
H	2.8 ±.2	.110 +.009 -.008
I	.7 min	.028 min
J	3.6	.142
K	1.27 [TP] Note 1	.050 [TP]
L	.7	.028
M	.40 ±.10	.016 +.004 -.005
N	15.50 ±.20	.610 +.009 -.008
O	.15 Note 2	.006
P	1.0	.040
Q	R.8	R.031
R	.20 +.10 -.05	.008 +.004 -.002

**Note:**

- [1] Each lead centerline is located within .12 mm [.005 inch] of its true position [TP] at maximum material condition.
- [2] Flat within .15 mm [.006 inch] total.



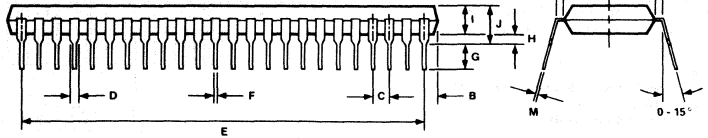
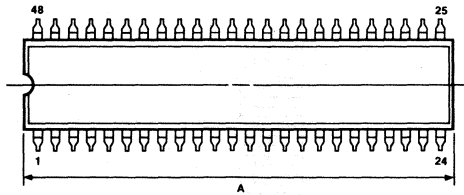
83-003793B

## 48-Pin Plastic DIP (600 mil)

Item	Millimeters	Inches
A	63.50 max	2.5 max
B	2.54 max	.10 max
C	2.54 [TP]	.10 [TP]
D	.5 ±.10	.02 +.004 -.005
E	58.42	2.3
F	1.1 min	.043 min
G	3.6 ±.3	.142 ±.012
H	.51 min	.02 min
I	4.31 max	.17 max
J	5.72 max	.226 max
K	15.24 [TP]	.60 [TP]
L	13.8	.543
M	.25 +.10 -.05	.01 +.004 -.003

**Notes:**

1. Each lead centerline is located within .25 mm [.01 inch] of its true position [TP] at maximum material condition.
2. Item "K" to center of leads when formed parallel.



83-001493B

## PACKAGING INFORMATION

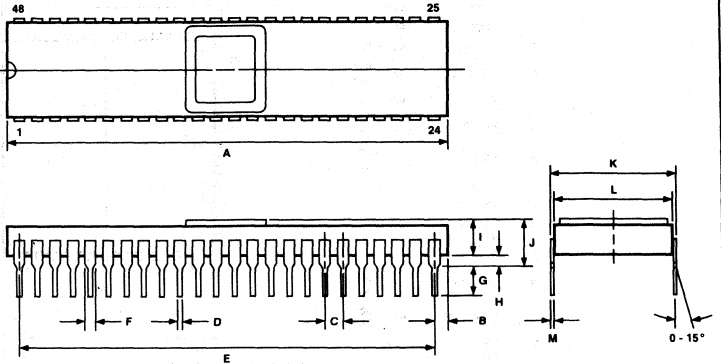
### 48-Pin Ceramic DIP (600 mil)

Item	Millimeters	Inches
A	63.50 max	2.500 max
B	2.54 max	.100 max
C	2.54 [TP]	.100 [TP]
D	.46 ±.05	.018 ±.002
E	58.42	2.300
F	.92 min	.036 min
G	3.5 ±.3	.138 ±.012
H	1.0 min	.039 min
I	2.74	.108
Note 1		
J	4.57 max	.180 max
K	15.24 [TP]	.600 [TP]
L	14.93	.588
M	.25 ±.05	.010 ±.002
	Note 2	-.003

**Note:**

[1] Each lead centerline is located within .25 mm (.01 inch) of its true position [TP] at maximum material condition.

[2] Item "K" to center of leads when formed parallel.



