μPD8259A
PROGRAMMABLE INTERRUPT CONTROLLER

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.

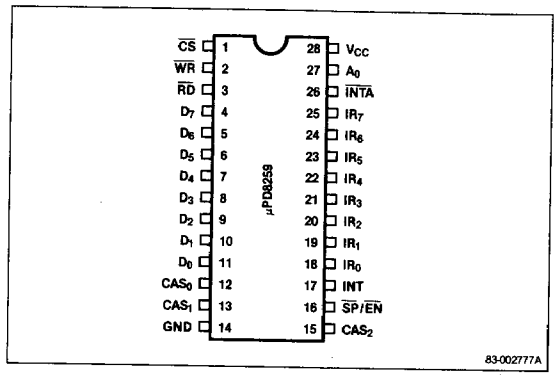
Features

- ☐ 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	$\overline{CS}$	Chip select input
2	$\overline{WR}$	Write input
3	$\overline{RD}$	Read input
4–11	D_7–D_0	Bidirectional data bus
12, 13, 15	CAS_0–CAS_2	Cascade lines
14	GND	Ground
16	$\overline{SP}/\overline{EN}$	Slave program input / enable buffer output
17	INT	Interrupt output
18–25	IR_0–IR_7	Interrupt request inputs
26	$\overline{INTA}$	Interrupt acknowledge input
27	A_0	Command select address input
28	V_{CC}	+5 V power supply

8

Pin Functions

Bidirectional Data Bus (D_7-D_0)

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_0-IR_7)

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_0-CAS_2)

These lines are used as a bus which controls multiple μPD8259As in a master/slave configuration. When an μPD8259A is a master, these lines are outputs. When a μPD8259A 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 μPD8259A. The INTA input operates independently of $\overline{CS}$.

Command Select Address Input (A_0)

The μPD8259A uses this input with $\overline{CS}$ and $\overline{WR}$ to decode command words written by the CPU. A_0 is used with $\overline{CS}$ and $\overline{RD}$ to decode controller status information for the CPU to read. Typically, A_0 is connected to the A_0 address lines on the CPU.

Interrupt (INT)

When the μPD8259A 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 μPD8259A. $\overline{INTA}$ enables interrupt vector data onto the data bus.

Read Input ($\overline{RD}$)

When both $\overline{RD}$ and $\overline{CS}$ are low, the μPD8259A sends its status information to the data bus so the CPU can read it.

Write Input ($\overline{WR}$)

The μPD8259A can receive command words from the CPU when both $\overline{WR}$ and $\overline{CS}$ are low.

Slave Program Input/Enable Buffer Output ($\overline{SP}/\overline{EN}$)

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 μPD8259A operates as a master and when the $\overline{SP}$ input is low, the μPD8259A operates as a slave.

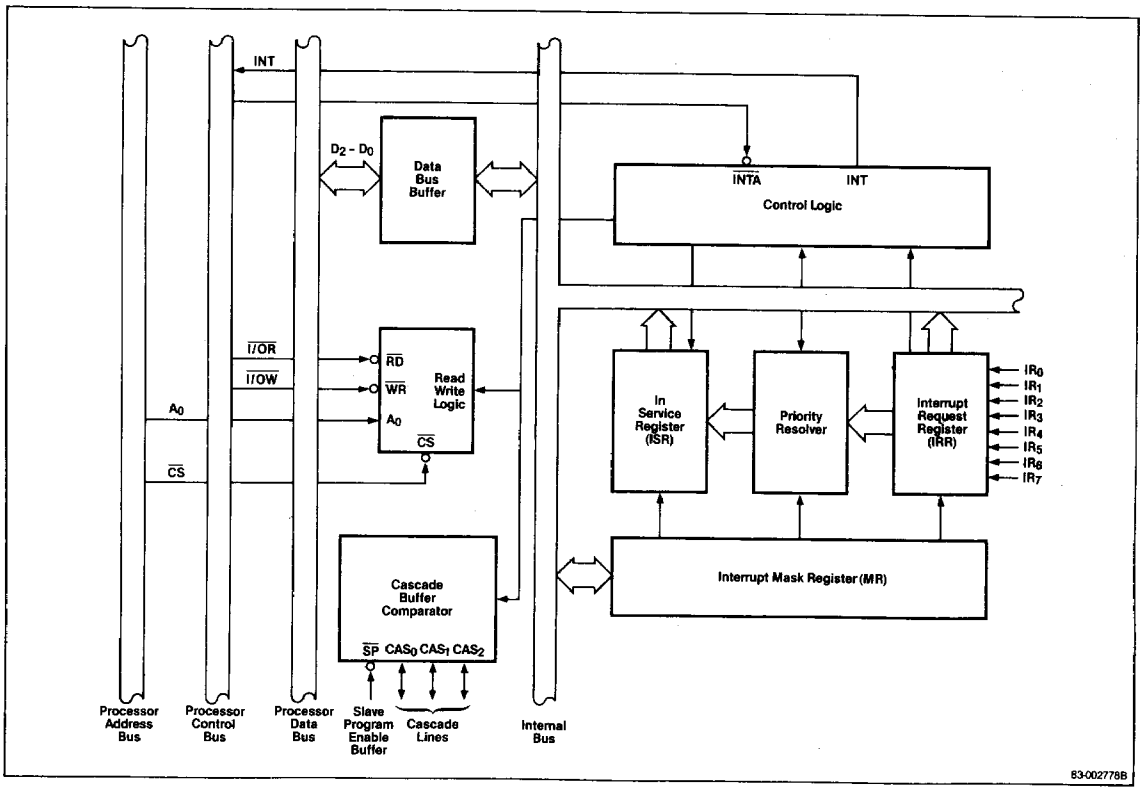
Ground (GND)

Ground

Power Supply (V_{CC})

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_0-IR_7 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 $\overline{INTA}$ input has been received. Multiple non-masked interrupts occurring simultaneously can be stored in the IRR. The incoming $\overline{INTA}$ 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 $\overline{INTA}$ 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.

8

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
						Processor Input (Read)
0			0	1	0	IRR, ISR or IR → data bus (Note 1)
1			0	1	0	IMR → data bus
						Processor Output (Write)
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)
						Disable Function
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 SP input must be set to a logic 1, since it is functioning as a master.

Absolute Maximum Ratings

$T_A = 25°C$

Power supply voltage, V_{CC}	−0.5 to +7.0 V(Note 1)
Input voltage, V_I	−1.0 V to V_{CC} + 1.0 V
Output voltage, V_O	−0.5 V to V_{CC} + 0.5 V
Operating temperature, T_{OPT}	0 to +70°C
Storage temperature, T_{STG}	−65 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

$T_A = 0$ to +70°C, $V_{CC} = +5\,V \pm 10\%$

Parameter	Symbol	Limits Min	Typ	Max	Unit	Test Conditions
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	$I_{OL} = 2.2$ mA
Output voltage high	V_{OH}	2.4			V	$I_{OH} = -400\,\mu A$
Interrupt output	V_{OH-INT}	2.4			V	$I_{OH} = -400\,\mu A$
High voltage		3.5			V	$I_{OH} = -100\,\mu A$
Input leakage current (Note 1)	I_{LI}	−10		10	μA	$0\,V \leq V_I \leq V_{CC}$
Output leakage current	I_{LO}	−10		10	μA	$0.45\,V \leq V_O \leq V_{CC}$
V_{CC} power supply current	I_{CC}			85	mA	

Note:
(1) For other inputs.

AC Characteristics

Timing Requirements

$T_A = 0°C$ to +70°C, $V_{CC} = +5\,V \pm 10\%$

Parameter	Symbol	Limits μPD8259A Min	Max	μPD8259A-2 Min	Max	Unit	Test Conditions
A0 / $\overline{CS}$ setup to $\overline{RD}$ / $\overline{INTA}$ ↓	t_{AHRL}	0		0		ns	
A0 / $\overline{CS}$ hold after $\overline{RD}$ / $\overline{INTA}$ ↑	t_{RHAX}	0		0		ns	
$\overline{RD}$ pulse width	t_{RLRH}	235		160		ns	
A0 / $\overline{CS}$ setup to $\overline{WR}$ ↓	t_{AHWL}	0		0		ns	
A0 / $\overline{CS}$ hold after $\overline{WR}$ ↑	t_{WHAX}	0		0		ns	
$\overline{WR}$ pulse width	t_{WLWH}	290		190		ns	
Data setup to $\overline{WR}$ ↑	t_{DVWH}	240		160		ns	
Data hold after $\overline{WR}$ ↑	t_{WHDX}	0		0		ns	
Interrupt request width low	t_{JLJH}	100		100		ns	(Note 1)
Cascade setup to second or third $\overline{INTA}$ ↓ (slave only)	t_{CVIAL}	55		40		ns	
End of $\overline{RD}$ to next command	t_{RHRL}	160		160		ns	
End of $\overline{WR}$ to next command	t_{WHRL}	190		190		ns	
End of command to next command (different type)	t_{CHCL}	500		500		ns	(Note 2)
End of $\overline{INTA}$ sequence to next $\overline{INTA}$ sequence	t_{CHCL}	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_{CHCL} 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).

8

AC Characteristics (cont)

Timing Responses

$T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 10\%$

Parameter	Symbol	Limits μPD8259A Min	Limits μPD8259A Max	Limits μPD8259A-2 Min	Limits μPD8259A-2 Max	Unit	Test Conditions
Data valid from $\overline{RD}$ / $\overline{INTA}$ ↓	t_{RLDV}		200		120	ns	(Notes 1–5)
Data float after $\overline{RD}$/$\overline{INTA}$ ↑	t_{RHDZ}	10	100	10	85	ns	(Notes 1–5)
Interrupt output delay	t_{JHIH}		350		300	ns	(Notes 1–5)
Cascade valid from first $\overline{INTA}$ ↓ (master only)	t_{IALCV}		565		360	ns	(Notes 1–5)
Enable active from $\overline{RD}$ ↓ or $\overline{INTA}$ ↑	t_{RLEL}		125		100	ns	(Notes 1–5)
Enable inactive from $\overline{RD}$ ↑ or $\overline{INTA}$ ↑	t_{RHEH}		150		150	ns	(Notes 1–5)
Data valid from stable address	t_{AHDV}		200		200	ns	(Notes 1–5)
Cascade valid to valid data	t_{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_{INT} = 100 pF
(5) $C_{CASCADE}$ = 100 pF