83-003853B

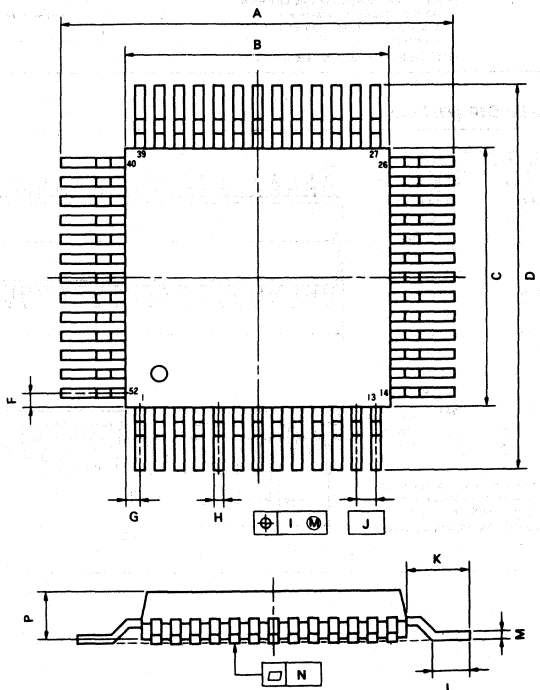
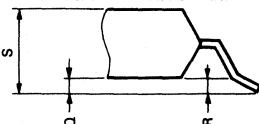
### 52-Pin Plastic Flat Package

Item	Millimeters	Inches
A	17.6±0.4	0.693±0.016
B	14±0.2	0.551 <sup>+0.009</sup> <sub>-0.008</sub>
C	14±0.2	0.551 <sup>+0.009</sup> <sub>-0.008</sub>
D	17.6±0.4	0.693±0.016
F	1.0	0.039
G	1.0	0.039
H	0.40±0.10	0.016 <sup>+0.004</sup> <sub>-0.005</sub>
I	0.20	0.008
J	1.0 (T. P.)	0.039 (T. P.)
K	1.8±0.2	0.071 <sup>+0.008</sup> <sub>-0.009</sub>
L	0.8±0.2	0.031 <sup>+0.009</sup> <sub>-0.008</sub>
M	0.15 <sup>+0.10</sup> <sub>-0.05</sub>	0.006 <sup>+0.004</sup> <sub>-0.003</sub>
N	0.15	0.006
P	2.7	0.106
Q	0.1±0.1	0.004±0.004
R	0.1±0.1	0.004±0.004
S	3.0 max	0.119 max

**NOTE**

Each lead centerline is located within 0.20 mm (0.008 inch) of its true position (T.P.) at maximum material condition.

**detail of lead end**



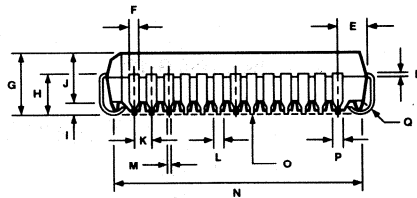
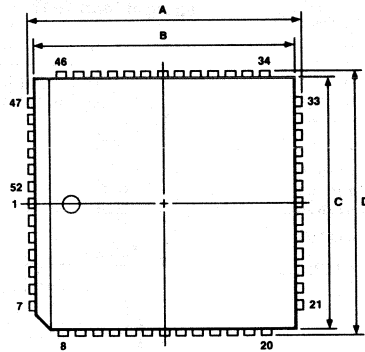
### 52-Pin Plastic Leaded Chip Carrier (PLCC)

Item	Millimeters	Inches
A	20.1 ±.2	.791 <sup>+.009</sup> <sub>-.008</sub>
B	19.12	.753
C	19.12	.753
D	20.1 ±.2	.791 <sup>+.009</sup> <sub>-.008</sub>
E	1.94 ±.15	.076 <sup>+.007</sup> <sub>-.006</sub>
F	.6	.024
G	4.4 ±.2	.173 <sup>+.009</sup> <sub>-.008</sub>
H	2.8 ±.2	.110 <sup>+.009</sup> <sub>-.008</sub>
I	.7 min	.028 min
J	3.6	.142
K	1.27 [TP] Note 1	.050 [TP]
L	.7	.028
M	40 ±.10	.016 <sup>+.004</sup> <sub>-.005</sub>
N	18.04 ±.20	.710 <sup>+.009</sup> <sub>-.008</sub>
O	.15 Note 2	.006
P	1.0	.040
Q	R.8	R.031
R	.20 <sup>+.10</sup> <sub>-.05</sub>	.008 <sup>+.004</sup> <sub>-.002</sub>

**Note:**

[1] Each lead centerline is located within .12 mm [.005 inch] of its true position [TP] at maximum material condition.