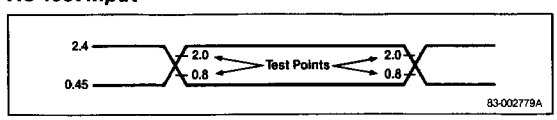
Capacitance

$T_A = 25°C$, $V_{CC} = GND = 0V$, fc = 1.0 MHz

Parameter	Symbol	Limits Min	Limits Typ	Limits Max	Unit	Test Conditions
Input capacitance	C_I			10	pF	(Note 1)
I/O capacitance	$C_{I/O}$			20	pF	(Note 1)

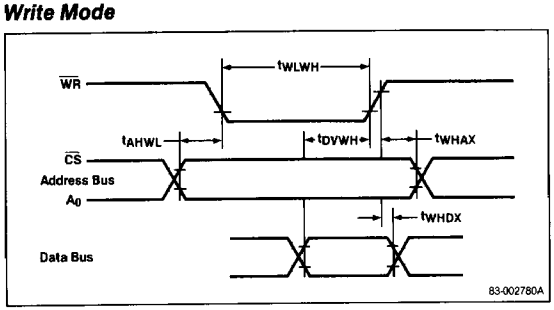
Note:
(1) Unmeasured pins returned to V_{SS}

AC Test Input

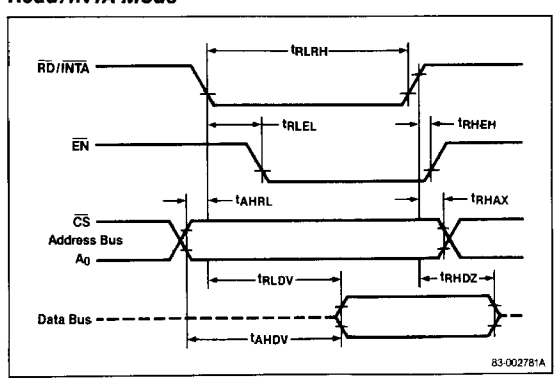


Timing Waveforms

Write Mode

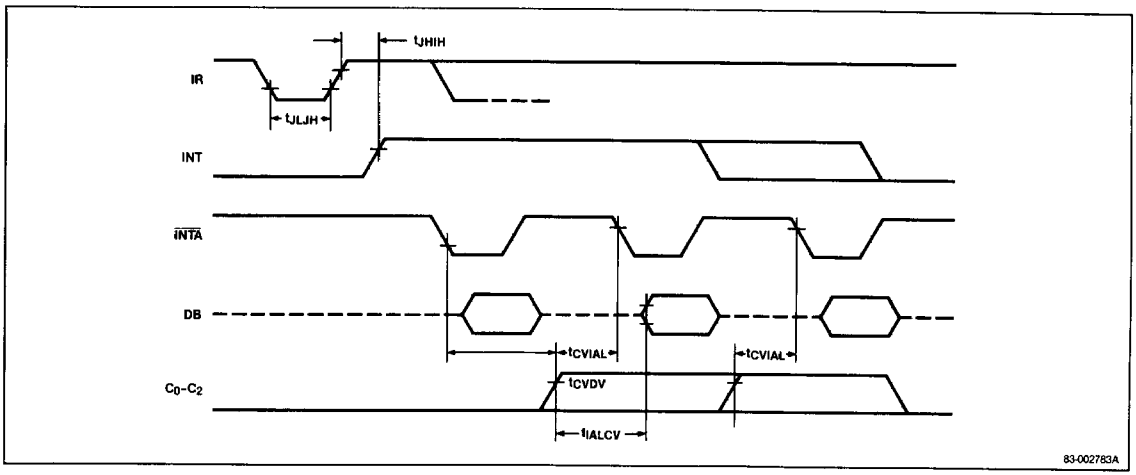


Read/$\overline{INTA}$ Mode

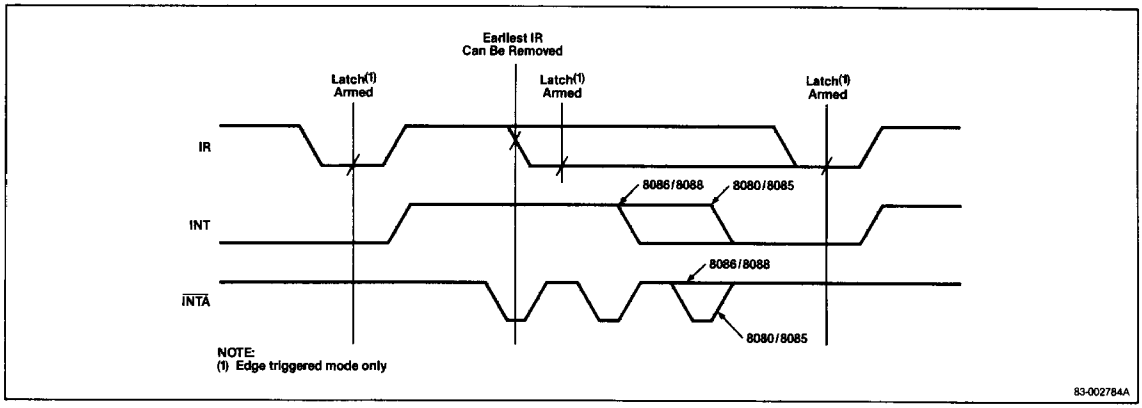


Timing Waveforms (cont)