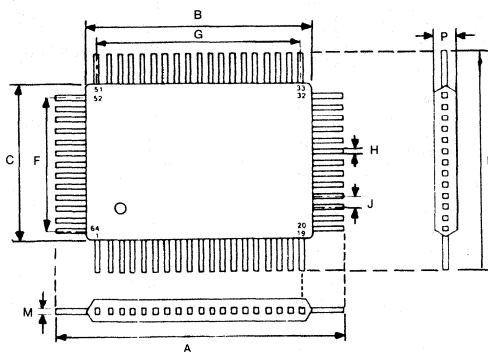
[2] Flat within .15 mm [.006 inch] total.



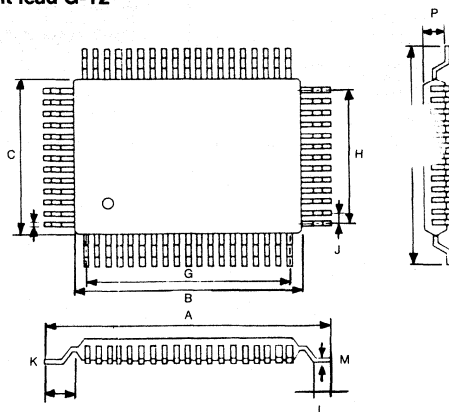
## 64-Pin Plastic Flat Package

Item	Millimeters	
	G-11	G-12
A	25.6 ±0.4	24.7 ±0.4
B	20	20
C	14	14
D	19.6 ±0.4	18.7 ±0.4
F	12.0 ±0.3	12.0 ±0.3
G	18.0 ±0.3	18.0 ±0.3
H	0.4 ±0.1	0.4 ±0.1
J	1.0 ±0.15	1.0 ±0.15
K	-	2.35 ±0.2
L	-	1.2 ±0.2
M	0.15 <sup>+0.10</sup> / <sub>-0.05</sub>	0.15 <sup>+0.10</sup> / <sub>-0.05</sub>
P	2.3 max	2.3 max

### straight lead G-11



### bent lead G-12



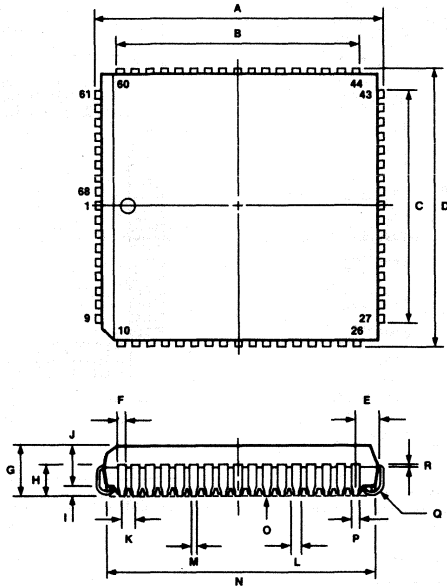
### 68-Pin Plastic Leaded Chip Carrier (PLCC)

Item	Millimeters	Inches
A	25.2 ±.2	.992 ±.008
B	24.20	.953
C	24.20	.953
D	25.2 ±.2	.992 ±.008
E	1.94 ±.15	.076 <sup>+007</sup> <sub>-.006</sub>
F	.6	.024
G	4.4 ±.2	.173 <sup>+009</sup> <sub>-.008</sub>
H	2.8 ±.2	.110 <sup>+009</sup> <sub>-.008</sub>
I	.7 min	.028 min
J	3.6	.142
K	1.27 [TP]	.050 [TP]
L	.7	.028
M	.40 ±.10	.016 <sup>+004</sup> <sub>-.005</sub>
N	23.12 ±.20	.910 <sup>+009</sup> <sub>-.008</sub>
O	.15	.006
P	1.0	.040
Q	R .8	R .031
R	.20 <sup>+010</sup> <sub>-.06</sub>	.008 <sup>+004</sup> <sub>-.002</sub>

**Note:**

[1] Each lead centerline is located within .12 mm [.005 inch] of its true position [TP] at maximum material condition.