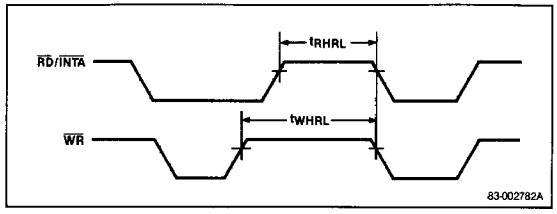
INTA Sequence



IR Triggering Timing Requirements



Other Timing



8

Functional Description

The μPD8259A 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 μPD8259A Status

Interrupt Sequence

The μPD8259A 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 μPD8259A to handle an interrupt depends upon whether an 8080A/8085A or 8080/8088 CPU is being used.

The following sequence demonstrates how the μPD8259A interacts with the 8080A/8085A systems.

(1) An interrupt(s) appearing on IR_0-IR_7 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 μPD8259A 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 $\overline{INTA}$ to the μPD8259A.

(4) The $\overline{INTA}$ input to the μPD8259A from the processor group sets the highest priority ISR bit and resets the corresponding IRR bit. The $\overline{INTA}$ also signals the μPD8259A 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 $\overline{INTA}$ pulses to the μPD8259A.

(6) The two $\overline{INTA}$ pulses signal the μPD8259A to place its preprogrammed interrupt vector address onto the data bus. The first $\overline{INTA}$ releases the low order 8 bits of the address and the second $\overline{INTA}$ releases the high order 8 bits.

(7) The μPD8259As CALL instruction sequence is complete. A preprogrammed EOI command is issued to the μPD8259A at the end of the interrupt service routine. This resets the ISR bit and allows the μPD8259A to service the next interrupt.

The following sequence demonstrates how the μPD8259A interacts with the 8086/8088 systems.

(1), (2), (3) Same as for 8080A/8085A.

(4) During the first $\overline{INTA}$ from the processor, the μPD8259A does not drive the data bus. The highest priority ISR bit is set and the corresponding IRR bit is reset.

(5) The μPD8259A puts vector information onto the data bus on the second $\overline{INTA}$ pulse from the 8086/8088.

(6) There is no third $\overline{INTA}$ pulse in this mode. In the AEOI mode the ISR bit is reset at the end of the second $\overline{INTA}$ pulse, or it remains set until an EOI command is issued.

8080/8085A Mode

For these processors, the μPD8259A is controlled by three $\overline{INTA}$ pulses. The first $\overline{INTA}$ pulse will cause the μPD8259A to put the CALL opcode onto the data bus. See table 2. The second and third $\overline{INTA}$ 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_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	0	0	1	1	0	1

Table 3. Contents of Second Interrupt Vector Byte

IR	Interval = 4							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
7	A_7	A_6	A_5	1	1	1	0	0
6	A_7	A_6	A_5	1	1	0	0	0
5	A_7	A_6	A_5	1	0	1	0	0
4	A_7	A_6	A_5	1	0	0	0	0
3	A_7	A_6	A_5	0	1	1	0	0
2	A_7	A_6	A_5	0	1	0	0	0
1	A_7	A_6	A_5	0	0	1	0	0
0	A_7	A_6	A_5	0	0	0	0	0

IR	Interval = 8							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
7	A_7	A_6	1	1	1	0	0	0
6	A_7	A_6	1	1	0	0	0	0
5	A_7	A_6	1	0	1	0	0	0
4	A_7	A_6	1	0	0	0	0	0
3	A_7	A_6	0	1	1	0	0	0
2	A_7	A_6	0	1	0	0	0	0
1	A_7	A_6	0	0	1	0	0	0
0	A_7	A_6	0	0	0	0	0	0

Table 4. Contents of Third Interrupt Vector Byte

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8

Table 5. Contents of Interrupt Vector Byte, 8086/8088 Mode

IR	Interval = 4							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
7	T_7	T_6	T_5	T_4	T_3	1	1	1
6	T_7	T_6	T_5	T_4	T_3	1	1	0
5	T_7	T_6	T_5	T_4	T_3	1	0	1
4	T_7	T_6	T_5	T_4	T_3	1	0	0
3	T_7	T_6	T_5	T_4	T_3	0	1	1
2	T_7	T_6	T_5	T_4	T_3	0	1	0
1	T_7	T_6	T_5	T_4	T_3	0	0	1
0	T_7	T_6	T_5	T_4	T_3	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.

A_5–A_{15} — 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.

The address form is two bytes long (A_0–A_{15}). When the routine interval is four, A_0–A_4 are automatically inserted by the μPD8259A, while A_5–A_{15} are programmed externally. When the routine interval is eight, A_0–A_5 are automatically inserted by the μPD8259A, while A_6–A_{15} 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_7–T_3 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.

8

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: μPM = 0 sets the μPD8259A for 8085A system operation; μPM = 1 sets the μPD8259A for 8086 system operation.

Figure 1 illustrates the command word initialization sequence.

Figure 1. Initialization Sequence

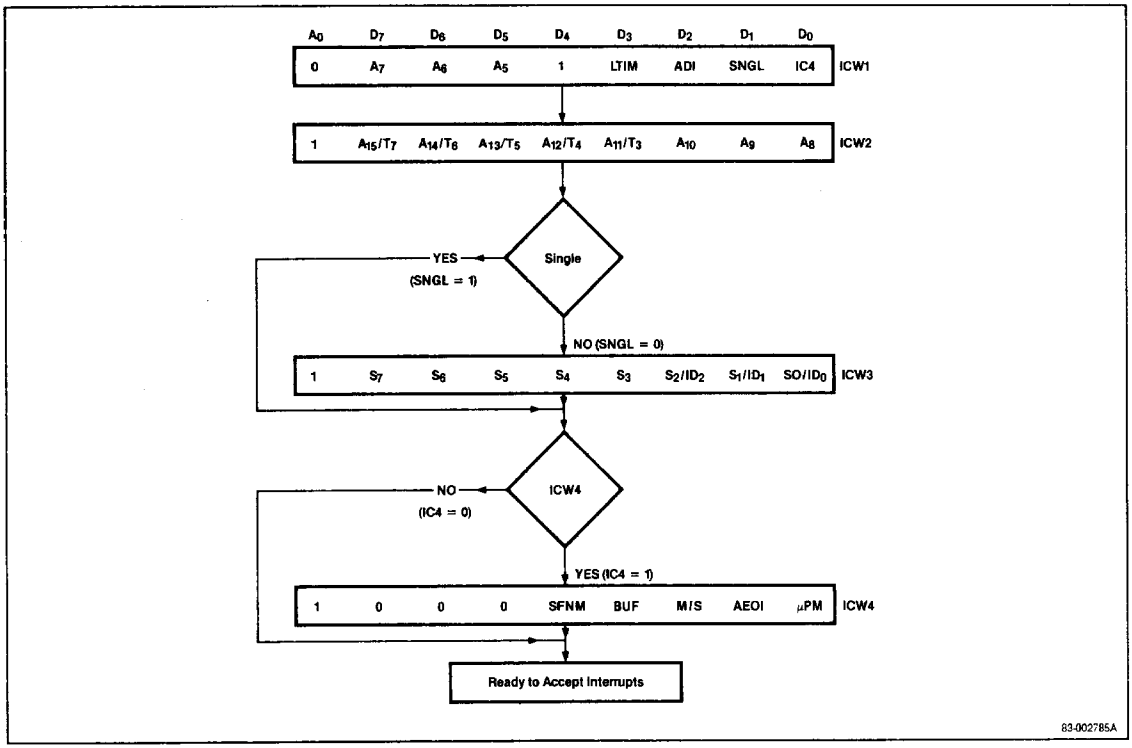
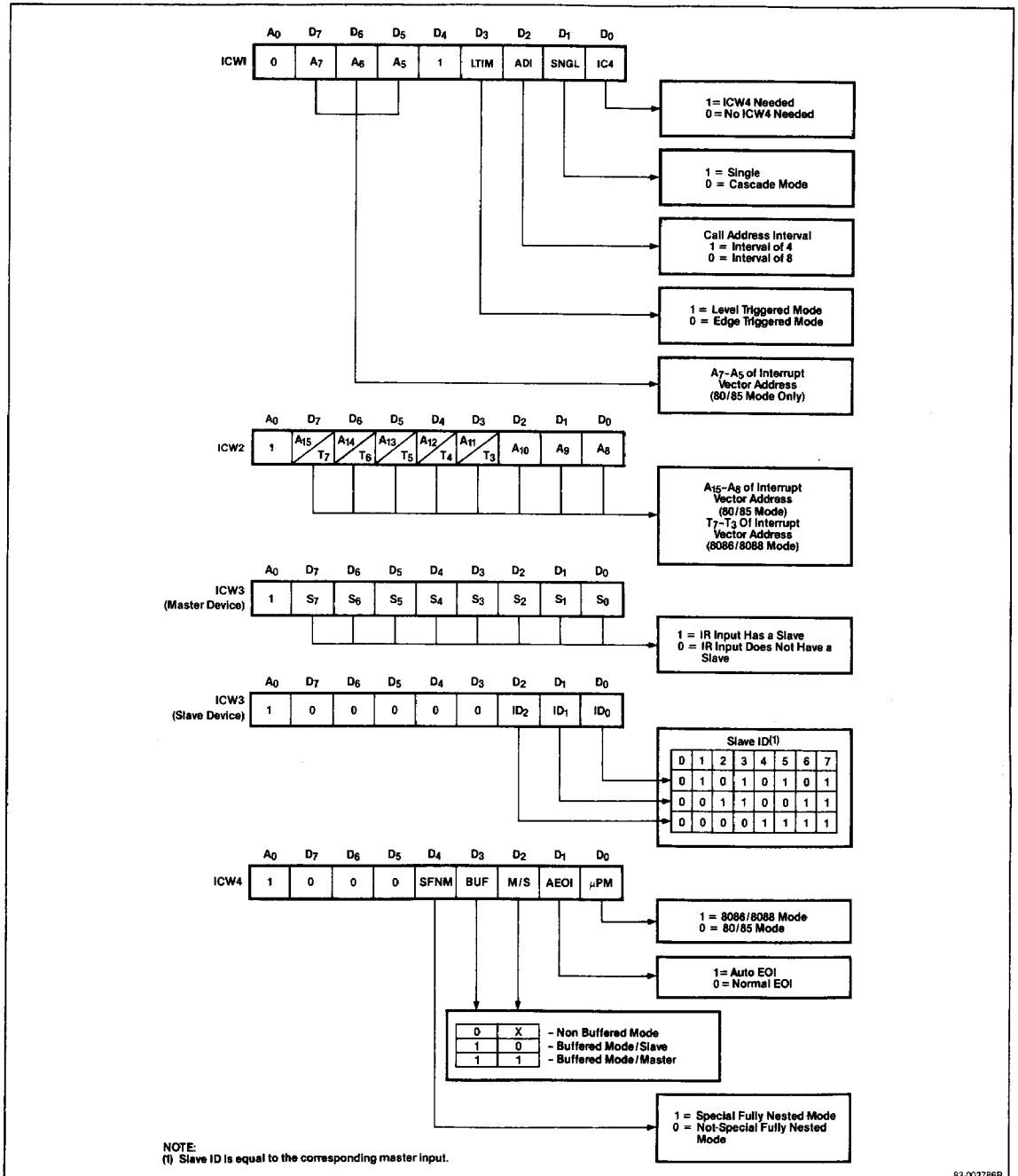