[2] Flat within .15 mm [.006 inch] total.



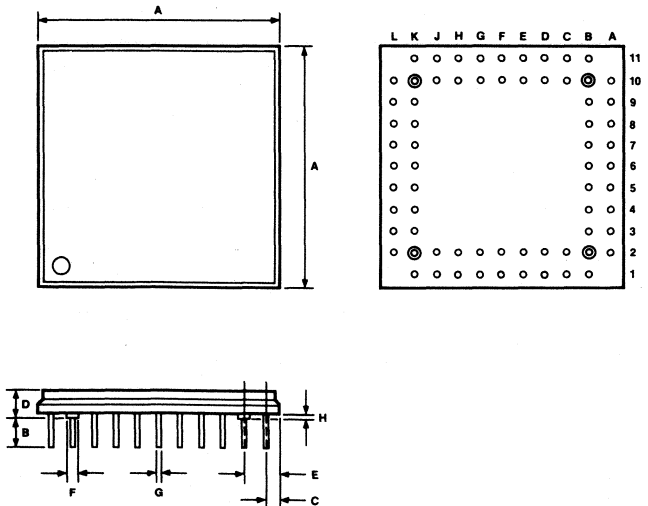
83-003792B

### 68-Pin Ceramic PGA

Item	Millimeters	Inches
A	27.9 <sup>+0.8</sup> <sub>-.4</sub>	1.10 <sup>+031</sup> <sub>-.015</sub>
B	2.8 ±.3	.110 ±.012
C	1.25	.049
D	5.08 max	.200 max
E	2.54 [TP]	.100 [TP]
F	φ 1.2 ±.2	.047 ±.008
G	φ .46 ±.05	.018 ±.002
H	.5 min	.020 min

**Note:**

[1] Each lead centerline is located within .25 mm [.010 inch] of its true position [TP] at maximum material condition.



83-003842B



# PACKAGING INFORMATION

# NEC

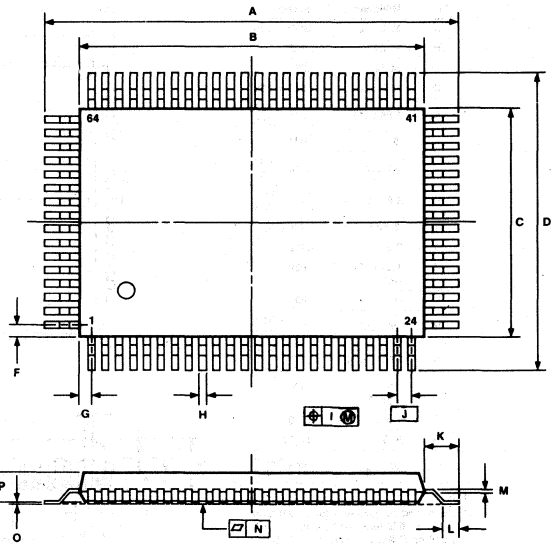
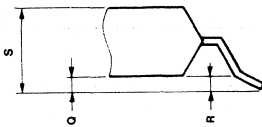
## 80-Pin Plastic Flat Package (bent lead):

Item	Millimeters	Inches
A	23.6 ±0.4	0.929 ±0.016
B	20 ±0.2	0.795 <sup>+0.009</sup> <sub>-0.008</sub>
C	14 ±0.2	0.551 <sup>+0.009</sup> <sub>-0.008</sub>
D	17.6 ±0.4	0.693 ±0.016
F	1.0	0.039
G	0.8	0.031
H	0.35 ±0.10	0.014 <sup>+0.004</sup> <sub>-0.005</sub>
I	0.15	0.006
J	0.8 (T.P.)	0.031 (T.P.)
K	1.8 ±0.2	0.071 <sup>+0.008</sup> <sub>-0.009</sub>
L	0.8 ±0.2	0.031 <sup>+0.009</sup> <sub>-0.008</sub>
M	0.15 <sup>+0.10</sup> <sub>-0.05</sub>	0.006 <sup>+0.004</sup> <sub>-0.003</sub>
N	0.15	0.006
P	2.7	0.106
Q	0.1 ±0.1	0.004 ±0.004
R	0.1 ±0.1	0.004 ±0.004
S	3.0 max	0.119 max

**Note**

Each lead centerline is located within 0.12 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

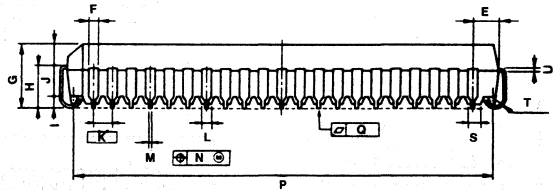
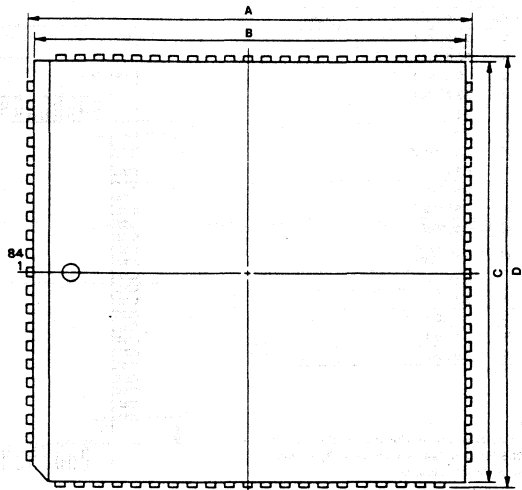
detail of lead end



### 84-Pin Plastic Leaded Chip Carrier (PLCC)

PLCC-84-100 (MIL-STD-188)

Item	Millimeters	Inches
A	30.4 ±0.2	1.197 ±0.006
B	29.30	1.154
C	29.30	1.154
D	30.4 ±0.2	1.197 ±0.006
E	1.95 ±0.16	0.077 <sup>+0.006</sup> -0.007
F	0.6	0.024
G	4.4 ±0.2	0.173 <sup>+0.009</sup> -0.006
H	2.5 ±0.2	0.098 <sup>+0.009</sup> -0.006
I	0.6 min.	0.024 max.
J	3.7	0.146
K	1.27 (T.P.)	0.050 (T.P.)
L	0.7	0.028
M	0.40 ±0.10	0.016 <sup>+0.004</sup> -0.006
N	0.12	0.005
P	28.51 ±0.20	1.122 <sup>+0.009</sup> -0.006
Q	0.15	0.006
S	1.0	0.040
T	R 0.8	R 0.031
U	0.20 <sup>+0.10</sup> -0.06	0.008 <sup>+0.004</sup> -0.002



**Note**  
Each lead centerline is located within 0.15 mm (0.006 inch) of its true position (T.P.) at maximum material condition.