Figure 2 illustrates the initialization command word format.

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 interrrupt, 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_0–IR_7. IR_0 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 $\overline{INTA}$, 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_0 is set to the highest priority and IR_7 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_2 and IR_5.

(1) Before interrupts are serviced:

In-service register

IS_7	IS_6	IS_5	IS_4	IS_3	IS_2	IS_1	IS_0
0	0	1	0	0	1	0	0

Priority status register — highest priority

IR_7	IR_6	IR_5	IR_4	IR_3	IR_2	IR_1	IR_0

According to the priority status register, IR_2 has a higher priority than IR_5 and will be serviced first.

(2) After interrupts are serviced:

In-service register

IS_7	IS_6	IS_5	IS_4	IS_3	IS_2	IS_1	IS_0
0	0	1	0	0	0	0	0

Priority status register — highest priority

IR_2	IR_1	IR_0	IR_7	IR_6	IR_5	IR_4	IR_3

At the completion of IR_2's service routine, the corresponding in-service register bit (IS_2) is reset to logic 0 by the preprogrammed EOI command. IR_2 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_5.

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_3 is set to the lowest priority (bits L_2, L_1, L_0 form the binary code of the bottom priority level), then IR_4 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_2, L_1, L_0 to the lowest priority level. If EOI is set to a logic 1, the ISR bit defined by L_2, L_1, L_0 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_2, L_1, L_0 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 $\overline{RD}$ 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_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
I	X	X	X	X	W_2	W_1	W_0

where:
I = 1 if there is an interrupt requesting service
I = 0 if there are no interrupts
W_2-W_0 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 $\overline{INTA}$ 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 $\overline{RD}$ 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 $\overline{RD}$ command, a $\overline{WR}$ 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 $\overline{WR}$ command must be issued with OCW3 prior to issuing the $\overline{RD}$ 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 $\overline{WR}$ pulse preceding the $\overline{RD}$ is not necessary to read the IMR status. The IMR data is available to the data bus when $\overline{RD}$ is asserted with A_0 at logic 1.

A single OCW3 is sufficient to enable succesive 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.