# PACKAGING INFORMATION

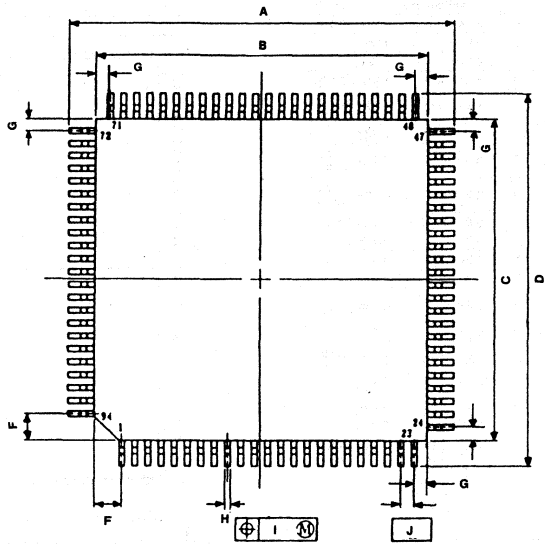
# NEC

## 94-Pin Plastic Flat Package

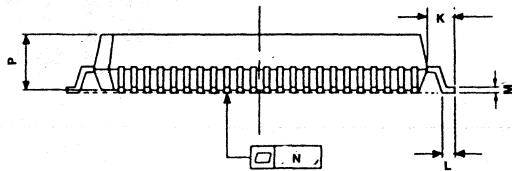
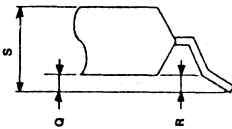
Item	Millimeters	Inches
A	23.2 ±0.4	
B	20.0 ±0.2	
C	20.0 ±0.2	
D	23.2 ±0.4	
F	1.6	
G	0.8	
H	0.35 ±0.10	
I	0.15	
J	0.8 (T.P.)	
K	1.6 ±0.2	
L	0.8 ±0.2	
M	0.15 <sup>+0.10</sup> <sub>-0.05</sub>	
N	0.15	
P	3.7	
Q	0.1 ±0.1	
R	0.1 ±0.1	
S	4.0 max	

**Note**

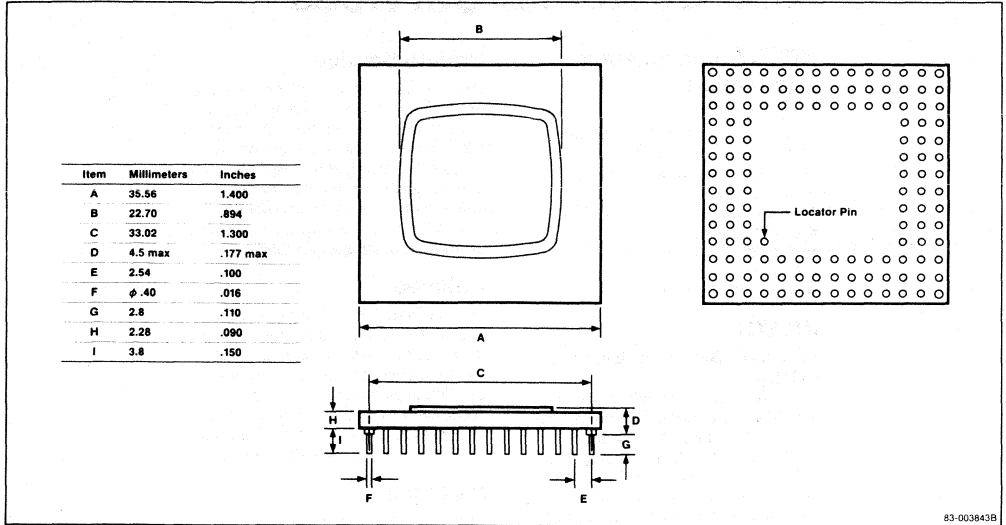
Each lead centerline is located within 0.15 mm (0.006 inch) of its true position (T.P.) at maximum material condition.



detail of lead end



### 132-Pin Ceramic PGA (Pin Grid Array)



83-003843B

# NEC Customer Services

## NEC's Commitment to Information

Our offices throughout Europe are always at your service for comprehensive support. Here are some of the technical services we provide:

- INSECT
- Seminars
- Update service
- Hotline
- Mailing list
- University program

### INSECT

**I**nformation **S**ystem and **E**lectronic **C**atalog.

This is an on-line information service. Via a telephone link you can call up the latest data on all VLSI devices available from NEC. This includes enhancements, news and the most recent application know-how.

The service is free to our customers and other interested parties. For a menu-driven guest session, you can dial in to INSECT via the international packet switching network – in Germany this is DATEX-P – using of these numbers  
45 21 10 13 020/030  
and responding to the request for USERNAME and PASSWORD simply with "customer".

### Seminars

No-one is more aware than NEC of the difference that a brief intensive training course can make to your mastery of advanced and often complex devices. We hold regular workshops and seminars at local NEC offices, in our Düsseldorf headquarters, or on customer premises. For information on NEC workshops and seminars, please contact your nearest office.

## Update Service

When you buy an evaluation package from NEC, you become automatically entitled to one year's free updates for both hardware and software. All updates reach you fast and reliably via a courier service. In a field where rapid changes are the norm, you can be thus be sure of working with the most up-to-date development tools.

## Hotlines

NEC's offices located throughout Europe are responsible for technical support and customer services. On the back cover of this brochure you can check which office is most convenient for you to contact. You are also welcome to contact our European headquarters in Düsseldorf directly.

## Mailing list

Our engineering staff produces frequent additions to the available technical documentation in the form of application notes, product news and technical letters. If you would like to be included on our mailing list for this documentation, please inform us.

## University Program

Many of our products, because of their complexity and dedicated application support through EB tools, are interesting subjects for graduate studies. NEC is always ready to discuss this possibility.

## EUROPEAN DISTRIBUTORS

### AUSTRIA

A & D  
ABRAHAMCZIK & DEMEL  
GES. MBH. & CO KG  
EICHENSTRASSE 58-64/1  
1120 WIEN  
TEL.: (222) 85 76 61  
TLX.: 134 273

### BELGIUM

INTRA ELECTRONICS P.V.B.A.  
BOSUIL 80, BUS 1  
2100 DEURNE  
TEL.: (03) 325 23 20  
TLX.: 31102  
MALCHUS ELECTRONICS B.V.B.A.  
PLANTIN EN MORETUSLEI 172  
2018 ANTWERPEN  
TEL.: (03) 2 35 32 72

INNOCIRCUIT BENELUX B.V.B.A.  
PLANTIN EN MORETUSLEI 172  
2018 ANTWERPEN  
TEL.: (03) 2 35 32 72 (B)  
TEL.: (00 31) 10-4 27 77 80  
TLX.: 33 637