8

Figure 3. Operation Command Word Format

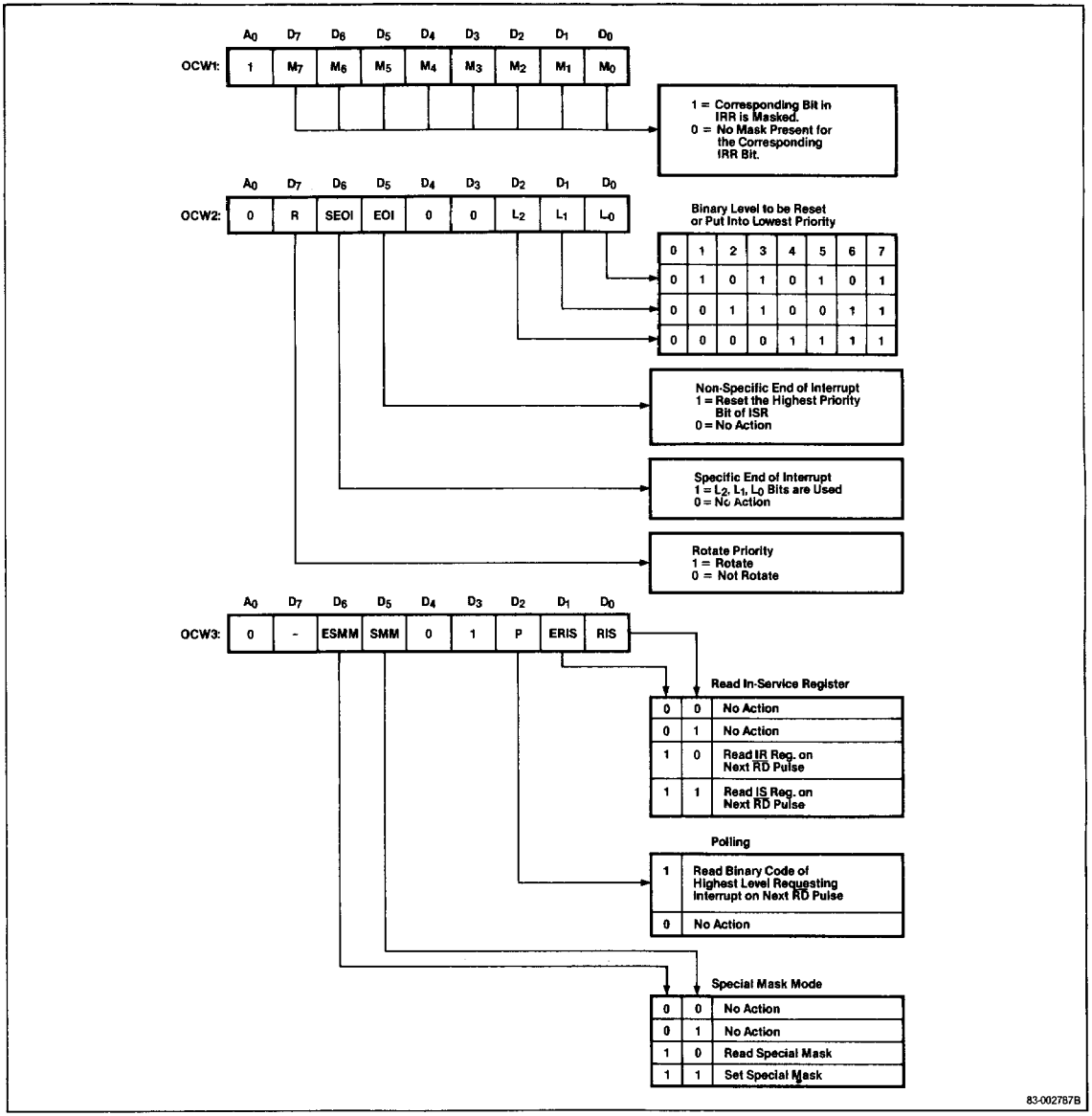
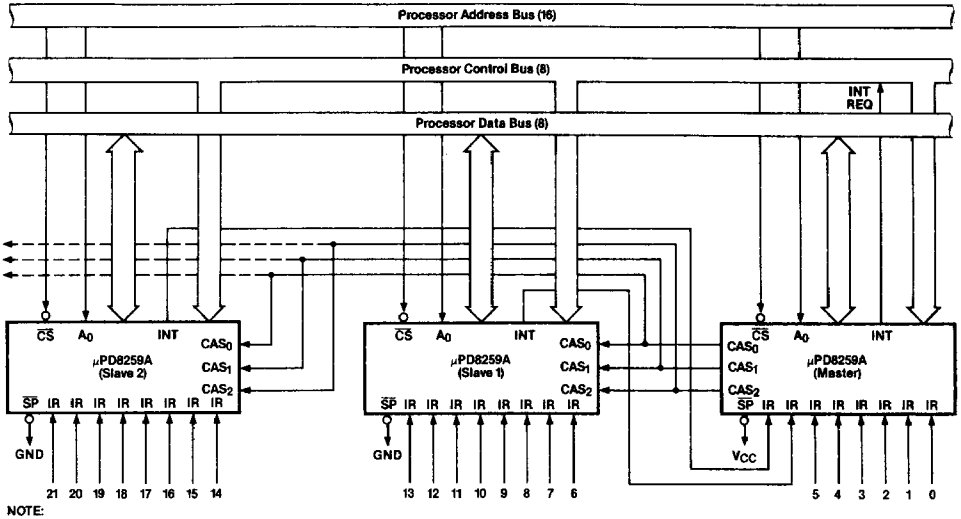


Table 6. Summary of Operation Command Word Programming

	A_0	D_4	D_3				
OCW1	1	X	X	M_7–M_0			IMR (interrupt mask register) $\overline{WR}$ loads IMR data while $\overline{RD}$ reads status
OCW2	0	0	0	R	SEOI	EOI	
				0	0	0	No action
				0	0	1	Non-specific end of interrupt
				0	1	0	No action
				0	1	1	Specific end of interrupt L_2, L_1, L_0 forms binary representation of level to be reset
				1	0	0	No action
				1	0	1	Rotate priority at end of interrupt (auto mode)
				1	1	0	Rotate priority, L_2, L_1, L_0 specifies bottom priority without end of interrupt
				1	1	1	Rotate priority at end of interrupt (specific mode). L_2, L_1, L_0 specifies bottom priority, and it is in-service register bit is reset.