### DENMARK

MER-EL A/S  
VED KLAEDBO 18  
2970 HOERSHOLM  
TEL.: (2) 57 10 00  
TLX.: 37 360

### FINLAND

OY COMDAX AB  
ITÄELÄHDENKATU 23A  
PI 1  
00210 HELSINKI  
TEL.: (0) 67 02 77  
TLX.: 57 125 876

### FRANCE

ASAP  
2 AVENUE DES CHAUMES  
78180 MONTIGNY LE BRETONNEUX  
TEL.: (1) 30 43 82 33  
TLX.: 698 887

### C.C.I.

5, RUE MARCELIN BERTHELOT  
B.P. 92  
92164 ANTONY  
TEL.: (1) 46 66 21 82  
TLX.: 203 881

CGE COMPOSANTS  
32 RUE GRANGE DAME ROSE  
92360 MEUDON  
TEL.: (1) 46 30 24 25  
TLX.: 632 118

PANTEC (GROUPE SCIENTECH)  
29 RUE MARCEL DASSAULT  
92100 BOULOGNE  
TEL.: (1) 46 20 06 74  
TLX.: 633 048

### GERMANY

BIT-ELECTRONIC AG  
DINGOLFINGER STRASSE 6  
8000 MÜNCHEN 80  
TEL.: (0 89) 41 80 07-0  
TLX.: 5 212 931

GLEICHMANN + CO ELECTRONICS  
GMBH  
WORMSER STRASSE 34  
6710 FRANKENTHAL  
TEL.: (0 62 33) 250 56  
TLX.: 4 65 270

GLYN GMBH  
SCHÖNE AUSSICHT 30  
6272 NIEDERNHAUSEN  
TEL.: (0 61 27) 80 77  
TLX.: 4 186 911

H3W ELEKTRONIK VERTRIEBS GMBH  
STAHLGRUBERRING 12  
8000 MÜNCHEN 82  
TEL.: (0 89) 42 92 71  
TLX.: 5 214 514

MICROSCAN GMBH  
ÜBERSEERING 31  
2000 HAMBURG 60  
TEL.: (0 40) 6 32 00 30  
TLX.: 2 13 288

REIN ELEKTRONIK GMBH  
LÖTSCHERWEG 66  
4054 NETTETAL 1  
TEL.: (0 21 53) 73 31 11  
TLX.: 8 54 251

SYSTEM ELEKTRONIK VERTRIEBS  
GMBH  
HEESFELD 4  
3300 BRAUNSCHWEIG  
TEL.: (05 31) 31 40 95  
TLX.: 9 52 351

ULTRATRONIK GMBH  
GEWERBESTRASSE 4  
8036 HERRSCHING  
TEL.: (0 81 52) 37 09-0  
TLX.: 5 26 459

UNIELECTRONIC VERTRIEBS GMBH  
IM GEFIERTH 11A  
6072 DREIEICH 1 B. FRANKFURT  
TEL.: (0 61 03) 3 51 75  
TLX.: 4 11 213

### ITALY

ADELSY S.R.L.  
VIA DEL FONDITORE, 5  
LOCALITA ROVERI  
40127 BOLOGNA  
TEL.: (51) 532119  
TLX.: 510 226

CLAITRON S.P.A.  
VIA GALLARATE, 211  
20151 MILANO  
TEL.: (01) 33404000  
TLX.: 313 843

DIS-EL S.P.A.  
VIA ALA DI STURA 71  
10148 TORINO  
TEL.: (011) 2201522  
TLX.: 215 118

MELCHIONI S.P.A.  
VIA COLLETTA, 37  
20135 MILANO  
TEL.: (01) 57941  
TLX.: 315 293

### NETHERLANDS

MALCHUS B.V.  
FOKKERSTRAAT 511  
3125 BD SCHIEDAM  
TEL.: (10) 4 27 77 77  
TLX.: 21 598

INNOCIRCUIT BENELUX B.V.  
POSTBUS 48  
3100 AA SCHIEDAM  
TEL.: (10) 4 27 77 80  
TLX.: 21 598

INTRA ELECTRONICS B.V.  
GULBERG 33  
5874 TE NUENEN  
TEL.: (40) 83 80 09  
TLX.: 59 418

### NORWAY

JAKOB HATTELAND ELECTRONIC A/S  
PB. 25  
5578 NEDRE VATS  
TEL.: (47) 63 111  
TLX.: 428 50

### PORTUGAL

AMPEREL S.A.  
AV. FONTES PEREIRA DE MELO 47, 4D  
1000 LISBOA  
TEL.: (1) 53 26 98  
TLX.: 18588

### SPAIN

COMELTA SA  
EMILIO MUNOZ 41. NAVE 1-1-2  
28037 MADRID  
TEL.: (1) 7 54 30 77  
TLX.: 42 007

VENCO  
CALLE GALILEO 249  
08028 BARCELONA  
TEL.: (3) 330 97 51  
TLX.: 98 266

### SWEDEN

NAX AB KOMPLEMENTBOLAGET  
BOX 4115  
17104 SOLNA  
TEL.: (8) 98 51 40  
TLX.: 17912

NORDQVIST & BERG  
BOX 1458  
17128 SOLNA  
TEL.: (8) 764 61 10  
TLX.: 10407

TH'S ELEKTRONIK  
BOX 3027  
16303 SPAANGA  
TEL.: (8) 36 29 70  
TLX.: 11145

### SWITZERLAND

MEMOTEC AG  
GASWERKSTRASSE 32  
4901 LANGENTHAL  
TEL.: (63) 28 11 22  
TLX.: 9 82 550

### UNITED KINGDOM

2001 ELECTRONIC COMPONENTS  
WOOLMERS WAY  
STEVENAGE  
HERTS  
SG1 3AJ  
TEL.: (4 38) 74 20 01  
TLX.: 827 701

ANZAC COMPONENTS LIMITED  
822, YEOVIL ROAD  
SLOUGH TRADING ESTATE  
SLOUGH  
BERKSHIRE  
SL1 4JA  
TEL.: (62 86) 44 11  
TLX.: 847 949

CELDIS LIMITED  
37, LOVEROCK ROAD  
READING  
BERKSHIRE  
RG3 1EL  
TEL.: (7 34) 58 51 71  
TLX.: 848 370

IMPULSE ELECTRONICS LIMITED  
HAMMOND HOUSE  
CATERHAM  
SURREY  
CR3 6XG  
TEL.: (8 83) 4 70 11  
TLX.: 291 496

S.T.C. MULTICOMPONENTS LIMITED  
EDINBURGH WAY  
HARLOW  
ESSEX  
CM20 2DF  
TEL.: (2 79) 44 11 44  
TLX.: 818 763

V.S.I. ELECTRONICS (UK) LIMITED  
ROYDONBURY INDUSTRIAL PARK  
HORSECROFT ROAD  
HARLOW  
ESSEX  
CM19 5BY  
TEL.: (2 79) 2 96 66  
TLX.: 81 387