	A_0	D_4	D_3			
OCW3	0	0	1	ESMM	SMM	
				0	0	} Special mask not affected
				0	1	Special mask not affected
				1	0	Reset special mask
				1	1	Set special mask
				ERIS	RIS	
				0	0	} No action
				0	1	No action
				1	0	Read IR register status
				1	1	Read IS register status

Figure 4. Cascading the μPD8259A



NOTE:
(1) Insure that the processor interrupt input is disabled during the execution of any control command and initialization sequence for all μPD8259A's.

83-002788B

8

Instruction Set

#	Mnemonic	Operation Description	Operation Code								A_0	Format
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
(Byte 1 Initialization, No ICW4 Required)												
1	ICW1 A	Single, edge triggered	A_7	A_6	A_5	1	0	1	1	0	0	4
2	ICW1 B	Single, level triggered	A_7	A_6	A_5	1	1	1	1	0	0	4
3	ICW1 C	Not single, edge triggered	A_7	A_6	A_5	1	0	1	0	0	0	4
4	ICW1 D	Not single, level triggered	A_7	A_6	A_5	1	1	1	0	0	0	4
5	ICW1 E	Single, edge triggered	A_7	A_6	0	1	0	0	1	0	0	8
6	ICW1 F	Single, level triggered	A_7	A_6	0	1	1	0	1	0	0	8
7	ICW1 G	Not single, edge triggered	A_7	A_6	0	1	0	0	0	0	0	8
8	ICW1 H	Not single, level triggered	A_7	A_6	0	1	1	0	0	0	0	8
(Byte 1 Initialization, ICW4 Required)												
9	ICW1 I	Single, edge triggered	A_7	A_6	A_5	1	0	1	1	1	0	4
10	ICW1 J	Single, level triggered	A_7	A_6	A_5	1	1	1	1	1	0	4
11	ICW1 K	Not single, edge triggered	A_7	A_6	A_5	1	0	1	0	1	0	4
12	ICW1 L	Not single, level triggered	A_7	A_6	A_5	1	1	1	0	1	0	4
13	ICW1 M	Single, edge triggered	A_7	A_6	0	1	0	0	1	1	0	8
14	ICW1 N	Single, level triggered	A_7	A_6	0	1	1	0	1	1	0	8
15	ICW1 O	Not single, edge triggered	A_7	A_6	0	1	0	0	0	1	0	8
16	ICW1 P	Not single, level triggered	A_7	A_6	0	1	1	0	0	1	0	8
(Byte 2 Initialization)												
17	ICW2	Initialize byte 2	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	1	
(Byte 3 Initialization)												
18	ICW3 M	Initialize byte 3 (master)	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0	1	
19	ICW3 S	Initialize byte 3 (slave)	0	0	0	0	0	S2	S_1	S_0	1	
(Byte 4 Initialization)												
20	ICW4 A	No action, redundant	0	0	0	0	0	0	0	0	1	
21	ICW4 B	Non-buffered, no AEOI, 8086 / 8088	0	0	0	0	0	0	0	1	1	
22	ICW4 C	Non-buffered, AEOI, 80 / 85	0	0	0	0	0	0	1	0	1	
23	ICW4 D	Non-buffered, AEOI, 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 AEOI, 8086 / 8088	0	0	0	0	0	1	0	1	1	
26	ICW4 G	Non-buffered AEOI, 80 / 85	0	0	0	0	0	1	1	0	1	
27	ICW4 H	Non-buffered, AEOI, 8086 / 8088	0	0	0	0	0	1	1	1	1	
28	ICW4 I	Buffered, slave, no AEOI, 80 / 85	0	0	0	0	1	0	0	0	1	
29	ICW4 J	Buffered, slave, no AEOI, 8086 / 8088	0	0	0	0	1	0	0	1	1	
30	ICW4 K	Buffered, slave, AEOI, 80 / 85	0	0	0	0	1	0	1	0	1	
31	ICW4 L	Buffered, slave, AEOI, 8086 / 8088	0	0	0	0	1	0	1	1	1	
32	ICW4 M	Buffered, master, no AEOI, 80 / 85	0	0	0	0	1	1	0	0	1	
33	ICW4 N	Buffered, master, no AEOI, 8086 / 8088	0	0	0	0	1	1	0	1	1	

Instruction Set (cont)

#	Mnemonic	Operation Description	Operation Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_0	Format
(Byte 4 Initialization) (cont)												
34	ICW4 O	Buffered, master, AEOI, 80/85	0	0	0	0	1	1	1	0	1	
35	ICW4 P	Buffered, master, AEOI, 8086/8088	0	0	0	0	1	1	1	1	1	
36	ICW4 NA	Fully nested, non-buffered, no AEOI, 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 AEOI, 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_7	M_6	M_5	M_4	M_3	M_2	M_1	M_0	1	
53	OCW2 E	Non-specific EOI	0	0	1	0	0	0	0	0	0	
54	OCW2 SE	Specific EOI, L_0–L_2 code of IS FF to be reset	0	1	1	0	0	L_2	L_1	L_0	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_0–L_2 code of line	1	1	1	0	0	L_2	L_1	L_0	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_2	L_1	L_0	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	

8